Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.4, Aug. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.4.687

넷플로우-타임윈도우 기반 봇넷 검출을 위한 오토엔코더 실험적 재고찰

강 구 홍†‡
서원대학교 (교수)

An Experimental Study on AutoEncoder to Detect Botnet Traffic Using NetFlow-Timewindow Scheme: Revisited

Koohong Kang†‡
Seowon University (Professor)

요 약

공격 양상이 더욱 지능화되고 다양해진 봇넷은 오늘날 가장 심각한 사이버 보안 위협 중 하나로 인식된다. 본 논문은 UGR과 CTU-13 데이터 셋을 대상으로 반지도 학습 딥러닝 모델인 오토엔코더를 활용한 봇넷 검출 실험결과를 재검토한다. 오토엔코더의 입력벡터를 준비하기 위해, 발신지 IP 주소를 기준으로 넷플로우 레코드를 슬라이딩 윈도우 기반으로 그룹화하고 이들을 중첩하여 트래픽 속성을 추출한 데이터 포인트를 생성하였다. 특히, 본 논문에서는 동일한 흐름-차수(flow-degree)를 가진 데이터 포인트 수가 이들 데이터 포인트에 중첩된 넷플로우 레코드 수에 비례하는 멱법칙(power-law) 특징을 발견하고 실제 데이터 셋을 대상으로 97% 이상의 상관계수를 제공하는 것으로 조사되었다. 또한 이러한 멱법칙 성질은 오토엔코더의 학습에 중요한 영향을 미치고 결과적으로 봇넷 검출 성능에 영향을 주게 된다. 한편 수신자조작특성(ROC)의 곡선아래면적(AUC) 값을 사용해 오토엔코더의 성능을 검증하였다.

ABSTRACT

Botnets, whose attack patterns are becoming more sophisticated and diverse, are recognized as one of the most serious cybersecurity threats today. This paper revisits the experimental results of botnet detection using autoencoder, a semi-supervised deep learning model, for UGR and CTU-13 data sets. To prepare the input vectors of autoencoder, we create data points by grouping the NetFlow records into sliding windows based on source IP address and aggregating them to form features. In particular, we discover a simple power-law; that is the number of data points that have some flow-degree is proportional to the number of NetFlow records aggregated in them. Moreover, we show that our power-law fits the real data very well resulting in correlation coefficients of 97% or higher. We also show that this power-law has an impact on the learning of autoencoder and, as a result, influences the performance of botnet detection. Furthermore, we evaluate the performance of autoencoder using the area under the Receiver Operating Characteristic (ROC) curve.

Keywords: Botnet Detection, Network Security, AutoEncoder, NetFlow, Power-law

I. 서 론

봇넷(botnet)은 공격자(botmaster)에 의해 감염되어 원격으로 제어되는 인터넷에 연결된 컴퓨터(봇 bot)의 집단이다. 공격자는 특정 서버를 공격하거나 혹은 더 많은 좀비(zombie) PC들을 감염시키

Received(05. 08. 2023), Modified(06. 19. 2023), Accepted(06. 19. 2023)

† 주저자, khkang@seowon.ac.kr
‡ 교신저자, khkang@seowon.ac.kr(Corresponding author)

기 위해 이들 봇넷을 조정하게 된다. 과거에는 분산 서비스거부공격(distributed denial of service)과 같은 단순한 공격 패턴을 보였지만, 최근에는 대량의 스팸메일 발송, 클릭 사기(ClickFraud), 신용카드 번호와 같은 개인정보 탈취, 그리고 불법적인 컴퓨팅 자원 사용 등 공격의 양상이 더욱 지능화되며 다양해지고 있다. 이러한 봇넷 피해 사례는 우리가 속한 사회와 특정 기관 및 조직을 넘어 일반 개인의 경제적 손실과 일상생활의 불편으로 이어지고 있어, 오늘날 봇넷은 가장 심각한 사이버보안 위협 중 하나로 인식되고 있다[1].

지난 수년간 봇넷을 검출하기 위한 다양한 방법들이 제안되고 있으나 크게 두 가지 카테고리, 즉 시그너처(signature) 기반과 어노멀리(anomaly) 기반으로 분류된다[2]. 초기 연구들은 대부분 악의적 시그너처(signatures)를 포함하는 TCP와 UDP의 콘텐츠를 직접 검사하는 페이로드(payload) 분석 방법을 사용하였다. 즉 특정 봇넷이 포함하는 특이 패턴을 페이로드 내 콘텐츠와 매칭하여 봇넷을 검출함으로써 높은 정확도를 보였다. 하지만 엄청난 양의 데이터 분석에 따른 컴퓨팅 자원 낭비 문제점들이 지적되어왔다. 뿐만 아니라 끊임없이 진화하는 새로운 봇넷 출현에 따른 즉각적인 시그너처 확보 방안의 어려움, 개인정보 보호 인식에 따른 페이로드 분석에 대한 법적 문제점, 그리고 페이로드 분석을 무력화시키기 위한 암호화 통신으로 인해 페이로드 분석 방법에 의한 봇넷 검출은 한계점을 보이고 있다.

어노멀리 기반 검출은 네트워크 내 봇이 존재할 때 나타나는 네트워크 지연 증가, 대량의 트래픽 발생, 흔하지 않는 포트 사용 트래픽, 그리고 비정상적인 시스템 행위와 같은 비정상 행위를 근거로 봇넷을 검출한다[2,3]. 이러한 어노멀리 기반 검출은 시그너처 기반 검출의 단점을 극복할 수 있는 대안으로 관심을 받고 있으며, 특히 기계학습(ML: Machine Learning)을 이용한 비정상적인 네트워크 트래픽 행위를 분류하는 트래픽 행위 분석 기반 봇 넷 검출 기법에 대한 다양한 연구들이 지난 수년간 진행되어 왔다[3,4,8,10,11,13,15]. 그러나 이러한 연구결과들이 실제 네트워크에 적용된 사례는 매우 제한적이다. 특히 지도학습(supervised learning)을 적용하는 ML 기법은 안전하게 운영되고 있는 실제 네트워크 환경에서 쉽게 확보하기 어려운 악의적인 트래픽을 포함한 정확하게 레이블링된 학습 데이터 셋 확보에 대한 문제점이 있다. 이러한 제약은 제로-데이 공격(zero-day attack)에 대응하기 위한 시그너처 확보에 어려움이 있는 시그너처 기반 검출의 단점과 동일한 수준으로 평가된다. 따라서 본 논문에서는 평상시 실제 네트워크 환경에서 수집되는 네트워크 트래픽과 레이블링 작업이 필요없는 반지도학습(semi-supervised) 딥러닝 (deep learning) 모델 오토엔코더(AE: AutoEncoder)를 활용한 네트워크-기반 봇넷 검출에 초점을 맞춘다.

