JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 5, pp. 1301-1313, May. 2024

Check for updates

기계학습 기반 클라우드 디지털 포렌식 적용 및 비교분석

신 현 욱[1] · 손 태 식[2*]
[1]아주대학교 사이버보안학과 학부과정
[2*]아주대학교 사이버보안학과 교수

Application and Comparative Analysis of Machine Learning-Based Cloud Digital Forensics

Hyenuk Shin[1] · Taeshik Shon[2*]

[1]Bachelor's Course, Department of Cyber Security, Ajou University, Suwon 16499, Korea
[2*]Professor, Department of Cyber Security, Ajou University, Suwon 16499, Korea

[요 약]

최근 클라우드 시장의 규모가 성장하면서 클라우드 환경에서의 디지털 포렌식이 주목 받고 있다. 그러나 클라우드 환경에서의 디지털 포렌식은 기존의 환경과는 다르게 데이터의 분산성, 멀티테넌시 환경, 데이터 수집의 법적인 문제 등과 같은 제약사항들이 존재한다. 본 논문에서는 전통적인 디지털 포렌식과 클라우드 환경에서의 디지털 포렌식의 차이를 분석하고 아티팩트를 수집하여 분석을 진행한다. 클라우드 환경에서 디지털 포렌식을 위한 아티팩트를 수집하기 위해 KVM(Kernel- based Virtual Machine)를 이용하여 하이퍼바이저 레벨에서 가상 환경을 구축하고 libvirt API를 사용해 디지털 포렌식에 적절한 아티팩트를 수집하였다. 수집한 아티팩트는 매우 방대한 메타 데이터를 포함하고 있으며 기존의 아티팩트들과는 차이점이 존재하기 때문에 효율적인 분석을 위해 기계학습을 이용한 분석을 진행한다. 이를 통해 아티팩트의 특성을 이해할 수 있으며 학습된 데이터를 사용하여 디지털 포렌식을 효율적으로 진행할 수 있다.

[Abstract]

With the cloud market continuing to grow, digital forensics within cloud environments is drawing attention. However, digital forensics in cloud settings differs significantly from traditional contexts due to challenges such as data dispersion, multi-tenancy, and legal complexities related to data collection. This paper analyzes the differences between conventional digital forensics and those practiced in cloud environments, focusing on the collection and analysis of artifacts. To facilitate this, a virtual environment was established at the hypervisor level using a kernel-based virtual machine (KVM), collecting artifacts suitable for digital forensics using the libvirt API. These artifacts, containing extensive metadata and differing from traditional ones, were analyzed using machine learning to enhance understanding of their characteristics and improve the efficiency of forensic investigations using trained data.

색인어 : 클라우드 환경, 디지털 포렌식, KVM, 하이퍼바이저, 기계학습

Keyword : Cloud Environments, Digital Forensics, Kernel-Based Virtual Machine, Hypervisor, Machine Learning

http://dx.doi.org/10.9728/dcs.2024.25.5.1301

Received 27 March 2024; **Revised** 16 April 2024
Accepted 10 May 2024

***Corresponding Author; Taeshik Shon**

Tel: +82-31-219-3321
E-mail: tsshon@ajou.ac.kr

Ⅰ. 서 론

최근 몇 년 동안 IT분야에서 클라우드 컴퓨팅 기술이 발전하면서 데이터 통신 영역의 핵심으로 자리잡았다. 이러한 클라우드 컴퓨팅 기술의 발전은 데이터 저장, 처리 및 접근 방식에서 많은 변화를 일으키며 그림 1과 같이 클라우드 관련 시장의 규모가 매우 커지는 결과로 이어졌다[1]. 클라우드를 통해서 처리되는 디지털 데이터의 규모가 증가하게 되면서 클라우드 환경에서의 데이터 보안과 개인정보 보호의 중요성도 증가했다. 이는 곧 디지털 포렌식의 영역이 클라우드 컴퓨팅 환경까지 확장된다는 것을 의미한다. 디지털 포렌식은 범죄 수사, 법적 분쟁 해결, 보안사고 대응 등에서 중요한 역할을 한다. 전통적인 디지털 포렌식 방법에서는 주로 물리적인 하드웨어, 로컬 네트워크, 개별 컴퓨터 시스템에서의 물리적인 접근과 시스템 이벤트 로그 분석 등의 방법을 이용하여 디지털 포렌식을 진행한다. 그러나 클라우드 컴퓨팅 환경에서의 디지털 포렌식은 클라우드 서비스 제공자의 인프라, 멀티테넌시(Multitenancy) 환경, 데이터 소유권, 접근 권한, 법적 제약 사항 등 여러 가지 요소들을 모두 고려해야 하며 물리적인 접근이 불가능한 경우도 존재한다[2]. 디지털 포렌식을 수행하기 위해서는 증거 데이터를 확보하고 수집하는 것이 매우 중요한데 기존의 전통적인 디지털 포렌식과는 다르게 클라우드 환경에서의 포렌식은 증거를 얻는 위치가 물리적, 논리적으로 매우 다양하며, 여러 사용자들이 동일한 물리적 인프라를 공유하는 멀티테넌시 환경이기 때문에 증거 수집에 많은 어려움이 존재한다. 또한 클라우드를 이용하는 사용자가 속한 국가와 정보를 저장한 매체가 존재하는 국가 사이에서의 관할권의 충돌이라는 법적인 문제가 발생한다[3]. 이러한 관할권의 충돌에 대응하는 법적인 절차가 국가마다 다르다는 점 역시도 클라우드 환경에서 증거 수집의 어려운 점이다. 또한 클라우드 환경에서 수집되는 아티팩트들은 매우 방대하기 때문에 이러한 데이터셋에서 유의미한 정보를 분석하는 것은 매우 어렵다.

따라서 본 논문에서는 클라우드 환경에서 디지털포렌식을 수행 하기 위한 아티팩트 수집분석과 이러한 아티팩트를 기계학습 알고리즘을 이용한 분석을 통해서 효율적인 디지털포렌식을 수행하는 방법을 연구한다. 이를 위해 가상머신 내에 실제 클라우드와 유사한 환경을 구축하여 디지털포렌식에 적합한 아티팩트를 수집하였다. 사용한 데이터셋은 KVM(Kernel-based Virtual Machine)으로 하이퍼바이저 레벨에서 libvirt API를 사용하여 클라우드 환경에서 발생하는 이벤트의 정상과 비정상 아티팩트를 수집하였다[4]. 수집한 아티팩트들은 매우 방대한 양의 데이터를 포함하고 있기에 적절한 분석을 위해 기계학습을 활용한 분석을 진행하였다.

본 논문은 2장에서 클라우드 컴퓨팅 환경의 배경지식 및 관련 연구를 통해서 기존 환경과 클라우드 환경에서의 디지털포렌식의 차이점을 구체화한다. 3장에서는 클라우드 환경에서의 포렌식에서의 고려사항을 살펴보고 기존의 환경과 클라우드 환경에서 수집하는 아티팩트의 차이점을 분석하며, 아티팩트 수집에 적합한 클라우드 환경을 구축한다. 4장에서는 연구에 사용할 데이터셋을 디지털 포렌식의 관점에서 분석을 진행한다. 5장에서는 디지털 포렌식에 사용될 기계학습 알고리즘을 살펴보며 본 연구에 적절한 알고리즘을 선정하는 과정을 거친다. 6장에서는 전처리 과정과 분석을 통해서 모델을 학습시키며, 성능평가를 진행하여 본 연구의 의의를 살펴본다. 마지막으로 7장에서는 결론에 대해 서술하였다.

