제26권 제1호
Vol. 26 No. 1

2024
제26회
한국 소프트웨어공학 학술대회 논문집

Proceedings of the 26th Korea Conference on Software Engineering (KCSE 2024)

- 일시: 2024년 1월 31일(수) ~ 2월 2일(금)
- 장소: 강원도 평창 한화리조트(휘닉스파크점)

주최: 한국정보과학회, 한국정보처리학회
주관: 한국정보과학회 소프트웨어공학 소사이어티
한국정보처리학회 소프트웨어공학연구회

후원: SOLUTIONLINK
㈜비트컴퓨터, ㈜다한테크, 브이플러스랩㈜,
슈어소프트테크㈜, 한국정보통신기술협회,
㈜이에스지, T3Q㈜

초대의 글

소프트웨어공학 학술대회(KCSE 2024) 참가자 여러분을 환영합니다.

KCSE (Korea Conference on Software Engineering)는 기업, 연구소 및 학계에서 활동하고 계신 소프트웨어공학 분야 전문가들의 모임으로, 한국정보과학회 소프트웨어공학 소사이어티와 한국정보처리학회 소프트웨어공학연구회가 소프트웨어공학 기술의 발전 및 적용 확산을 위하여 1999년부터 매년 개최하는 학술대회입니다.

이번 제26회 학술대회는 “거대 인공지능과 소프트웨어공학 기술” 주제로, 기조 연설, 튜토리얼, 신진 연구자 발표, 우수 논문 발표 등의 초청 세션과 소프트웨어공학 분야의 각계에서 제출한 논문 발표로 구성될 예정이며, 2024년 1월 31일부터 3일간에 걸쳐 진행합니다.

이번 KCSE 2024 학술대회가 소프트웨어공학을 연구하고, 적용하는 모든 연구자 그리고 전문가 여러분께 즐겁고 활기찬 학술 교류 및 기술 협력의 장이 될 수 있도록 다양한 프로그램으로 진행하고자 하오니 여러분의 많은 관심과 참여를 부탁드립니다.

제26회 KCSE 학술행사를 위해 수고해 주신 조직위원회와 학술위원회 위원들, 후원 기관 관계자 여러분, 그리고 기조 연설을 포함한 학술대회 모든 발표자분들께 깊이 감사드리며 건승을 기원합니다.

한국정보과학회 소프트웨어공학 소사이어티 회장 고인영

한국정보처리학회 소프트웨어공학연구회 운영위원장 이은서

학술대회 준비 위원회

공동대회장: 고인영 교수 (KAIST), 이은서 교수 (안동대)

조직위원장: 이관우 교수(한성대)

조직위원: 백종문 교수(KAIST), 김정아 교수(가톨릭 관동대), 류덕산 교수(전북대),
유철중 교수(전북대), 김순태 교수(전북대), 이지현 교수(전북대),
민상윤 대표(솔루션링크), 전진옥 사장(비트컴퓨터)

학술위원장: 남재창 교수(한동대)

학술위원: 강지훈 교수(KAIST), 강종구 교수(성신여대), 김동선 교수(경북대),
김문주 교수(KAIST), 김미정 교수(UNIST), 김윤호 교수(한양대),
김진대 교수(서울과학기술대), 김진현 교수(경상국립대), 김태호 박사(ETRI),
김택수 박사(삼성전자), 마유승 박사(ETRI), 박수진 교수(서강대),
박지훈 교수(충남대), 배경민 교수(POSTECH), 서영석 교수(영남대),
송지영 교수(한남대), 안성수 교수(경상국립대), 양근석 교수(한경대),
오학주 교수(고려대), 유신 교수(KAIST), 이선아 교수(경상국립대),
이우석 교수(한양대), 이은주 교수(경북대), 이정원 교수(아주대), 이주용 교수(UNIST),
이찬근 교수(중앙대), 이희진 교수(동양미래대), 정우성 교수(서울교대),
정필수 교수(경상국립대), 지은경 교수(KAIST), 차상길 교수(KAIST),
차수영 교수(성균관대), 최윤자 교수(경북대), 채흥석(부산대), 한종대 교수(방송대),
홍신 교수(한동대), 홍장의 교수(충북대)

문의사항 연락처

학술대회 홈페이지: http://www.sigsoft.or.kr/kcse2024/
조 직: 이관우 교수 (Email: kwlee@hansung.ac.kr, Tel. 02-760-5864)
학 술: 남재창 교수 (Email: jcnam@handong.edu, Tel. 054-260-1404)

KCSE 2024 프로그램

1월 31일 (수)			
시 간	**행 사 내 용**		
12:00–13:00	**KCSE 2024 등록**		
	튜토리얼 T1 좌장: 김태호 (ETRI) 장소: 세미나실 1	**튜토리얼 T2** 좌장: 이선아 (경상국립대) 장소: 세미나실 2	**튜토리얼 T3** 좌장: 이희진 (동양미래대) 장소: 세미나실 4 (지하1층)
13:00-16:00 (180분)	LLM을 활용한 소프트웨어 공학의 문제 해결 박재호 전무 (㈜레인보우브레인)	유무인 복합체계의 소개와 참조 아키텍처 개발 전략 김동환 연구위원 (LIG넥스원)	미래 모빌리티의 소프트웨어 및 데이터 기반 혁신 강종구 교수 (성신여대)
16:00-16:20	**휴식**		
	개회식 장소: 그랜드홀 2	사회: 이관우 조직위원장 (한성대)	
16:20-16:40 (20분)	개회사:	고인영 회장 (한국정보과학회 소프트웨어공학 소사이어티) 이은서 운영위원장 (한국정보처리학회 소프트웨어공학연구회)	
	기조강연 1 장소: 그랜드홀 2	사회: 남재창 학술위원장 (한동대)	
16:40-17:30 (50분)	대규모언어모델과 소프트웨어 공학 - 역사, 현황, 전망 유신 교수 (KAIST)		
17:30–17:40	**휴식**		