ML 기술을 활용한 IRC 기반 봇넷 검출을 위한 초기연구[10]부터 딥러닝(deep learning) 기술을 활용한 P2P 기반 봇넷 검출 연구[11]까지 대부분 기존 연구들은 네트워크 트래픽을 IP 패킷 단위가 아닌 플로우(Flow) 단위로 처리하였다. 즉 IP 주소 및 포트 번호 등 동일한 네트워크 혹은 전송계층의 헤더정보를 가진 일련의 IP 패킷 스트림인 플로우를 입력 특징(features)으로 사용하여 봇넷 검출을 시도하였다. 그러나 최근에는 이들 플로우 단위 트래픽 특징들을 발신지 IP 주소 기준으로 일정 시간(time window) 중첩하여 봇넷 행위를 검출할 수 있는 중첩된 플로우 (aggregated flows) 특징들을 이용하고 있다[3,4,6,8,11]. 즉 중첩되어 가공된 새로운 플로우 특징들은 봇넷의 시간 의존적인 행위를 반영하기 때문에 중첩되지 않은 플로우 단위 트래픽 특징과 비교해 봇넷 검출에 훨씬 유리한 것으로 조사되었다. 이러한 타임윈도우 단위로 중첩된 플로우(본 논문에서는 '데이터 포인트'라고 칭함)는 타임윈도우 내 발신지 IP 주소 기준으로 다양한 네트워크 트래픽 통계치를 나타낸다. 특히 Nguyen et al.[4]는 10개 미만 플로우를 가진 데이터 포인트를 제거한 환경에서 AE 모델의 매우 성공적인 실험 결과를 보였다. 그러나 이러한 넷플로우-타임윈도우 기반 봇 넷 검출 연구들이 데이터 포인트 즉 타임윈도우 기반의 중첩된 플로우 트래픽 특징을 사용하기 위한 충분한 연구가 이루어지지 않고 있다. 본 논문에서는 데이터 포인트 내 플로우 수의 멱법칙(power-law) 특징을 발견하고 이러한 멱법칙 성질이 딥러닝 성능에 미치는 영향을 확인하고 이를 극복하기 위한 실험을 진행하였다. 특히 이러한 멱법칙은 Tier 3 ISP의 실제 네트워크에서 수집된 데이터 셋에 대해 97% 이상의 상관계수(correlation coefficient) 값을 가지는 것을 확인하였다. 또한 현재 운영 중인 네트워크 환경에서 수집 가능한 네트워크 트래픽과 레이블링된 학습 데이터 셋이 필요 없는 AE 모델을 활용함으로써 실제 네트워크에 적용 가능한 수준의 기술과 봇넷

검출 성능에 대한 결과를 보였다.

서론에 이어, 제2장에서는 ML을 이용한 넷플로우-타임윈도우 기반의 기존 봇넷 검출 연구에 대해 간략히 기술하고, 제3장에서는 본 논문에서 사용하는 AE 모델을 보였다. 제4장에서는 봇넷 연구에서 가장 많이 사용되고 있는 UGR16과 CTU-13 데이터 셋을 설명하고, 이들로부터 생성되는 데이터 포인트의 멱법칙 특성을 제5장에서 기술한다. 제6장에서는 기존의 AE 성능과 멱법칙 특성을 고려한 성능을 비교 분석하고 마지막으로 제7장에서 결론을 맺는다.

II. 관련 연구

오늘날 더욱 가속화되고 있는 초고속 대용량 네트워크 인프라 구축과 개인정보보호에 대한 관심으로 원시수준(raw level) 트래픽 수집 및 사용은 실현성이 극히 제한적이다. 이에 반하여, 라우터와 같은 네트워크 장비 또는 Zeek와 같은 오픈 소스 프로그램들을 통해 쉽게 수집 가능한 넷플로우[12]를 이용한 네트워크 트래픽 모니터링은 봇넷 검출에 있어 매우 현실적인 대안이 될 수 있다. 또한 봇넷 행위를 잘 반영할 수 있는 타임윈도우 기반의 중첩된 플로우 특성을 활용하는 연구들은 실제 네트워크에 적용 가능한 기술로 인식되고 있다. 본 절에서는 넷플로우-타임윈도우 기반의 ML 적용 봇넷 검출에 대한 기존 연구에 대해 살펴보고 이들 연구의 문제점을 살펴본다.

Zhao et al.[3]은 타임윈도우 내 플로우의 12개 속성(attributes)과 의사결정트리(decision tree) 분류기를 사용하여 봇넷을 검출하였다. 특히 이들 연구는 잘 알려진 봇넷과 프로토콜 행위를 근거로 분류기의 입력 벡터로 사용할 속성을 선정했으며 타임윈도우의 크기에 따른 봇넷 검출 정확도 차이를 보였다. 지나치게 짧은 타임윈도우는 트래픽 특성을 잘 반영하지 못하고, 반대로 지나치게 긴 타임윈도우는 빠른 검출의 방해 요인으로 작용함을 지적하였다. 한편 허니팟 프로젝트를 통해 확보한 Storm과 Waledac 봇넷 트래픽과 Ericsson Research와 LBNL의 백그라운드 트래픽을 합쳐서 만들어진 레이블된 데이터 셋을 활용한 실험을 진행하여 낮은 오탐율과 90%이상의 높은 탐지율을 보였다.

