논문 2025-4-1 http://dx.doi.org/10.29056/jsav.2025.12.01

악성코드 이상 징후 탐지를 위한 LLM 로그 임베딩 기반 표현 학습 기법

김동완*, 김현수*, 박경엽*, 김민수*, 신동명*†

An LLM-based Log Embedding Representation Learning Approach for Detecting Early-stage Malware Anomalies

Dong-Wan kim*, Hyun-Soo Kim*, Kyung-Yeob Park*, MinSoo Kim*, Dong-Myung Shin*†

요 약

대규모 정보 시스템과 웹 서비스 환경에서는 로그 기반 이상 탐지가 랜섬웨어 등 악성코드 침투의 초기 징후를 포착하기 위한 핵심 수단이지만, 원문 로그 텍스트와 제한적인 특징 공학에 의존하는 기존 비지도 이상 탐지 구성은 불균형한 라벨 분포와 복잡한 공격 단계가 뒤섞인 실제 보안 로그에서 이상 행위를 안정적으로 구분하기 어렵다. 본 논문은 웹 접근 로그와 시스템 감사 로그 약 80만 건을 대상으로 원문 로그 표현, 사전학습 Llama 계열 언어모델 임베딩, 보안 로그 도메인 파인튜닝 임베딩으로 구성된 세 가지 로그 표현과 통계 및 심층 표현 학습 기반 이상 탐지 모델을 결합한 LLM 로그 임베딩 표현 학습 기법을 제안한다. 공통 데이터 분할과 학습 조건에서 정상·비정상 판별 성능을 비교한 결과, 원문 로그 기반 구성에서는 전반적으로 F1-Score가 낮았으나 임베딩 기반 표현에서는 모델 대부분이 이진 F1-Score가 원본 로그 대비 약 2.5배, 저빈도 공격 유형에서는 약 3배 이상 향상됨을 확인하였다. 이를 통해 보안 로그 도메인에 적응된 LLM 로그 임베딩이 악성코드 이상 징후를 구조적으로 구분하는 공통 입력 표현으로 활용가능함을 시사한다.

Abstract

In large-scale information systems and web services, log-based anomaly detection is a key means of capturing early signs of ransomware and other malware. However, unsupervised methods that rely on raw log text and limited feature engineering perform poorly on real security logs with imbalanced labels and multi-stage attacks. This paper proposes an LLM-based log embedding pipeline that combines three representations-raw logs, embeddings from pre-trained Llama language models, and domain-fine-tuned embeddings for security logs-with statistical and deep anomaly detection models, using about 800,000 web access and system audit log entries. Under a common data split, embedding-based representations raise the binary F1-score of most models to roughly 2.5 times the raw-log baseline and more than threefold for rare attack types, demonstrating their effectiveness as a common input representation for malware anomaly detection and early-warning systems.

한글키워드 : 이상 탐지, 대규모 언어모델, 로그 임베딩, 악성코드 침입 탐지, 표현 학습

keywords : Anomaly detection, LLM, Log embedding, Malware intrusion detection, Representation learning

* 엘에스웨어㈜ 소프트웨어연구소 연구개발본부
† 교신저자: 신동명(email: roland@lsware.co.kr)

접수일자: 2025.12.01. 심사완료: 2025.12.10.
게재확정: 2025.12.20.

1. 서 론

대규모 정보 시스템과 클라우드 환경에서 서비스 가용성과 보안성을 유지하려면 로그 기반 이상 징후 탐지 기술이 요구된다. 그러나 분산 아키텍처와 마이크로서비스 확산으로 로그 규모와 구조적 복잡성이 증가하면서 규칙 기반 탐지나 단순 통계 기법만으로는 다양한 장애와 공격 징후를 식별하기가 어려워졌다. 이를 보완하기 위해 로그 시퀀스를 순환 신경망이나 오토인코더와 같은 심층 학습 모델로 학습하여 정상 패턴을 모형화하고, 재구성 오차 또는 예측 실패를 이상으로 판정하는 방법이 제안되었다[1]. 예를 들어, DeepLog는 로그 시퀀스를 언어 모델링 문제로 간주하여 LSTM(Long Short-Term Memory)으로 정상 패턴을 학습한 뒤, 예측 실패 시 이상으로 판단한다[2]. 한편, 트랜스포머 기반 사전학습 언어모델의 발전으로 로그 메시지의 의미 정보를 직접 인코딩하는 표현 학습 기법이 제안되었다. LogBERT는 BERT(Bidirectional Encoder Representations from Transformers) 기반 자기지도 학습으로 정상 로그 시퀀스 분포를 학습하고, 임베딩 공간의 편차를 통해 이상 여부를 판정한다[3]. 또한, LAnoBERT는 로그 파서에 의존하지 않고 BERT 기반 사전학습을 활용하여 기존 비지도 학습 기반 방법보다 높은 이상 탐지 성능을 보였고[4], LogLLM은 BERT 계열과 Llama 계열 대규모 언어모델을 결합하여 의미 벡터 추출과 시퀀스 수준 분류를 통합하는 연구를 제시하였다[5]. 이러한 연구들은 로그를 자연어로 간주하여 파서 의존성을 완화하고 의미 정보를 활용하는 방향으로 이상 탐지 기법을 확장하였다. 그러나 표현 학습 관점에서 설계 선택을 체계적으로 비교한 사례는 제한적이다.

또한 선행 연구는 주로 공개 벤치마크 환경에서 일반적인 시스템 장애 탐지 성능을 평가한다. 실제 운영 환경의 보안 침해, 특히 악성코드 침투와 관련된 이상 징후 로그는 서비스 도메인과 공격자 행위 패턴의 영향을 강하게 받으므로, 템플릿 기반 표현이나 일반 도메인 코퍼스로 학습된 임베딩만으로는 미세한 이상 징후를 안정적으로 포착하는데 한계가 있다. 사전학습 언어모델을 보안 로그 도메인에 어떤 방식으로 적응시키는 것이 공격 징후 탐지에 유리한지, 사전학습 임베딩, 도메인 파인튜닝 임베딩, 원문 로그 표현 간 설계 선택을 어떻게 할 것인지에 대한 정량적 비교 연구는 부족하다[6].

따라서 본 논문에서는 악성코드 이상 징후 탐지를 위해 수집한 웹 접근 로그와 시스템 로그를 대상으로 LLM 기반 로그 임베딩 표현 학습 기법을 제안한다. 제안 기법은 LLM을 통해 로그 행 단위 의미 벡터를 추출하여 원문 로그 기반 표현, 사전학습 임베딩, 보안 로그 도메인 파인튜닝 임베딩으로 구성된 세 가지 표현을 정의한다. 각 표현은 비지도 이상 탐지 모델에 공통 입력하여 동일한 데이터셋과 평가 지표에서 표현 학습 전략에 따른 성능 차이를 정량적으로 평가한다. 본 논문은 LLM 로그 임베딩 기반 표현 학습 기법과 실험 결과를 제시하여 악성코드 이상 징후를 포함한 보안 로그 분석에 사전학습 언어모델 적용 시, 표현 방식과 도메인 적응 전략을 선택하기 위한 설계 기준을 제공한다.

2. 관련 연구

2.1 로그 임베딩 벡터 추출과 LLM 모델 기반 접근

로그 데이터를 벡터 공간에서 표현하는 연구는 고전적인 단어 빈도 기반 표현에서 BERT, GPT(Generative Pre-trained Transformer) 계열과 같은 LLM을 활용하는 방식으로 확장되었다. 초기에는 로그 시퀀스에 BERT를 파인튜닝하여

시퀀스 수준의 의미 정보를 반영하는 이상 탐지 구조가 제안되었고[7], 이후 로그 도메인 적응과 파라미터 효율적 파인튜닝 기법을 결합하여 LLaMA 계열 언어모델을 로그 분석에 적용해 로그 전용 임베딩 벡터를 활용하는 방향으로 발전하였다[8],[9]. 예를 들어, RoBERTa에 LoRA (Low-Rank Adaptation)를 적용하여 OpenStack 로그 이상 탐지 환경에서 소수의 추가 파라미터만으로 성능을 개선시킨 사례가 있으며[10], Llama-2 모델에 지시문 기반 추가 학습을 적용해 네트워크 로그 파싱과 원인 분석 정확도를 향상시킨 연구도 있다[11]. 한편 라벨과 원인 설명을 동시에 생성하는 LLM-LADE 및 제한된 라벨 환경에서 LLM에 기계학습 모델을 결합해 로그 이상 탐지를 수행하는 FlexLog 등은 LLM을 로그 임베딩 추출의 핵심 모듈로 사용하고, 로그 시퀀스의 의미 구조를 반영한 표현 공간을 구성한 뒤 이를 다양한 이상 탐지 모형의 공통 입력으로 활용한다[12]. 이러한 연구 흐름은 LLM이 로그 이상 탐지에서 규칙 기반 또는 수작업 특징 공학을 대체하는 공통 표현 계층으로 자리 잡고 있음을 시사한다.