	신진 연구자 초청 세미나 N1 좌장: 이정원 (아주대) 장소: 그랜드홀 2	신진 연구자 초청 세미나 N2 좌장: 지은경 (KAIST) 장소: 세미나실 1	A1: AI와 SE 1 좌장: 양근석 (한경대) 장소: 세미나실 2
17:40-18:30 (50분)	데이터-기반 소프트웨어 테스팅 (Data-Driven Software Testing) 차수영 교수 (성균관대학교)	엣지 컴퓨팅 환경에서의 인공지능 시스템 연구 박지훈 교수 (충남대학교)	[우수 일반논문] 확률적 모델 체킹을 이용한 깃허브 기반의 프로젝트 참여자 기여 분석 조수희, 장지영, 박소희, 권기현 (경기대) [산업체 논문] 로우코드개발플랫폼과 RPA, ML을 융합한 Intelligent Hyper Automation 박병용 (㈜지산웨어)
18:30-19:30	**석식**		

2월 1일 (목)			
시 간	**행 사 내 용**		
	논문 발표		
	B1: AI와 SE 2 좌장: 김진현 (경상국립대) 장소: 그랜드홀 2	**B2: 결함 예측** 좌장: 양근석 (한경대) 장소: 세미나실 1	**B3: 요구 공학 1** 좌장: 지은경 (KAIST) 장소: 세미나실 2
09:30-10:35 (65분)	[단편 논문] 드럼 용기 라벨 육안 검사 AI pipeline 설계 및 구축 구민, 허의남 (경희대) [우수 학부생논문] Explainable AI 기법을 활용한 다중 오믹스 데이터 기반 COVID-19 중증도 예측 모델 장호중, 주정진, 김준구, 조겨리 (충북대) [산업체 논문] API Management를 위한LLM 기반 Test Case 자동 생성 기법 정하늘, 백두산, 전하영, 박성준 (KT)	[학부생 논문] ARM-Net 기반의 소프트웨어 결함 예측 이정화, 주은정, 류덕산 (전북대) [우수 단편논문] SAINT 기반의 소프트웨어 결함 예측 스리만 모하파트라, 김지영, 류덕산 (전북대) [일반 논문] 소프트웨어 결함 예측을 위한 XAI 기반 특징 공학 (XAI-based Feature Engineering for Software Defect Prediction) 아비섹 차우다리, 류덕산, 최선오 (전북대)	[우수 학부생논문] SAND: 거대 언어 모델을 이용한 소프트웨어 요구사항 명세서에서의 비원자 문장 탐지 및 수정 박상준, 이선구, 백종문 (KAIST) [단편 논문] 비정형 자연어 요구사항으로부터 UML 시퀀스 다이어그램 및 툰 이미지 생성 메커니즘 김현태 (홍익대), 공지훈 (툰스퀘어), 박영식, 박찬솔 (홍익대), 이상호 (라스테크), 김영철 (홍익대) [최우수 일반논문] STPA, FTA, FMEA 분석 기법을 연계한 소형 항공기 충돌 회피 소프트웨어 안전 요구사항 도출 이종원, 이의천, 김태환, 이선아 (경상국립대)
10:35-10:45	**휴식**		

	C1: AI와 SE 3 좌장: 양근석 (한경대) 장소: 그랜드홀 2	**C2: 프로그램 분석과 디버깅** 좌장: 유신 (KAIST) 장소: 세미나실 1	**C3: 요구 공학 2** 좌장: 배경민 (POSTECH) 장소: 세미나실 2
10:45-11:45 (60분)	[단편 논문] 모바일 환경에서 실시간 족부 불균형 판별을 위한 딥러닝 앙상블 시스템 설계 및 구현 정혜선, 김태구, 조용훈, 신기훈, 신채림 (부산대), 이수경 (동의대), 백윤주 (부산대) [우수 단편논문] 기저선 변동 잡음 제거와 딥러닝을 이용한 심전도 QT 간격 예측 이승준, 박진주 (전남대), 김유리 (전남대병원), 양형정 (전남대) [단편 논문] 차량 횡 방향 제어를 위한 신경망 기반 차량 슬립 각도 예측 김성현, 유승한, 강승우 (한국기술교육대)	[초청 논문] FunProbe: Probing Functions from Binary Code through Probabilistic Analysis (ESEC/FSE2023) Soomin Kim, Hyungseok Kim, Sang Kil Cha (KAIST) [초청 논문] WINE: Warning Miner for Improving Bug Finders (IST) Yoon-ho Choi and Jaechang Nam (한동대)	[단편 논문] 언어학적 의미 분석 기반 요구 공학을 통한 만화(Toon) 이미지 생성 김장환 (홍익대), 공지훈 (툰스퀘어), 장우성 (홍익대), 이근상 (전북테크노파크), 김기두 (한국정보통신기술협회), 김영철 (홍익대) [단편 논문] 자연어 기반 요구 사항을 활용한 UML 상태 다이어그램을 통해 만화 이미지 생성 진예진 (홍익대), 공지훈 (툰스퀘어), 김기두 (한국정보통신기술협회), 장우성, 김영철 (홍익대) [단편 논문] 문제 기술서에서 자연어 처리와 구조화를 이용한 요구사항 자동화 백영윤, 박용범 (단국대))
11:45-13:15	**중식**		