Nguyen et al.[4]은 타임윈도우 내 발신지 IP 주소를 기준으로 53개의 중첩된 속성을 추출하고 반지도 딥러닝 모델인 AE와 변이 오토엔코더(VAE: Variational AE)를 활용한 봇넷 검출을 시도하였다. 이들 연구는 gradient-based 핑거프린팅 기술을 활용하여 비정상 행위를 일으키는 주요 특징을 확인하고 검출된 어노멀리를 설명할 수 있는 방법을 제시하였다. 한편 CTU-13 데이터 셋[7]으로 부터 Neris 멜웨어 넷플로우 트래픽과 Tier 3 ISP 백그라운드 트래픽 UGR 데이터 셋[5]을 합쳐서 만든 데이터 셋을 대상으로 0.9 이상의 ROC AUC 값을 보였다. 그러나 이들은 사용된 53개 속성 중에서 주요 속성만 관련 연구를 통해 제시하고 있으며 10개 미만의 플로우를 가진 데이터 포인트를 제거하여 전체 데이터 셋 중에서 88% 이상의 백그라운드와 16% 이상의 봇넷 데이터 포인트를 제외한 실험을 진행하였다.

Ongun et al.[13]은 연결-레벨의 26개 속성과 타임윈도우 기반의 756개의 중첩된 트래픽 속성을 분리하여 다양한 ML 모델(logistic regression, random forest, 그리고 gradient boosting)을 적용해 봇넷 검출 성능을 검증하였다. 특히 이들은 목적지 포트 번호를 기준으로 중첩된 속성을 정의하였고 입력과 출력 연결을 분리하여 이들 속성을 생성하였다. 또한 지도학습을 이용한 봇넷 검출을 시도하기 때문에 학습 데이터 셋에 대한 레이블 작업의 중요성을 강조하고 있다. 한편 이들 연구는 CTU-13 데이터 셋을 대상으로 0.9 이상의 ROC AUC 값을 보였다. Kim et al.[8]은 봇넷 활동의 순차적(sequential) 특징을 효과적으로 이용하기 위해 Recurrent VAE 모델을 이용하고 추가적으로 전이학습(transfer learning) 프레임워크를 제안하였다. 그러나 이들 연구에서는 사용된 속성의 개수 및 특징에 대한 구체적인 설명이 부족하다. 또한 레이블된 학습 데이터의 정상과 봇넷 샘플로부터 추정된 확률밀도함수를 활용하여 어노멀리 검출을 수행함으로써 레이블된 데이터 셋 확보가 필요하다.

AE의 재생오류(reconstruction error)를 활용하여 봇넷 검출을 시도한 Nguyen et al.[4]를 제외한 이들 기존 연구들은 레이블링된 데이터 셋을 필요로 한다. 그러나 오늘날 다양한 보안장비 구축에 따른 안정적인 네트워크 환경에서는 봇넷 트래픽과 같은 악성행위를 반영한 트래픽 확보가 용이하지 않다. 따라서 정확하게 레이블링 된 학습 데이터가 필요한 ML 지도학습 알고리즘은 실제 네트워크 적용에 어려움이 있다. 한편 앞에서 설명한 바와 같이 Nguyen et al.[4]는 AE의 봇넷 검출 성능을 확보

하기 위해 특별한 이유를 밝히지 않고 10개 미만의 플로우를 가지는 데이터 포인트를 제외하고 실험을 진행하였다. 그러나 본 연구를 통해 이렇게 제거되는 봇넷 데이트 포인트는 UGR의 경우 16% 그리고 CTU-13의 경우 77%에 이르는 것으로 조사되었다. 결국 본 논문은 이들 데이터 포인트에 대해 필터링 처리되는 이유를 멱법칙 특성을 통해 확인하고 이를 극복할 수 있는 실험을 진행하였다.

III. 오토엔코더 모델

AE는 세 개의 주요 계층, 즉 입력 속성 벡터를 수신하는 입력 계층, 속성을 재 표현(representation)하는 잠재(latent) 계층, 그리고 입력 속성 벡터를 다시 재생하는 출력 계층으로 구성된다[14,15]. Fig. 1.은 Nguyen et al.[4]이 사용한 MLP(Multi-layer Perceptron) AE 구조를 보여준다. AE를 구성하는 엔코더(입력 벡터와 잠재 벡터 사이)와 디코더(잠재 벡터와 출력 벡터 사이) 구조, 그리고 잠재 계층의 크기는 AE 성능에 영향을 미치게 된다. 따라서 본 연구에서는 참고문헌[4]에서 사용한 동일 데이터 셋과 AE 네트워크 구조, 그리고 AE 관련 하이퍼 파라미터 값들을 사용하여 동일한 실험 결과를 보일 수 있도록 재현성에 초점을 맞추었다.

다음은 본 논문에서 사용된 AE의 주요 파라미터 값들이다.

- ◉ 잠재계층 크기 :100
- ◉ 엔코더와 디코더는 각각 512, 512, 그리고 1,024 크기의 세 개 히든 레이어로 구성
- ◉ 활성함수는 ReLU (rectified linear unit), 그리고 출력을 위해서는 선형 활성함수 사용
- ◉ 랜덤 초기화와 역전파(backpropagation)를 통한 AE 재생오류의 평균제곱오차(mean square errors) 최소화
- ◉ 미니배치 크기 300과 에포크(epochs) 수 50
- ◉ 학습률 0.01을 적용한 추계적 경사 하강법(stochastic gradient descent)으로 최적화

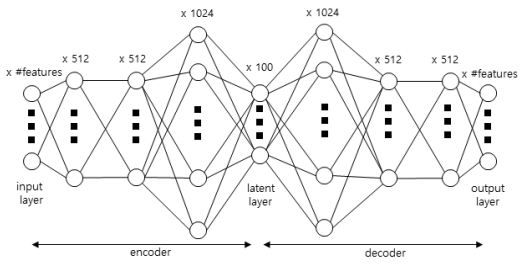

Fig. 1. The autoencoder (AE) architecture

IV. 데이터 셋

본 논문에서는 실험을 위해 두 가지 데이터 셋을 사용한다. 먼저, 최종 사용자(end customers)에게 인터넷 서비스를 제공하는 Tier 3 ISP의 실제 네트워크에서 넷플로우를 수집한 UGR16 데이터 셋[5]과 실제 네트워크에서 다양한 봇넷 멜웨어를 가상머신에 설치해 운영하여 봇의 악성 행위를 수집한 CTU-13 데이터 셋[7]을 활용하였다.