2.2 통계 기반 비지도 이상 탐지 기법

본 논문에서는 로그 임베딩 표현의 효과를 평가하기 위해 대표적인 통계 기반 비지도 이상 탐지 기법인 Isolation Forest와 COPOD(Copula Based Outlier Detection)를 사용한다. Isolation Forest는 무작위 분할을 반복하는 결정 트리 집합에서 적은 깊이로 고립되는 샘플을 이상으로 간주하는 비지도 이상 탐지 알고리즘으로, 구조가 단순하고 연산 효율이 높다[13]. COPOD는 Copula 이론에 기반하여 관측치의 꼬리 확률을 추정하고 통계적 극단값으로 이상치를 판별하는 비모수 이상 탐지 기법이다. Copula 이론은 주변 분포를 유지한 채 변수 간 의존 구조를 분리하여 표현하는 방법이며, 꼬리는 상위 몇 퍼센트 영역과 같이 분포의 상위와 하위 극단 구간을 의미한다[14]. 이러한 알고리즘은 라벨 정보가 제한된 보안 로그 환경에서 경량 연산으로 이상 신호를 선별할 수 있는 통계 기반 비지도 기법이며, 본 논문에서는 로그 임베딩 기반 표현 학습과 결합하여 실험의 비교 대상 모형으로 활용한다.

2.3 심층 표현 학습 기반 이상 탐지 기법

Deep SVDD(Deep Support Vector Data Description), 심층 오토인코더(Deep Autoencoder), LSTM-AE는 모두 잠재 공간에서의 거리 또는 재구성 오차를 이상도로 사용하는 심층 표현 학습 계열 기법이며, 단일 로그 행이나 로그 시퀀스를 입력으로 사용하는 이상 탐지 모델의 기본 구성 요소로 활용 가능하다. 이에 본 논문에서는 입력 시퀀스와 복원 시퀀스 간 재구성 오차를 이상도로 정의한다.

Deep SVDD는 정상 데이터를 잠재 공간의 원 인근에 집중되도록 학습하고, 원 중심으로부터의 거리를 이상도로 사용하는 이진 분류 기법으로 정상 라벨만 제공되는 환경에서도 학습이 가능하다[15].

심층 오토인코더는 인코더 - 디코더 구조를 이용하여 입력을 저차원 잠재 공간으로 압축한 뒤 복원하고, 원본과 복원 값의 차이를 재구성 오차로 정의하여 이를 이상 지표로 사용하는 기법이다[16]. 이 같은 심층 오토인코더 계열 기법은 시계열, 이미지, 로그 등 서로 다른 도메인에 비교적 쉽게 적용 가능하다[17].

LSTM-AE는 LSTM 인코더와 LSTM 디코더로 구성된 시퀀스 오토인코더 구조를 사용하여 시계열 패턴을 학습하고, 입력 시퀀스와 복원 시퀀스 간 재구성 오차를 이상도로 활용한다. 인접 시점 간 의존성이 강한 센서 데이터나 네트워크 트래픽과 같이 시퀀스 특성이 뚜렷한 데이터에서 주로 사용된다[18],[19].

3. 제안 표현 학습 기법

본 장에서는 보안 로그에서 악성코드 관련 이상 징후를 탐지하기 위한 LLM 기반 로그 표현 학습 기법을 제안한다. 제안 기법은 웹 접근 로그와 시스템 감사 로그를 입력으로 하여 각 로그 행을 고정 차원의 의미 벡터로 변환하고, 사전학습 단계와 보안 로그 도메인 파인튜닝 단계에서 획득한 임베딩을 이상 탐지 모델의 공통 입력으로 활용하는 구조를 갖는다.

3.1 설계 목표 및 개요

제안 학습 기법의 설계 목표는 다음과 같다. 첫째, 웹 서버 접근 로그와 시스템 감사 로그처럼 문법이 불규칙하고 도메인 특화 용어가 많은 보안 로그에 대해 자연어 수준의 의미 정보를 반영하는 고차원 벡터 표현을 구성한다. 둘째, 동일한 로그 데이터에 대해 원문 로그 기반 표현, 일반 도메인에 사전 학습된 언어모델 임베딩, 보안 로그 코퍼스로 추가 학습한 도메인 파인튜닝 임베딩을 정의하여, 표현 전략 간 차이를 정량적으로 비교한다. 셋째, 표현 학습 단계에서는 정상 여부나 공격 유형 라벨을 입력부에 노출하지 않고 출력부에서만 활용하여, 임베딩이 로그 본문의 의미 구조를 중심으로 형성되도록 유도한다.

이를 위해 본 논문에서는 Meta가 공개한 Llama-3.1-8B 튜닝 모델[20]을 사용한다. 로그 한 건을 하나의 입력 시퀀스로 간주하고, 선택한 은닉 계층의 토큰별 은닉 상태를 평균 풀링(Average Pooling) 하여 4,096차원 로그 임베딩을 정의 한다. 사전학습 상태에서 얻은 임베딩을 기준선으로 사용하고, 동일한 구조를 유지한 상태에서 보안 로그 분류 과제에 대해 추가 학습한 이후의 임베딩을 도메인 파인튜닝 임베딩으로 구분한다. 이렇게 얻은 두 임베딩은 다양한 이상 탐지 모델의 입력으로 사용되며, 표현 공간의 구조와 성능 차이는 실험을 통해 비교한다.

제안 표현 학습 및 이상 탐지 절차의 전체 구조는 그림 1과 같다. 그림의 왼쪽에는 웹 접근 로그와 시스템 감사 로그로 이루어진 원본 로그 데이터가 배치되며, A, B, C는 각각 세 가지 입력 표현 조건을 나타낸다. A 조건은 원문 로그를 토큰 단위로 분할한 후 TF-IDF(Term Frequency-Inverse Document Frequency) 기반 Bag-of-Words 벡터로 변환하는 전통적인 표현 방식이고, B와 C 조건은 전처리된 로그 문자열을 토크나이저와 Llama-3.1-8B 디코더에 입력하여 마지막 은닉 계층의 토큰별 표현을 계산한 뒤 평균 풀링을 적용함으로써 로그 행 단위 임베딩 벡터를 생성하는 LLM 기반 표현 방식이다. B 조건은 사전학습 상태에서 직접 추출한 임베딩을 사용하고, C 조건은 보안 로그 도메인에 대해 LoRA 기반 파인튜닝을 수행한 후 동일한 임베딩 블록을 통해 얻은 임베딩을 사용한다.