	D1: AI와 SE 4 좌장: 류덕산 (전북대) 장소: 그랜드홀 2	**D2: AI와 데이터 분석** 좌장: 강종구 (성신여대) 장소: 세미나실 1	**D3: SW 안전** 좌장: 배경민 (POSTECH) 장소: 세미나실 2
13:15-14:50 (95분)	[우수 일반논문] 사전 훈련된 기계 학습 모델의 효과적인 조합을 위한 공유 모델 허브 분석 Arogya Kharel, 고인영 (KAIST) [우수 단편논문] Variational Autoencoder 를 통한 대면 커뮤니케이션에서 다중 청취자 얼굴 생성 응웬민득, 응웬 당 칸, 파우델 쁘러베스, 양형정 (전남대) [단편 논문] 그래프 합성곱 신경망을 활용한 웹 데이터 추출 김창영, 조영우 (전남대), 김명석 (㈜나로수), 양형정 (전남대) [초청 논문] Deceiving Humans and Machines Alike: Search-based Test Input Generation for DNNs using Variational Autoencoders (TOSEM) Sungmin Kang (KAIST), Robert Feldt (Chalmers University of Technology), Shin Yoo (KAIST)	[산업체 논문] EAGLE: 고가용 AI모델 서빙을 위한 마이크로서비스 아키텍쳐 기반 API 게이트웨이 연구 박진우, 이원영, 이상정, 이동훈, 윤형화, 최호영 (LG 전자) [학부생 논문] 스마트 시티 실현을 위한 성북구 도로의 혼잡시간과 혼잡시간이 아닐 때의 교통사고 비교 분석 정가은, 이다영, 강종구 (성신여대) [학부생 논문] 스마트 시티 실현을 위한 전국 졸음운전 다발 구역과 졸음 쉼터 위치 비교 분석 이채원, 강종구 (성신여대) [초청 논문] An Empirical Study on the Performance of Individual Issue Label Prediction (MSR2023) Jueun Heo, Seonah Lee(경상국립대)	[일반 논문] STPA를 활용한 협업 가상물리시스템의 안전성 테스트 케이스 생성 허윤아, 유준범 (건국대) [학부생 논문] 가속도 센서를 이용한 CNN 기반의 스마트폰 터치 에러 감소 기법 개발 김은호, 박상근 (경희대) [일반 논문] 하이브리드 안전 분석을 통한 시각 기반 자율 주행 시스템의 적대적 견고성 향상 후세인 만주루, 이브라힘 아메드, 김경민, 상정위, 홍장의 (충북대) [우수 산업체논문] SDV 개발을 위한 차량 제어 소프트웨어 리팩토링 구태완, 김백준, 성병준 (현대자동차), 조광현, 이승준, 손정호 (에어플러그)
14:50-15:00	**휴식**		

	E1: 자동 디버깅 좌장: 유준범 (건국대) 장소: 그랜드홀 2	**E2: AI와 SE 5** 좌장: 유신 (KAIST) 장소: 세미나실 1	**E3: 블록 체인** 좌장: 박용범 (단국대) 장소: 세미나실 2
15:00-16:30 (90분)	[초청 논문] Automated Program Repair from Fuzzing Perspective (ISSTA2023) YoungJae Kim, Seungheon Han, Askar Yeltayuly Khamit, Jooyong Yi(UNIST) [초청 논문] A Bayesian Framework for Automated Debugging (ISSTA2023) Sungmin Kang, Wonkeun Choi, Shin Yoo (KAIST) [초청 논문] Poracle: Testing Patches Under Preservation Conditions to Combat the Overfitting Problem of Program Repair (TOSEM) Elkhan Ismayilzada (UNIST), Md Mazba Ur Rahman (UNIST), Dongsun Kim (경북대), Jooyong Yi (UNIST)	[초청 논문] Intent-Driven Mobile GUI Testing with Autonomous Large Language Model Agents (ICST2024) Juyeon Yoon (KAIST), Robert Feldt (Chalmers University of Technology), Shin Yoo (KAIST) [우수 단편논문] 감정인식 기반 Text-to-Speech : 감정표현 및 인식 훈련을 위한 플랫폼 개발 양현지, 조영우, 유연수, 양형정 (전남대) [단편 논문] 다양한 적절한 얼굴 반응 생성을 위한 향상된 트랜스포머 변이 오토 인코더 응웬당칸, 응웬민득, 파우델 쁘러베스, 양형정 (전남대) [학부생 논문] KNN 기반의 가짜 리뷰 계정 분류 모델 및 시스템 개발 한철현, 권세빈, 박상근 (경희대)	[초청 눈문] EtherDiffer: Differential Testing on RPC Services of Ethereum Nodes (FSE2023) Shinhae Kim (국가보안기술연구소), Sungjae Hwang (성균관대) [단편 논문] 분산 식별자를 사용한 머신 러닝 데이터 수집 과정의 개선 한윤경, 고한경, 이주희, 서중원, 조성우, 박수용 (서강대) [단편 논문] 허가형 블록체인의 트랜잭션 로그 정제 방법 강등원, 정수민, 박준석, 염근혁 (부산대)
16:30-16:40	**휴식**		
	기조강연 2 장소: 그랜드홀 2		사회: 고인영 대회장 (KAIST)
16:40-17:30 (50분)	ChatGPT 1년, 초거대 AI 시대가 불러온 변화와 우리의 대응전략 하정우 센터장 (Naver Cloud)		
17:30-18:00	**휴식**		
18:00-20:00	**석식 (Banquet) 및 시상식**		

2월 2일 (금)