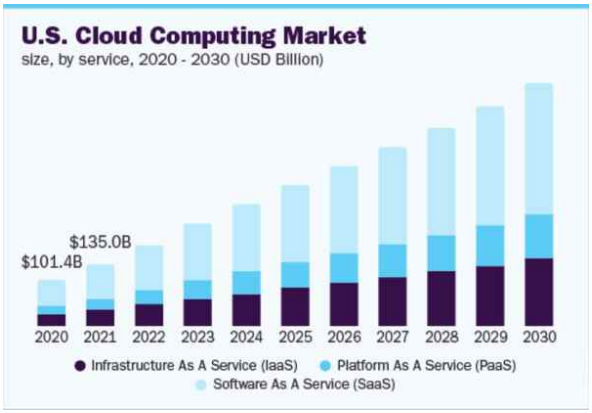


그림 1. 클라우드 컴퓨팅 시장 규모
Fig. 1. Cloud computing market size

Ⅱ. 배경 지식 및 관련 연구

2-1 배경 지식

1) 클라우드 컴퓨팅 환경

클라우드 컴퓨팅 환경이란 인터넷을 통해서 컴퓨팅 리소스를 제공하는 기술로, 사용자는 인터넷을 통해 IT 리소스를 제공받고 사용한 만큼의 비용을 지불하는 것을 의미한다[5]. 이러한 클라우드 컴퓨팅에는 제공하는 IT 리소스의 종류에 따라 IaaS(Infrastructure as a Service), PaaS(Platform as a Service), SaaS(Software as a Service) 등 다양한 서비스가 존재한다. 클라우드 컴퓨팅 기술을 활용하기 위해서는

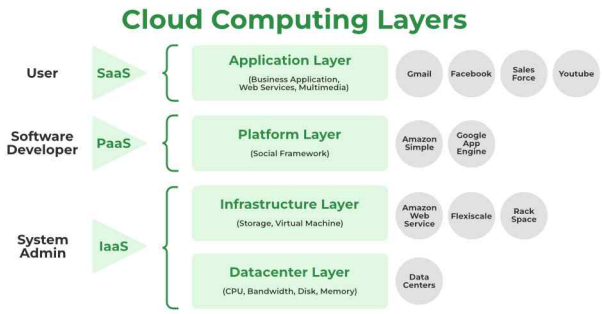


그림 2. 클라우드 계층 구조
Fig. 2. Layered architecture of cloud

사용자의 용도에 따라 적절한 클라우드 컴퓨팅 서비스를 선택하는 것이 중요하다. 그림 2는 클라우드 컴퓨팅의 계층화된 아키텍처를 설명한다[6]. 클라우드 아키텍처는 그림 2와 같이 네 개의 주요 계층으로 구성된다. 클라우드 서비스의 최상단 계층인 Application Layer에서는 사용자에게 직접적으로 서비스를 제공하며 SaaS가 해당 계층에 포함된다. Platform Layer에서는 공급자가 제공하는 플랫폼을 이용하여 애플리케이션과 서비스를 개발하는 영역으로 PaaS가 해당 계층에 포함된다. Infrastructre Layer에서는 사용자가 가상머신, 네트워크 등의 IT 리소스를 할당 받아 사용하는 계층으로 IaaS가 해당 계층에 포함된다. 클라우드 아키텍처의 최하단 계층으로는 Datacenter Layer로 실제 제공하는 클라우드 컴퓨팅의 하드웨어 및 리소스를 포함한다. 해당 계층은 클라우드 서비스의 인프라를 형성하는 기본 요소이다.

이러한 클라우드 환경에서는 리소스들이 여러 서버와 지역에 걸쳐 분산되어 있는 데이터의 분산성, 여러 사용자들이 접근하여 수정하는 멀티테넌시 환경, 법적인 문제 등의 문제로 인하여 기존의 방식과 같은 데이터 수집이 어렵다. 또한 클라우드 환경은 데이터를 수집할 때 데이터의 무결성을 어떻게 고려할 것인지에 대한 문제도 존재한다. 클라우드 서비스는 사용자의 요구에 따라 리소스를 유동적으로 할당하고 조정하기 때문에 이러한 문제는 더욱 부각된다[7],[8].

2) 전통적인 디지털 포렌식 방법론

전통적인 디지털 포렌식 방법론은 기기 자체 메모리에서 내부 데이터와 네트워크 통신 데이터를 중심으로 포렌식 분석 과정을 거친다.

정준호 등 1명은 IoT 기기의 플래시 메모리 데이터 추출을 통해 디지털 포렌식을 진행했다[9]. 직접 플래시 메모리를 디솔더링하고, 덤프한 바이너리 파일로부터 ext4 파일시스템을 추출한 뒤 마운트하는 과정을 통해서 포렌식을 진행하였다. 해당 연구는 실제 IoT 기기에서 플래시 메모리를 추출하여 해당 메모리에서 아티팩트를 수집했다.

김민주는 네트워크 기반 스마트 홈 기기 포렌식 기법과 관련된 연구를 진행했다[10]. 해당 연구는 공격자가 스마트 홈 IoT 기기에서의 TLS 통신을 가로채 데이터를 유출하는 것을 막기 위해서 먼저 스마트 홈 IoT 네트워크 통신을 분석을 통해 스마트 홈 IoT 에코시스템을 구성하여 네트워크 스니핑과 MITM 공격을 이용해서 네트워크 패킷 데이터를 수집하여 포렌식을 진행하였다. 해당 연구에서는 IoT 기기와 클라우드 서버와의 통신에서 Transport Layer Security(TLS)를 사용하여 통신하는 사례 위주로 연구를 진행했다.

3) 클라우드 환경에서의 디지털 포렌식 방법론

클라우드 환경에서의 디지털 포렌식은 최근에서야 관련 연구가 많아지고 있으나, 클라우드 컴퓨팅 환경 특성 상 정보 수집 계층, 각종 제약 조건 등으로 인하여 지속적인 연구가 필요하다. 본 장에서는 클라우드 환경에서의 디지털 포렌식과 관련된 여러 연구를 소개한다.

Liwen Peng 등은 데이터 마이닝 기술을 이용하여 클라우드 환경에서의 디지털 포렌식을 진행하였다[11]. 클라우드 컴퓨팅 환경에서의 데이터들은 다양한 서버 주소에 분산되어 있기 때문에 데이터 수집에 어려움이 존재한다. 해당 연구에서는 클러스터링 알고리즘(Clustering Algorithm)을 이용하여 증거가 될 수 있는 아티팩트를 수집하였다. 그러나 해당 연구는 데이터 마이닝 기술을 이용해 데이터를 수집하고 분석하여 증거로 활용될 수 있도록 라벨링을 하는 과정만을 진행하였다. 본 연구에서는 한 발 더 나아가 추가적인 패턴을 학습하여 분석하고, 클라우드 환경에서 사용 가능한 일반적인 기계학습 알고리즘을 개발을 목표로 한다.

Prasad Purnaye 등은 클라우드 환경에서 가상화 기술을 이용한 포렌식 방법을 제시한다[12]. 클라우드 컴퓨팅 환경은 멀티테넌트 환경으로 인하여 증거 수집을 하는데 있어서 어려움이 존재한다. 해당 연구는 가상화 환경에서 하이퍼바이저에서의 API를 이용하여 아티팩트를 수집한다. 이때 'ALmaNebula'라는 하이퍼바이저의 API를 이용하여 가상 머신 로그에 접근하여 아티팩트를 수집한다. 그림 3은 하이퍼바이저 수준에서 아티팩트를 수집하는 과정을 보여준다. 하이퍼바이저 수준에서 아티팩트를 수집하면 가상 머신 내부에 에이전트를 설치할 필요 없이 아티팩트를 수집할 수 있기 때문에 아티팩트를 수집하는 과정이 시스템에 주는 영향이 최소화된다.

Ezz EL-Din Hemdan 등은 가상 머신의 스냅샷을 주기적으로 캡처하고, 이를 TGS에 저장하는 방식을 이용하여 증거를 수집하는 새로운 클라우드 포렌식 조사 모델(CFIM)을 제시한다[13]. CFIM(Cloud Forensics Investigation Model)이란 가상머신의 스냅샷을 이용하여 VM의 상태를 기록한 데이터를 포렌식 서버를 이용하여 디지털 증거를 분석하는 것을 의미한다. 이때 클라우드 리소스를 사용하여 디지털 조사를 가능하게 하는 FaaS를 사용한다. 가상머신은 VMware을 이용하여 ESXi-5서버와 포렌식 서버를 구축하여 CPU 및 네트워크 성능을 모니터링 한 후 AccessDataFTK 툴킷, Encase, Autopsy와 같은 도구를 사용하여 스냅샷을 분석하여 포렌식을 진행한다.

