논문 24-27-01-05 KNOM Review '24-01 Vol.27 No.01
https://doi.org/10.22670/knom.2024.27.1.49

딥러닝 기반 응용 트래픽 분석을 위한 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템

장 윤 성◆, 이 윤 서*, 최 유 진*, 박 선 민*, 김 주 성**, 백 의 준**, 김 명 섭°

Multi-host Dataset Collection System Using Process Log for DL-Based Application Traffic Analysis

Yoon-Seong Jang◆, Yoon-Seo Lee*, Yu-Jin Choi*, Seon-Min Park*,
Ju-Sung Kim**, Ui-Jun Baek**, Myung-Sup Kim°

요 약

네트워크 트래픽 분류는 네트워크 관리 분야의 핵심 기술로 최근 딥러닝 기법을 적용하여 다양한 공개 데이터셋을 활용하여 발전하고 있다. 그러나 현존 공개 데이터셋의 대부분은 단일 호스트 데이터셋으로 특정 사용자 행동 패턴에 과적합될 가능성이 있다. 본 논문에서는 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템을 제안하며, 이를 활용하여 수집한 데이터셋을 활용하여 다양한 사용자 행동 패턴을 학습하여 모델의 일반화 능력을 향상시키고, 2D-CNN 분류기 모델의 정확도를 약 27~29% 향상시켰다. 이러한 시스템을 통해 다중 호스트 데이터셋의 구축을 지원하고, 분류기 모델의 일반화 능력을 향상시키고자 한다.

Key Words : Machine Learning, Dataset, Network Traffic Classification, Multi-host, Generalization

ABSTRACT

Network traffic classification is a core technology in network management, and recent advancements have applied deep learning techniques using various public datasets. However, most existing public datasets are single-host datasets, which may lead to overfitting to specific user behavior patterns. In this paper, we propose a multi-host dataset collection system utilizing process logs. Using the collected dataset, we aim to enhance the model's generalization capability by learning various user behavior patterns and improve the 2D-CNN Classifier model's accuracy by approximately 27-29%. This system supports the construction of multi-host datasets and aims to improve the generalization capability of classifier models.

※ 본 연구는 2024년도교육부의재원으로한국연구재단의지원을받아수행된지자체-대학협력기반지역혁신 사업(2021RIS-004), 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(00235509, ICT융합 공공 서비스●인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관제 기술 개발)

◆ First Author : Korea University Department of Computer Convergence Software, brave1094@korea.ac.kr
° Corresponding Author : Korea University Department of Computer and Information Science, tmskim@korea.ac.kr
* Korea University of Department of Computer and Convergence Software, {yoongbb, yujin0706, parksm1101}@korea.ac.kr
** Korea University Department of Computer Information Science, {jsung0514, pb1069}@korea.ac.kr

논문번호 : KNOM2024-01-08 Received July 16, 2024; Revised August 10, 2024; Accepted August 20, 2024

Ⅰ. 서 론

트래픽 분류는 침입 탐지와 같은 네트워크 보안, 성능 최적화, 서비스 품질 관리 등 다양한 네트워크 관리 및 보안 목적을 위해 중요한 역할을 하는 분야이다. 이 분야는 네트워크에서 전송되는 데이터 패킷을 그 특성과 행동에 따라 다른 카테고리 또는 클래스로 분류하는 프로세스를 포함하며, 이러한 분류는 데이터의 특성을 이해하고 관리하기 위해 필수적이다. 컴퓨터 네트워크에서 발생하는 트래픽은 다양한 출발지과 목적지 간에 전송되며, 웹 브라우징, 비디오 스트리밍, 파일 전송, 음성 통화 등 다양한 응용 프로그램 및 서비스에서 발생한다. 트래픽 분류는 이러한 다양한 트래픽 유형을 식별하고 분석함으로써 네트워크 관리자에게 중요한 정보를 제공하고, 보안 위협을 탐지하고 대응하는 데 도움을 준다. 트래픽 분류에는 포트 기반 분류, 패킷 헤더 기반 분류, 통계적 분류, 심층 패킷 분석 분류 등 다양한 분류 기법이 존재한다. 이러한 기법들은 최근 고도화된 프로토콜 보안에 의해 그 정확도가 현저히 떨어지게 되었다. 이에 최근에는 머신러닝, 딥러닝 분야를 트래픽 분류에 적용하여 정확도를 향상시키는 방향으로 연구가 활발히 진행되고 있다.

딥러닝 기법은 최근 몇 년 동안 컴퓨터 비전, 자연어 처리, 음성 인식 등 다양한 분야에서 혁신적인 성과를 거두며 주목받고 있는 인공지능 기술 중 하나이다. 이러한 딥러닝 기술은 트래픽 분류 분야에서도 중요한 역할을 하며, 네트워크 관리와 보안에 관련된 여러 측면에서 큰 잠재력을 가지고 있다. 딥러닝을 활용한 트래픽 분류는 기존의 통계적 방법과 비교하여 더 높은 정확도와 효율성을 제공할 수 있다. 딥러닝 모델은 데이터로부터 복잡한 패턴을 스스로 학습하며, 이를 기반으로 트래픽을 분류하는 데 사용된다. 이러한 모델은 대규모 데이터셋과 충분한 학습을 통해 네트워크 트래픽의 다양한 특징과 동작을 이해하고 식별할 수 있다. 또한, 딥러닝 기반 트래픽 분류는 다양한 네트워크 관리 및 보안 시나리오에서 적용된다. 예를 들어, 악성 코드나 DDoS 공격과 같은 보안 위협을 탐지하고 차단하는 데 사용될 수 있으며, 서비스 품질(QoS) 관리나 네트워크 최적화에도 활용된다.