시 간	행 사 내 용		
	논문 발표		
	F1: IoT와 CPS 좌장: 강종구 (성신여대) 장소: 그랜드홀 2	**F2: AI와 SE 6** 좌장: 남재창 (한동대) 장소: 세미나실 1	**F3: 구현 및 유지보수** 좌장: 김진대(서울과기대) 장소: 세미나실 2
9:30–11:20 (110분)	[단편 논문] AIS데이터를 이용한 대형선망어선의 조업 해역 분석 송은아, 정은주, 김광일 (제주대) [학부생 논문] 모빌리티 데이터를 활용한 MTGNN, AGCRN 모델의 하이퍼파라미터 최적화를 위한 예비 연구 양하늘, 김효은, 정민서, 강종구 (성신여대) [단편 논문]오픈스택 기반 컨테이너 인프라 관리 방법 및 시스템 정수민, 박준석, 염근혁 (부산대) [우수 단편논문] 360도 어라운드 뷰 시스템의 SIFT 특징점 기반 동적 캘리브레이션 알고리즘 강대웅 , 조상훈, 이학승, 한정우, 염지환, 국중진 (상명대) [초청 논문] Dynamic and Effect-driven Output Service Selection for IoT Environments Using Deep Reinforcement Learning (IEEE IoT Journal) KyeongDeok Baek, In-Young Ko (KAIST)	[단편 논문] ChatGPT 활용에 대한 고찰: 다양한 도메인에 적용되는 프롬프트 엔지니어링 전략 분석 Ivan Stanislavov Ivanov, 송지영 (한남대) [단편 논문] 클러스터링 알고리즘에 대한 성능 비교 평가 문지원 (애리조나주립대) [단편 논문] 한국어 음성으로부터의 조음기관 시각화를 통한 마비말장애 환자의 심각도 측정 주윤지, Rodrigo Piicinini Mexas, 심윤섭, 박운상 (서강대) [일반 논문] 오토인코더 기반 조인트 상태 진단을 통한 협동 로봇의 작업 수행 성능 및 건전성 평가 기법 최민서, 김진세, 이정원 (아주대)	[단편 논문] 선박무선통신 음성인식 기술 개발 연구 김광일, 유상록 (㈜)미래해양정보기술), 문일주 (제주대) [최우수 단편논문] 원자력 안전 소프트웨어 대상 신뢰도 평가 도구 Lingjun Liu, 최우영, 지은경 (KAIST), 류덕산(전북대) [학부생 논문] 오픈소스 시뮬레이터 기반 자율주행 구현 및 검증 윤수한, 기석철 (충북대) [우수 일반논문] 모바일 에지 컴퓨팅환경에서의 서비스 품질 연구 고찰 김지영, 남준성, 이재혁, 류덕산 (전북대) [일반 논문] 소프트 랜딩 스케일러: 웹 트래픽 폭주에 대한 효율적 대응을 위한 적응적 자원 할당 전략 강병욱, 고인영 (KAIST)
11:20-11:30	**휴식**		
11:30-11:55 (25분)	**폐회식** 장소: 그랜드홀 2		사회: 이관우 조직위원장 (한성대)

KCSE 2024 튜토리얼

튜토리얼 T1: LLM을 활용한 소프트웨어 공학의 문제 해결

- 일시: 1월 31일(수) 13:00~16:00
- 장소: 세미나실 1
- 제목: LLM을 활용한 소프트웨어 공학의 문제 해결
- 연사: 박재호 전무(㈜레인보우브레인)
- 튜토리얼 초록:

  코드 작성과 수정 과정에서 여러 IDE에 결합 가능한 플러그인 형태의 깃헙 코파일럿이나 웹 브라우저 상에서 챗GPT를 활용해 도움을 받는 경우를 주변에서 흔히 찾아볼 수 있게 되었다. 실제로 LLM의 가장 성공적인 상용화 시장은 프로그래머 대상이라고 할 만큼 탄탄하게 자리를 잡았으며 최근에는 자바 통합 개발 환경인 IntelliJ IDEA로 유명한 젯브레인이 공개한 AI 어시스턴트(IDE에 통합된 형태의 채팅)는 코드 설명/리팩토링 제안/잠재적 문제 발견/메소드 작성/이름 제안/커밋 메시지 생성/문서 작성 기능을 포함시켜서 개발자들의 좋은 반응을 이끌어내고 있다. 하지만 이는 시작일 뿐이며, LLM은 소프트웨어 공학 분야에서 난해한 문제라고 여겨지던 설계, 검증, 분석, 유지보수, 프로토타입 분야까지 점차 외연을 넓혀가고 있다. 현재 기술 발전 속도를 감안할 때 소프트웨어 개발 전구간에 걸쳐 개발자를 지원하게 될 날이 멀지 않았다. 본 튜토리얼에서는 소프트웨어 공학의 문제를 해결하기 위해 LLM을 활용하는 다양한 기법을 실제 사례로 보면서 최신 LLM 기술 현황을 파악해보기로 한다.

  소프트웨어 공학 관점에서 다시 바라보는 LLM:
  - 추론을 위한 코드 학습 - 명세와 코드 이해는 LLM의 기본
  - 코드 인터프리터의 도입으로 인한 예상치 못한 (좋은) 부작용
  - 멀티모달의 파급 효과 – 21세기에 새롭게 등장한 CASE 도구로서 LLM

  LLM을 활용한 소프트웨어 공학의 문제 해결 사용 사례
  - 명세를 코드로 변환
  - 언어간 코드 변환
  - 보안/성능 관련 문제 추적
  - 코드 리팩토링
  - 단위 테스트

  소프트웨어 공학의 전구간을 넘나드는 LLM의 위력
  - 환각을 억제하고 전문적인 지식 기반(Knowledge Base) 구축 목적으로 등장한 RAG(Retrieval-Augmented Generation) 응용
  - 스테이트 차트로 명세한 실시간 시스템의 liveness/safety 분석과 명세를 따르는 파이썬 코드 구현
  - 프로토타이핑(스테이트 차트와 와이어프레임으로 명세하고 실행가능한 코드를 생성하기)