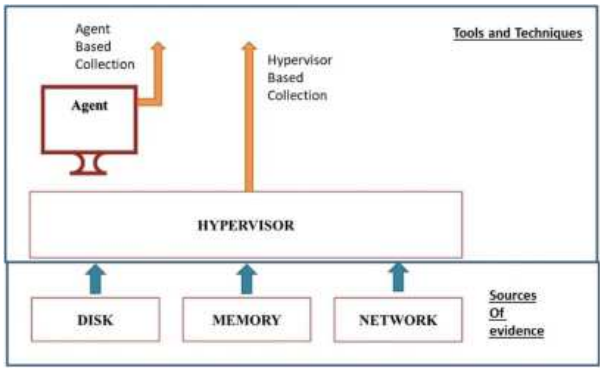


그림 3. 증거 수집 과정
Fig. 3. The process of collecting evidence

Ⅲ. 클라우드 환경을 위한 포렌식 방안

3-1 클라우드 환경에서의 포렌식 고려사항

본 논문에서는 클라우드 환경에서 아티팩트를 수집하기 위해 BiSHM(Binary Smart Hypervisor Monitoring)시스템을 통해 하이퍼바이저 레벨에서 생성된 데이터를 모니터링하여 AI 에이전트를 사용한 증거 탐지 및 보존에 대한 새로운 접근 방식을 사용했다. 하이퍼바이저란 가상머신(VM)을 생성하고 구동하는 소프트웨어로써 할당되는 리소스를 각 가상 머신에 제공하고 리소스를 관리하여 가상화를 수행하는 소프트웨어이다[14]. BiSHM시스템은 클라우드에서 생성된 데이터에서 증거 데이터를 구별하여 포렌식을 수행하기 위해 커널 레벨에서 API를 이용하여 데이터를 수집한다. 본 논문에서는 클라우드 환경에서의 문제점과 더불어, 접근 권한, 개인정보 보호, 프라이버시 등을 모두 고려하여 IaaS의 Infrastructure 계층의 하이퍼바이저 레벨에서 모니터링을 수행하는 BiSHM시스템을 사용했다. 해당 계층에서 모니터링을 수행하면 사용자는 가상머신의 내부에 직접 접근하지 않고도 필요한 데이터와 증거를 수집할 수 있다.

클라우드 환경에서는 2장에서 설명했듯이 구조적인 문제로 인하여 아티팩트 수집이 어렵다. 데이터의 분산성, 멀티테넌시 환경 등의 구조적인 문제가 존재한다. BiSHM 시스템은 가상머신의 하이퍼바이저 레벨에서 아티팩트를 수집하는데 이것은 가상화 기술을 사용하는 off-premise 환경인 클라우드 환경과 매우 유사하게 아티팩트를 수집할 수 있다. BiSHM 시스템은 가상머신의 내부에 직접 접근하지 않고, 하이퍼바이저 레벨에서 모니터링을 수행하며 여러 가상머신을 관리하기 때문에 클라우드 환경의 멀티테넌시 환경 및 데이터의 분산성의 성질을 모두 고려한 아티팩트 수집이 가능하다. 이러한 점을 토대로 클라우드 환경에서의 디지털포렌식에 유의미한 결과가 나올것으로 예상된다.

본 논문에서의 증거수집은 하이퍼바이저 레벨에서 데이터를 수집하기 위해 KVM(Kernel Virtual Machine)에서 libvirt API를 통해 증거를 수집했다. KVM이란 리눅스 커널 기반의 오픈 소스 가상화 기술로 가상머신과 그 내부에서 발생하는 이벤트들을 모니터링 하는 하이퍼바이저 역할을 수행하며 libvirt API는 이러한 가상 환경의 관리를 적절하게 하기 위한 인터페이스를 제공한다. libvirt API를 사용하면 가상머신의 스냅샷을 캡쳐하고, 실시간으로 메모리를 덤프하여 현재 실행 중인 프로세스와 상태의 정확한 아티팩트를 수집할 수 있다. 이는 클라우드 환경에서 서비스 연속성을 유지하면서 필요한 아티팩트를 확보하는 데 필수적이다. 그림 4는 하이퍼바이저 레벨에서 libvirt API를 사용하여 아티팩트를 수집하는 원리를 나타내는 그림이다. KVM은 리눅스 호스트 위에서 하이퍼바이저의 역할을 수행하며 개별 도메인 가상머신을 통해서 가상 인스턴스를 생성하고 관리한다. libvirt API는 관리 응용 프로그램(Mgmt app)를 통해 하이퍼바이저 레벨에서 인스턴스를 수집한다[15]. libvirt API는 가상머신을 직접적으로 관리하고 제어하므로 이는 가상화 환경을 이용하는 클라우드 환경에서도 동일하게 적용 가능하다. 즉 본 논문에서는 KVM을 이용하여 클라우드 환경과 유사한 도메인 가상머신을 구축하여 libvirt API를 이용하여 도메인 가상머신에서 아티팩트를 수집하는 것으로 클라우드 환경에서의 디지털 포렌식을 위한 아티팩트를 수집한다.

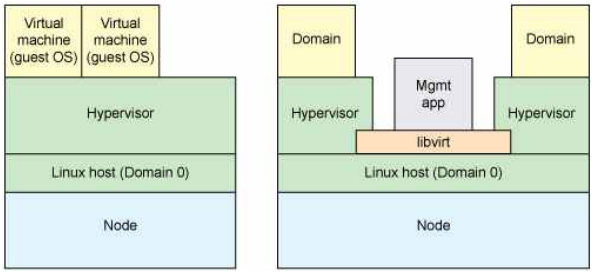


그림 4. KVM 가상화 구조
Fig. 4. KVM virtualization architecture

3-2 기존의 아티팩트 수집분석과의 차이점

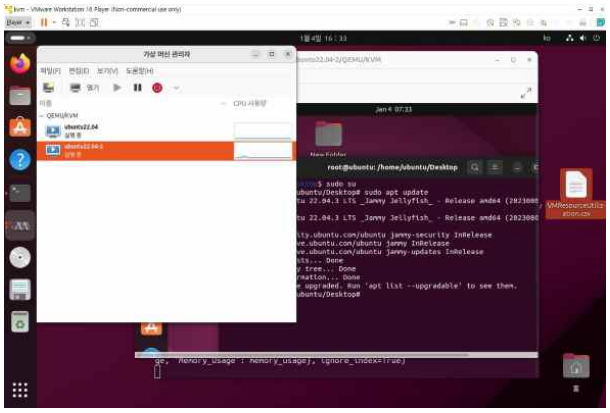
클라우드 환경에서 아티팩트를 수집하는 것은 기존의 디지털 포렌식을 하기 위한 아티팩트 수집분석과 차이점이 존재한다. 기존환경에서는 파일 시스템 데이터, 삭제된 파일, 시스템 로그 파일, 메모리 덤프를 통해 수집된 데이터, 실행 중인 프로그램과 프로세스에 대한 정보, 라우터나 스위치의 로그 파일, 패킷 캡처 데이터 등과 같은 네트워크 장비등에서 직접 아티팩트를 수집하여 분석했다. 클라우드 환경에서는 클라우드 환경을 이루고 있는 가상화 서버의 스냅샷, 설정 파일, 로그 등과 같은 메타데이터, 클라우드 스토리지 로그, 사용자 접근 로그, 네트워크 트래픽 데이터, 메모리 아티팩트, 디스크 아티팩트 등 기존 환경보다 더 복잡하고 포괄적인 아티팩트를 수집해야 한다. 그러나 앞서 설명했듯이 클라우드 환경의 구조적인 문제로 인하여 아티팩트 수집의 어려움이 존재하며, KVM과 livirt API를 이용하여 수집한 아티팩트는 매우 방대한 양의 데이터를 포함하고 있으며 이러한 아티팩트간의 상관관계를 종합적으로 분석하는 것은 매우 어렵다. 수집한 아티팩트에 기계학습을 활용하여 분석하면 기존의 환경보다 더욱 복잡하고 방대한 아티팩트를 다차원적인 분석을 통해 클라우드 환경의 디지털 포렌식에서 필요한 아티팩트을 명확히 파악할 수 있다. 따라서 본 논문에서는 수집한 아티팩트를 토대로 추가적인 기계학습을 진행하여 클라우드 환경에서의 디지털 포렌식을 위한 적절한 방안을 연구한다.