4.1 UGR16

UGR16 데이터 셋 내에는 스캔공격, 스팸공격, DoS공격, 그리고 봇넷 트래픽 등 실제 공격 시나리오를 포함시키고 레이블 작업을 수행하였다. 그러나 봇넷 트래픽의 경우, 공개된 네트워크에서 봇넷 멜웨어가 발생시킬 수 있는 잠재적 악영향을 완벽하게 제어할 수 있는 환경 하에서 컴퓨터를 봇에 감염시키는 것이 불가능하다. 이러한 문제점을 해결하기 위해 Fernandez et al.[5]은 CTU-13 데이터 셋 내 Neris로 알려진 봇넷 트래픽을 자신의 백그라운드 트래픽에 혼합하는 방식을 사용하였다. 그러나 이러한 방법은 봇넷 트래픽이 다른 트래픽에 미치는 영향을 정확하게 반영하지 못하는 문제점이 있어 실제 봇넷 환경을 완벽하게 재현하지 못한다.

Nguyen et al.[4]은 UGR16 데이터 셋 중 5일치 데이터셋을 활용하여 네트워크 어노멀리를 검출할 수 있는 AE와 VAE에 관한 연구를 진행하였다. 본 연구에서도 동일하게 봇넷 프래픽이 포함된 July 30(Sat)과 July 31(Sun) 이틀치 데이터 셋을 사용한다. 즉 학습용 데이터 셋은 봇넷 플로우 레코드가 모두 제거된 30일 데이터 셋, 검증용 데이터 셋은 봇넷 플로우 레코드가 포함된 30일 데이터 셋, 그리고 시험용 데이터 셋은 31일 데이터 셋을 각각 사용하였다. Table 1.은 이들 데이터 셋에 포함된 넷플로우 레코드 수를 보여준다. 본 연구에서는 학습용 플로우 109,110,343개 그리고 시험용 플로우 103,448,109개 이며, 검증 및 시험을 위한 봇넷 플로우 수는 각각 151,619개 그리고 151,562이다.

Table 1. Volume comparison of the NetFlow records in UGR16 (in thousands)

Date(2016)		Total	Botnet
Training Set July 30(Sat)	[4]	110M	152
	this paper	109M	151
Test Set July 31(Sun)	[4]	105M	152
	this paper	103M	151

이러한 수치는 Nguyen et al.[4]이 사용한 UGR16 데이터 셋의 넷 플로우 개수와 일치하는 것으로 조사되었다. 그러나 본 연구에서는 봇넷 이외 다른 공격은 제외하였기 때문에 전체 넷플로우 수는 약간 감소하였다(Table 1.참고). 한편 Table 1.에서 보듯이, 봇넷 학습 및 시험용 데이터셋의 넷플로우 수가 동일함을 볼 수 있다. 이것은 앞에서 설명한 바와 같이 UGR16 데이터셋 생성 시 CTU-13 데이터 셋이 제공하는 동일한 Neris 봇넷 트래픽을 30일과 31일 반복해 백그라운드 트래픽과 혼합한 결과인 것으로 조사되었다.

4.2 CTU-13

CTU-13 데이터 셋[7]은 봇넷 검출을 위한 연구에서 가장 활발히 사용되고 있다. 이 데이터 셋은 특정 멜웨어를 사용하는 가상 머신을 이용하여 봇넷의 행위를 13개 시나리오로 만들었다. 이들 13개 시나리오들의 특징으로는 C&C 통신에 사용되는 IRC, P2P, 그리고 HTTP 등 다양한 프로토콜과 봇이 발생시키는 SPAM, CF(Click Fraud), PS(Port Scan), FF(Fast Flux), 그리고 자체 제어되는 다양한 공격 유형 등 오늘날 발생되는 대부분 봇넷 시나리오를 잘 반영하고 있다.

본 논문에서 사용한 학습, 검증, 그리고 시험을 위해 사용한 시나리오는 Table 3.과 같이 Garcia et al.[7]가 제시한 동일한 시나리오 조합을 사용하였다. 따라서 학습에 전혀 사용되지 않은 새로운 봇넷을 시험을 통해 검출함으로써 새로운 봇넷에 AE 모델이 얼마나 효과적으로 대응할 수 있는지 확인할 수 있다. 그러나 표에서 보듯이 CTU-13 데이터 셋 제공 웹 사이트[1]에서 시나리오 1과 12에 해당하는 레이블링된 단방향(unidirectional) 플로우 정보를 제공하지 않는 이유로 본 연구에서는 제외되었다. 한편, Table 3.은 각 시나리오별 정상 및 봇넷 플로우 수를 보여준다.

1) https://www.stratosphereips.org

Table 2. Data separation into training, validation and testing in CTU-13

Scenario	Dataset
3,4,5,7,10,11,13	Training and validation
2,6,8,9	Testing

Table 3. Distribution of labels (No. of NetFlows) for each scenario in CTU-13

Scenario	Normal	Botnet	Bot
2	225,336	54,433	Neris
3	744,270	75,891	Rbot
4	336,103	6,472	Rbot
5	37,144	2,129	Virut
6	149,931	5,877	Menti
7	28,270	293	Sogou
8	530,666	12,063	Murlo
9	386,889	383,219	Neris
10	321,917	322,158	Rbot
11	11,010	277,892	Rbot
13	59,190	21,760	Virut

V. 데이터 포인트의 멱법칙

본 논문에서는 AE의 입력벡터를 준비하기 위해 발신지 IP 주소를 기준으로 넷플로우 레코드를 슬라이딩 윈도우 기반으로 그룹화하고 이들을 중첩하여 특징을 추출한 데이터 포인트를 생성하였다. Table 4.는 UGR16 데이터 셋에 대해 타임윈도우 3분 주기로 만들어진 데이터 포인트 수를 보여준다. 한편 검증 및 시험을 위한 레이블링 작업은 참고문헌[4]와 같이 데이터 포인트 내에 중첩된 플로우 수의 절반 이상이 봇넷 플로우인 경우 해당 데이터 포인트를 어노멀리 데이터 포인트로 지정하였다. Table 1.을 통해 이미 예상한 바와 같이 UGR16 데이터 셋은 백그라운드와 봇넷의 데이터 포인트 수가 지나치게 불균형되어 있다. 특히 Nguyen et al.[4]의 실험에서 제안된 데이터 포인트 내 10개 미만의 플로우를 가진 데이터 포인트를 잡음으로 간주하고 제거하면 Table 4.에서 보듯이 백그라운드는 88% 그리고 어노멀리는 16% 이상의 데이터 포인트가 손실되는

Table 4. Volume of background and botnet data points in UGR16 (The number in parentheses is the number of botnet flows)

	Total	Filtering
train	8,463,959	1,015,455
validation	8,463,933 (1,208)	1,015,461 (983)
test	8,277,806 (1,193)	955,907 (985)

것을 확인할 수 있다. 이러한 필터링에 의해 발생되는 엄청난 정보손실은 생성된 데이터 포인트의 멱법칙 특성[9]에 의해 잘 설명된다. 멱법칙은 $y \propto x^a$ 의 형태(여기서, a 는 상수값)를 가지고 x 와 y 는 관심의 대상, 그리고 $\propto$ 는 "비례"를 나타낸다. 따라서 이러한 멱법칙 특징을 가지는 현상은 로그-로그 그림에서 선형적 근사(linear fitting)가 가능하다.