데이터셋은 딥러닝 기반 트래픽 분류 연구에 있어서 핵심적인 구성 요소 중 하나로, 모델 학습 및 평가를 위한 핵심 자원이다. 트래픽 분류를 위한 데이터셋은 실제 네트워크 트래픽을 포함하고 있으며, 다양한 트래픽 클래스 또는 카테고리로 구분되어 있다. 현재 다양한 연구에서 높은 빈도로 사용되는 공개 데이터셋은 ISCX VPN 2016, ISCX TOR 2016, CIC-IDS-2018 등이 존재하는데, 이러한 공개 데이터셋의 대부분은 단일 호스트에서 수집된 데이터셋이다. 다중 호스트 데이터셋으로 발표된 데이터셋들도 다양한 호스트 PC에서 각각 하나의 응용 트래픽만을 수집한 것으로, 이 또한 단일 호스트 데이터셋에 포함된다. 딥러닝 모델이 이러한 단일 호스트 데이터셋을 학습하게 된다면, 응용 프로그램을 사용하는 사용자의 행동 패턴에 의해 트래픽을 분류할 가능성이 있으며, 이로 인한 일반화 성능의 저하로 실제 환경의 다양한 사용자 환경에서 제대로 분류하지 못할 수 있다. 따라서 이를 대체하기 위한 다중 호스트 데이터셋의 구축이 필요한 상황이다.

다중 호스트 데이터셋은 현재 이를 수집하기 위한 많은 시간과 자원으로 인해 제한점이 존재하고, 이러한 점 때문에 다중 호스트 데이터셋을 구축하는 방법에 대한 연구가 필요하다. 네트워크 트래픽 데이터셋을 구축하는 도구로는 현재 Wireshark, Microsoft Network Monitor 등 다양한 수집 도구가 존재하지만, 단순히 이러한 도구만으로는 다중 호스트 데이터셋 수집에 어려움이 있다. 예를 들어 Wireshark의 경우 각 응용 프로그램별로 수동으로 라벨링하여 수집해야 하고, Microsoft Network Monitor의 경우 데스크톱, 노트북 등의 다양한 종류의 호스트 PC에서 수집할 때 저장되는 트래픽의 프레임 구조가 달라서 이에 대한 복잡한 전처리 과정이 요구된다. 따라서 이러한 점들을 보완할 새로운 다중 호스트 데이터셋을 구축하는 시스템에 대한 연구가 필요한 상황이다.

본 논문은 단일 호스트 데이터셋과 다중 호스트 데이터셋으로 학습한 모델의 성능을 비교하며, 다중 호스트 데이터셋 구축을 위한 프로세스 로그 활용 수집 시스템을 제안하고, 이를 통해 다중 호스트 데이터셋을 활용한 딥러닝 기반 트래픽 분류기 모델의 발전에 기여하고자 한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 관련 연구에는 최신 연구 논문에서 발표된 단일 호스트 데이터셋과 다중 호스트 데이터셋의 비교 연구를 서술하고, 이를 보완할 방향에 대하여 논의한다. 본론에서는 다중 호스트 데이터셋을 효율적으로 구축하기 위해 제안하는 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템을 소개한다. 또한, 실험

을 통해 단일 호스트 데이터셋과 다중 호스트 데이터셋 각각으로 학습한 모델의 성능을 비교하여 다중 호스트 데이터셋과 그 수집 시스템의 필요성을 강조한다.결론에서는 이 연구의 기여와 한계점, 향후 연구과제를 제시하며 마무리를 맺는다.

Ⅱ. 관련 연구

Wei Wang 등은 CNN을 사용하여 악성 트래픽을 분류하는 방법을 제안하였다 [1]. Wei Wang 등은 벡터화한 트래픽 데이터를 이미지 분류 모델인 CNN에 입력하여 모델이 그 특징들을 학습함으로써 악성 트래픽을 높은 정확도로 분류할 수 있다고 보고하였다. 또한, 각각 1차원 CNN과 2차원 CNN을 사용하여 네트워크 트래픽 데이터를 이미지 형태로 변환한 후, 이를 통해 악성 트래픽 분류를 연구하였으며, 높은 정확도를 달성하였다 [2][3].

머신러닝과 딥러닝을 활용한 네트워크 트래픽 분류기를 연구하는 논문에서는 다양한 공개 데이터셋을 사용하여 분류기를 학습시킨다. Jingjing Zhao 등은 다양한 공개 데이터셋에 대한 조사를 통해 연구자들의 데이터셋 선정을 돕고자 하였다 [4]. 또한, 본 논문의 저자는 최근 연구를 통해 비교적 높은 빈도로 사용되는 UNSW-NB15, KDDCup99, NSL-KDD, CICIDS2017, ISCX VPN 2017, USTC - TFC2016, CMU-SynTraffic-2022의 7가지 공개 데이터셋에 대한 비교를 제시하였다 [5].

현재 트래픽을 수집하는 도구로 가장 널리 사용되는 도구는 Wireshark로 네트워크 패킷을 실시간으로 캡처하고 분석하는 도구이다. Sourit Singh Rajawat 등은 Wireshark를 사용하여 패킷 스니핑 도구를 설계하고 구현하는 방법에 대해 연구하였다. [6]. 또한, T. P. Fowdur 등은 다양한 네트워크 트래픽 캡처 도구의 성능을 분석하고, 여러 머신러닝 알고리즘을 사용하여 응용 프로그램, 상태 및 이상 상태를 분류하는 방법을 평가하였다 [7].

김태훈 등은 NIDS(Network Instrusion Detection System) 분야에서 단일 호스트 데이터셋과 다중 호스트 데이터셋으로 학습시킨 모델의 성능을 비교하였고, 각각의 특징을 혼합하여 최적의 NIDS 시스템을 연구하였다 [8].