3-3 데이터 수집을 위한 클라우드 환경 구축

KVM을 이용하여 하이퍼바이저 레벨에서 아티팩트를 수집하기 위해서는 하이퍼바이저 레벨에서 모니터링을 수행해야 한다. 본 장에서는 VMware를 통해서 KVM을 통해 아티팩트

를 수집하기 위한 클라우드 환경 구축과정을 설명한다. 가상머신으로는 Ubuntu 22.04.3 LTS를 호스트 운영체제로 사용한다. KVM을 통해 클라우드 환경을 구축하기 위해서는 먼저 호스트 시스템에 KVM을 설치하고, 필요한 네트워크 설정과 스토리지 리소스를 구성한다. KVM을 가상화 플랫폼으로 사용하여 아티팩트를 수집하기 위한 도메인 가상 머신을 생성한다. 생성된 도메인 가상 머신은 실제 네트워크 환경과 유사한 시뮬레이션을 위해 호스트 가상 머신과 같은 물리적 네트워크에 연결된 것처럼 통신할 수 있도록 하는 브리지 네트워킹을 이용하여 네트워크 설정을 구성한다. 그림 5는 KVM을 이용하여 여러 개의 도메인 가상머신을 모니터링 하고 있는 모습을 나타낸다. 이는 클라우드 환경에서의 멀티 테넌시 환경과 매우 유사하다.

구축한 클라우드 환경에서 아티팩트를 수집하기 위한 알고리즘은 그림 6과 같다. QEMU 하이퍼바이저와 연결을 설정한 후 클라우드 환경을 구축한 가상머신에서 마지막 폴링 시간을 기록한 LAST_POLL, 가상머신의 네트워크 인터페이스 상태, CPU 사용 상태, 메모리 사용 상태, 하드 디스크에 대한 블록 상태, 가상 디스크에 대한 블록 디바이스 상태 등을 수

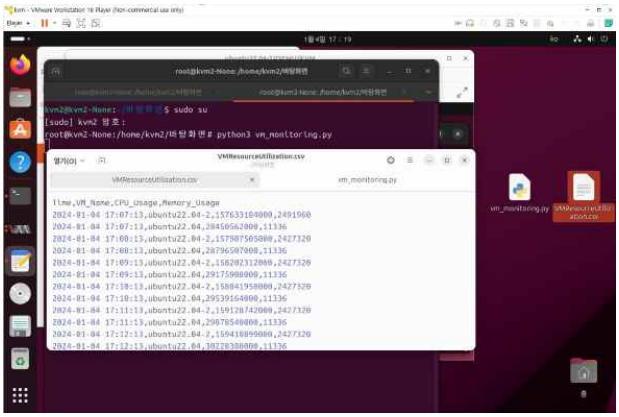


*Due to the operating system being in Korean, some text is inevitably included in Korean.

그림 5. KVM를 이용하여 도메인 가상머신 모니터링
Fig. 5. KVM domain virtual machines by KVM

Algorithm 1 KVM Monitoring Algorithm
Require: QEMU Hypervisor connection established
Ensure: MonitoringDatabase updated
1: **for each** VM deployed on the cloud **do**
2: find devices of VM;
3: get_LAST_POLL_value();
4: get_interfaces_stats();
5: get_CPU_stats();
6: get_memory_stats();
7: get_blocks_stats(hda);
8: get_blocks_stats(vda);
9: store_data_VMResourceUtilization();
10: **end for**

그림 6. KVM 모니터링 알고리즘
Fig. 6. KVM monitoring algorithm



*Due to the operating system being in Korean, some text is inevitably included in Korean.

그림 7. libvirt API를 이용하여 아티팩트 수집
Fig. 7. Collecting artifacts using libvirt API

집하며 수집한 아티팩트들을 VMResourceUtilization에 저장하는 과정을 거친다.

본 논문에서는 KVM 하이퍼바이저를 사용하여 모니터링 하고 있는 도메인 가상머신에서 아티팩트를 수집하기 위해서 호스트 가상머신에서 libvirt API를 이용하였다. 메타 데이터, 네트워크 데이터, 메모리 데이터, 디스크 데이터 등의 아티팩트를 수집하였다. 아티팩트 수집은 파이썬을 이용한 아티팩트 스크립트를 호스트 가상머신의 터미널에서 실행하여 모니터링 하고 있는 도메인 가상머신의 데이터를 수집한다.

Ⅳ. 데이터 세트

4-1 데이터 세트 정보

본 논문에서의 목적은 클라우드 환경에서의 디지털 포렌식을 진행하는 것이므로 정상 데이터와 비정상 데이터를 수집하였다. 비정상 데이터를 판단은 다변량 통계적 변화 추적 방법을 이용하여 하이퍼바이저로부터 네트워크 트래픽, 시스템 로그, 메모리 사용 데이터 등을 통해 판단한다[16]. 이러한 하이퍼바이저 기반의 IDS를 이용하면 디지털 포렌식에 사용 가능한 정상 데이터와 비정상 데이터를 수집할 수 있다. 총 수집한 데이터의 샘플수는 9610개이며, 피쳐는 총 43개로 피쳐에 대한 자세한 정보는 표 1에 나와있다. 정상 데이터는 약 7300개이고 비정상 데이터는 약 2310개다. 그림 8은 데이터의 일부분을 나타내는 예시 이미지이다.

디지털 포렌식을 수행하기 위해서는 정상 데이터와 비정상 데이터의 차이가 구별되는 피쳐를 아티팩트로 사용하는 것이 매우 중요하다. 아티팩트는 도메인 가상 머신의 정보를 나타내는 메타데이터 아티팩트, 네트워크 트래픽 아티팩트, 메모리 아티팩트, 하드 디스크와 가상 디스크에서의 아티팩트를 수집하였다. 이때 수집한 아티팩트 중에서 대부분의 수치가 0

표 1. KVM 데이터셋의 피쳐값
Table. 1. Category and description of features of KVM dataset

Num	Category	Feature	Description
1	Meta-data	LAST_POLL	Epoch timestamp
2		VMID	The ID of the VM
3		UUID	Unique identifier of the domain
4		DOM	Domain name
5	Network	Rxbytes	Received bytes from the network
6		rxpackets	Received packets from the network
7		rxerrors	Number of received errors from the network
8		rxdrops	Number of received packets dropped from the network
9		txbytes	Transmitted bytes from the network
10		txpackets	Transmitted packets from the network
11		txerrors	Number of transmission errors from the network
12		txdrops	Number of transmitted packets dropped from the network
13	Memory	timecpu	Time spent by vCPU threads executing guest code
14		timesys	Time spent in kernel space
15		timeusr	Time spent in userspace
16		state	Running state
17		memmax	Maximum memory in kilobytes
18		mem	Memory used in kilobytes
19		cpus	Number of virtual CPUs
20		cputime	CPU time used in nanoseconds
21		memactual	Current balloon value (in KiB)
22		Memswap_in	The amount of data read from swap space (in KiB)
23		Memswap_out	The amount of memory written out to swap space (in KiB)
24		Memmajor_fault	The number of page faults where disk IO was required
25		Memminor_fault	The number of other page faults
26		memunused	The amount of memory left unused by the system (in KiB)
27		memavailable	The amount of usable memory as seen by the domain (in KiB)
28		memusalbe	The amount of memory that can be reclaimed by balloon without causing host swapping (in KiB)
29		Memlast_update	The timestamp of the last update of statistics (in seconds)
30		Memdist_cache	The amount of memory that can be reclaimed without additional I/O, typically disk caches (in KiB)
31		Memhugetlb_pgalloc	The number of successful huge page allocations initiated from within the domain
32		Memhugetlb_pgfail	The number of failed huge page allocations initiated from within the domain
33		memrss	Resident set size of the running domain's process (in KiB)
34	Disk	Vdard_req	Number of read-requests on the vda block device
35		Vdard_bytes	Number of read-bytes on the vda block device
36		Vdawr_reqs	Number of write requests on the vda block device
37		Vdawr_bytes	Number of write requests on vda the block device
38		vdaerror	Number of errors in the vda block device
39		Hdard_req	Number of read requests on the hda block device
40		Hdard_bytes	Number of read bytes on the had block device
41		Hdard_reqs	Number of write requests on the hda block device
42		Hdawr_bytes	Number of write bytes on the hda block device
43		hdaerror	Number of errors in the hda block device

	LAST_POLL	VMID	UUID	dom	rxbytes_slope	rxpackets_slope	rxerrors_slope	rxdrops_slope	txbytes_slope	txpackets_slope	...
0	1604455173	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	88.2065	30.1414	0.0	0.0	79.8981	5.5275	...
1	1604455142	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	87.8708	27.3499	0.0	0.0	0.0000	0.0000	...
2	1604455113	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	87.8865	27.2996	0.0	0.0	0.0000	0.0000	...
3	1604455082	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	87.8760	27.4076	0.0	0.0	0.0000	0.0000	...
4	1604455055	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	87.7241	25.8210	0.0	0.0	0.0000	0.0000	...
5	1604455024	7	"2bc1fde1-28d9-454e-8029-21a138714234"	one-33	87.7128	25.7100	0.0	0.0	0.0000	0.0000	...