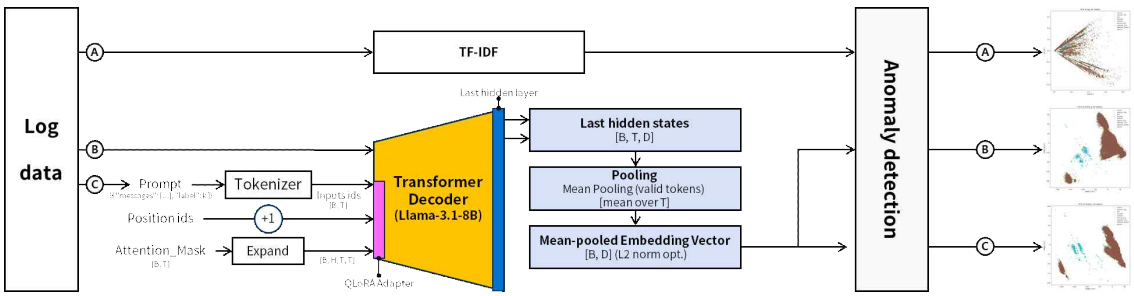

그림 1. LLM 로그 임베딩을 이용한 로그 표현 학습 및 이상 탐지 절차 개요

Fig. 1. Overview of log representation learning and anomaly detection using LLM-based log embeddings

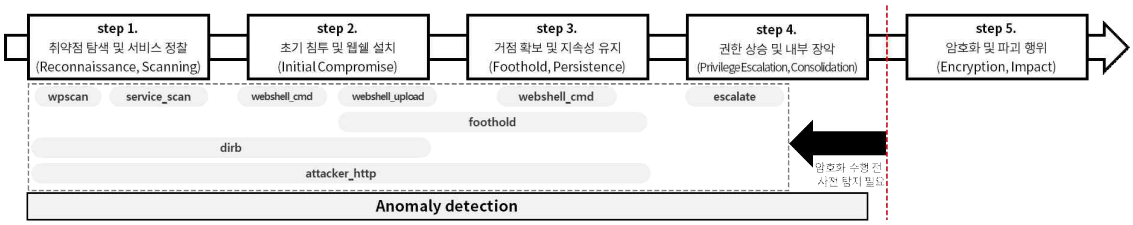

그림 2. 랜섬웨어 공격 단계와 로그 라벨 간 매핑 및 이상 징후 탐지 구간

Fig. 2. Mapping between ransomware attack stages and log labels with the anomaly detection range

그림 오른쪽의 Anomaly detection 영역에서는 TF-IDF 벡터와 두 종류의 LLM 임베딩을 이상 탐지 알고리즘의 공통 입력으로 사용하여 성능을 비교하며, 예시로 제시된 A, B, C 시각화는 각 표현에서 추출한 임베딩을 주성분 분석으로 투영한 결과를 통해 정상 로그와 공격 로그의 분리 양상을 나타낸다.

랜섬웨어 등 악성 코드의 공격 절차 및 로그 데이터셋의 라벨 간 관계는 그림 2와 같다. 공격자는 취약점 탐색과 서비스 정찰 단계(step 1)에서 wpscan, service_scan, dirb, attacker_http 라벨에 해당하는 요청을 반복적으로 발생시키고, 초기 침투 및 웹쉘 설치 단계(step 2)에서 webshell_upload, webshell_cmd 라벨이, 거점 확보 및 지속성 유지 단계(step 3)에서 foothold와 webshell_cmd 라벨이 주로 등장한다. 이후 권한 상승 및 내부 장악 단계(step 4)에서는 escalate 라벨이 나타나며, 최종적으로 암호화 및 파괴 행위 단계(step 5)에서 랜섬웨어 본연의 암호화 공격이 수행된다. 본 논문에서 제안하는 악성코드 이상 징후 탐지 절차는 그림 1의 로그 표현 학습 파이프라인과 그림 2의 공격 단계 구성을 결합한 구조로, step 1부터 step 4까지 구간에서 발생한 로그를 LLM 기반 임베딩과 원문 로그 표현으로 변환한 뒤 통계 기반 및 심층 표현 학습 기반 이상 탐지 모델에 공통 입력하여, 암호화 단계에 도달하기 이전 시점에서 악성 코드 관련 이상 징후를 조기에 탐지하는 것이다.

3.2 LLM 임베딩 블록 구조

본 논문에서 제안하는 LLM 기반 로그 표현 기법은 그림 1과 같이 세 단계로 구성된다. 우선 웹 서버 접근 로그와 시스템 감사 로그처럼 형식이 서로 다른 원본 로그를 행 단위 문자열 x로 정규화하고, 로그 유형 정보를 접두부에 명시한다. 예를 들어, Apache 접근 로그는 '[log type: apache2_access.log]||<로그>'이며, 감사 로그는 '[log type: audit.log]||<로그>'과 같다. 이와 같이 전처리된 로그 문자열은 '다음 로그를 정상 또는 특정 공격 유형으로 분류하라.'와 같은 지시 문장을 포함하는 프롬프트에 삽입하여, 대화형 입력 형식의 사용자 메시지로 구성한다.

사용자 메시지 구성은 일반적인 대화형 언어 모델 포맷을 따른다. 사용자 메시지는 로그 분류 지시와 로그 본문으로 구성되고, 모델 응답은 해당 로그의 정상 여부 또는 공격 유형 목록으로 구성된다. 정상 로그의 경우 'normal'이라는 단일 토큰을 응답하고, 공격 로그의 경우 'attacker_http, dirb, foothold'와 같이 공격 유형 태그를 나열한다. 학습 시에는 이러한 프롬프트 쌍을 사용하여 언어모델이 텍스트 로그에 대해 적절한 라벨 토큰을 생성하도록 한다.

표현 모델 관점에서 로그의 한 행 x는 토크나이저를 통해 토큰 시퀀스 $(t_1, \ldots, t_L)$로 변환된다. 이 시퀀스를 lama-3.1-8B 디코더에 입력하면 각 토큰 위치 i에 대해 마지막 은닉 계층의 은닉 상태 $h_i \in R^D$가 도출된다. 배치 크기 B, 토큰

길이 T, 은닉 차원 D 일 때, 마지막 은닉 계층의 전체 출력은 $R^{\{B\times T\times D\}}$과 같은 3차원 텐서이다. 본 논문에서는 이 중 개별 로그 행에 해당하는 유효 토큰 위치의 은닉 상태만을 모아 평균 풀링을 적용하고, 식 1과 같이 로그 행 단위 임베딩 벡터 $z\in R^{D}$를 정의한다.

$$z = \frac{1}{L}\sum_{i=1}^{L} h_i \qquad (1)$$

식 1에서 L은 패딩을 제외한 유효 토큰의 개수이다. 이후 실험에서는 그림 1의 동일한 임베딩 블록을 사전학습 Llama-3.1-8B와 보안 로그 도메인에 대해 파인튜닝한 Llama-3.1-8B에 공통으로 적용하여 로그 임베딩을 생성한다. 원문 로그 텍스트와 두 종류의 임베딩 표현은 모두 동일한 로그 집합에 대해 정의되므로, 5장에서 입력 표현 차이가 이상 탐지 성능에 미치는 영향을 직접 비교가 가능하다.

3.3 보안 로그 도메인 적응 파인튜닝

보안 로그 도메인 적응 파인튜닝은 3.2절에서 정의한 프롬프트 기반 로그 분류 과제를 대상으로 수행한다. 사전 학습된 Llama-3.1-8B의 기본 가중치는 모두 고정하고, LoRA를 이용하여 일부 투영 행렬에 대해서만 파라미터를 갱신한다. 본 논문에서는 self-attention 모듈의 질의(query) 계층과 값(value) 투영 계층에 랭크 16, 스케일 계수 32, 드롭아웃 0.05를 갖는 LoRA 어댑터를 삽입하고, 나머지 계층은 동결한다.

학습 데이터는 AIT Logdataset V2.0[21]에서 추출한 웹 접근 로그와 시스템 감사 로그로 구성한다. 각 로그는 3.2절에서 설명한 형식의 사용자 메시지와 정상 여부 또는 공격 유형을 나타내는 응답 메시지 쌍으로 변환한다. 이 데이터는 토크나이저를 거쳐 입력 토큰, 어텐션 마스크, 레이블 토큰 시퀀스로 변환되며, 인과 언어모델(causal LM) 손실 함수를 사용하여 학습에 활용한다. 모델은 주어진 로그와 지시 문장을 입력으로 받아 전체 응답 토큰을 순차적으로 예측하도록 학습하고, 이 과정에서 로그 텍스트와 공격 레이블 간의 의미적 대응 관계를 내재화한다.