표 1. 데이터셋 별 호스트 수
Table 1. Number of hosts of each datasets

데이터셋	연도	호스트 수
UNSW-NB15 [9]	2015	Multi-host
USTC-TFC2016 [10]	2016	1
ISCX VPN 2016 [11]	2016	2
ISCX TOR 2016 [12]	2016	1
CIC-IDS2017 [13]	2017	Multi-host
CSE-CIC-IDS2018 [14]	2018	Multi-host
CSTNET-TLS1.3 [15]	2021	Multi-host

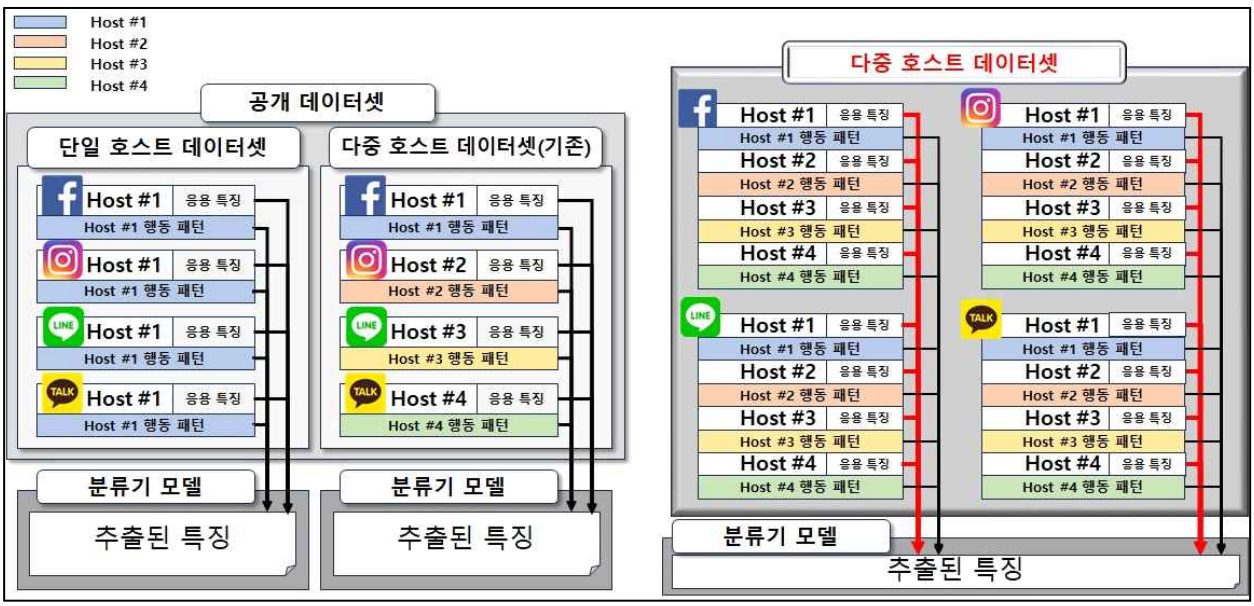


그림 1. 단일 호스트 데이터셋과 기존의 다중 호스트 데이터셋의 비교
Fig. 1. Comparison of single-host and multi-host datasets

Ⅲ. 본 론

3.1 다중 호스트 데이터셋의 필요성

딥러닝을 활용한 네트워크 트래픽 분류를 위해 많은 대학과 연구기관에서는 공개 데이터셋을 발표하였고, 단일 호스트 데이터셋인지, 다중 호스트 데이터셋인지를 추가로 명시하였다. 표 1에서 언급하는 데이터셋들은 최근 10년간의 연구에서 높은 빈도로 사용된 7가지 공개 데이터셋이고, 그 중 UNSW-NB15, CIC-IDS2017, CSE-CIC-IDS2018, CSTNET-TLS1.3 데이터셋은 모두 다중 호스트 데이터셋으로 발표되었다. 그림 1과 같이 이러한 기존의 다중 호스트 데이터셋의 개념은 각각의 응용에 여러 개의 수집 호스트 IP가 존재함을 의미하는 것이 아닌, 하나의 수집 호스트 IP마다 단 하나의 응용 트래픽만을 수집한 형태로 단일 호스트 데이터셋와 동일하게 취급될 수 있다. 기존의 단일 호스트 데이터셋과 다중 호스트 데이터셋으로 학습을 시키게 되면, 모델이 응용 자체의 특징과 더불어, 특정 호스트의 행동 패턴을 학습하게 되어 이에 과적합 될 가능성이 있다.또한, 하나의 호스트에서 수집한 응용 트래픽에 다양한 기능이 포함되지 않을 수 있으며, 이는 실제 환경에서 다양한 호스트의 기능 사용과 행동 패턴들을 분류할 트래픽 분류기를 학습시키기에 적합하지 않다. 4장의 실험에 의하면, 실제로 다중 호스트 데이터셋으로 학습된 모델을 단일 호스트 데이터셋 A,B로 평가할 때에 단일 호스트 데이터셋 A로 학습된 모델을 단일 호스트 데이터셋 B로 평가할 때보다 약 27-29% 더 높은 정확도가 확인되었다.

본 논문에서는 다중 호스트 데이터셋의 정의를 '각각의 서비스 유형, 응용 프로그램마다 여러 개의 다양한 수집 호스트 IP를 사용하여 수집한 데이터셋'로 설정하여 표현한다. 이를 통해, 다양한 호스트에서 발생하는 다양한 기능에 대한 응용 트래픽을 포함하도록 다중 호스트 데이터셋을 구축하여 네트워크 트래픽 분류기를 학습시키는 연구가 필요하다.

3.2 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템

다중 호스트 데이터셋을 구축하는 것은 많은 시간과 자원이 필요하다는 한계가 존재한다. 여러 대의 호스트 PC에서 특정 응용 트래픽만을 라벨링하여 저장하는 것은 Wireshark, Microsoft Network Monitor 등 기존의 수집 도구로 수집하기에 어려운 작업이다. 따라서 이를 효율적으로 저장하기 위한 다중 호스트 데이터셋 수집 도구나 시스템이 필요하다. 본 논문에서는 이를 해결하기 위해 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템을 제안하고, 이를 통해 다중 호스트 데이터셋 수집의 구축을 지원하고자 한다.