하나의 데이터 포인트에 중첩된 플로우 수(흐름-차수 flow-degree) d에 대한 빈도수 f_d는 데이터 포인트 내 d개 플로우를 가진 데이터 포인트 개수로 정의하고 빈도수 f_d 대 흐름-차수 d를 로그-로그 스케일로 그리면 Fig. 2.의 왼쪽 그림과 같다. 근사화 정확도를 높이기 위해서 빈도수가 1인 매우 높은 흐름-차수(3000개 이상)를 가진 데이터 포인트와 흐름-차수가 1인 데이터 포인트는 그림에서 제외하였다. 그림에서 보듯이, 선형회귀(linear regression) 결과로부터 얻은 실선이 실제 데이터를 잘 표현함을 알 수 있으며 상관계수값이 0.97로 계산되었다. 따라서 다음과 같은 멱법칙 정의를 내릴 수 있다.

멱법칙 1. (흐름-차수 지수)
데이터 포인트 내 d개 플로우를 가진 데이터 포인트의 빈도수 f_d는 데이터 포인트 내 플로우 수(흐름-차수, flow-degree)의 상수 Ω의 거듭제곱(power)에 비례한다.

$$f_d \propto d^{\Omega}$$

정의 1. 흐름-차수 지수 Ω를 로그-로그 스케일에서 흐름-차수 빈도수 대 흐름-차수 그래프의 기울기로 정의할 수 있다.

Fig. 2.에서 보듯이, Tier 3 ISP의 실제 트래픽에 대해 데이터 포인트의 흐름-차수의 지수

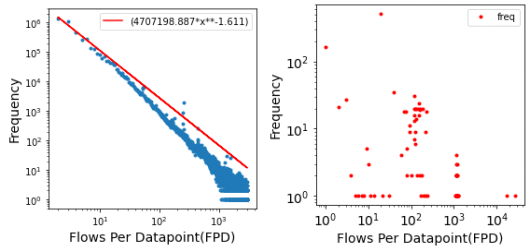

Fig. 2. The number of data points plots in UGR dataset July 30(Sat): Log-log plot of frequency versus the number of flows per data point(Left: the train data points, Right: the botnet data points in the validation dataset)

(exponent) 값은 -1.611로 확인되었다. 한편 검증용 데이터 포인트 내 봇넷의 경우, 멱법칙 특징이 불명확하지만 플로우 수가 많은 데이터 포인트의 빈도수가 급격히 감소하는 긴꼬리(long-tail) 분포의 경향이 있음을 확인할 수 있다(Fig. 2.의 오른쪽 그림 참조 - 데이터 포인트 대부분이 그래프의 좌하 삼각 영역에 존재).

UGR16과 동일한 방법으로 CTU-13 데이터 셋을 대상으로 Table 5.와 같이 학습, 검증 그리고 시험 데이터 포인트 그리고 데이터 포인트 당 플로우 수(FPD : Flows Per Datapoint) 10개 미만을 제거한 데이터 포인트를 생성하였다. 표에서 보듯이, 필터링된 데이터 포인트 경우 시험(혹은 검증)용 데이터 포인트의 백그라운드 48%(혹은 55%) 그리고 봇넷 77%(혹은 85%)의 데이터 포인트가 제거되는 것을 확인할 수 있다. UGR 데이터 포인트와는 달리 이러한 심각한 봇넷 데이터 포인트 손실은 Fig. 3.의 오른쪽 그림 멱법칙 특징을 통해 확인할 수 있다. 즉 Fig. 2.의 오른쪽 그림(UGR16 봇넷 데이터 포인트)과 비교해 CTU-13의 봇넷 데이터 포인트의 멱법칙 특징은 보다 분명하게 확인되며 이로 인해 필

Table 5. Volume of the data points that contain too few flows (less than 10 in this case) are removed from the CTU-13 dataset (The number in parentheses is the number of botnet flows)

	Total	Filtering
train	56,535	25,189
validation	56,486 (8,334)	25,190 (1,192)
test	33,918 (8,091)	17,444 (1,798)

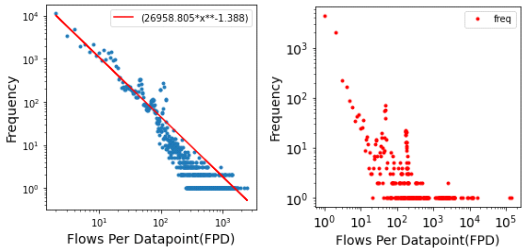

Fig. 3. The number of data points plots in CTU-13 data points: Log-log plot of frequency versus the number of flows per data point(Left: the train data points, Right: the botnet data points in the validation dataset)

터링된 데이터 포인트 손실이 확대된 것이다. 한편, Fig. 3.(왼쪽 그림)으로부터 CTU-13의 학습 데이터 포인트에 대한 흐름-차수의 지수 값은 -1.388, 그리고 상관계수가 92%로 각각 확인되었다.

VI. 실 험

6.1 입력 벡터

넷플로우는 기본 필드 - 시작시간, 지속시간, 발신지 및 목적지 IP 주소, 프로토콜, 발신지 및 목적지 포트 번호, 패킷 수, 바이트 수, 그리고 상태정보(State) - 로 구성된다. 물론 넷플로우 버전 및 종류에 따른 추가적인 정보를 사용할 수 있으나 본 연구에서는 이들 기본 필드 정보만 이용하였다. 한편, 제5장에서 설명한 바와 같이 타임윈도우 기반 봇넷 검출을 위해 3분 주기로 발신지 IP 주소를 기준으로 플로우 레코드들을 그룹화하고 동일 그룹 내 플로우 레코드들의 기본 속성을 중첩하여 다음과 같은 데이터 포인트의 32개 속성(AE 입력 벡터)을 생성하였다.

