AI 프레임워크 기반 ML 프로젝트 이슈 정량적 리스크 평가에 관한 실증 연구

김태영*, 박진만, 서영진
*국방기술품질원 품질기획실
e-mail:rlaxodud1200@dtaq.re.kr, (박진만)jinman.bak@dtaq.re.kr
, (서영진)mulle0514@dtaq.re.kr

An Empirical Study on Quantitative Risk Assessment of AI Framework-based ML Project Issues

Taeyoung Kim*, Jinman Baek, Youngjin Seo
*Dept. of Quality Planning, Defense Agency for Technology and Quality

요 약

본 연구는 AI 프레임워크 기반 ML 프로젝트 이슈 데이터 기반으로 수집·요약한 후, UMAP과 HDBSCAN을 활용해 결함 패턴을 식별한다. 이후 BERTopic을 통해 패턴 내 주요 내용을 도출하고, 발생 가능성과 영향도를 기준으로 정량적 위험 평가를 수행하였다. 분석 결과, 모델 학습 오류, 분산 학습 문제 등이 고위험군으로 나타났으며, 이는 AI 프레임워크 기반 시스템 개발 시 주요 리스크를 사전 식별하고 대응 전략을 마련하는 데 기여할 수 있음을 확인하였다.

1. 서론

Keras, TensorFlow, PyTorch 등 대표적 AI 프레임워크의 비약적 발전은 머신러닝(ML)·딥러닝 응용을 폭넓게 확산시켰다. ISO/IEC 23053:2022가 제시하는 AI 시스템 수명주기 참조 모델과 ISO/IEC 42001:2023이 요구하는 AI 관리 시스템 관점에서도 이들 프레임워크는 권장 구현 수단으로 자리매김하고 있다[1][2].

그러나 표준에서 권장된다는 사실이 곧바로 현업 운용의 안전성을 담보하는 것은 아니다. 실무 환경에서는 프레임워크 자체의 버그뿐 아니라, 이를 기반으로 개발된 애플리케이션·서비스·도구 전반에 걸쳐 다양한 이슈가 발생하며, 이러한 문제들이 결국 더 큰 운영 리스크로 이어진다.

예를 들어 프레임워크의 버전 불일치로 인해 PyTorch 모델을 TensorRT로 변환한 뒤 추론 결과가 모두 동일한 값으로 왜곡되는 치명적 정확도 저하가 된 사례가 있으며[3], GPU 메모리 부족 때문에 YOLOv8 추론 서비스가 반복적으로 중단된 이슈도 있다[4]. 더나아가, CUDA 드라이버/라이브러리 버전 불일치로 CrashLoopBackOff 상태에 빠져 자동 복구가 지연되기도 한다[5]. 이러한 문제는 방위산업·의료·자동차처럼 안전성 요구가 높은 도메인에서 AI 프레임워크 도입을 주저하게 만드는 요인으로 작용할 수 있다.

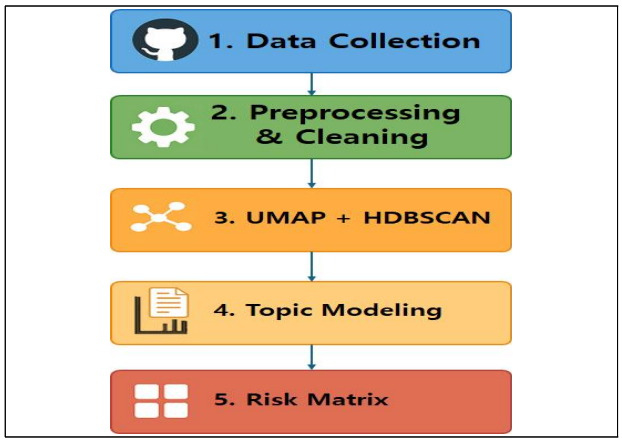
이러한 문제를 분석하기 위해 기존 연구들은 AI 프레임워크와 그 활용 프로젝트에서 발생하는 결함을 다각도로 탐구하였다. 예를 들어 Tambon 등[6]은 Keras와 TensorFlow의 핵심 API를 대상으로, 실행 중 오류 메시지는 나타나지 않으나 결과값을 왜곡해 성능을 잠식하는 silent bugs을 정밀 추적하였다. Long · Chen[7]은 이슈를 성능·정확도 버그로 구분하고 각 버그의 보고 빈도와 해결 시간을 분석하였다. 또한 Lai 는[8] 다양한 오픈소스 저장소의 이슈를 범주별·빈도별로 비교해 프로젝트 유형에 따른 결함 특성을 탐색하였다. 그러나 이 연구들은 각 프레임워크를 도입 시 반복적으로 재현되는 이슈들의 제공이 어려우며 계량적 위험 지표를 제시하지 않아 결함 간 상대적 우선순위를 가늠할 수 없었다.