본 논문에서는 다양한 호스트 PC에서 발생하는 다양한 응용 트래픽 중 특정 응용에 대해서만 라벨링하여 수집할 수 있도록 프로세스 로그를 활용하여

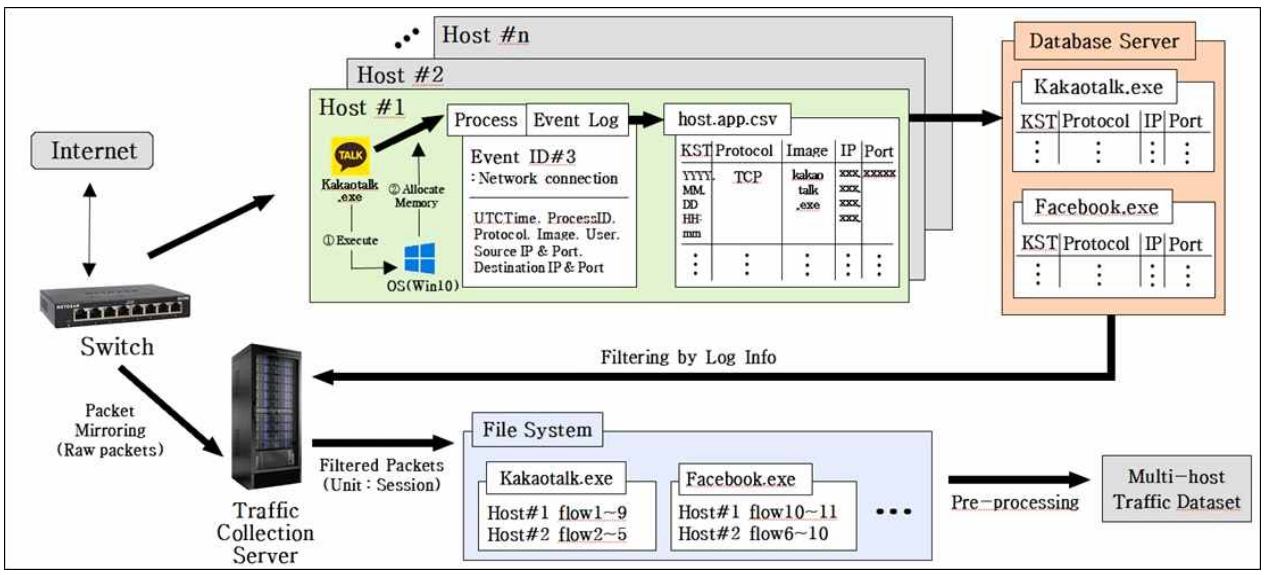


그림 2. 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템
Fig. 2. Number of hosts of each datasets

트래픽을 수집하는 시스템을 구축하였다. 그림 2는 제안된 수집 시스템의 구성도이다. 이 수집 시스템은 2가지의 환경 설정과 4단계의 수집 과정을 거친 뒤 전처리를 거쳐 다중 호스트 데이터셋을 구축한다.

(1) 환경설정 :

첫 번째로, 수집할 호스트 PC들을 연결하는 스위치 상에 연결된 트래픽 수집 서버가 필요하다. 이 수집 서버에서는 스위치에 연결된 다양한 호스트 PC에서 발생하는 트래픽을 패킷 미러링을 통하여 저장하게 된다. 미러링한 패킷을 저장할 때 .pcap 파일 형식으로 저장하게 되며, 1분마다 'pcap_YYYY_MM_DD_HH_MM.pcap'의 파일명을 생성하며, 해당 시간 동안 발생한 모든 트래픽을 모두 파일 내에 저장하게 된다. 이를 구현하기 위해, 파이썬의 Scapy 라이브러리를 활용하여 Flow Generator 프로그램을 개발하였다. 이 프로그램은 미러링한 패킷을 양방향 흐름(Bi-flows) 단위로 결합하여 각 분 단위의 파일에 저장한다.

두 번째로, 특정 응용 프로그램에 대한 네트워크 연결 로그를 저장하도록 하는 설정이 필요하다. 이 시스템에서는 각 호스트에서 발생하는 네트워크 연결 프로토콜에 대한 로그 정보를 저장하고, 이 로그 정보를 바탕으로 다량의 트래픽 중에서 특정 응용 프로그램에 대한 로그를 활용하여 필터링하므로, 네트워크 연결 정보를 저장하는 과정이 필수적이다. 이를 위해 Windows Event Viewer, Microsoft Sysmon, Log Parser 2.2, 총 3가지의 수집도구를 사용하였다. 먼저 Windows Event Viewer는 Windows 운영체제에서 모든 시스템, 보안 및 응용 프로그램 이벤트를 로깅하고 보여주는 관리 도구이며, Microsoft Sysmon과 결합한 형태로 사용하게 되면, 각 호스트 PC의 프로세스 별로 로그를 확인하는 것이 가능하다. 이 때, Sysmon의 설정파일 'sysmonconfig-export.xml'에서 네트워크 연결 정보를 의미하는 Event ID가 3인 항목에서 각종 응용 프로그램의 실행 파일 위치를 적용시키면, 해당 실행 파일이 프로세스로 등록되어 네트워크를 연결할 때 그 로그를 저장할 수 있게 된다. 또한, Log Parser 2.2의 경우, Sysmon으로 확인된 로그 정보를 다양한 형태로 파싱해주는 도구로, 각각의 호스트 PC에 .csv 파일 형식으로 저장하기 위해 사용되었다. 위와 같은 2가지 환경 설정을 마치게 되면, 다음의 4가지 수집 과정을 통해 트래픽을 수집하게 된다.

(2) 수집 과정 :

첫 번째로, 스위치를 통해 연결된 여러 대의 호스트 PC에서 'sysmonconfig-export.xml' 설정 파일에 적용된 다양한 응용 프로그램이 실행되면 .exe 파일 형식의 실행파일이 운영체제에 의해 메모리를 할당받아 프로세스가 생성된다. 생성된 프로세스 별로 Microsoft Sysmon을 통해 네트워크 연결 정보에 대한 로그를 저장하게 되며, 이는 Windows Event Viewer로 열람이 가능하다.