- 플로우 지속기간, 패킷 수, 바이트 수, 패킷율, 그리고 바이트율의 평균과 표준편차
- 프로토콜 타입, 목적지 IP 주소, 발신지 포트, 목적지 포트, 그리고 상태정보에 대한 엔트로피
- 플로우 수
- 잘 알려진(well-known) 포트 (ftp, ssh, smtp, dns, http, pop, telnet, winrpc)를 사용하는 발신지 및 목적지 포트 사용 비율

이러한 32개 입력 벡터의 개수는 참고문헌[4]와 비교해 다소 적은 수이며 참고문헌[7]과 비교해서는 약간 많은 개수이다. 참고문헌[7]의 경우 양방향 넷플로우를 사용함에 따라 단방향 넷플로우를 사용하는 본 연구와는 입력 벡터에 차이가 있다. 이들 속성값들은 Min-Max Scaler를 이용한 정규화 과정을 통해 0과 1사이 값으로 변환되어 AE에 입력된다. 한편, 속성값들이 여러 개의 플로우를 중첩하여 생성되는 평균, 표준편차, 엔트로피, 그리고 비율 값을 가짐에 따라 만약 데이터 포인트 내에 하나의 플로우만 존재하는 경우 입력 벡터가 가져야 하는 속성 값에 대한 의미가 사라지게 된다. 결국 FPD=1인 데이터 포인트는 적절한 의미의 입력 벡터 값을 가지지 못하게 되며 본 연구에서는 이들 데이터 포인트는 실험에서 제외하였다.

6.2 AE 성능 분석

6.2.1 UGR16 봇넷 검출 성능

ROC(Receiver Operating Characteristic) 곡선은 거짓 양성 비율(FPR: False Positive Rate)에 대한 진짜 양성 비율(TPR: True Positive Rate)의 곡선으로 이진 분류기의 성능을 나타내기 위해 널리 사용되는 도구이다[16]. Fig. 4.는 FPD 10개 미만의 데이터 포인트가 제거된 UGR16에 대한 AE의 ROC 곡선을 보여준다. 이러한 결과는 참고문헌[4]의 ROC 곡선과 거의 일치하는 것으로 조사되었다. 따라서 본 논문에서 구현된 AE 모델은 기존연구의 결과를 정확하게 재현하고 있다. 한편 Fig. 4.에서 보듯이 검증(왼쪽 그림) 및 시험(오른쪽 그림) 데이터 포인트에 대한 두 ROC 곡선이 거의 동일한 모양을 보이는 것을 확인할 수 있다. 이러한 이유는 제4.1절에서 설명한 바와 같이 UGR16 데이터 셋은 동일한 Neris 봇을 이틀간에 걸쳐 반복 사용함에 따라서 검증과 시험 데이터 셋의 백그라운드 트래픽만 서로 다르고 봇넷 트래픽은 동일하기 때문이다. 그러나 시험 데이터 셋에 대해서도 검증 데이터 셋과 동일한 수준의 성능을 보장함으로써 동일한 네트워크에서 서로 다른 요일에 수집된 백그라운드 트래픽 환경에서 학습된 AE 모델이 같은 종류의 봇넷을 성공적으로 검출할 수 있는 사실을 확인할 수 있다. Table 6.은 ROC 곡선의 AUC(Area Under the Curve) 값을 보여준다. 참고문헌[4]의 AE 학습과정에서 설정한 에포크 수

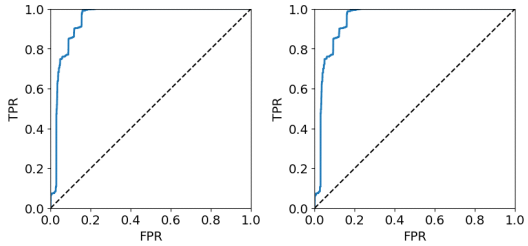

Fig. 4. ROC(Left: the validation data, Right: the test data) of the filtered UGR16 data points

50을 동일하게 적용한 경우, 약간 낮은 수준의 성능을 확인할 수 있으나 에포크 수를 증가시키면 검증 및 시험 데이터 포인트 모두 0.94 이상의 높은 성능을 보이는 것으로 확인되었다. 그러나 앞에서 설명한 바와 같이 필터링 처리로 인해 전체 봇넷 데이터 포인트의 16%는 기본적으로 검출하지 못하는 결과가 된다.

Table 7.은 FPD=1을 제외한 전체 데이터 포인트에 대해 학습한 AE의 ROC AUC 값을 보인다. 표에서 보듯이 FPD=1을 제외한 결과 13% 정도의 봇넷 데이터 포인트의 손실이 확인되고 이러한 손실은 참고문헌[4]의 필터링 손실 16%와 비교해 큰 차이는 아닌 것으로 확인되었다. 그러나 Table 7.을 통해 ROC AUC 성능을 살펴보면, 에포크 수 50에서는 Table 6.과 비교해 더 많은 봇넷 데이터 포인트을 고려했음에도 불구하고 비슷한 성능을 보이고 있으며 에포크 수 500인 경우 매우 높은 성능을 보이는 것을 알 수 있다. 에포크 50과 500에서 AE 모델이 학습한 학습 데이터 포인트를 대상으로 AE의 재생오류의 평균을 확인하면 Fig. 5.와 같다. 그림에서 보듯이 학습 에포크수 50의 경우 FPD가 증가하면 재생오류값이 서서히 증가하는 패턴을 보인다. 이러한 현상은 4.1절에서 설명한 FPD의 멱법칙 특성으로 설명된다. 즉 데이터 포인트 내 플로우 수가 많은 학습 샘플 수는 플로우 수가 적은 학습 샘플 수에 비해 지수 함수적으로 적기 때문에 AE의 학습

Table 6. Area under the curve (AUC) of the ROC of the filtered UGR16 data points

	Validation	Test
[4]	0.931	0.915
epoch=50	0.911	0.908
epoch=200	0.949	0.946
epoch=500	0.955	0.953

Table 7. Area under the curve (AUC) of the ROC of the UGR16 data points except FPD=1

	Validation	Test
# Data points	4,958,433 (1,042)	4,784,168 (1,035)
epoch=50	0.935	0.934
epoch=500	0.973	0.975

에포크 수가 적은 경우 FPD가 큰 데이터 포인트들은 충분한 학습이 이루어질 수 없다. 따라서 Fig. 5.에서 보듯이 FPD의 증가에 따른 학습 데이터 포인트들의 AE 재생오류 차이가 크지 않도록 에포크 수를 500 이상으로 충분히 늘려 실험을 진행할 필요가 있다.