파인튜닝은 배치 크기 16, gradient accumulation 단계 4, 학습 epoch 1로 설정하고 AdamW 옵티마이저와 16비트 부동소수점(FP16) 연산을 사용하였다. 이를 통해 Llama-3.1-8B의 사전학습 가중치는 그대로 유지하면서, LoRA 어댑터 파라미터만 효율적으로 갱신되도록 구성하였다. 동일한 로그 집합에 대해 사전학습 상태와 도메인 파인튜닝 이후 상태에서 각각 임베딩을 추출하고, 5장에서 두 표현이 라벨 구조와 이상 징후 패턴을 얼마나 다르게 나타내는지 실험을 통해 비교한다.

4. 실험 설계

본 장에서는 제안한 로그 임베딩 기반 표현 학습 기법의 효과를 검증하기 위한 실험 설계를 기술한다. 입력 표현 조건, 데이터셋 구성과 LLM 파인튜닝 범위, 로그 표현 방식과 임베딩 추출 절차, 이상 탐지 모델 구성, 파인튜닝 비적용 시나리오에서의 일반화 평가 절차를 설명한다.

4.1 실험 조건 및 비교 대상

로그 표현 방식과 LLM의 학습 상태에 따라 원문 로그 기반, 사전학습 임베딩 기반, 파인튜닝 임베딩 기반의 세 가지 비교 조건을 정의한다. 첫째, 원문 로그 기반 조건은 자연어 로그 텍스트를 그대로 사용하는 기준선이다. 로그 파일에서 한 줄 단위로 로그를 읽어들인 후 개행 문자와 불필요한

공백을 정리하고, 토큰화 결과를 이상치 탐지 모델의 입력으로 사용한다. 둘째, 사전학습 임베딩 기반 조건은 사전학습 상태의 Llama-3.1-8B에서 로그 임베딩을 추출하여 사용하는 조건이다. 셋째, 파인튜닝 임베딩 기반 조건은 AIT Logdataset V2.0의 라벨링된 로그를 이용하여 Llama-3.1-8B를 지시 기반으로 파인튜닝한 후, 동일한 절차로 임베딩을 추출하는 조건이다.

세 조건 모두 동일한 데이터 분할, 동일한 이상치 탐지 모델, 동일한 학습과 평가 설정을 적용하며, 입력 표현만 변경한다. 이를 통해 로그 표현 방식과 LLM 학습 상태가 이상치 탐지 성능에 미치는 영향을 정량적으로 비교한다.

4.2 데이터셋 구성 및 파인튜닝 범위

실험에는 정상 동작, 공격 전 징후, 본격 공격 단계 등 다양한 시나리오가 포함된 AIT Logdataset V2.0을 사용한다. 각 로그에는 정상 여부와 함께 attacker_http, dirb, foothold, service_scan, escalate, webshell_cmd, webshell_upload, wpscan과 같은 공격 유형 태그가 부여되어 있다. 본 논문에서는 먼저 시나리오 단위로 파인튜닝 학습용 집합과 파인튜닝 비적용 집합을 구분한다. 파인튜닝 학습용 집합은 다시 학습 영역과 검증 영역으로 분할하여 Llama-3.1-8B의 지시 기반 대화형 파인튜닝에 사용한다. 파인튜닝에 사용하지 않는 시나리오 집합은 도메인 외 일반화 평가에만 사용한다.

로그와 임베딩 기반 데이터셋의 구성은 표 1과 같다. 표 1에서 D1부터 D3까지는 정상 라벨만 포함된 데이터셋이다. D1은 원문 로그 텍스트로 구성되며, D2와 D3은 각각 사전학습 Llama-3.1-8B와 파인튜닝 Llama-3.1-8B에서 추출한 4,096차원 로그 임베딩으로 구성된다. D1은 재구성 기반 오토인코더 학습에 사용하고, D2와 D3은 이상치 탐지 모델의 학습과 성능 비교에 사용한다. D4부터 D6까지는 정상과 비정상 라벨이 모두 포함된 데이터셋이다. D4는 원문 로그 텍스트를 기반으로 공격 단계별 라벨 분포와 공격 흐름을 분석하는 데 사용한다. D5와 D6은 각각 사전학습 임베딩과 파인튜닝 임베딩으로 구성되며, 동일한 라벨 구성을 유지한 상태에서 임베딩 표현 차이가 이상치 탐지 성능과 공격 유형별 분리도에 미치는 영향을 비교한다.

각 데이터셋 내부에서는 로그 라인 인덱스의 홀수와 짝수를 기준으로 학습 영역과 시험 영역을 분할한다. 이때 정상 로그와 비정상 로그의 비율이 두 영역에서 가능한 한 유사하도록 유지하여, 표현 방식의 차이만이 성능 차이에 영향을 주도록 설계한다.

표 1. 로그와 임베딩 데이터셋 구성
Table 1. Configuration of log and embedding datasets

ID	라벨 구성	데이터 표현	주요 용도
D1	정상	원문 로그 텍스트	오차 재구성 모델 학습
D2	정상	사전학습 Llama-3.1-8B 임베딩(4,096차원)	이상 탐지 모델 학습과 평가
D3	정상	파인튜닝 Llama-3.1-8B 임베딩(4,096차원)	이상 탐지 모델 학습과 평가
D4	정상/비정상	원문 로그 텍스트	공격 단계 분석과 라벨 분포 통계
D5	정상/비정상	사전학습 Llama-3.1-8B 임베딩(4,096차원)	사전학습 임베딩 기반 성능 비교
D6	정상/비정상	파인튜닝 Llama-3.1-8B 임베딩(4,096차원)	파인튜닝 임베딩 기반 성능 비교

4.3 로그 표현 방식 및 임베딩 추출

원문 로그 기반 조건에서는 별도의 임베딩 변환 없이 로그를 이상치 탐지 모델의 입력 특징으로 사용한다. 로그 텍스트를 한 줄 단위로 읽어 들인 뒤, 타임스탬프와 IP 주소는 각각 <TIME>, <IP> 토큰으로 치환하여 로그의 구조적 패턴은 유지하면서 개별 시스템과 시간 정보에 대한 의존성을 줄인다. 이후 토큰화 결과를 바탕으로 단어 빈도 기반 희소 벡터를 구성하여 벡터 입력 모델과 시퀀스 모델의 입력으로 사용한다.

LLM 임베딩 기반 조건에서는 동일한 정규화 로그 텍스트를 고정된 프롬프트 템플릿에 삽입하여 Llama-3.1-8B에 입력한다. 프롬프트는 로그를 정상 또는 특정 공격 유형으로 분류하라는 지시 문장과 로그 타입 정보를 포함하며, 전체 로그 라인을 하나의 사용자 발화로 취급한다.

사전학습 임베딩 조건에서는 사전학습 상태의 Llama-3.1-8B를 그대로 사용하고, 파인튜닝 임베딩 조건에서는 AIT Logdataset V2.0의 라벨링된 로그로 지시 기반 파인튜닝을 수행한 Llama-3.1-8B를 사용한다. 두 조건 모두 선택한 은닉 계층에서 로그 토큰에 대응하는 은닉 상태를 수집한 뒤 평균 풀링을 적용하여 4,096차원 고정 길이 로그 임베딩을 생성한다.

생성된 임베딩 벡터는 원문 로그 기반 조건에서의 로그 단위 벡터와 동일한 역할을 가지며, Isolation Forest, COPOD, 심층 오토인코더(Deep AutoEncoder), Deep SVDD, 에는 개별 로그 단위로 입력한다. LSTM-AE에서는 로그 임베딩 벡터를 시간 순서대로 정렬하고, 고정 길이 또는 슬라이딩 윈도우 단위의 시퀀스로 구성하여 입력으로 사용한다.

4.4 이상치 탐지 모델 구성

이상치 탐지 모델은 원문 로그, 사전학습 임베딩, 파인튜닝 임베딩에 대하여 동일한 구조와 학습 절차를 적용한다. 데이터 분할, 학습 에포크 수, 최적화 알고리즘과 같은 학습 설정을 모든 조건에서 고정하고, 입력 표현만 변경하여 표현 방식에 따른 성능 차이를 비교할 수 있도록 구성한다.