- 약력:
  - 2022~현재 ㈜레인보우브레인 CTO
  - 2015~2017 ㈜엑셈 CTO
  - 1995~1997 포항공과대학교 컴퓨터공학과(석사, 소프트웨어공학 전공)
  - 1991~1995 포항공과대학교 컴퓨터공학과(학사)
- 관심분야:

  인공지능(LLM), 클라우드, 고성능/고가용성 소프트웨어 아키텍처

튜토리얼 T2: 유무인 복합체계의 소개와 참조 아키텍처 개발 전략

- 일시: 1월 31일(수) 13:00~16:00
- 장소: 세미나실 2
- 제목: 유무인 복합체계의 소개와 참조 아키텍처 개발 전략
- 연사: 김동환 연구위원 (한국정보과학회 종신회원)
- 튜토리얼 초록:

K-방산이라는 무기체계의 수출은 사회적으로 많은 관심을 갖게 되었다. 산업적으로는 큰 성과이기는 하나, 사회적으로는 무기가 전쟁의 도구라는 부정적 시각이 있는 것도 사실이다. 구글의 연구원들이 Maven이라는 인공지능기반 체계 개발에 반대를 했던 것도 이러한 이유 중에 하나일 것이다. 그러나 인공지능은 핵과 맞먹는 수준의 기술이라는 평가로 이를 둘러싼 무기체계의 개발은 마치 냉전시대에 미국과 소련의 과학기술 전쟁을 연상하게 할 정도로 큰 이슈가 되고 있다. 국방은 우리의 안전을 지키는 수단이지 전쟁을 위한 수단이 아니다. 이러한 관점에서 인공지능을 적용하되 엄격한 개발과 공정한 개발이 되도록 시스템 및 소프트웨어공학자들이 적극적으로 기술개발에 참여해야 한다.

미래의 무기체계는 인공지능과 같은 4차산업기술의 발전으로 디지털변혁을 통해 유무인복합체계라는 보다 복잡한 체계로 발전해 가고 있다. 또한 상대적으로 개발난이도의 증가에도 불구하고 첨단 기술을 신속히 적용하기 위해 전통적인 개발방법이 아닌 애자일방법으로 개발해야 하는 도전을 받고 있다. 이러한 유무인 복합체계를 다양한 품질을 고려하여 Model-based system engineering(MBSE) 기반으로 개발하는 것은 옵션이 아니라 필수적인 요소가 되고 있다.

본 튜토리얼에서는 미래의 무기체계로 구현될 전장의 모습과 인공지능으로 구현될 유무인복합체계의 사례 및 미국의 방산유니콘 기업의 동향을 소개한다. 이를 기반으로 인공지능 기반 유무인복합체계의 Product Line Engineering 관점에서 공통적으로 적용될 수 있는 참조 아키텍처 개발 전략을 소개한다. 이를 통해 소프트웨어시스템공학자들이 인공지능 기반 체계 개발에서의 고려사항 및 미래의 무기체계라는 도메인 지식을 습득할 수 있다. 또한 실제 사례 중심의 MBSE 개발을 통해 Software Engineering의 한계를 이해하고 Software System Engineering으로의 사고를 넓힐 수 있는 기회를 제공하고자 한다.

튜토리얼은 다음과 같은 내용으로 구성된다.

미래 무기체계 및 전장의 특성
미래의 유무인 복합체계의 사례 소개
미국 방산유니콘의 소개
유무인 복합체계의 참조 System Analysis Model
유무인 복합체계의 참조 Contextual Knowledge Model
유무인 복합체계의 주요 아키텍처 설계 결정사항
유무인 복합체계의 참조 시스템 아키텍쳐

- 약력:

- 2020 ~ 현재: LIG넥스원 C4ISTAR부분.연구개발2본부 연구위원
- 2003~2009: ㈜ 히어솔루션코리아 부사장
- 2000~2003: 팅크웨어㈜ 상무
- 1996~2000: 대우통신㈜ 부장
- 1995~1996: ㈜한조엔지니어링 부장
- 1986~1995: 국방과학연구소 선임연구원
- 2008: 정보시스템수석감리원
- 1995: 전자계산기조직응용기술사
- 1998: KAIST 정보및통신공학과(박사과정수료)
- 1986: KAIST 전산학과(석사)
- 1984: 인하대학교 전산학과(학사)

- 연구분야:

Dependable SW시스템 개발방법론 및 품질보증, 무인화 전투체계 및 자율주행 시스템

튜토리얼 T3: 미래 모빌리티의 소프트웨어 및 데이터 기반 혁신

- 일시: 1월 31일(수) 13:00~16:00
- 장소: 세미나실 4 (지하 1층)
- 제목: 미래 모빌리티의 소프트웨어 및 데이터 기반 혁신
- 연사: 강종구 (성신여자대학교 AI융합학부 조교수)
- 튜토리얼 초록:

  인터넷 시대, 스마트폰 시대를 지나 스마트 모빌리티 시대가 되고 있습니다. 과거 신생 인류의 출현은 모든 것들을 바꾸어 놓고는 하였습니다. 도구를 만들어 사용하는 인간의 출현은 가장 나약한 신체 조건에도 불구하고 모든 동물 위에 군림하게 했으며, 현재 스마트폰을 들고 있는 인간은 언제든 네트워크로 연결되어 있으며, 검색 한 번으로 원하는 지식을 바로 습득할 수 있고 컨텐츠를 실시간으로 확장해 나갈 수 있습니다. 이제 스마트 모빌리티 시대의 인간은 스스로 만들어 낸 AI비서를 데리고 언제든 최적의 방식으로 노력 없이 이동 가능하며, 심지어 물리적인 이동이 필요 없이 가상에서 모든 경험을 가능하게 할 것으로 예측해볼 수 있습니다. 이러한 변화하는 신인류의 출현에 모빌리티의 소프트웨어와 데이터 기반 혁신이 핵심적으로 자리하고 있습니다.