UGR16 데이터 셋을 이용한 봇넷 검출 실험 결과에 대한 타당성은 다소 부족해 보인다. 즉 실제 네트워크에서 수집된 백그라운드 트래픽은 충분한 가치를 가지고 있으나 한 가지 종류의 봇넷 트래픽을 매우 짧은 시간에 추가함으로써 시험 데이터 셋 내에 검출 대상 샘플 수가 지나치게 적다. 따라서 이러한 AE 성능 검증은 CTU-13과 같이 여러 가지 봇넷 시나리오와 많은 어노멀리 데이터 포인트를 제공하는 데이터 셋을 통해 결과를 확인하는 것이 필요하다.

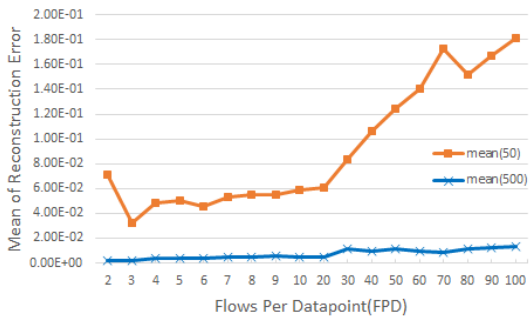

Fig. 5. Mean of AE reconstruction error by FPD for the training data points

6.2.2 CTU-13 봇넷 검출 성능

Table 8.은 참고문헌[4]에서 제시한 FPD 10 미만의 데이터 포인트를 제거한 CTU-13에 대한 학습, 검증 그리고 시험을 실시한 AE의 ROC AUC 값을 보여준다. 이때 학습 및 시험에 사용된 시나리오 별 조합은 Table 2.와 같으며, Table 9.는 시험 시나리오별 ROC AUC 결과 값을 보여준다. 제

Table 8. Area under the curve (AUC) of the ROC of the filtered CTU-13 data points

	Validation	Test
ROC AUC	0.897	0.920

Table 9. Area under the curve (AUC) of the ROC for each scenario in the filtered CTU-13 data points

Scenario	S2	S6	S8	S9
ROC AUC	0.957	0.995	0.853	0.906

4.2절에서 설명한 바와 같이 시험용 봇넷 시나리오를 학습에 사용된 봇넷 시나리오와 완전히 분리하여 실험함으로써 AE의 새로운 봇넷 검출 능력을 확인하였다. Table 9.에서, 시나리오 8(Murlo 봇넷)을 제외하면 모든 시험 데이터 포인트에 대해 0.9 이상의 높은 ROC AUC 값을 보이고 있으나 제4.2절에서 설명한 바와 같이 77%의 봇넷 데이터 포인트가 제거된 시험 결과이다. 따라서 UGR16 데이터 셋과는 달리 많은 봇넷 데이터 포인트가 손실되는 CTU13 데이터 셋에서는 이러한 실험 결과 값이 큰 의미를 가지지 못하게 된다.

Table 10.은 FPD=1을 제외한 모든 데이터 포인트를 대상으로 학습, 검증 그리고 시험한 결과를 보여준다. 표에서 보듯이, FPD=1 데이터 포인트 제거에 따라 검증 및 시험 봇넷 데이터 포인트가 각각 39% 그리고 53% 제외되어 참고문헌[4]의 77%와 85% 비교해 보면 많은 수의 봇넷 데이터 포인트를 대상으로 검출을 시도하고 있음을 확인할 수 있다. 한편, UGR16 데이터 포인트에 대한 실험(Fig. 5. 참고)과 같이 FPD에 따른 AE 재생오류의 차이를 최소화할 수 있도록 에포크 수를 조절(본 연구에서는 에포크 수 2,000 적용)하여 Table 10.과 같은 ROC AUC 값을 구하였다. 시험용 봇넷 시나리오의 경우 ROC AUC 값이 0.832 로 Table 8.과 비교해 다소 낮은 성능을 보이고 있으나 앞에서 설명한 바와 같이 4,927개 봇넷 샘플에 대한 실험 결과 값이다.

한편, FPD의 멱법칙 특징으로부터 10 미만의 많은 학습 데이터 포인트 수에 대한 불균형(imbalanced) 문제를 해소하기 위해 과샘플(over sampling) 혹은 저-샘플링(under sampling) 기법을 시도해 볼 수 있다[17]. 본 연구에서는

Table 10. Area under the curve (AUC) of the ROC of the CTU-13 data points except FPD=1

dataset	# Data points	ROC AUC
Validation	53,795(3,887)	0.879
test	32,299(4,927)	0.832

Table 11. Area under the curve (AUC) of the ROC of the CTU-13 data points under-sampled by FPD 10(no. of data point 1,184)

dataset	# Data points	ROC AUC
Validation	53,795(3,887)	0.920
Test	32,299(4,927)	0.861

FPD=10을 기준(데이터 포인트 수 1,184)으로 FPD 2 ~ 10 까지 동일한 학습 샘플 개수를 유지(저-샘플링)하도록 pandas Dataframe sample 함수를 사용하여 34,661개 학습 데이터 포인트를 생성하였다. Table 11.에서 보듯이, 검증 및 시험 데이터 포인트 모두 ROC AUC 성능 지표가 향상되는 것을 확인할 수 있었다. 이러한 실험 결과는 데이터 포인트의 멱법칙 성질이 AE 성능에 미치는 영향을 최소화하기 위해 더욱 다양한 데이터 포인트 불균형 해소 방안에 대한 향후 연구가 필요하다는 것이다.