실험에 사용한 이상치 탐지 모델 구성은 표 2와 같다. 표 2에서 M1은 로그 시퀀스를 입력으로 사용하는 LSTM-AE이며, 정상 시퀀스에 대한 재구성 오차를 이상 점수로 사용한다. M2는 개별 로그 벡터를 입력으로 하는 심층 오토인코더로서 정상 로그의 재구성 오차를 이상 점수로 정의한다. M3은 Deep SVDD로, 정상 로그 임베딩이 하나의 중심에 밀집하도록 학습한 후 중심으로부터의 거리를 이상 점수로 사용한다.

M4와 M5는 비지도 트리 기반 모델이다. M4인 Isolation Forest는 평균 경로 길이에 기반한 점수를 이상도로 사용하고, M5인 COPOD는 꼬

표 2. 실험에 사용한 이상치 탐지 모델 구성
Table 2. Configure the outlier detection model used in the experiment

번호	모델명	입력 단위	학습 방식	이상 점수 정의
M1	LSTM-AE	로그 시퀀스	정상 시퀀스 재구성	시퀀스 재구성 오차
M2	Deep AutoEncoder	개별 로그 벡터	정상 로그 재구성	로그 단위 재구성 오차
M3	Deep SVDD	개별 로그 벡터	정상 로그 원클래스	임베딩 중심으로부터의 거리
M4	COPOD	개별 로그 벡터	비지도	꼬리 확률에 기반한 점수
M5	Isolation Forest	개별 로그 벡터	비지도	평균 경로 길이에 기반한 점수

리 확률에 기반한 점수를 이상도로 사용한다. 다섯 모델 모두 D1부터 D3까지의 데이터셋을 사용하여 정상 로그를 기반으로 학습하며, D4부터 D6까지의 데이터셋을 이용해 정상 로그와 비정상 로그에 대한, Precision, Recall, F1- Score 등 성능 지표를 산출한다.

4.5 파인튜닝 비적용 영역에서의 일반화 평가

LLM 임베딩 표현이 파인튜닝에 사용되지 않은 시나리오에서도 안정적인 이상치 탐지 성능을 유지하는지 평가한다. 이를 위하여 4.2절에서 분리한 파인튜닝 비적용 시나리오 집합을 도메인 외 평가 구간으로 정의한다.

평가 절차는 다음과 같이 진행한다. 먼저 파인튜닝 비적용 시나리오에 포함된 로그만 선택하고, 각 로그에 대하여 4.3절과 동일한 방식으로 원문 로그 기반 특징, 사전학습 임베딩, 파인튜닝 임베딩을 생성한다. 이후 세 표현을 D4부터 D6까지에 대응하도록 구성하여 표 2의 M1부터 M5까지의 모델에 입력하고, 로그 단위 또는 시퀀스 단위 이상 점수를 계산한다.

각 조합에 대해 비정상/정상 라벨을 기준으로 Precision, Recall, F1-Score를 산출하고, 파인튜닝에 사용된 시나리오 영역과의 성능 차이를 함께 측정한다. 이를 통해 원문 로그 표현, 사전학습 임베딩, 파인튜닝 임베딩 간의 도메인 간 일반화 격차를 정량화하고, 파인튜닝이 특정 시나리오에 대한 과적합을 유발하는지 또는 로그 도메인 전반의 표현력을 향상시키는지 분석한다. 나아가 정상 동작 구간, 공격 전 징후, 본격 공격 단계와 같은 시나리오 유형별 성능을 함께 관찰하여 제안 임베딩 기반 이상치 탐지 구조의 실질적인 적용 가능성을 검증한다.

4.6 실험 환경

제안 기법의 학습과 평가는 단일 GPU 워크스테이션에서 수행하였다. GPU는 NVIDIA RTX 3090(24 GB VRAM)을 사용하였고, CPU는 16코어 AMD Ryzen 9 5950X를 사용하였다. 시스템 메모리는 128 GB RAM으로 구성하였다. 운영체제는 64비트 Windows 환경이며, PyTorch 2.x와 Hugging Face Transformers, Datasets, PEFT 라이브러리를 이용하여 Llama 3.1 8B 모델의 파인튜닝과 로그 임베딩 추출을 수행하였다. LLM 파인튜닝은 배치 크기 16, gradient accumulation 단계 4, 학습 에폭 1로 설정하고 AdamW 옵티마이저와 16비트 부동소수점(FP16) 연산을 사용하였다. LoRA 어댑터는 self attention 모듈의 질의 계층과 값 투영 계층에만 삽입하여 파라미터 수를 제한하였다. 이상 탐지 알고리즘인 Deep AutoEncoder, Deep SVDD, LSTM AE는 PyTorch 기반으로 구현하였고, Isolation Forest와 COPOD는 scikit learn과 PyOD 라이브러리를 사용하여 구현하였다. 모든 실험은 동일한 하드웨어와 소프트웨어 환경에서 수행하여, 로그 표현 방식(D4~D6)과 이상 탐지 알고리즘 조합에 따른 성능 차이가 구현 환경이 아니라 입력 표현과 모델 구성의 차이에 의해 발생하도록 통제하였다.

5. 성능 검증

5.1 데이터셋 분석

본 논문에서는 단일 웹 서비스에서 수집한 약 80만 건의 HTTP 접근 로그를 데이터셋으로 사용하였다. 각 로그 라인은 공격 단계에 해당하는 여러 라벨을 동시에 가질 수 있으며, 동일한 요청에 대해 공격자의 초기 스캐닝 행위와 지속성 확보 행위가 함께 부여되는 경우도 존재한다. 그림 3은 로그 라인 인덱스에 따른 라벨 분포를 나타낸 것이다. 데이터 수집 구간 전반에 걸쳐

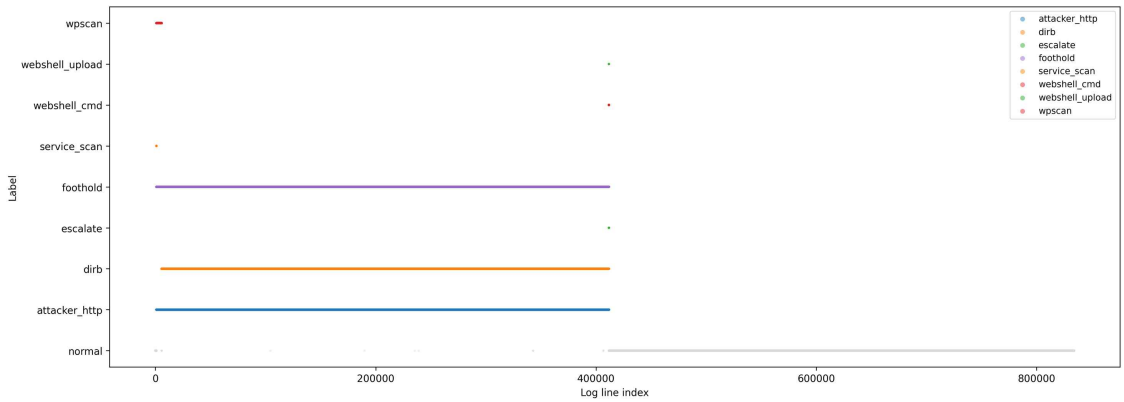

그림 3. 로그 데이터셋의 라벨별 시퀀스 분포 및 공격 단계별 발생 위치
Fig. 3. Label sequence distribution of the log dataset and occurrence positions of each attack stage

attacker_http, dirb, foothold 라벨이 연속적으로 등장하는 반면, wpscan, service_scan, webshell_cmd, webshell_upload, escalate 라벨은 특정 구간에 국소적으로 집중되어 나타난다. 이는 장기간에 걸친 지속적인 침투 시도와 함께, 일부 시점에 한정된 고위험 행위가 함께 기록된 시나리오 구조임을 뜻한다.