그림 8. KVM 데이터셋
Fig. 8. Dataset in KVM

을 나타내는 희소 아티팩트가 상당수 존재했다. 본 연구에서는 데이터의 99% 이상이 0으로 구성된 경우를 희소 아티팩트로 판단하였다. 네트워크 아티팩트 중에서 수신 및 송신할 때 발생하는 에러와 패킷 드롭의 변화율이 희소 아티팩트라는 것은 수집된 데이터가 네트워크를 이용한 공격이 아니라는 것을 의미한다. 메모리의 아티팩트 중에서 메모리 사용량의 최대치, 현재 사용량, 실제 사용량의 변화율이 희소 아티팩트라는 것은 수집된 데이터가 메모리 누수와 관련이 없다는 것을 의미한다. 할당된 가상 CPU의 수가 희소 아티팩트라는 것은 정상과 비정상 일때 모두 동일한 가상 CPU가 할당되었다는 것을 의미한다. 메모리 스왑 인과 메모리 스왑 아웃의 변화율이 희소 아티팩트라는 것은 시스템의 성능 저하가 일어나지 않았다는 것을 의미한다. 가상 드라이브와 하드 드라이브의 에러 변화율이 희소 아티팩트라는 것은 모니터링 과정에서 장비의 오작동이 없었다는 것을 의미한다.

이러한 희소 아티팩트들은 기계학습 및 디지털 포렌식에 의미있는 아티팩트로 활용할 수 없기 때문에 데이터 전처리 과정에서 제거하는 과정을 거친다.

수집한 아티팩트 중에서 디지털 포렌식의 관점에서 의미있는 아티팩트들은 정상 데이터와 비정상 데이터에서 유의미한 차이를 보일 것으로 예상된다. 네트워크 아티팩트 중에서 수신 및 송신 바이트 수와 패킷 수는 비정상적인 데이터의 흐름을 파악하는데 사용될 수 있다. 메모리 아티팩트 중에서 현재 사용중인 메모리의 양은 비정상 데이터는 이상 행동을 하고 있는 경우이므로 정상 데이터에 비해 메모리 사용량이 급증할 것으로 예상된다. CPU 아티팩트 중에서 CPU 사용 시간과 관련된 아티팩트는 가상 CPU가 사용되는 시간의 변화율을 의미하며 이것이 정상 데이터와 비정상 데이터간에 차이가 존재한다면 비정상 데이터의 이상 행동으로 인하여 CPU 사용량의 차이가 존재하는 것으로 해석 가능하다. 가상 드라이브와 하드 드라이브의 아티팩트 중에서 드라이브 블록 장치의 읽기와 쓰기 바이트의 변화율의 아티팩트는 디스크 I/O 패턴을 분석하고, 데이터의 무단 변경이나 불법 복사 시도를 추적하는데 중요한 단서가 되며 이는 곧 디지털 포렌식에서 매우 큰 역할을 하는 아티팩트이다.

수집된 아티팩트들은 디지털 포렌식 프로세스에서 근본적인 역할을 하게 된다. 특히 클라우드 환경에서 발생한 사건의 정확한 원인을 규명하거나, 사건 발생 전후의 데이터 상태를 비교하고 분석하는 데 있어서 중요한 정보를 제공한다. 정상 및 비정상 데이터의 차이를 드러내는 이러한 아티팩트들은 사건의 연관성 및 영향을 평가하는데 필수적인 요소로 작용한다. 이러한 아티팩트들은 클라우드 환경 특성에 최적화된 포렌식 접근 방식을 개발하는데 활용된다.

V. 기계학습 알고리즘

데이터셋에는 클라우드 환경에서의 가상 머신의 리소스 사용량과 네트워크 및 디스크 활동등에 관한 시계열 데이터 등이 포함되어 있다. 본 논문에서는 이러한 데이터들을 적절하게 분석하기 위해 로지스틱 회귀(Logistic Regression), SVM (Support Vector Machine), KNN(K-Nearest Neighbors), 순환 신경망(Recurrent Neural Network, RNN), LSTM (Long Short-Term), GRU(Gated Recurrent Unit)과 같은 여러 가지 머신러닝과 딥러닝 알고리즘을 모두 고려해서 적절한 알고리즘을 선정하여 학습을 진행한다.

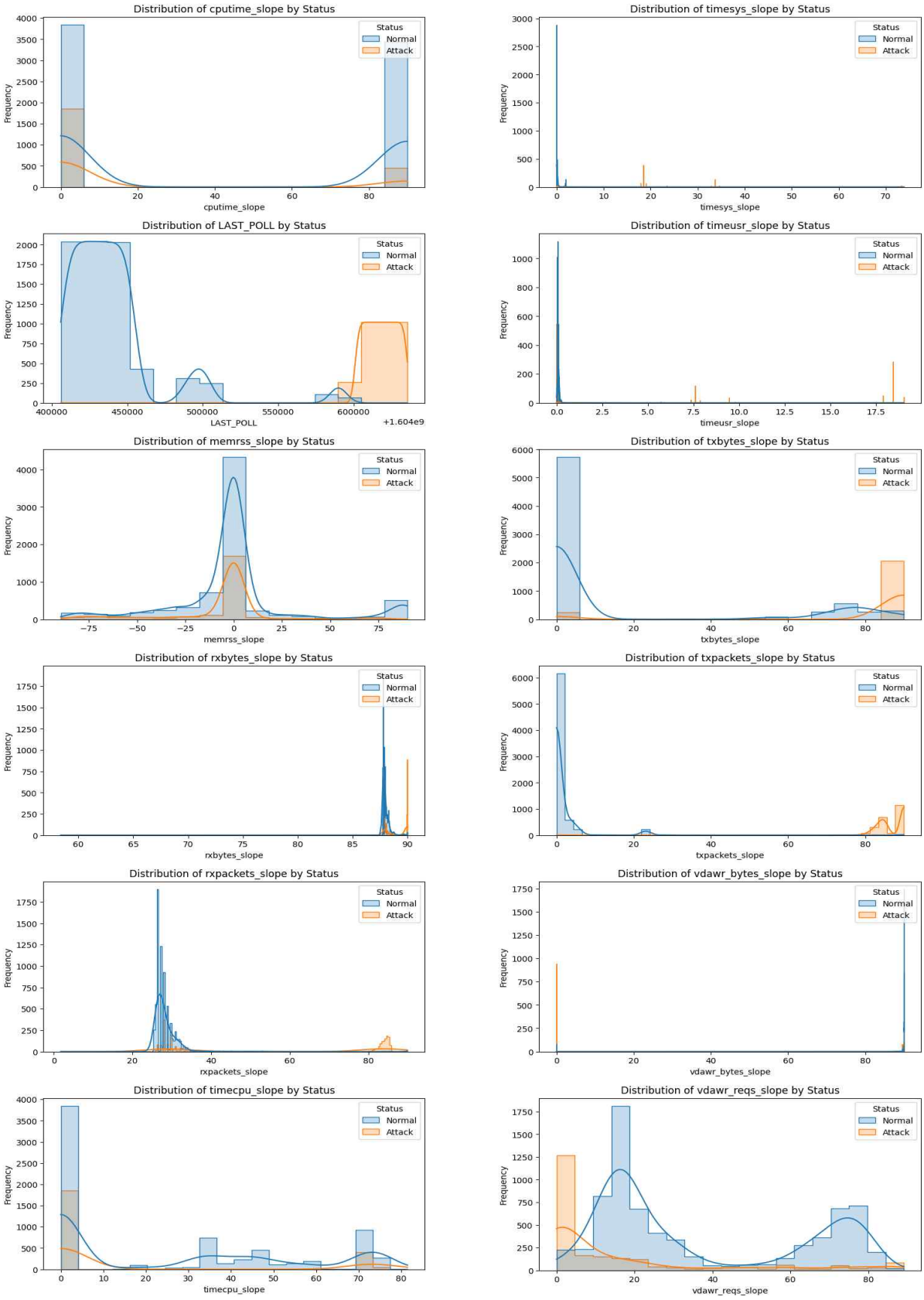


그림 9. 피쳐값 분포

Fig. 9. Distribution of features

5-1 로지스틱 회귀

로지스틱 회귀는 출력 값을 0과 1 사이로 제한하고 이를 시그모이드 함수를 이용하여 확률로 해석하여 주어진 입력 변수들의 선형 조합을 사용하여 시그모이드 함수를 계산하고 나온 확률 값을 임계값을 기준으로 이진 레이블을 결정한다. 로지스틱 회귀는 출력값이 확률이기 때문에 분류 결정뿐만이 아니라 예측의 불확실성을 평가하는 데에도 유용하다.