  미래의 모빌리티는 소프트웨어 중심 플랫폼과 발생하는 데이터의 처리와 응용을 수단으로 혁신이 이루어지고 있습니다. 본 튜토리얼에서는 SDV, 커넥티드/자율주행전기차와 같은 미래자동차와 드론과 같은 무인항공기 등 미래 모빌리티의 소프트웨어 플랫폼 혁신과 모빌리티 데이터 공학의 현재와 미래를 다룹니다. 1) 우선, 발표자가 15년 이상 미래모빌리티 산업현장에서 수행했던 프로젝트 및 보유 공인자격에 대해 소개합니다. 2) 미래 모빌리티 비전과 사례, 그리고 SDV로 불리는 소프트웨어 정의 모빌리티의 출현과 데이터 기반 혁신 동향을 중심으로 최근 동향을 소개합니다. 3) 이를 기반으로 모빌리티 중심 AIoT, 스마트시티, 디지털트윈, 소프트웨어 실제 등 산업과 밀접한 주제를 제시하고 인간 중심 확장의 방향성을 모색하고자 합니다. 이를 통해 큰 변화 속에서 미래 소프트웨어 공학인의 사고를 넓히고 소프트웨어 및 데이터 기반 혁신을 능동적으로 준비할 수 있는 기회를 제공하고자 합니다.

  튜토리얼은 다음과 같은 내용으로 구성됩니다.

  1. 발표자의 경험 중심 모빌리티 프로젝트 소개
  2. 발표자의 경험 중심 모빌리티 공인자격 소개
  3. 미래 모빌리티 비전 및 사례 소개
  4. 소프트웨어 정의 모빌리티 동향
  5. 데이터 기반 혁신 동향
  6. 미래 방향성 예측 - 모빌리티/도시/인간/디지털트윈 등
  7. 인간 중심 확장 방향성

- 약력:
  - 현재: 성신여자대학교 AI융합학부 조교수
  - 2020~2023년: 현대자동차 전략기술본부/GSO 책임연구원
  - 2008~2020년: 현대중공업그룹 책임연구원
  - 2022년: KAIST 전산학부/미래자동차학제(박사)
  - 2008년: KAIST 전산학부(석사)
  - 2006년: KAIST 전산학부(학사)
- 연구분야:

  Future Mobility, Smart City, Digital Twin, Software in Practice

KCSE 2024 기조강연

기조강연 I

- 일시: 1월 31일(수) 16:40-17:30
- 장소: 그랜드홀 2
- 제목: 대규모 언어 모델과 소프트웨어 공학 - 역사, 현황, 전망
- 연사: 유신 교수 (KAIST)
- 초록:

  대규모 언어 모델은 현재 소프트웨어 공학 연구에서 단연 가장 각광받는 신기술 중 하나이다. 대규모 언어 모델은 자연어와 코드의 경계에 구애받지 않고 프로그램의 의미를 이해하는 듯한 추론 능력을 보이며, 별도의 훈련을 거치지 않은 업무도 프롬프트 엔지니어링을 통해 수행하는 창발적 능력을 지니고 있기 때문이다. 본 발표는 높은 수준의 코드 합성을 가능하게 하는 소스 코드의 통계적 속성에 대한 연구를 역사적으로 살펴보고, 현재 폭발적인 속도로 진행되고 있는 대규모 언어 모델 기반 소프트웨어 공학 연구를 소개하며, 마지막으로 통계적 언어 모델이 가지는 한계를 극복하기 위해 기존에 진행된 소프트웨어 공학 연구가 어떻게 쓰일 수 있는지에 대한 전망을 제시한다.
- 약력:
  - 현. ACM Transactions on Software Engineering and Methodology (TOSEM) 편집위원
  - 현. Springer Journal of Empirical Software Engineering (EMSE) 편집위원
  - 2015 ~ 현재: KAIST 전산학부 부교수
  - 2012 ~ 2015: University College London 전산학과 조교수
  - 2018 ~ 2023: IEEE ICST학회 Steering Committee Chair
  - 2006 ~ 2009: 영국 King's College London (박사)

기조강연 II

- 일시: 2월 1일(목) 16:40-17:30
- 장소: 그랜드홀 2
- 제목: ChatGPT 1년, 초거대 AI 시대가 불러온 변화와 우리의 대응전략
- 연사: 하정우 센터장 (Naver Cloud)
- 초록:

ChatGPT가 출시된 지 1년이 지났고 지난 1년간 전세계는 초거대 언어모델로 대변되는 생성AI에 의한 패러다임 전환이 가속화되고 있다. 2023년이 생성AI기술이 어떤 변혁을 불러올 수 있는 가능성을 보여준 해였다면 2024년은 산업과 사회가 본격적인 생성AI 전환의 시대가 될 것으로 예상된다. 본 강연에서는 현재 생성AI 기술이 산업과 일상생활에서 불러온 변화를 소개하고 특히 네이버의 하이퍼클로바X를 중심으로 B2C, B2B, 및 B2G에서의 활용사례등을 포함하여 한국의 생성AI 경쟁력에 대해 공유한다. 마지막으로 생성AI 전환시대 AI안전성을 포함한 우리가 준비해야 할 대응 전략에 대해 논의한다.