표 3. 라벨별 로그 샘플 수 분포
Table 3. Distribution of log samples by label

라벨	의미	샘플 수	비율(%)
wpscan	워드프레스 취약점 스캐닝	4,764	0.34
webshell_upload	웹쉘 업로드	3	<0.01
webshell_cmd	웹쉘 명령 실행	17	<0.01
service_scan	서비스 포트 스캔	12	<0.01
foothold	침투 후 지속성 확보 행위	410,837	28.9
escalate	권한 상승 시도	4	<0.01
dirb	디렉터리 열거·무차별 대입 공격	406,045	28.6
attacker_http	공격자 HTTP 요청·스캐닝	410,831	28.9
normal	일반 서비스 접근 로그	189,159	13.3

표 3은 각 라벨 별 샘플 수와 전체 대비 비율을 요약한 결과이다. foothold, attacker_http, dirb 라벨의 샘플 수는 각각 약 40만 건 수준으로 가장 크며, normal 라벨은 약 19만 건으로 전체의 약 13%를 차지한다. wpscan 라벨은 4천여 건에 불과하지만, 특정 취약점을 대상으로 하는 집중적인 취약점 스캐닝 행위를 반영한다. webshell_cmd, webshell_upload, service_scan, escalate 라벨은 샘플 수가 매우 적어 장기적으로 희소 공격 유형에 대한 검출 성능을 평가하는 데 활용된다. 로그 한 건이 복수의 라벨을 가질 수 있으므로 표 3의 샘플 수 합계는 실제 로그 라인 수보다 크며, 이는 다단계 공격 과정이 단일 요청 내에 함께 표현되는 데이터셋 특성을 반영한다.

그림 4는 원문 로그 기반 표현, 사전학습 언어 모델에서 추출한 임베딩 D5, 파인튜닝을 통해 얻은 임베딩 D6에 대해 각각 주성분 분석을 수행한 후 이차원 공간에 투영한 결과를 비교한 것이다. 원문 로그의 주성분 공간에서는 대부분의 라벨이 원점 부근에서 방사형으로 뻗는 선형 구조를 형성하며 서로 강하게 중첩되어, 클래스 간

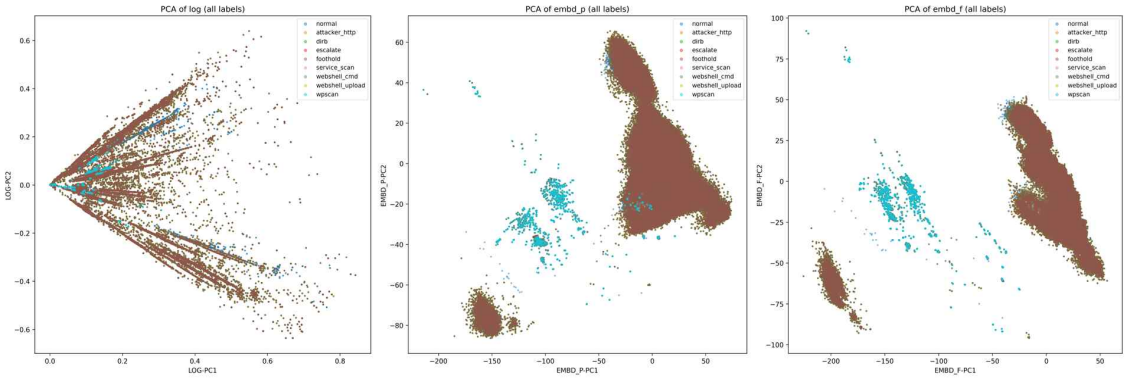

그림 4. D1, D4(좌측), D2, D5(중앙), D3, D6(우측) 전체 라벨 PCA 분석 시각화
Fig. 4. PCA visualizations with all labels for D1 and D4 (left), D2 and D5 (center), and D3 and D6 (right)

공간적 분리가 거의 관찰되지 않는다. 이에 비해 D5와 D6에서는 라벨에 따라 여러 개의 밀집 클러스터가 형성되고 normal 라벨과 주요 공격 라벨이 부분적으로 분리되는 양상이 나타난다. 특히 attacker_http, dirb, foothold, wpscan 라벨은 주성분 공간에서 상대적으로 응집된 군집을 이루며, D6의 경우 D5에 비해 군집 경계가 다소 뚜렷하다. 이는 언어 모델 기반 임베딩이 원문 로그 표현보다 공격 유형 간 구조적 차이를 더 잘 보존하며, 파인튜닝을 통해 동일 공격 시나리오 내 세부 패턴이 보다 선명하게 분리되는 방향으로 표현 공간이 재구성되었음을 시사한다.

5.2 원본 로그 텍스트 기반 이상 탐지 성능

표 4는 D4 조건에서 다섯 가지 이상 탐지 모델의 이진 분류 성능을 요약한 결과이다. 모든 모델에서 F1-score가 0.16 이하에 머무르며, 그 중 COPOD가 F1-score 0.156으로 가장 높은 값을 보인다. 실험에서 LSTM-AE와 Deep AutoEncoder는 각각 정밀도 0.701, 0.620을 기록하지만 재현율이 0.023, 0.017 수준에 불과하여 F1-score는 0.044, 0.034에 그쳤다. Deep SVDD는 재현율과 F1-score가 0으로 수렴하여 사실상 모든 샘플을 정상으로 판단하는 퇴행 해(collapse) 양상을 보이며, Isolation Forest 또한

표 4. D4 로그 텍스트 기반 이상 탐지 성능
Table 4. Anomaly detection performance on D4

모델 \ 지표	Acc	Prec	Rec	F1
LSTM-AE	0.324	0.701	0.023	0.044
Deep AE	0.320	0.620	0.017	0.034
Deep SVDD	0.315	0.000	0.000	0.000
COPOD	0.349	0.694	0.088	0.156
Isolation Forest	0.319	0.577	0.018	0.036

정밀도 0.577, 재현율 0.018로 F1-score 0.036을 나타냈다. 전반적으로 원문 로그 텍스트만을 입력으로 사용할 경우, 서로 다른 알고리즘 간 차이는 존재하나 어느 경우에도 실용적인 수준의 이진 검출 성능에 도달하지 못함을 확인하였다.

5.3 사전학습 임베딩 기반 이상 탐지 성능

표 5는 사전학습 D5 조건을 입력으로 사용했을 때 각 이상 탐지 모델의 성능을 요약한 결과이다. 모든 알고리즘에서 원본 로그 텍스트(D4)에 비해 이진 F1-score가 증가하였다. 실험에서, Deep AutoEncoder는 정확도 0.328, 정밀도 0.515, 재현율 0.316, F1-score 0.392를 기록하여 가장 높은 F1-score를 달성하였다. Deep SVDD와

표 5. D5 사전학습 임베딩 기반 이상 탐지 성능
Table 5. Anomaly detection performance on D5

모델 \ 지표	Acc	Prec	Rec	F1
LSTM-AE	0.327	0.532	0.149	0.233
Deep AE	0.328	0.515	0.316	0.392
Deep SVDD	0.327	0.524	0.195	0.284
COPOD	0.349	0.694	0.088	0.156
Isolation Forest	0.333	0.547	0.148	0.232

LSTM-AE, Isolation Forest는 각각 F1-score 0.284, 0.233, 0.232를 보여 Deep AutoEncoder 다음으로 높은 성능을 나타낸다. COPOD의 F1-score는 0.156으로 D4와 유사한 수준이지만, 정밀도가 0.694로 높아 순위 기반 탐지 관점에서는 의미 있는 성능을 유지하였다.

실험에서는 동일 알고리즘에 대해 D4와 D5를 비교하여, 사전학습 임베딩의 효과를 확인하였다. Deep AutoEncoder의 F1-score는 원문 로그 기반 조건에서 0.034에 불과하였으나, D5의 경우 0.392로 약 10배 이상 상승하였다. LSTM-AE와 Isolation Forest 역시 각각 0.044에서 0.233, 0.036에서 0.232로 증가하여, 표현 방식만 변경하여도 성능 향상이 발생함을 확인하였다. D4에서 비정상 로그를 거의 검출하지 못한 Deep SVDD는 D5 조건에서 F1-score 0.284까지 개선되어, 사전학습 임베딩이 단일 분류 기반 모델에도 충분한 정보를 제공함을 보였다.