두 번째로, 주요 네트워크 연결 정보를 각 호스트 PC에 임시 저장한다. 이는 갑작스러운 통신 장애나 패킷 누락 시에 정상적으로 복구하는 데에 사용된다. 저장되는 주요 로그 정보는 UTC Time, Protocol, Image, IP, Port로 5가지이며, UTC Time은 한국 표준시로 변환되어 필터링할 대상 pcap파일을 선택할 때에 사용된다. Image은 해당 응용 프로그램명을 식별하여 자동으로 라벨링하기 위해 필요한 로그정보이며, IP, Port, Protocol 3가지 정보는 실질적으로 필터링 대상 pcap파일에서 해당 응용 프로그램에 대한 로그를 사용하여 필터링 할 때에 사용되는 주요 정보이다.

세 번째로, 위 과정으로 생성된 파일에 저장된 네트워크 연결 정보를 DB서버(DataBase Server)에 저장한다. 이 때, Image 정보에 의해 각각의 응용 프로그램명을 그 이름으로 하는 테이블에 분류되어 저장되게 된다. 예를 들어 카카오톡 응용 트래픽의 경우 Kakaotalk 테이블에, 페이스북 응용 트래픽의 경우 Facebook에 저장되는 것이다. 이러한 과정으로 인해, 이 시스템은 응용 프로그램별로 자동으로 라벨링하는 특징을 가진다. 저장된 네트워크 연결 로그들은 각 테이블 내에서 KST Time, IP, Port, Protocol 4가지 항목에 대한 정보를 저장하게 된다.

네 번째로, 트래픽 수집 서버에 미러링 되어 저장된 패킷들 중에서 KST Time정보와 일치하는 시간대의 파일은 필터링 대상 pcap파일이 된다. 예를 들어 저장된 로그의 KST Time 정보가 2024.06.08.11:29일 경우, pcap_2024_06_08_11_29.pcap 파일이 필터링 대상 pcap파일이 되며, 해당 파일로부터 IP, Port, Protocol 3가지 정보를 바탕으로 필터링 하게 되는 것이다. 해당 부분의 구현은 파이썬 프로그램을 개발하여 DB서버에 저장된 로그를 바탕으로 수집 서버에 있는 pcap파일들로부터 특정 응용 트래픽을 추출하였다.

(3) 전처리 단계 :

이 단계에서는 저장된 파일들을 전처리하여 불필요한 부분을 제거한다. 필요한 전처리 과정은 2가지로 VLAN(Virtual LAN) 헤더의 제거와 MTU(Maximum Transmission Unit)에 맞게 패킷을 재조립하는 것이다. 첫 번째로, 패킷 미러링을 하게 되는 과정에서 미러링된 패킷은 VLAN 헤더가 추가된다. 이 VLAN 헤더는 이더넷 헤더에 추가되는 4바이트 길이의 헤더로, TPID(Tag Protocol IDentifier), PCP (Priority Code Point) 등의 필드를 가지는 헤더이다. 이러한 VLAN 헤더가 붙는 모든 패킷들에 대해서, 헤더를 제거하는 전처리가 필요하다. 일반적으로 각각의 호스트 PC에서 수집된 패킷을 원형(Original)이라고 가정하였을 때, 이러한 원형 패킷에서는 VLAN 헤더가 추가되지 않는다. 따라서 이에 대한 제거를 통해 원형 패킷과 동일한 구조를 가지도록 패킷을 전처리하는 것이다. 두 번째로는, MTU에 맞게 패킷을 재조립하는 것이다. 일반적으로 호스트 PC는 기본적으로 1,500 bytes의 MTU를 가진다. MTU는 네트워크를 통해 전송될 수 있는 패킷의 최대 크기로, MTU 값이 너무 작으면 많은 패킷을 전송해야 하므로 오버헤드가 증가하고, 너무 큰 MTU는 패킷이 손실되었을 때 재전송해야 할 데이터의 양이 많아져 비효율적일 수 있다. 따라서, 별도로 설정하지 않는다면 MTU 값은 이더넷 표준에 의해 1,500 bytes로 고정되어있다. 이러한 MTU는 미러링하여 패킷을 저장하는 트래픽 수집 서버와 다를 수 있다. 예를 들어, 트래픽 수집 서버에서 수집될 때에는 3,014 bytes 크기로 저장된 패킷이 호스트 PC에서 수집될 때에는 1,514 bytes를 가진 2개의 패킷으로 분할되어 저장된다. 따라서 1,514 bytes가 넘는 크기의 패킷은 원본과 동일한 헤더를 페이로드 중간에 추가하여 2개의 패킷으로 인식되도록 재조립하는 과정이 필요하다. 이러한 두 가지 전처리 과정을 거치면, 이 다중 호스트 데이터셋 수집 시스템으로 통하여 수집한 응용 트래픽이 각각의 호스트 PC에서 수집한 응용 트래픽과 동일한 형식과 내용을 가지게 되고, 이를 통해 다중 호스트 데이터셋을 효율적으로 구축할 수 있다.

현재 약 500대의 호스트 PC를 연결하는 스위치 상에서 미러링하여 패킷을 수집하고 .pcap파일로 저장하는 시스템을 가동 중이며, 패킷이 누락되는지에 대한 검사를 반복적으로 수행하고 있다. 이러한 검사에서 누락된 패킷은 존재하지 않았고, 안정적으로 패킷을 모두 수집할 수 있는 것으로 확인되었다.

제시된 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템은 다음과 같은 장점을 가진다.