5-2 SVM(Support Vector Machine)

SVM은 마진 최대화 원리를 이용하여 클래스를 가장 잘 구분하는 초평면을 찾는 것이다. 이때 SVM은 초평면과 가장 가까운 훈련 데이터 사이의 거리인 마진을 최대화하는 초평면을 찾는다. 이렇게 구성한 SVM 모델은 결정 경계와 가장 가까운 각 클래스의 데이터 포인트에 의해 결정된다. SVM은 기본적으로 선형 분류기로 사용되지만 커널 트릭을 사용하거나 유사도 특성을 사용하면 비선형 데이터셋에서도 학습이 가능하다.

5-3 순환 신경망(Recurrent Neural Network, RNN)

순환 신경망은 이전 단계의 출력을 현재 단계의 입력으로 사용하는 신경망 구조로 순차적인 데이터를 처리하는 데 강점이 있는 모델이다. 각 단계에서 입력과 이전 단계의 출력을 사용하여 현재 단계의 출력을 계산하여 이를 통해서 이전의 정보를 유지하고 시간적인 의존성을 모델링 할 수 있다. 그러나 이러한 RNN은 과적합에 취약할 수 있다는 단점이 존재한다. 본 논문에서 사용하는 데이터셋에는 시계열 데이터도 포함되어 있으며 시퀀스의 각 요소를 순차적으로 처리하는 이러한 모델이 효과적일 것으로 예상된다. 일반적으로 RNN은 Gradient Vanising이라는 문제를 가지고 있다. 이는 시퀀스가 길어질 수록 이전 정보를 잃어버려서 시퀀스의 앞부분의 정보가 뒷부분에 영향을 적절한 학습이 되지 않는다는 것을 의미한다. 이러한 문제를 해결하기 위해서 게이트라는 구조를 통해 정보를 효과적으로 학습하는 LSTM(Long Short-Memory)과 GRU(Gated Recurrent Unit)과 같은 RNN 구조가 고안되었다. 본 논문에서는 RNN과 더불어 LSTM과 GRU를 이용하여 학습을 진행한다.

5-4 LSTM(Long Short-Memory)

LSTM은 입력 게이트(input gate), 망각 게이트(forget gate), 출력 게이트(output gate)로 이루어져 있다. 이 게이트들은 셀 상태를 업데이트하고 데이터가 셀을 통과하는 방식을 조절한다. 망각 게이트는 셀 상태에서 어떤 정보를 제거할 지를 결정하고, 입력 게이트는 새로운 정보를 어떻게 추가할 것인지를 결정한다. 출력 게이트는 다음 층으로 전달될 출력을 결정한다. 이러한 방식을 통해서 LSTM은 Gradient Vanising 문제를 해결한다.

5-5 GRU(Gated Recurrent Unit)

GRU는 LSTM보다 간단한 구조로 업데이트 게이트(update gate), 재설정 게이트(reset gate)의 2개의 게이트로 이루어져 있다. 업데이트 게이트는 셀의 상태르 얼마나 업데이트할 것인지를 결정하고, 재설정 게이트는 이전 상태를 얼마나 고려하여 재설정할지를 결정한다. GRU는 셀 상태와 hidden state를 분리하지 않고 하나의 벡터로 사용한다. 이러한 GRU는 파라미터가 더 적어 LSTM보다 구조가 간단하고 계산 효율이 높다는 장점이 있다.

Ⅵ. 클라우드 디지털 포렌식 시나리오 및 검증

6-1 전처리 과정

전처리 과정은 다음과 같다. 먼저 학습에 사용할 수 없는 희소 데이터를 제거한다. 그후 NaN값과 같은 결측치와 이상치 및 중복된 값을 제거한다. 그 후 디지털 포렌식 관점에서 의미있는 피처를 분석하기 위해서 정상 데이터와 비정상 데이터의 분포를 히스토그램을 통해서 나타냈다. cputime_slope, rxpackets_slope 와 같은 몇몇 피처들을 제외하고는 대부분의 피처들이 정상과 비정상 데이터로 구분되는 것으로 확인되었다. 이것은 4장에서 설명한 것과 동일하게 정상과 비정상 데이터의 성질이 다르다는 것을 확인 가능하다. 이것은 기계학습을 이용하여 데이터를 분석할 수 있다는 것을 의미한다. 이후 전체 데이터셋을 훈련 데이터, 검증 데이터, 테스트 데이터로 6:2:2의 비율로 분할한다. 학습 모델의 성능을 최적화 하기 위해 정규화와 표준화를 진행하여 데이터를 정규 분포와 유사하게 만들어준다.

6-2 PCA 분석

의미없는 피처를 드랍하여 특성 선택을 진행한 이후 해당 피쳐들을 토대로 PCA(Principal Component Analysis) 분석을 진행하였다. PCA 분석은 공분산 행렬과 고유값 및 고유벡터의 계산을 통해 가장 중요한 주성분을 선택하여 차원축소를 진행한다(식 (1)). 이를 통해서 고차원의 데이터를 데이터의 주요한 특성은 유지하면서 저차원 데이터로 변환하여 데이터 시각화를 진행한다.

$$\sum_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \overline{x_i})(x_{kj} - \overline{x_j}) \quad (1)$$

PCA 분석을 진행한 결과는 그림 10과 같다. 정상과 비정상은 두 개의 주성분을 기준으로 어느정도 구분되는 것으로

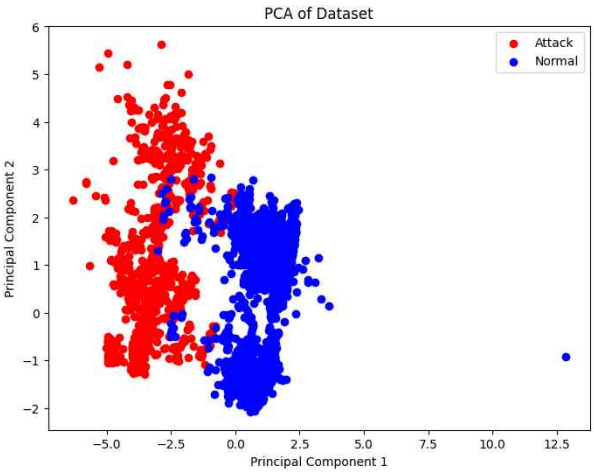


그림 10. 데이터셋의 PCA
Fig. 10. PCA of dataset

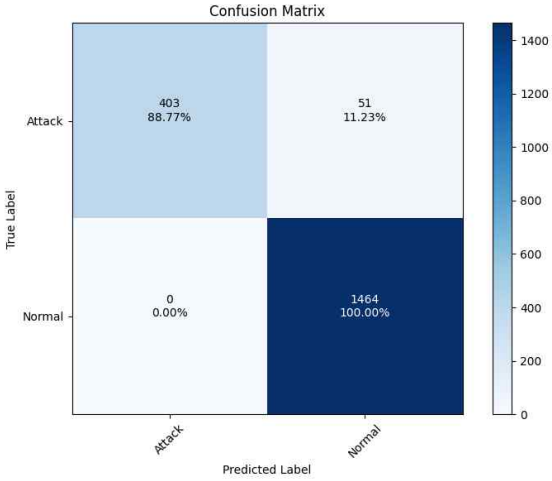


그림 11. SVM의 혼동행렬
Fig. 11. Confusion matrix of SVM

보이며 이는 PCA가 두 상태를 구분하는데 효과가 있다는 것을 의미한다. Normal 상태의 점이 멀리 떨어져 있는 이상치 데이터도 확인 가능하다. 이러한 PCA는 해당 데이터셋의 핵심적인 패턴이나 구조를 파악하는데 도움이 된다. PCA는 가상머신의 다양한 성능 지표 사이의 상관관계를 파악하는 데에도 도움을 주며 어떤 지표들이 성능에 가장 큰 영향을 미치는지를 확인하는데 사용된다.