본 연구는 이러한 한계를 보완하기 위해 PyTorch, TensorFlow, Keras를 활용한 프로젝트의 이슈를 LLM을 통해 요약화 하고 임베딩한 후 UMAP(Uniform Manifold Approximation and Projection) 과 HDBSCAN(Hierarchical Density-Based Spatial Clustering of Applications with Noise)을 통해 군집화를 수행한다. 다음으로 BERTopic을 활용해 각 군집 별 주제어를 추출하여 각 프레임워크를 활용하는 프로젝트 들에서 반복적으로 나타나는 결함 패턴을 식별한다. 마지막으로 군집별 상대 빈도(발생가능성)와 공통 심각도 척도(영향도)를 정의 후 Risk Matrix를 구성하여 발생된 결함들에 대한 계

량적 위험도를 제시하여 AI 프레임워크를 기반으로 시스템 개발 시 발생할 수 있는 이슈들과 위험도를 제시해 사전 대응을 할 수 있도록 한다.

2. 실험 설계

본 장에서는 AI 프레임워크 기반 오픈소스 프로젝트의 문제 영역을 파악하기 위해 다음 그림 1과 같이 실험을 설계하였다.



[그림 1] 실험 설계 절차

먼저, GitHub REST API로 Keras·TensorFlow·PyTorch 저장소를 탐색해 이슈를 크롤링한 뒤, ChatGPT로 본문을 요약하고 SentenceTransformer 기반 임베딩으로 변환한다. 이어 UMAP으로 차원 축소 후 HDBSCAN으로 밀도 기반 군집을 형성한 후 BERTopic을 통해 군집 별 반복 오류 패턴을 도출한다. 마지막으로 이슈 발생 빈도와 해결 기간을 바탕으로 확률과 영향도를 산정하고, 리스크 매트릭스로 등급화하여 AI 프레임워크 적용 시 고려·보완해야 할 사항들을 제시한다.

2.1 이슈 수집

이 단계에서는 AI 프레임워크와 연관된 오픈소스 저장소에서 이슈 데이터를 수집한다. 먼저 GitHub에서 각 프레임워크 키워드로 저장소를 검색한 뒤, 이름이나 설명에 “tutorial” “handbook” “UI” 등이 포함된 학습용·문서용 레포지토리는 제외한다. 또한 전체 공개 이슈 중 영어 데이터가 많으므로 연구 대상의 일관성을 확보하기 위해 영어 이외 언어가 포함된 저장소 및 이슈도 제외하였다.

선정된 저장소에서 이슈의 title, body, state (open/closed), resolution time 등을 수집한다. 이러한 데이터는 문제의 내용과 복잡도뿐 아니라 개발자 참여 수준과 해결에 든 시간을 파악하여 복합적인 문제 유형과 프레임워크 별 해결 난이도를 식별할 수 있으며, 각 위험 요소를 정량적으로 분석할 수 있다.

2.2 데이터 가공 및 정제

이슈는 종종 장황하고 불필요한 세부 정보가 포함되어 있어 그대로는 활용이 어렵기 때문에 대규모 언어 모델(ChatGPT-4o)을 사용해 각 이슈의 title과 body를 그림 2와 같이 정의 된 프롬프트에 따라 요약한다.

> Please summarize the following issue title and description in a concise manner, focusing on the key points, problem descriptions, and any technical details that are essential for understanding the issue. If possible, highlight the most relevant aspects of the issue.