첫 번째로, 시스템을 통해 수집한 다중 호스트 데이터셋으로 학습된 모델은 기존 연구의 모델들과 비교하여 더 높은 일반화 능력을 가진다. 본 논문에서 제안된 다중 호스트 데이터셋 수집 시스템은 동시에 더 많은 호스트 PC에서 더 많은 응용 프로그램을 사용하면서 발생하는 응용 트래픽에 대한 수집이 가능하다. 기존의 수집 도구와 시스템은 다양한 응용 트래픽 수집을 위해 여러 대의 수집용 호스트 PC가 필요하기 때문에 많은 자원을 필요로 한다는 한계점이 존재하였지만, 제안된 수집 시스템은 한 대의 호스트 PC에서 동시에 사용되는 다양한 응용 프로그램에 대한 응용 트래픽을 분류하여 저장할 수 있으므로, 기존의 방법보다 효율적으로 자원을 사용할 수 있다. 또한, 필요에 따라 더 많은 호스트 PC에서 수집함으로써 더욱 다양한 사용자의 행동 패턴을 학습하도록 확장할 수 있다. 이것은 모델이 특정 사용자의 행동 패턴에 과적합되는 것이 아니라, 다양한 사용자의 행동 패턴에서의 특정 응용 트래픽에 대한 특징을 모델이 더 잘 학습할 수 있도록 도우므로, 제안된 시스템에 의해 일반화 능력을 향상시킬 수 있다.

두 번째로, 각각의 응용 트래픽에 대해서 자동으로 라벨링이 가능하다. Wireshark와 같은 기존의 수집 도구는 별도로 라벨링 과정이 필요하였지만, 제안된 시스템은 응용 프로그램이 메모리를 할당받고 프로세스로 등록되는 순간부터의 네트워크 연결 정보를 모두 분류하여 저장하게 된다. 따라서 이를 추가적인 수동 라벨링 과정에 대한 제약이 해소된다.

Ⅳ. 실험

4.1 실험 설계

본 논문에서는 제안하는 프로세스 로그 활용 다중 호스트 데이터셋 수집 시스템을 활용한 실험을 제공한다. 이 실험은 다중 호스트 데이터셋 수집 시스템의 필요성을 강조하기 위해 진행되었다. 동일한 모델을 단일 호스트 데이터셋과 다중 호스트 데이터셋으로 학습시키고 평가하며 성능에 대한 비교를 제공한다.

4.2 데이터셋

실험을 위해 제안된 다중 호스트 데이터셋 수집 시스템으로 4대의 호스트 PC에서 발생하는 트래픽 데이터를 수집하여 여러 개의 단일 호스트 데이터셋을 수집하였다. 호스트 A, 호스트 B, 호스트 C, 호스트 D 각각의 호스트 PC에서 발생한 Google Drive, Instagram, Kakaotalk, Riot Games, Steam, Youtube의 총 6개의 응용 트래픽에 대해 수집하고 각각을 A, B, C, D 데이터셋으로 구축하였다. 그리고, 이 4가지 데이터셋을 통합하여 다중 호스트데이터셋 KorUniv_Multi 데이터셋을 구축하였다. 수집할 때 동일한 기능을 처리할 수 있도록 동일한 시나리오로 수집되었다. 예를 들어, 카카오톡 응용 트래픽을 캡쳐하는 동안에는 친구 추가, 상태메세지 변경, 메세지 전송, 사진 전송, 동영상 전송 등의 시나리오를 적용하여 수집하였다. 그러나, 각 기능을 사용하는 방식과 세부적인 행동 패턴에 대해서는 제한하지 않았다. 예를 들어, 카카오톡 대화창을 계속해서 팝업시키고 다른 작업을 하는 패턴을 보이는 사용자가 있는 반면, 또 다른 사용자는 대화가 끝나자마자 대화창을 닫는다. 이는 각 사용자들의 행동 패턴이 각각의 데이터셋에 포함되도록 한 것인데, 같은 기능에 대해서 여러 가지의 사용자의 행동 패턴을 학습하도록 유도한 것이다. 구축된 데이터셋은 모두 .pcap 파일 형식으로 저장되었다.

4.3 전처리

수집된 A, B, C, D, KorUniv_Multi 데이터셋은 이더넷 헤더 14 bytes를 제거하고, IP 헤더, TCP or UDP 헤더와 Payload를 포함하도록 하는 JSON 파일 형식으로 변환되었다. 이더넷 헤더는 물리적 네트워크 레벨의 불필요한 정보를 담고 있으므로, 제거하여 데이터를 처리하는 효율성을 높이고자 하였다. 그 뒤 각 파일은 5-튜플 (출발지 주소, 출발지 포트, 목적지 주소, 목적지 포트, 프로토콜)을 기반으로 양방향 흐름으로 분할되고, 양방향 흐름으로 나뉜 각각의 파일을 초기 784 bytes만 사용하였다. 이 때 분할된 파일들 중 784 bytes을 초과하는 파일은 후반부를 제거하고, 784 bytes 미만인 파일은 제로 패딩(Zero-Padding)을 통하여 784 Bytes로 파일 크기를 통일하였다. 그 후 마지막으로 CNN의 입력인 텐서(Tensor) 형식으로 변환하여 .npy 파일로 저장한 뒤 학습에 사용할 데이터셋으로 저장하였다.

4.4 평가 지표

모델의 성능을 평가하는 지표로서 본 논문에서는 기존의 연구에서 사용되는 네 가지 평가 지표인 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1-score)를 활용하였다. 아래 수식 1, 2, 3, 4는 각각의 계산식이다. TP는 True Positive로 모델이 무언가를 긍정적으로 올바르게 분류한 경우이고, TN은 True Negative로 모델이 무언가를 부정적으로 올바르게 분류한 경우이다. FP는 False Positive로 모델이 부정적인 것을 긍정적으로 잘못 분류한 것이고, FN는 False Negative로 모델이 긍정적인 것을 부정적으로 잘못 분류한 경우이다.

$$Accuracy = \frac{TP + TN}{(TP + FN + FP + TN)} \quad (1)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

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

$$F1\,Score = \frac{2 \times Recall \times \mathrm{Precision}}{(Recall + \mathrm{Precision})} \quad (4)$$