전반적으로 사전학습 LLM 임베딩을 이용할 경우 재현율이 원문 로그 대비 크게 증가하는 동시에 정밀도도 0.5 내외 수준을 유지하여, 이진 F1-score 기준에서 균형 잡힌 탐지 성능을 확보한다. 공격 유형별 세부 분석에서도 attacker_http, dirb, foothold와 같은 주요 라벨에 대해 F1-score가 0.38 전후로 수렴하는 경향이 나타나며, 원문 로그 조건에서의 F1-score 0.03대와 비교하면 실질적인 운용 관점에서 의미 있는 개선으로 해석된다. 이러한 결과는 사전학습 LLM 임베딩이 보안 로그의 문맥 구조를 벡터 공간에서 적절히 분리하여, 다양한 비지도 이상 탐지 모델의 공통 입력 표현으로 기능할 수 있음을 뒷받침한다.

5.4 파인튜닝 임베딩 기반 이상 탐지 성능

표 6은 D6을 입력으로 사용했을 때 각 이상 탐지 모델의 이진 분류 성능을 요약한 결과이다. Deep AutoEncoder는 정확도 0.329, 정밀도 0.515, 재현율 0.314, F1-score 0.390으로 가장 높은 F1-score를 나타냈으며, Deep SVDD는 정밀도 0.506, 재현율 0.307, F1-score 0.383, PR AUC 0.589로 두 번째로 높은 성능을 보였다. Isolation Forest와 LSTM-AE는 각각 F1-score 0.254, 0.212 수준이며, COPOD는 F1-score 0.202로 상대적으로 낮지만, 정밀도와 PR AUC 측면에서는 여전히 일정 수준 이상의 순위를 유지하였다. 전반적으로 D6에서는 Deep AutoEncoder와 Deep SVDD가 상위권을 형성하고, 나머지 모델이 뒤따르는 구조를 보인다.

D4와 비교하면, Deep AutoEncoder의 F1-score는 0.034에서 0.390으로 약 11배 이상 증가하고, Deep SVDD는 사실상 비정상을 검출하

표 6. D6 파인튜닝 임베딩 기반 이상 탐지 성능
Table 6. Anomaly detection performance on D6

모델 \ 지표	Acc	Prec	Rec	F1
LSTM-AE	0.327	0.536	0.132	0.212
Deep AE	0.327	0.515	0.314	0.390
Deep SVDD	0.328	0.516	0.305	0.383
COPOD	0.343	0.602	0.121	0.202
Isolation Forest	0.331	0.537	0.166	0.254

지 못하던 상태에서 F1-score 0.383까지 상승한다. Isolation Forest와 LSTM-AE은 각각 0.036에서 0.254, 0.044에서 0.212로 향상되어, 표현 방식만 변경해도 대부분의 모델에서 F1-score가 개선되는 양상을 보였다. COPOD의 경우 0.156에서 0.202로 개선 폭은 상대적으로 작지만, D4 조건과 달리 재현율이 의미 있는 수준으로 회복되어 실질적으로 이진 탐지 성능이 향상되었음을 확인하였다.

한편 표 5와 표 6에서 모든 모델의 정확도는 대략 0.33 수준에 머문다. 이는 데이터셋에서 정상 로그 비율이 약 13%로 매우 낮은 반면 비정상 로그가 다수를 차지하는 심한 라벨 불균형 구조이며, 공격 로그를 놓치지 않도록 재현율을 우선하는 방향으로 임계값을 설정한 결과이다. 이 설정에서는 정상 구간에 대한 오탐지 증가로 전체 정확도가 낮게 나타나지만, 악성 코드 이상 징후 탐지 관점에서는 F1-score와 PR AUC가 성능을 판단하는 보다 적절한 지표이다. 실제 운영 환경에서는 라벨 불균형을 완화하기 위한 정상 로그 샘플 추가 확보, 비용 민감 임계값 조정, 규칙 기반 필터와의 하이브리드 탐지 구조를 결합하여 오탐지를 줄이면서 정확도를 보완하는 방향으로 확장할 수 있다.

5.5 실험 결과 분석

표 7은 입력 표현(D4~D6)과 이상 탐지 모델별 이진 F1 점수를 비교한 결과이다. 원본 로그 텍스트를 사용한 D4에서는 모든 모델의 bin-F1이 0.16 이하에 머무르며, Deep AE 0.034, Deep SVDD 0.000, Isolation Forest 0.036과 같이 전반적으로 성능이 낮다. 이에 비해 사전학습 임베딩(D5)과 파인튜닝 임베딩(D6)을 사용한 경우 bin-F1이 일괄적으로 상승한다. Deep AE는 D4에서 0.034에 불과하지만 D5와 D6에서 각각 0.392, 0.390을 기록하며, Deep SVDD 역시 0.000에서 0.284, 0.383으로 크게 증가한다. LSTM-AE와 Isolation Forest도 D5, D6에서 0.23 내외 수준으로 개선되며, COPOD는 D4와 D5가 0.156으로 동일하지만 D6에서 0.202까지 상승한다. 표현 방식을 D4에서 D5·D6로 변경했을 때의 F1 증가폭이 알고리즘 간 차이보다 훨씬 크다는 점에서, 본 실험에서는 모델 선택보다 LLM 임베딩 기반 표현 설계가 성능을 결정하는 주된 요인임을 확인할 수 있다.

표 8은 워드프레스 취약점 스캔(wp_scan) 라벨에 대한 F1 점수를 정리한 결과이다. 해당 라벨은 전체 로그에서의 비율이 0.3% 수준에 불과한 희귀 공격 유형으로, 모든 표현·모델 조합에서

표 7. D4~D6 및 알고리즘별 F1-score 비교
Table 7. F1-score comparison across D4 - D6 and algorithms

지표 / 모델	bin-F1 (D4)	bin-F1 (D5)	bin-F1 (D6)
LSTM-AE	0.044	**0.233**	0.212
Deep AE	0.034	**0.392**	0.390
Deep SVDD	0.000	0.284	**0.383**
COPOD	0.156	0.156	**0.202**
Isolation Forest	0.036	0.232	**0.254**

표 8. wp_scan 라벨에 대한 입력 표현별 F1-Score 비교
Table 8. Comparison of F1-scores for the wp_scan label across input representations

지표 / 모델	wp_scan F-1(D4)	wp_scan F-1(D5)	wp_scan F-1(D6)
LSTM-AE	0.031	0.079	**0.090**
Deep AE	0.000	**0.037**	**0.037**
Deep SVDD	0.000	**0.060**	0.038
COPOD	0.029	0.100	**0.109**
Isolation Forest	0.000	**0.082**	0.072

F1이 0.11 이하의 낮은 값을 보인다. 그럼에도 불구하고 원본 로그(D4)에 비해 임베딩 기반 표현(D5, D6)에서는 일정 수준의 개선이 나타난다. 예를 들어 LSTM-AE는 D4에서 0.031이지만 D5와 D6에서 0.079, 0.090으로 증가하고, COPOD는 0.029에서 0.100, 0.109로 상승한다. 이는 LLM 임베딩이 저빈도 공격 라벨에 대해서도 일정 부분 분리 능력을 제공했음을 의미한다.