VII. 결 론

고속 대용량 네트워크에서도 쉽게 수집 가능한 넷플로우 레코드를 활용한 봇넷 검출 연구는 개인정보 보호 인식 확대와 암호화 통신으로 인해 더욱 주목을 받고 있다. 최근 실시간 수집되는 넷플로우 레코드를 발신지 IP 주소 기준으로 슬라이딩 윈도우 단위로 그룹화하고 이렇게 동일한 발신지 IP 주소를 가진 넷플로우의 중첩된 트래픽 속성을 데이터 포인트(입력 벡터)로 사용하는 오토엔코더 딥러닝 모델에 대한 연구가 활발히 진행되고 있으나, 주어진 데이터 셋에 대한 임의적인 변경과 제시된 성능을 재현할 수 없는 실험 환경의 모호함 그리고 세분화된 실험결과의 부재 등의 문제점이 있다. 본 논문에서는 실험을 세분화하여 성능 평가 결과를 분석하여 보여줌으로써 재현 실험을 통해 해당 기술을 보다 깊이 있게 이해하는 데 기여하였다. 특히, 실제 네트워크 환경에서 수집한 UGR16 데이터 셋을 대상으로 생성된 데이터

포인트의 멱법칙 특징을 발견하였다. 이 멱법칙은 데이터 포인트 내 플로우 수가 고도로 구부러진(skewed) 분포, 즉 긴 꼬리 분포(long tail distribution)를 가지고 있으며 동일한 흐름-차수를 가진 데이터 포인트의 빈도수는 이들 데이터 포인트에 중첩된 넷플로우 레코드 수에 비례하는 멱법칙 지수를 도출하였다. 한편 이러한 멱법칙은 97% 이상의 상관계수를 제공하는 것으로 조사되었다. 많은 기존 연구들이 자체 데이터 셋을 활용하여 모델 개발 및 검증 작업을 진행하고 있으나 실제 네트워크에서 발생되는 이러한 멱법칙 특징을 확인하여 자체 데이터 셋의 유효성을 검증할 필요가 있다.

기계학습 및 딥러링 모델을 활용한 봇넷 검출에 대한 연구가 매우 활발하게 진행되고 있다. 그러나 많은 연구들이 정확하게 레이블링된 데이터 셋과 원시수준의 트래픽을 이용하고 있어 실제 네트워크에 적용하기 위해서는 어려움이 있는 것이 사실이다. 따라서 대부분의 네트워크 장비에서 지원하는 넷플로우 정보와 타임윈도우 기반으로 한 차원 높은 트래픽 속성을 이용하여 반지도학습 딥러닝 모델 오토엔코더를 활용하는 것은 실제 네트워크에 구현 가능한 기술 수준으로 인식된다. 넷플로우-타임윈도우 기반의 기존 연구에서는 적은 수의 플로우(예를 들어, 10개 미만)를 가진 데이터 포인트를 잡음으로 처리해 필터링하여 학습,검증 그리고 시험 데이터 셋에서 제거함으로써 딥러닝 모델 성능을 일정 수준 보장하고 있다. 그러나 이러한 필터링 작업은 본 논문에서 확인된 데이터 포인트의 멱법칙 특징을 고려하면 매우 심각한 정보 손실이 발생하게 된다. 본 논문에서는 이러한 멱법칙 성질이 딥러닝 모델 오토엔더의 학습에 미치는 영향을 분석하였고 이를 극복할 수 있는 다양한 실험 결과를 제시하였다.

References

[1] K. Alieyan, A. ALmomani, A. Manasrah, and M.M. Kadhum, "A survey of botnet detection based on DNS," Neural Computing and Applications, vol. 28, no. 7, pp. 1541-1558, July 2017.

[2] H.R. Zeidanloo, M.J.Z. Shooshtari, P.V. Amoli, M. Safari, and M. Zamani, "A taxonomy of botnet detection techniques," Proceedings of the International Conference on Computer Science and Information Technology, vol. 2, pp. 158-162, July 2010.

[3] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," Computers & Security, vol. 39, pp. 2-16, Nov. 2013.

[4] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A gradient-based explainable variational Autoencoder for network anomaly detection," Proceedings of the CNS, pp. 91-99, March 2019.

[5] G. M. Fernabdez, J. Camacho, R. Magan-Carrion, P. Garcia-Teodoro, and R. Theron, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," Computer & Security, vol. 73, pp. 441-424, March 2018.

[6] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting botnet command and control servers through large-scale NetFlow analysis," Proceedings of the ACSAC, pp. 129-138, Dec. 2012.

[7] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Comp. & Sec., vol. 45, pp. 100-123, Sep. 2014.

[8] J. Kim, A. Sim, J. Kim, K. Wu, and J. Hahm, "Improving botnet detection with recurrent neural network and transfer learning," arXiv preprint arXiv:2104.12602, April 2021.

[9] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology,"

ACM SIGCOMM Computer Communication Review, vol. 29, no. 4, pp. 251-262, Oct. 1999.
[10] L. Carl, R. Walsh, D. Lapsley, and W. Strayer, "Using machine learning techniques to identify botnet traffic," Proceedings of the Local Computer Networks, pp. 967-974, Nov. 2006.
[11] V. Roosmalen, J. Vranken, and M. Eekelen, "Applying deep learning on packet flows for botnet detection," Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 1629-1636, April 2018.
[12] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes, "Review: A survey of network flow applications," Journal of Network and Computer Applications, vol. 36, no. 2, pp. 567-581, March 2013.
[13] T. Ongun, T. Sakharaov, S. Boboila, A. Oprea, and T. Eliassi-Dad, "On designing machine learning models for malicious network traffic classification," arXiv preprint arXiv:1907.04846, July 2019.
[14] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptron and singular value decomposition," Biological cybernetics, vol. 59, no. 4-5, pp. 291-294, Sep. 1988.
[15] K. Kang, "Network anomaly detection technologies using unsupervised learning AutoEncoders," Journal of the Korea Institute of Information Security & Cryptology," vol. 30, no. 4, pp. 617-629, Aug. 2020.
[16] A. Geron, Hands-On machine learning with Scikit-Learn, Keras & TensorFlow: Concepts, tools, and techniques to build intelligent systems, 2nd Edition, O'Reilly Media, 2019.
[17] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," Proceedings of the 14th International Conference on Machine Learning, pp. 179-186, July 1997.

〈저자소개〉

강 구 홍 (Koohong Kang) 정회원
1985년 8월: 경북대학교 전자공학과 졸업
1990년 2월: 충남대학교 전자공학과 석사
1998년 2월: 포항공과대학교 컴퓨터공학과 박사
1985년 9월~1999년 3월: 한국전자통신연구원 선임연구원
2000년 9월~현재: 서원대학교 소프트웨어학부 교수
〈관심분야〉 성능분석, 네트워크 보안, 인공지능