6-3 데이터 스케일링

데이터를 표준화 하기 위하여 데이터 스케일링을 진행한다. 본 논문에서는 scikit-learn 라이브러리에서 제공하는 'StandardScaler'를 데이터 전처리 클래스로 특성의 표준화를 진행한다.

6-4 모델 학습 진행

5장에서 선정한 기계학습 알고리즘을 모델로 선택하여 학습을 진행한다.

1) SVM(Support Vector Machine)

SVM을 학습하는데 사용한 하이퍼파라미터는 rbf 커널, C=10, gamma=5를 사용하였다. 실제 성능평가를 진행한 결과는 그림 11과 같다.

본 연구는 이러한 데이터를 통해서 디지털 포렌식을 진행하는 것이기 때문에 Attack에 대한 데이터를 수집하는 것이 목표이므로 성능 평가는 Attack을 위주로 분석한다. 성능의 정확도는 0.9734가 나왔으며 precision은 1.00, recall은 0.89, f1-score는 0.94가 나왔다. 디지털 포렌식에서는 Attack을 놓치지 않는 recall이 매우 중요하므로 이러한 recall값을 올리기 위한 과정이 필요할 것으로 예상된다.

2) 순환 신경망(Recurrent Neural Network, RNN)

RNN을 학습하는데 사용된 하이퍼파라미터는 다음과 같다. 레이어의 뉴런의 수는 50개로 50차원의 벡터로 학습을 진행하였다. 활성화 함수로는 하이퍼볼릭 탄젠트(tanh)를 사용하여 -1과 1 사이의 값을 출력하도록 했다. 옵티마이저는 adam(Adaptive Moment Estimation)을 사용하여 학습률을 각 매개변수에 대해 조정하는 방식을 사용하였다. 규제 방법에는 과적합을 방지하기 위해서 L2 규제를 사용했다. 배치 크기는 64로 설정하였으며 에폭은 10으로 설정하였다. 실제 성능평가를 진행한 결과는 그림 12와 같다. 성능의 정확도는 학습이 진행될수록 loss값이 크게 줄어드며 정확도는 0.999에 근접하는 결과가 나왔다. 이는 이러한 결과는 데이터셋이 시계열의 데이터셋의 성질을 가지고 있기에 머신러닝을 사용한 SVM에 비해서 딥러닝을 사용한 RNN이 더 높은 정확도를 보이는 것으로 예상된다. 또한 과적합을 방지하기 위해서 규

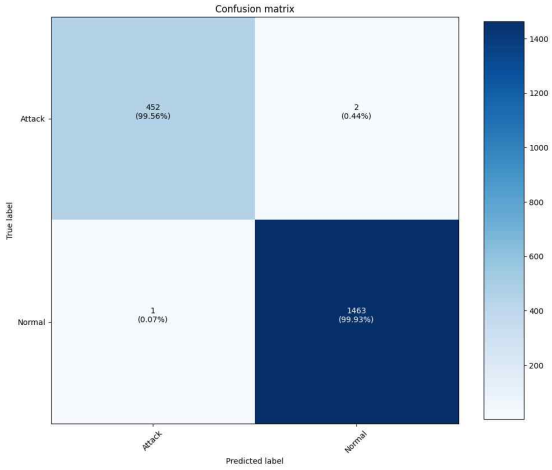


그림 12. RNN의 혼동행렬
Fig. 12. Confusion matrix of RNN

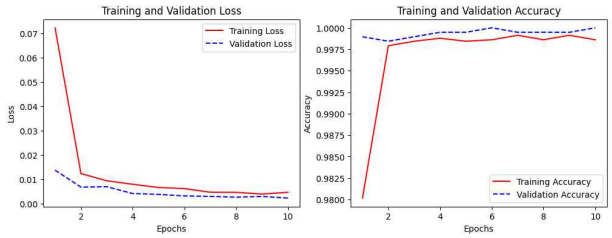


그림 13. RNN의 손실함수와 정확도
Fig. 13. Loss and accurancy of RNN

제를 사용하였다. SVM에 비해서 Attack을 Normal로 오진하는 경우가 매우 줄어든 것을 확인할 수 있다. 이는 디지털 포렌식을 하는데 있어서 중요한 증거 데이터를 놓치는 경우가 매우 적다는 것을 의미하므로 의미있는 결과값이다.

또한 그림 13을 통해서 에폭이 증가함에 따라서 Loss값이 감소하고 정확도가 증가하는 모습을 확인할 수 있다.

3) LSTM(Long Short-Memory)

LSTM 역시 RNN과 비슷한 하이퍼파라미터와 옵티마이저 및 규제를 사용했다. 실제 성능평가를 진행한 결과는 그림 14와 같다. 성능의 정확도는 학습이 진행될수록 loss값이 크게 줄어드며 정확도는 0.995에 근접하는 결과가 나왔다. 이는 RNN보다 약간 더 낮은 정확도 수치인데 Confusion Matrix는 동일한 결과값이 나왔지만 LSTM은 게이트를 이용하여 좀더 복잡한 연산을 진행하여 RNN의 장기 의존성 문제에 대응하여 좀더 복잡한 시퀀스 데이터를 처리하면서 과적합이 줄어든 것이라고 볼 수 있다. 결과적으로 그림 15를 보듯이 Loss값이 이전보다 줄었으며 에폭이 진행될수록 정확도가 이전의 데이터의 내용을 반영하는 것을 알 수 있다.

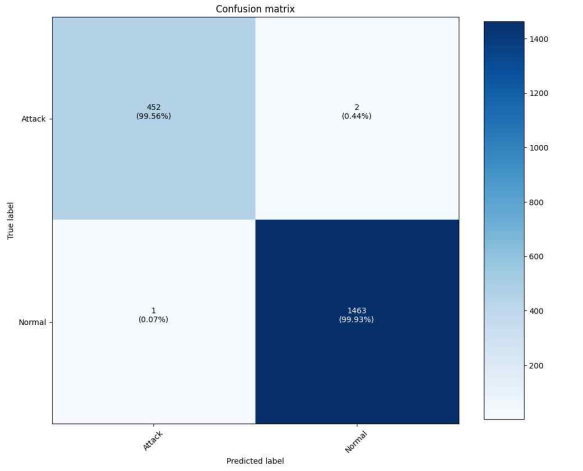


그림 14. LSTM의 혼동행렬
Fig. 14. Confusion matrix of LSTM

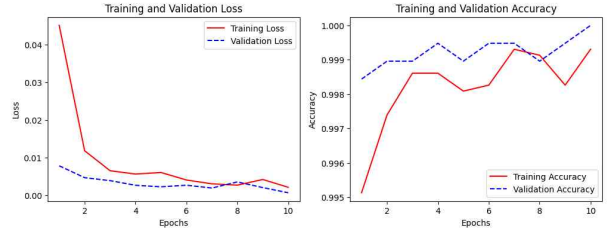


그림 15. LSTM의 손실함수와 정확도
Fig. 15. Loss and accurancy of LSTM

4) GRU(Gated Recurrent Unit)

GRU 역시 RNN과 비슷한 하이퍼파라미터와 옵티마이저 및 규제를 사용했다. 실제 성능평가를 진행한 결과는 그림 16과 같다. 성능의 정확도는 학습이 진행될수록 loss값이 크게 줄어드며 정확도는 0.996에 근접하는 결과가 나왔다. 이는 이는 LSTM과 마찬가지로 RNN보다 약간 더 낮은 정확도 수치인데 Confusion Matrix는 동일한 결과값이 나왔지만 GRU는 게이트를 이용하여 연산을 진행하지만 LSTM보다 간단한 구조로 LSTM보다 빠른 학습이 가능했다. GRU로 학습을 진행했을때는 특이하게 에폭이 5 일때 손실 데이터가 증가하는 과적합이 발생했지만 그 이후 추가적인 학습을 통해서 다시 회복하여 일반화를 진행한다는 것을 의미한다. 이는 RNN의 단점인 장기 의존성을 추가적인 학습을 통해서 해결 가능하다는 것을 의미한다.