6. 결 론

대규모 정보 시스템과 웹 서비스 환경에서 로그 기반 이상 탐지는 랜섬웨어를 비롯한 악성코드 침투를 조기에 식별하기 위한 핵심 수단이다. 그러나 원본 로그 텍스트와 제한적인 특징 공학에 의존하는 기존 비지도 이상 탐지 구성만으로는, 라벨 불균형과 복잡한 공격 단계가 뒤섞인 실제 보안 로그에서 초기 이상 징후를 안정적으로 포착하기 어렵다. 이에 본 논문에서는 약 80만 건의 웹 접근 로그와 감사 로그를 대상으로, 대규모 언어모델에서 로그 행 단위 임베딩을 추출하고 사전학습 임베딩과 보안 로그 도메인 파인튜닝 임베딩을 통계 기반 및 심층 표현 학습 기반 이상 탐지 모델의 공통 입력으로 사용하는 LLM 로그 임베딩 기반 표현 학습 절차를 제안한다. 실험 결과, 원본 로그 텍스트만을 사용할 때 모든 알고리즘의 이진 F1-score가 0.16 이하에 머문 반면, 제안 임베딩 기반 표현을 적용한 경우 대표 조합인 심층 오토인코더에서 이진 F1-score가 약 2.5배 향상되었고 빈도 수가 적은 공격 유형에 대해서도 F1-score가 크게 증가하였다. 또한 주성분 분석에서 임베딩 기반 표현이 정상 로그와 주요 공격 행위 간 군집을 보다 뚜렷하게 분리하는 것으로 나타나, 제안 기법이 악성코드 이상 징후 조기 탐지를 위한 유효한 설계 기준으로 기능함을 확인하였다. 향후에는 다양한 데이터셋을 대상으로 표현 학습 전략과 이상 탐지 모델 구성을 비교하는 추가 연구를 수행할 예정이다.

이 논문은 2025년도
정부(과학기술정보통신부)의 재원으로
정보통신기획평가원의 지원을 받아 수행된
연구임 (No. RS-2024-00469698, 사이버보안
해외진출 촉진을 위한 해외시장 수요기반
국제공동 기술개발)

참 고 문 헌

[1] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey”, Machine Learning with Applications, vol. 12, p. 100470, 2023, DOI: https://doi.org/10.1016/j.mlwa.2023.100470

[2] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning”, in Proc. 24th ACM Conf. Computer and Communications Security (CCS), 2017, pp. 1285 - 1298, DOI: https://doi.org/10.1145/3133956.3134015

[3] H. Guo, S. Yuan, and X. Wu, “LogBERT: Log anomaly detection via BERT”, 2021 international joint conference on neural networks (IJCNN). IEEE, 2021, DOI: https://ieeexplore.ieee.org/document/9534113/

[4] Y. Lee, J. Kim, and P. Kang, “LAnoBERT: System log anomaly detection based on BERT masked language model”, Applied Soft Computing, vol. 146, p. 110689, 2023, DOI: https://doi.org/10.1016/j.asoc.2023.110689

[5] W. Guan, J. Cao, S. Qian, J. Gao, and C. Ouyang, “LogLLM: Log-based anomaly detection using large language models”,

arXiv preprint arXiv:2411.08561, 2024, DOI: https://doi.org/10.48550/arXiv.2411.08561

[6] M. De la Cruz Cabello, T. P. Sales, and M. R. Machado, “AIOps for log anomaly detection in the era of LLMs: A systematic literature review”, Intelligent Systems with Applications, vol. 22, p. 200608, 2025, DOI: https://doi.org/10.1016/j.iswa.2025.200608

[7] J. Lim, J. Ahn, and H. Lee, “Adapting large language models for parameter-efficient log anomaly detection”, Pacific-Asia Conference on Knowledge Discovery and Data Mining. Singapore: Springer Nature Singapore, 2025, DOI: https://doi.org/10.1007/978-981-96-8298-0_26

[8] M. Liu et al., “LLM meets ML: Data-efficient anomaly detection on unstable logs”, ACM Transactions on Software Engineering and Methodology, 2025, DOI: https://doi.org/10.1145/3771283

[9] C. Han, et al., “Few-shot log anomaly detection based on matching networks”, IEEE Transactions on Network and Service Management vol. 21, no. 3, pp. 2909-2925, 2024, DOI: https://doi.org/10.1109/TNSM.2024.3363626

[10] J. Rajkarnikar, N. Poudel, and N. Rahimi, “Unsupervised anomaly detection in OpenStack logs via fine-tuned RoBERTa embeddings”, J. Cybersecurity, Digit. Forensics Jurisprudence, vol. 1, pp. 9 - 20, 2025, URL: https://www.cdfjjournal.com/index.php/cdfj/article/view/3/2

[11] L. Pang et al., “Large language model based optical network log analysis using LLaMA2 with instruction tuning”, Opt. Express, vol. 32, no. 6, pp. 7809 - 7823, 2024, DOI: https://doi.org/10.1364/JOCN.527874

[12] Z. Zhang et al., “LLM-LADE: Large language model-based log anomaly detection with explanation”, Knowl.-Based Syst., vol. 326, p. 114064, 2025, DOI: https://doi.org/10.1016/j.knosys.2025.114064

[13] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest”, in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, pp. 413 - 422, 2008, DOI: https://doi.org/10.1109/ICDM.2008.17

[14] Y. Li et al., “COPOD: Copula based outlier detection”, in Proc. IEEE ICDM, 2020, DOI: https://doi.org/10.1109/ICDM50108.2020.00135

[15] Y. Boateng et al., “Deep one class classification model assisted by radius constraint for ICS anomaly detection”, Eng. Appl. Artif. Intell., vol. 126, 2024, DOI: https://doi.org/10.1016/j.engappai.2024.109357

[16] M. Alaghbari et al., “Deep autoencoder based integrated model for anomaly detection and efficient feature extraction in IoT networks”, Comput., vol. 4, no. 3, pp. 255 - 273, 2023, DOI: https://doi.org/10.3390/iot4030016

[17] H. Rhachi, Y. Balboul, and A. Bouayad, “Enhanced anomaly detection in IoT networks using deep autoencoders with feature selection techniques”, Sensors, vol. 25, no. 10, p. 3150, 2025, DOI: https://doi.org/10.3390/s25103150

[18] D. Shin et al., “LSTM autoencoder based detection of time series noise signals for water supply and sewer pipe leakages”, Water, vol. 16, no. 18, p. 2631, 2024, DOI: https://doi.org/10.3390/w16182631

[19] M. N. Amin et al., “Real-time anomaly detection with LSTM-Autoencoder network on microcontrollers for industrial applications”, in Proc. 8th Int. Conf. Graphics Signal Process. (ICGSP), 2024, DOI: https://doi.org/10.1145/3694875.36948

[20] Meta AI, “Llama 3.1 - 8B”, Hugging Face, 2025. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.1-8B

[21] M. Landauer, F. Skopik, M. Frank, W. Hotwagner, M. Wurzenberger, and A. Rauber. “Maintainable Log Datasets for Evaluation of Intrusion Detection Systems”. IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 3466-3482, doi: 10.1109/TDSC.2022.3201582.

저 자 소 개

김동완(DongWan Kim)

2022.02 한국성서대학교 컴퓨터소프트웨어학과 졸업
2024.02 숭실대학교 컴퓨터학과 석사
2024.01 - 현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 주임 연구원
<주관심분야> 블록체인, 시계열 분석, 빅데이터, 인공지능, 자연어처리, AIOps

김현수(Hyun-Soo Kim)

2019.02 단국대학교 소프트웨어학과 졸업
2023.08 숭실대학교 AI·SW융합학과 석사
2024.03-현재 숭실대학교 AI·SW융합학과 박사과정
2019.01-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 연구팀장

박경엽(Kyung-Yeob Park)

2019.2 서울과학기술대학교 컴퓨터공학과 석사
2019-현재 엘에스웨어(주) 선임 연구원
<주관심분야> IoT 보안, 블록체인, 빅데이터, 메타버스

김민수(MinSoo Kim)

1996.02 성균관대학교 화학과 졸업
2002.02 성균관대학교 컴퓨터공학과 석사
2002 - 2005 데이터게이트 인터네셔널 연구소장
2005 - 현재 엘에스웨어㈜ 대표이사
2015 - 현재 한국정보보호산업협회(KISIA) 이사
2024 - 현재 벤처기업 확인위원회 위원
2024 - 현재 정보보호 인적자원개발위원회 위원

신동명(Dong-Myung Shin)

2003.02 대전대학교 컴퓨터공학과 박사
2001-2006 한국정보보호진흥원 응용기술팀 선임연구원
2006-2014 한국저작권위원회 저작권기술팀 팀장
2014-2016 한국스마트그리드사업단보안인증팀 팀장
2016-현재 엘에스웨어(주) 소프트웨어연구소 연구소장/전무이사