- 약력:
  - 2023. 1 – 현재 NAVER Cloud AI Lab 연구소장
  - 2020.10 – 2022.12 NAVER AI Lab 연구소장
  - 2020.3 – 2020.10 NAVER CLOVA AI Research 책임리더 (이사)
  - 2017.1 – 2020.2 NAVER CLOVA AI Research 리더
  - 2015. 3 – 2016. 12 NAVER Labs 책임연구원
  - 2015. 2 서울대학교 컴퓨터공학부 박사 (최우수박사학위 논문 수상)
  - 2004. 2 서울대학교 컴퓨터공학부 학사

KCSE 2024 신진 연구자 초청 발표

신진 연구자 초청 발표 N1

- 일시: 1월 31일(수) 17:40-18:30
- 장소: 그랜드홀 2
- 제목: 데이터-기반 소프트웨어 테스팅 (Data-Driven Software Testing)
- 연사: 차수영 교수 (성균관대학교)
- 초록:

소프트웨어 테스팅(Software Testing)은 소프트웨어의 품질을 향상하기 위해 오류를 검출하거나 코드 커버리지를 높이는 테스트-케이스를 자동으로 생성하는 것을 목표로 한다. 그리고 이 목표를 달성하기 위한 다양한 테스팅 방법론이 존재한다. 예를 들어, 기호 실행(Symbolic Execution)은 하나의 대표적인 화이트-박스 테스팅 기법으로서 산업계와 학계 모두에서 널리 이용되고 있다. 구체적으로, Microsoft사의 기호실행 기반 테스팅 도구 SAGE는 Windows 7의 개발동안 발견된 전체 취약점의 약 30%를 찾는데 성공하여 Microsoft사는 수백만 달러를 아낄 수 있었다. 기호 실행의 핵심 아이디어는 테스팅을 수행할 프로그램의 실제 입력을 기호 입력(Symbolic Input)으로 대체한 후에 프로그램을 그 기호 입력으로 실행하는 것이다. 이를 통해, 본 기술은 프로그램의 다양한 실행 경로들에 각각 도달할 수 있는 테스트-케이스를 체계적으로 생성할 수 있는 큰 이점을 지닌다. 또 하나의 대표적인 소프트웨어 테스팅 기법으로는 그레이-박스 테스팅 또는 퍼징(Fuzzing)이라고 불리는 기술이 있다. 퍼징은 기본적으로 무작위 값으로 테스트-케이스를 생성하는 기법으로서 기호 실행과 비교하여 더 빠르게 다양한 입력들을 만들어내는 장점이 있다. 퍼징 기법 역시 그 실용성을 학계와 산업계에서 다양한 방식으로 입증하고 있다. 본 세미나에서는 두 가지 대표적인 소프트웨어 테스팅 기법들 "기호실행"과 "퍼징"의 간단한 동작 방식 및 핵심 도전과제를 먼저 알아본다. 그리고 해당 도전과제들에 대한 현존 해결책들과 그 해결책들의 한계점을 논의해본다. 마지막으로, 그 한계를 극복하기 위해 근본적으로 다른 방향성을 추구하는 기술 "데이터 기반 소프트웨어 테스팅"을 소개하고 그 기술의 구체적인 사례 기술들을 소개하고자 한다.

- 약력:
  - 2021.09 ~ 현재: 성균관대학교, 소프트웨어학과 조교수
  - 2021.03 ~ 2021.08: 고려대학교, 소프트웨어보안연구소 연구교수
  - 2016.03 ~ 2021.02: 고려대학교, 컴퓨터학과, 박사
  - 2014.03 ~ 2016.02: 고려대학교, 컴퓨터학과, 석사
  - 2008.03 ~ 2014.02: 세종대학교, 컴퓨터공학과, 학사
- 연구분야:

소프트웨어 테스팅, 기호실행, 퍼징, 정적 분석

신진 연구자 초청 발표 N2

- 일시: 1월 31일(수) 17:40-18:30
- 장소: 세미나실 1
- 제목: 엣지 컴퓨팅 환경에서의 인공지능 시스템 연구
- 연사: 박지훈 교수 (충남대학교)
- 초록:

  최신 급격히 발전한 인공지능 기술의 다양한 응용 분야에 대해 기존 직장의 경험과 앞으로의 연구 계획을 포함하여 설명한다. 약 6년간의 국방과학연구소 재직 경험을 바탕으로 적대 드론 방어 시스템, 군용 무인 차량 자율 주행, 은닉 물체 자동 탐지/추적, 드론에서의 엣지컴퓨팅 환경 물체 인식/추적 기술 등 다양한 인공지능 활용 분야의 경험에 대해 문제 정의 및 접근 방법에 대해 설명한다. 데이터 희소 분야의 가상데이터의 활용 방안 및 인공지능 시스템이 탑재되는 엣지 컴퓨팅 환경에서의 적용/응용에 대해 소개한다.

- 약력:
  - 2023 ~ 현재: 충남대학교 컴퓨터 융합학부
  - 2022 ~ 2023: ㈜에이투마인드
  - 2016 ~ 2022: 국방과학연구소
  - 박사: KAIST 전산학부 (지도교수: 배두환교수님)
  - 석사: KAIST 전산학과 (지도교수: 배두환교수님)
  - 학사: KAIST 전산학과
- 연구분야:

  소프트웨어 저장소 마이닝, 적대 드론 방어 시스템, 자율 주행 인식, 엣지 컴퓨팅 환경 물체 인식/추적 등

우수 국제학회/학술지 초청 논문발표

- **FunProbe: Probing Functions from Binary Code through Probabilistic Analysis** - 2월 1일 10:45 세미나실 1
  - The 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2023
  - Soomin Kim, Hyungseok Kim, Sang Kil Cha (KAIST)

- **WINE: Warning Miner for Improving Bug Finders** - 2월 1일 11:15 세미나실 1
  - Information and Software Technology (IST), 2023
  - Yoon-ho Choi and Jaechang Nam (한동대)

- **Deceiving Humans and Machines Alike: Search-based Test Input Generation for DNNs using Variational Autoencoders** - 2월 1일 14:20 그랜드홀 2
  - ACM Transactions on Software Engineering and Methodology (TOSEM), 2023
  - Sungmin Kang (KAIST), Robert Feldt (Chalmers University of Technology), Shin Yoo (KAIST)