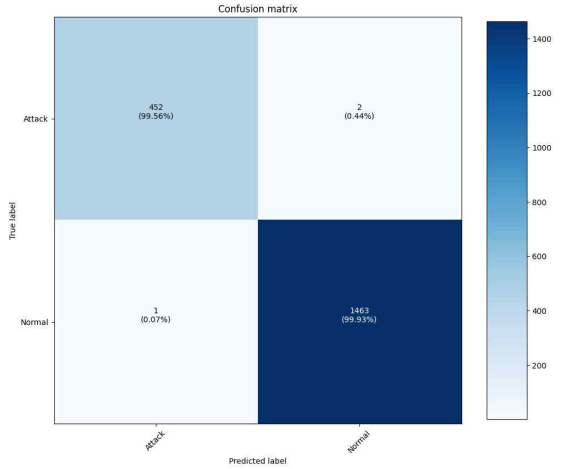


그림 16. GRU의 혼동행렬
Fig. 16. Confusion matrix of GRU

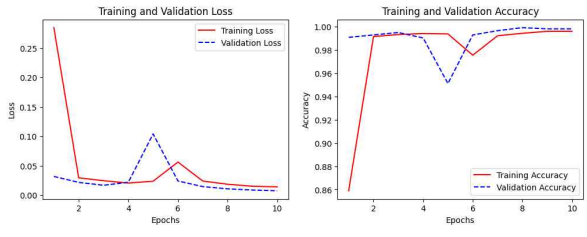


그림 17. GRU의 손실함수와 정확도
Fig. 17. Loss and accurancy of GRU

6-5 성능 평가

표 2에 따르면 SVM과 같은 머신러닝 모델에서는 약 97%의 정확도를 보였으며, RNN, LSTM과 같은 딥러닝 모델을 사용하여 학습을 진행하였을 때는 약 99.6%의 정확도라는 매우 높은 성능을 보였다. 이것은 학습한 모델이 정상과 비정상의 데이터가 매우 확실하게 구분된다는 것이며 이것은 디지털 포렌식의 관점에서 기계학습이 효과적이라는 것을 의미한다.

표 2. 각 모델별 성능 지표
Table 2. Performance for each model

Model	F1 Score	Precision	Recall	Accuracy
SVM	0.940	1.0	0.89	0.973
RNN	0.998	0.998	0.998	0.999
LSTM	0.998	0.998	0.998	0.995
GRU	0.998	0.998	0.998	0.996

Ⅶ. 결 론

클라우드 환경의 제약사항과 복잡한 서비스 구조로 인하여 기존의 디지털 포렌식 방법론으로는 아티팩트 수집과 분석의 어려움이 존재한다. 본 논문에서는 이러한 아티팩트 수집에서의 어려움을 해결하기 위해 KVM을 이용하여 클라우드 환경을 구축하였으며, 하이퍼바이저 레벨에서 디지털 포렌식에 적합한 아티팩트를 수집하였다. 수집한 아티팩트들이 디지털 포렌식에 적합한지에 대한 분석을 진행했으며, 수집된 대규모의 아티팩트를 효과적으로 처리하고 분석하기 위해 여러 기계학습 알고리즘을 적용하였다. 데이터의 패턴을 인식하고, 정상과 비정상 행동을 식별하는 모델을 개발하였으며, 이 과정에서 PCA와 같은 차원 축소 기법을 사용하였다. 개발된 모델은 클라우드 환경이 제공하는 대용량 데이터의 특성을 반영하면서 디지털 포렌식의 정확도를 향상시키는 데 중점을 두었다. 연구 결과는 기계학습 알고리즘이 클라우드 환경에서의 디지털 포렌식 프로세스를 개선할 수 있음을 보여주었다. 정확한 분류와 신속한 패턴 인식 능력은 클라우드 환경에서 발생하는 보안 위협에 대한 이해를 심화시키고 효과적인 사건 조사 및 분석에 필수적인 인프라를 제공한다. 기계학습 기반 클라우드 디지털 포렌식은 기존의 디지털 포렌식에서 수행하기 어려운 클라우드 환경에서의 디지털 포렌식을 수행하는데 효과적인 프로세스를 제공한다. 본 연구의 기계학습 기반 클라우드 디지털 포렌식은 포렌식 전문가들에게 클라우드 환경에서 디지털 포렌식을 수행하는데 강력한 도구를 제공할 것으로 기대된다.

감사의 글

본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었습니다(2022-0-01077).

참고문헌

[1] Cloud Computing Market Size, Share & Trends Analysis Report by Service (SaaS, IaaS), by Deployment, by Enterprise Size, by End-use, by Region, and Segment Forecasts, 2023-2030 [Internet]. Available: http://www.grandviewresearch.com/industry-analysis/cloud-computing-industry.

[2] H. J. Chung and S. J. Lee, “Digital Forensics Trends and Prospects in Cloud Computing Environments,” *Review of KIISC,* Vol. 22, No. 7, pp. 7-13, November 2012.

[3] C. B. Lee, A Study on the Improvement of the Legal System to Revitalize Cloud Computing, KISA, Technical Report KISA-RP-2010-0055, March 2011.

[4] P. Purnaye and V. Kulkarni, “BiSHM: Evidence Detection and Preservation Model for Cloud Forensics,” *Open Computer Science*, Vol. 12, No. 1, pp. 154-170, December 2022. https://doi.org/10.1515/comp-2022-0241

[5] Google Cloud. What Is Cloud Computing? [Internet]. Available: https://cloud.google.com/learn/what-is-cloud-computing?hl=ko.

[6] GeeksforGeeks. Layered Architecture of Cloud [Internet]. Available: https://www.geeksforgeeks.org/layered-architecture-of-cloud.

[7] S. Syed and V. Anu, “Digital Evidence Data Collection: Cloud Challenges,” in *Proceedings of 2021 IEEE International Conference on Big Data,* Orlando, FL, USA, pp. 6032-6034, December 2021. https://doi.org/10.1109/BigData52589.2021.9672014

[8] S. A. Ali, S. Memon, and F. Sahito, “Challenges and Solutions in Cloud Forensics,” in *Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing*, Barcelona, Spain, pp. 6-10, August 2018. https://doi.org/10.1145/3264560.3264565

[9] J. H. Jeong, “Flash Memory Data Extraction for Digital Forensic of IoT Device,” in *Proceedings of Korea Software Congress 2020,* pp. 798-800, December 2020.

[10] M. Kim and T. S. Shon, “Research on Network-based Smart Home Device Forensic Technology,” *Journal of Digital Forensics,* Vol. 15, No. 4, pp. 84-94, December 2021.

[11] L. Peng, “Information Fusion-Based Digital Forensics Framework in Cloud Environment,” in *Proceedings of 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, pp. 279-283, May 2020. https://doi.org/10.1109/ICAIBD49809.2020.9137434

[12] P. Purnaye, “A Comprehensive Study of Cloud Forensics,” *Archives of Computational Methods in Engineering,* Vol. 29, pp. 33-46, 2022. https://doi.org/10.1007/s11831-021-09575-w

[13] D. Hemdan, “An Efficient Digital Forensic Model for Cybercrimes Investigation in Cloud Computing,” *Multimedia Tools and Applications*, Vol. 80, pp. 14255-14282, January 2021. https://doi.org/10.1007/s11042-020-10358-x

[14] Red Hat. What Is a Hypervisor? [Internet]. Available: https://www.redhat.com/ko/topics/virtualization/what-is-a-hypervisor.

[15] DATA ON-AIR. libvirt Virtualization Library Analysis [Internet]. Available: https://dataonair.or.kr/db-tech-reference/d-lounge/technical-data/?mod=document&uid=236897.

[16] A. Aldribi, and I. Traoré, B. Moa, and O. Nwamuo, “Hypervisor-Based Cloud Intrusion Detection through Online Multivariate Statistical Change Tracking,” *Computers & Security*, Vol. 88, 101646, January 2020. https://doi.org/10.1016/j.cose.2019.101646

신현욱(Hyenuk Shin)

2021년～현　재: 아주대학교 소프트웨어융합대학 사이버보안학과 학사과정

※관심분야 : 디지털 포렌식, 정보보호, 기계학습

손태식(Taeshik Shon)

2000년 : 아주대학교 정보및컴퓨터공학부 졸업(학사)

2002년 : 아주대학교 정보통신전문대학원 졸업(석사)

2005년 : 고려대학교 정보보호대학원 졸업(박사)

2004년～2005년: University of Minnesota 방문연구원

2005년～2011년: 삼성전자 통신·DMC 연구소 책임연구원

2017년～2018년: Illinois Insitute of Technology 방문교수

2011년～현　재: 아주대학교 정보통신대학 사이버보안학과 교수

※관심분야 : Digital Forensics, ICS/Automotive Security