4.5 실험 결과 및 분석

표 2는 다중 호스트 데이터셋 KorUniv_Multi와 단일 호스트 데이터셋 A, B을 활용한 실험을 통해 성능을 비교하는 실험의 결과이다. A모델을 평가 데이터셋으로 단일 호스트 데이터셋 A로 학습된 모델이 단일 호스트 데이터셋 B로 평가될 때와 단일 호스트 데이터셋 B로 학습된 모델이 단일 호스트 데이터셋 A로 평가될 때, 각각 0.4055, 0.4371로 매우 낮은 정확도를 보였다. 그러나, 다중 호스트 데이터셋인 KorUniv_Multi 데이터셋으로 학습된 모델이 A와 B 각각의 단일 호스트 데이터셋으로 평가했을 때에는 정확도가 각각 0.6957, 0.7029로 약 27-29%정도의 정확도 향상이 확인되었다. 위 실험에서 전처리, 모델, 파라미터는 모두 통제 변인에 해당한다. 그 효과를 확인하고자, 독립 변인으로 설

표 2. 단일 호스트 데이터셋으로 학습된 모델과 다중 호스트 데이터셋으로 학습된 모델의 교차검증을 통한 성능 비교
Table 2. Performance Comparison of Models Trained with Single-Host Dataset and Multi-Host Dataset through Cross-Validation

훈련(Train)	평가(Test)	정확도 (Accuracy)	F1-점수 (F1-score)	재현율 (Recall)	정밀도 (Precision)
KorUniv_Multi	KorUniv_Multi	0.646	0.6493	0.646	0.6858
KorUniv_Multi	A	0.7029	0.6998	0.7029	0.7486
KorUniv_Multi	B	0.6957	0.6963	0.6957	0.7183
A	A	0.8544	0.8448	0.8544	0.8407
A	B	0.4055	0.3084	0.4055	0.3238
B	A	0.4371	0.4301	0.4371	0.4741
B	B	0.7314	0.7239	0.7314	0.727

정한 것은, KorUniv_Multi 데이터셋이 다중 호스트 데이터셋이며, A와 B는 단일 호스트 데이터셋이라는 것이 유일하다. 따라서 다중 호스트 데이터셋으로 학습시킨 모델이 더 높은 분류 정확도를 보인 것은 모델이 다양한 사용자의 행동 패턴을 학습하기 때문인 것으로 추측할 수 있다. 이처럼 딥러닝 기반의 네트워크 트래픽 분류기 모델은 특정 사용자의 행위 패턴에 과적합 되지 않도록 다중 호스트 데이터셋을 학습하고, 평가하는 방향으로 향후의 연구가 이루어져야 한다.

V. 결 론

본 논문에서는 프로세스 로그 활용 다중 호스트 데이터셋 구축 시스템을 제안한다. 각 호스트 PC에서 실행되는 응용 프로그램에 의해 트래픽 수집 서버에는 다양한 호스트 PC에서 발생하는 다양한 응용 트래픽이 저장되고, DB서버에는 응용 프로그램별로 각각의 호스트에서 발생하는 응용 트래픽의 프로세스 로그 정보를 저장한다. 그 후, DB서버에 저장된 로그 정보를 통해 수집 서버의 응용 트래픽을 필터링한 뒤 자동으로 분류하여 저장한다. 이 시스템은 기존 수집 도구와 시스템보다 더 효율적으로 자원을 사용하며, 필요할 시 더 다양한 사용자의 행동을 포함시킴으로써 모델의 일반화 능력을 향상시키는 장점을 가진다. 또한, 각각의 응용 트래픽에 대하여 자동으로 라벨링하여 저장함으로써, 별도의 수동 라벨링이 필요한 기존의 수집 도구의 한계점을 극복하였다. 논문에서 제시한 실험의 결과에 의하면, 이 시스템으로 수집된 다중 호스트 데이터셋으로 학습된 모델은 단일 호스트 데이터셋으로 학습된 모델보다 27-29%의 정확도 향상을 보이며, 따라서 학습된 모델이 실제 환경에서 네트워크 트래픽 분류기 모델로 사용될 때 더 향상된 분류 성능을 보일 것으로 판단된다.

이 연구의 결과로 몇 가지의 향후 연구과제를 다음과 같이 제시한다. 첫 번째로, 수집과 필터링이 동시에 불가능하다. 이 시스템은 가동과 동시에 수집되지만, 필터링은 수집 이후 최소 3분 정도의 지연 후 실행 가능하다. 이는 스위치와 연결된 트래픽 수집 서버가 얼마나 많은 호스트 PC와 연결되어 있는지, 수집 서버의 성능 등 다양한 변수에 따라 달라진다. 따라서, 코드의 최적화 및 변형 등 수집과 필터링을 동시에 하기 위한 연구가 추가로 필요한 상황이다. 두 번째로, 캡슐화, 서비스 유형 및 다른 Task에 대한 자동 라벨링이 불가하다. 이 시스템에서 자동으로 라벨링이 가능한 Task는 응용 프로그램 별 분류이다. 그러나, 이것은 다양한 연구에서 진행된 캡슐화 여부에 따른 분류, 서비스 유형 분류에 대해서는 자동 라벨링이 불가능하다. 따라서 이러한 한계점에 대해서 추가적인 연구가 필요하다. 마지막으로, 모델이 트래픽의 어떠한 특징을 학습하는 지에 대한 연구가 필요하다. 논문에서 제시한 실험의 결과 분석은 단일 호스트 데이터셋으로 학습시킨 모델의 과적합의 가능성을 제시하고, 다중 호스트 데이터셋으로 학습시킬 때에 모델이 다양한 호스트의 행동 패턴을 학습하기 때문에, 향상된 결과를 보인다고 설명하였다. 그러나, 실질적으로 모델이 어떤 특징을 얼마나 학습하는지 등에 대해서는 정확하게 알 수 없기 때문에, XAI(Explainable Artificial Intelligence) 연구의 방향성과 같이 모델의 해석가능성에 대한 연구가 필요하다.