[그림 2] 이슈 요약 시스템 프롬프트

요약된 이슈는 SentenceTransformer를 사용하여 벡터 데이터로 변환한다. 이는 대용량 텍스트 데이터를 학습한 Transformer 기반의 Encoder 모델로 문제 설명의 맥락과 뉘앙스를 포착할 수 있다[9]. 그러므로 모델은 HuggingFace leader board 10위 안에 드는 intfloat/ multilingual-e5-large-instruct를 사용한다.

2.3 UMAP과 HDBSCAN을 활용한 군집화

이슈 군집화 위해 임베딩 값들을 StandardScaler로 분산 편차에 따른 거리 왜곡을 제거한 후 UMAP[10]을 적용하였다. 스케일을 통일한 뒤 고차원을 저차원으로 투영하면, 데이터의 지역-글로벌 구조를 동시에 보존하면서도 군집화 연산이 빨라지기 때문이다. 그 후 HDBSCAN을 사용하여 군집화를 수행한다. HDBSCAN은 계층적 접근으로 서로 다른 밀도의 군집을 탐지하고, 군집에 속하지 않는 점을 노이즈로 처리해 유사 이슈를 명확히 구분할 수 있다[9].

2.4 BERTopic 기반의 Topic Modeling

군집화 후, BERTopic을 사용하여 군집된 이슈에 대한 Topic Modeling을 수행한다. BERTopic은 사전 계산된 임베딩과 UMAP 변환을 활용하여 문제 내 잠재적 주제를 파악한다[11]. 이 단계를 통해 군집 내 핵심 내용을 식별할 수 있다.

2.5 위험 식별 Matrix

식별된 군집을 기반으로 정량적 위험 매트릭스를 만들기 위해 발생 가능성 P 와 영향성 I를 정의한다. 발생 가능성을 정의하기 위해 모든 이슈에 대한 군집 k의 이슈의 빈도를 활용해 아래 수식과 같이 정의한다.

$$P_k = \frac{n_k}{N_{total}},\ N_{total} = \Sigma_i n_i$$

여기서 n_k는 군집 k에 속한 이슈의 개수를 나타내며 N_{total}은 AI 프레임워크 별 ML 프로젝트 이슈들의 총 개수를 나타낸다.

다음으로 MIL-STD-882E[12] and NASA guidance[13]에서 제시하는 thresholds 기준에 따라 구간을 설정하였다.

$$B(P_k) = \begin{cases} P_1, & 0 \le P_k < 0.01, \\ P_2, & 0.01 \le P_k < 0.05, \\ P_3, & 0.05 \le P_k < 0.15, \\ P_4, & P_k \ge 0.15. \end{cases}$$

영향성을 정의하기 위해 각 군집 k 별 $MTTR_k$를 삼분위수로 나누어 다음 수식과 같이 영향도를 구분하였다.

$$I_k = \begin{cases} I_1, & \mathrm{MTTR}_k \le q_{33}, \\ I_2, & q_{33} < \mathrm{MTTR}_k \le q_{66}, \\ I_3, & \mathrm{MTTR}_k > q_{66}. \end{cases}$$

마지막으로 발생 가능성과 영향성을 맵핑해 그림 3과 같이 위험도를 산출한다.

발생 가능성 p (0 ≤ p ≤ 1)	영향도 Impact (MTTR 기준)		
	I₁ Low (MTTR ≤ Q₃₃)	I₂ Medium (Q₃₃ < MTTR ≤ Q₆₆)	I₃ High (MTTR > Q₆₆)
P₁ Rare (0 ≤ p < 0.01)	L1 (허용)	L2 (허용)	M1 (모니터)
P₂ Occasional (0.01 ≤ p < 0.05)	L2 (허용)	M2 (모니터)	H1 (조치)
P₃ Frequent (0.05 ≤ p < 0.15)	M1 (모니터)	H1 (조치)	H2 (차단)
P₄ Systemic (p ≥ 0.15)	H1 (조치)	H2 (차단)	H3 (전면 재검토)

[그림 3] AI 프레임워크 별 이슈 위험 평가 Matrix

3. 실험

설계한 실험 방법을 기반으로 연구를 수행하기 위해 우리는 다음과 같은 연구 질문을 정의한다.