References

[1] Pathmaperuma, Madushi H., Yogachandran Rahu lamathavan, Safak Dogan, and Ahmet M. Kondo z. 2022. "Deep Learning for Encrypted Traffic C lassification and Unknown Data Detection" Sens ors 22, no. 19: 7643. (https://doi.org/10.3390/s22 197643)

[2] W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Ya ng, "End-to-end encrypted traffic classification with one-dimensional convolution neural networ ks," 2017 IEEE International Conference on Inte lligence and Security Informatics (ISI), Beijing, China, 2017, pp. 43-48, (https://doi.org/10.1109/I SI.2017.8004872)

[3] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng, "Malware traffic classifi cation using convolutional neural network for re presentation learning," 2017 International Confer ence on Information Networking (ICOIN), Da N ang, Vietnam, 2017, pp. 712-717, (https://doi.org /10.1109/ICOIN.2017.7899588)

[4] Jingjing Zhao, Xuyang Jing, Zheng Yan, Witold Pedrycz, "Network traffic classification for data fusion: A survey", Information Fusion, Volume 72, 2021, Pages 22-47, ISSN 1566-2535.(https:// doi.org/10.1016/j.inffus.2021.02.009)

[5] 장윤성, 박지태, 백의준, 김주성, 이대국, 김명 섭, "딥러닝 기반의 응용 프로그램 트래픽 분 류를 위한 데이터셋 사용 및 전처리 방법", K NOM Review, Vol.26, No.2, Dec., 2023, pp.33 -48. (https://doi.org/10.22670/knom.2023.26.2.33)

[6] Rajawat, Sourit & Khatri, Pallavi & Surange, Ge etanjali. (2023). Sniffit: A Packet Sniffing Tool Using Wireshark. (https://doi.org/10.1007/978-3 -031-43140-1_18)

[7] Fowdur, T. & Baulum, B. & Beeharry, Yogesh. (2020). Performance analysis of network traffic capture tools and machine learning algorithms f or the classification of applications, states and anomalies. International Journal of Information Technology. 12. (https://doi.org/10.1007/s41870 -020-00458-0)

[8] Kim, Taehoon & Pak, Wooguil. (2023). Integrate d Feature-Based Network Intrusion Detection S ystem Using Incremental Feature Generation. E lectronics. 12. 1657. (https://doi.org/10.3390/ele ctronics12071657)

[9] N. Moustafa and J. Slay, "UNSW-NB15: a comp rehensive data set for network intrusion detectio n systems (UNSW-NB15 network data set)," 20 15 Military Communications and Information S ystems Conference (MilCIS), Canberra, ACT, Australia, 2015, pp. 1-6, (https://doi.org/10.1109 /MilCIS.2015.7348942)

[10] USTC-TFC2016,https://github.com/yungshenglu/ USTC-TFC2016 (Accessed 14 December 2023)

[11] Gerard Drapper Gil, Arash Habibi Lashkari, Mo hammad Mamun, Ali A. Ghorbani, "Characteriz ation of Encrypted and VPN Traffic Using Tim e-Related Features", In Proceedings of the 2nd International Conference on Information System s Security and Privacy(ICISSP 2016) , pages 40 7-414, Rome, Italy, February 2016. (https://doi. org/10.5220/0005740704070414)

[12] Arash Habibi Lashkari, Gerard Draper-Gil, Moh ammad Saiful Islam Mamun and Ali A. Ghorba ni, "Characterization of Tor Traffic Using Time Based Features", In the proceeding of the 3rd I nternational Conference on Information System Security and Privacy, SCITEPRESS, Porto, Por tugal, 2017. (https://doi.org/10.5220/000610560 2530262)

[13] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New I ntrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (I CISSP), Portugal, January 2018. (https://doi.org/ 10.5220/0006639801080116)

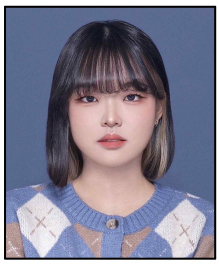
[14] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New I ntrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (I CISSP), Portugal, January 2018. (https://doi.org/ 10.5220/0006639801080116)

[15]CSTNET-TLS1.3, https://github.com/linwhitehat/ ET-BERT/tree/main/datasets/CSTNET-TLS%20 1.3 (Accessed 1 July 2024)

장 윤 성 (Yoon-Seong Jang)

2023년 : 고려대학교 컴퓨터융합소프트웨어학과 학사
2023년 ~ 현재 : 고려대학교 컴퓨터정보학과 석박사 통합과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

이 윤 서 (Yoon-Seo Lee)



2022년 ~ 현재 : 고려대학교 컴퓨터융합소프트웨어학과 학사 과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

최 유 진 (Yu-Jin Choi)

2021년 ~ 현재 : 고려대학교 컴퓨터융합소프트웨어학과 학사 과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

박 선 민 (Seon-Min Park)

2020년 ~ 현재 : 고려대학교 컴퓨터융합소프트웨어학과 학사 과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

김 주 성 (Ju-Sung Kim)

2023년 : 고려대학교 컴퓨터융합소프트웨어학과 학사
2023년 ~ 현재 : 고려대학교 컴퓨터정보학과 석박사 통합과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

백 의 준 (Ui-Jun Baek)

2018년 : 고려대학교 컴퓨터정보학과 학사
2018년 ~ 현재 : 고려대학교 컴퓨터정보학과 석박사 통합과정
<관심분야> 블록체인 거래 모니터링, 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

김 명 섭 (Myung-Sup Kim)

1998년 : 포항공과대학교 전자계산학과 학사
2000년 : 포항공과대학교 전자계산학과 석사
2004년 : 포항공과대학교 전자계산학과 박사
2006년 : Dept. of ECS, Univ of Toronto Canada
2006년 ~ 현재 : 고려대학교 컴퓨터정보학과 교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 단일미디어 네트워크