- RQ1. 어떤 종류의 이슈 패턴들이 AI 프레임워크 기반 프로젝트들로부터 식별되었는가?
- RQ2. 리스크 매트릭스를 적용 시 위험 등급 분포는 어떻게 되는가?

실험 데이터는 장수의 제한으로 인해 Keras 기반 프로젝트만을 대상으로 실험을 진행하였다. 총 64,751건 이슈(862개 프로젝트)를 수집하였고 2.1.1에서 소개한 절차를 통해 필터링을 수행하여 60,625건을 최종적으로 사용한다.

3. 실험

3.1 RQ1. 어떤 종류의 이슈 패턴들이 AI 프레임워크 기반 프로젝트들로부터 식별되었는가?

우리는 표 1과 같이 총 34개의 군집을 식별했으며 BERTopic을 통해 군집 별 주제어를 뽑아냈다.

[표 1] Keras 기반 ML 프로젝트 군집과 주제어

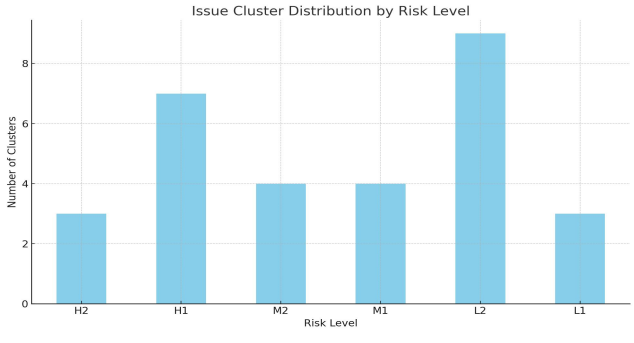
군집	주제어	개수
1	issue, user, model, training, loss	4,832
2	runs, weights biases, logging, wandb	3,062
3	operator, onnx model, conversion	2,081
4	layer, dimensions, model, input, shape	1,923
5	mpi, distributed training horovod	1,828
6	cpu, performance, training, gpu	990
7	object, keras, attribute error	974
8	output, masking, input, layer, lstm	1,096
9	function, model, type, error	737
10	tqdm library, progress bar, progress	524
11	library, import error, module	661
12	memory usage, allocation, gpu, memory	548
13	conversion, tflite, tensorflow, model, quantization	428
14	model, tasks, keras, models, bert	487
15	script, error, path, directory, file	327
16	learning, tuning, rate, hyperparameter	296
17	mmdnn, convert, model, caffe, conversion	204
18	tensorflow, versions, compatibility, python	276
19	layer, serialization, loading, custom, model	327
20	index range, list, index error	274
21	pip, version, package, install, installation	252
22	cmake, installation, build, protobuf, onnx	324
23	data, bounding, data augmentation, image	287
24	voice, speech, synthesis, merlin, audio	198
25	training process, epoch, process, generator	222
26	camera, throttle, donkeycar	213
27	mismatch, load, model, weight, weights	199
28	saved, save, training, model, saving	191
29	adversarial examples, attacks, foolbox	240
30	tensorflow, cuda, error, cudnn, gpu	196
31	dictionary, error, model, keyerror, talos	196
32	slack, slack channel, keras, users, github	219
33	project, issue titled, docstrings, documentation	181
34	bn, batch normalization	145

대표적 패턴으로는 모델 학습 과정에서의 loss 관련 오류(군집 1), ONNX, TFLite, Caffe 등의 모델 변환과 호환성 문제(군집 3, 13, 17), layer의 입력 차원(dimension)과 shape 불일치(군집 4, 8, 27)가 있었다. 또한 GPU의 메모리 할당 및 성능 문제(군집 6, 12), TensorFlow, CUDA 라이브러리 및 패키지 설치 과정에서의 호환성 오류(군집 18, 21, 22, 30)도 자주 나타났다. 그 밖에 hyperparameter 튜닝(군집 16), 모델의 serialization 및 loading 이슈(군집 19), distributed training 환경 문제(군집 5), adversarial attack 관련 문제(군집 29), documentation 미흡(군집 33) 등도 확인되었다. 이러한 결과는 AI 프레임워크 사용성과 신뢰성을 높이기 위해 중점적으로 개선이 필요한 영역을 제시한다.

3.2 RQ2. 리스크 매트릭스를 적용 시 위험 등급 분포는 어떻게 되는가?

질문에 답하기 위해 우리는 우선 식별된 군집에서 ML 문제와 연관이 적은 군집들(15, 24, 26, 32, 33)을 제외 하였다. 해당 군집들은 프로젝트 파일 설정, 도메인 이슈 그리고 사용자 소통과 문

서화에 해당하였다. 다음으로 위험 식별 Matrix를 통해 다음 그림 4와 같은 분포를 확인하였다.



[그림 4] AI 프레임워크 기반 프로젝트 이슈 등급별 분포

고위험군(H2, H1)은 ML 모델 학습, 분산 학습, 메모리 관리, 하드웨어 최적화와 관련된 이슈였으며, 발생 빈도는 Systemic 또는 Frequent, 해결 기간은 장기화하는 경향을 보였다. 중위험군(M2, M1)은 데이터 처리, 모델 변환, 저장 과정 등의 이슈가 많았으며, Occasional 발생이 일반적이었지만 해결에는 여전히 상당한 시간이 소요되었다. 저위험군(L2, L1)은 주로 개별 코드 오류나 부가적 기능 개선과 관련된 이슈로 구성되어 있었으며, 발생 빈도와 영향도 모두 낮았다.

이러한 등급별 군집 분포는 아래 표 2에서 표시하였다.

[표 2] 위험 등급 별 군집 분포

위험 등급	군집 번호
H1	2, 5, 6, 8, 12, 16, 22
H2	1, 3, 4
M1	25, 27, 30, 34
M2	10, 14, 19, 23
L1	17, 31, 35
L2	7, 9, 11, 13, 18, 20, 21, 28, 29

4. 결론

본 연구는 AI 프레임워크 기반 프로젝트 발생 이슈를 기반으로 반복적 결함 패턴을 식별하고, 정량적 위험 평가를 통해 발생 가능성과 영향도를 산출하였다. 실험 결과, 모델 학습 오류, 분산 학습 환경, 메모리 관리 등이 고위험으로 도출되었으며, 데이터 처리 및 모델 변환 이슈는 중위험에 분포하였다. 이는 AI 프레임워크 기반 시스템 개발 시 잠재적 리스크를 사전에 식별하고 대응하는데 기여할 수 있음을 시사한다.

참고문헌

[1] ISO/IEC. (2022). ISO/IEC 23053:2022 – Framework for the governance of AI systems. International Organization for Standardization.

[2] ISO/IEC. (2023). ISO/IEC 42001:2023 – AI management systems — Requirements. International Organization for Standardization.

[3] Wrong Predictions PyTorch vs TensorRT, https://github.com/NVIDIA/TensorRT/issues/1025

[4] CUDA out of memory during inference. https://github.com/ultralytics/ultralytics/issues/4057

[5] nvidia-container-cli: initialization error: nvml error: driver/library version mismatch: unknown, https://github.com/NVIDIA/gpu-operator/issues/898

[6] Tambon, F., Nikanjam, A., An, L. et al. Silent bugs in deep learning frameworks: an empirical study of Keras and TensorFlow. Empir Software Eng 29, 10 (2024).

[7] Guoming Long and Tao Chen. 2022. On Reporting Performance and Accuracy Bugs for Deep Learning Frameworks: An Exploratory Study from GitHub. In Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering (EASE '22). Association for Computing Machinery, New York, NY, USA, 90-99.

[8] Lai, T.D., Simmons, A., Barnett, S. et al. Comparative analysis of real issues in open-source machine learning projects. Empir Software Eng 29, 60 (2024).

[9] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[10] McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018).

[11] McInnes et al, (2017), hdbscan: Hierarchical density based clustering, Journal of Open Source Software, 2(11), 205, doi:10.21105/joss.00205

[12] Defense Department. "Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs". Government. Defense Department, January 1, 2015.

[13] Malone Jr, Roy W., and Kelly Moses. "Development of risk assessment matrix for NASA Engineering and Safety Center." In Risk Analysis: The Profession and the Future. 2004.