- **An Empirical Study on the Performance of Individual Issue Label Prediction** - 2월 1일 14:20 세미나실 1
  - The 20th IEEE/ACM International Conference on Mining Software Repositories (MSR), 2023
  - Jueun Heo, Seonah Lee(경상국립대)

- **Automated Program Repair from Fuzzing Perspective** - 2월 1일 15:00 그랜드홀 2
  - The 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), 2023
  - YoungJae Kim, Seungheon Han, Askar Yeltayuly Khamit, Jooyong Yi(UNIST)

- **Intent-Driven Mobile GUI Testing with Autonomous Large Language Model Agents** - 2월 1일 15:00 세미나실 1
  - The 17th IEEE International Conference on Software Testing, Verification and Validation (ICST), 2024
  - Juyeon Yoon (KAIST), Robert Feldt(Chalmers University of Technology), Shin Yoo (KAIST)

- **EtherDiffer: Differential Testing on RPC Services of Ethereum Nodes** - 2월 1일 15:00 세미나실 2
  - The 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2023
  - Shinhae Kim (국가보안기술연구소), Sungjae Hwang (성균관대)

- **A Bayesian Framework for Automated Debugging** - 2월 1일 15:30 그랜드홀 2
  - The 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), 2023
  - Sungmin Kang, Wonkeun Choi, Shin Yoo (KAIST)

- **Poracle: Testing Patches Under Preservation Conditions to Combat the Overfitting Problem of Program Repair** - 2월 1일 16:00 그랜드홀 2
  - ACM Transactions on Software Engineering and Methodology (TOSEM), 2023
  - Elkhan Ismayilzada (UNIST), Md Mazba Ur Rahman (UNIST), Dongsun Kim (경북대), Jooyong Yi (UNIST)

- **Dynamic and Effect-driven Output Service Selection for IoT Environments Using Deep Reinforcement Learning** - 2월 2일 10:50 그랜드홀 2
  - IEEE Internet of Things Journal
  - KyeongDeok Baek, In-Yong Ko (KAIST)

KCSE 2024 논문 목차

일반 논문

소프트 랜딩 스케일러: 웹 트래픽 폭주에 대한 효율적 대응을 위한 적응적 자원 할당 전략..... 강병욱,고인영 1

확률적 모델 체킹을 이용한 깃허브 기반의 프로젝트 참여자 기여 분석.....조수희, 장지영, 박소희, 권기현 10

STPA, FTA 및 FMEA 분석 기법을 연계한 소형 항공기 충돌 회피 소프트웨어 안전 요구사항 도출.....

.....이종원, 이의천, 김태환, 이선아 12

오토인코더 기반 조인트 상태 진단을 통한 협동 로봇의 작업 수행 성능 및 건전성 평가 기법.....

..... 최민서, 김진세, 이정원 14

소프트웨어 결함 예측을 위한 XAI 기반 특징 공학 (XAI-based Feature Engineering for Software Defect Prediction).....아비섹 차우다리, 류덕산, 최선오 24

모바일 에지 컴퓨팅환경에서의 서비스 품질 연구 고찰.....김지영, 남준성, 이재혁, 류덕산 34

사전 훈련된 기계 학습 모델의 효과적인 조합을 위한 공유 모델 허브 분석.....Arogya Kharel, 고인영 36

STPA를 활용한 협업 가상물리시스템의 안전성 테스트 케이스 생성.....허윤아, 유준범 38

하이브리드 안전 분석을 통한 시각 기반 자율 주행 시스템의 적대적 견고성 향상.....

..... 후세인 만주루, 이브라힘 아메드, 김경민, 상정위, 홍장의 47

단편 논문

드럼 용기 라벨 육안 검사 AI pipeline 설계 및 구축.....구민, 허의남 56

한국어 음성으로부터의 조음기관 시각화를 통한 마비말장애 환자의 심각도 측정.....

..... 주윤지, Rodrigo Picinini Mexas, 심윤섭, 박운상 60

기저선 변동 잡음 제거와 딥러닝을 이용한 심전도 QT 간격 예측..... 이승준, 박진주, 김유리, 양형정 63

차량 횡 방향 제어를 위한 신경망 기반 차량 슬립 각도 예측.....김성현, 유승한, 강승우 66

클러스터링 알고리즘에 대한 성능 비교 평가..... 문지원 69

문제 기술서에서 자연어 처리와 구조화를 이용한 요구사항 자동화.....백영윤, 박용범 73

오픈스택 기반 컨테이너 인프라 관리 방법 및 시스템..... 정수민, 박준석, 염근혁 77

허가형 블록체인의 트랜잭션 로그 정제 방법.....강등원, 정수민, 박준석, 염근혁 81

360도 어라운드 뷰 시스템의 SIFT 특징점 기반 동적 캘리브레이션 알고리즘..... 강대웅, 조상훈, 이학승, 한정우, 염지환, 국중진 84
Variational Autoencoder 를 통한 대면 커뮤니케이션에서 다중 청취자 얼굴 생성..... 응웬민득, 응웬 당 칸, 파우델 쁘러베스, 양형정 86
다양한 적절한 얼굴 반응 생성을 위한 향상된 트랜스포머 변이 오토 인코더 ..... 응웬당칸, 응웬민득, 파우델 쁘러베스, 양형정 88
그래프 합성곱 신경망을 활용한 웹 데이터 추출 .....김창영, 조영우, 김명석, 양형정 90
감정인식 기반 Text-to-Speech : 감정표현 및 인식 훈련을 위한 플랫폼 개발..... 양현지, 조영우, 유연수, 양형정 94
자연어 기반 요구 사항을 활용한 UML 상태 다이어그램을 통해 만화 이미지 생성..... 진예진, 공지훈, 김기두, 장우성, 김영철 96
비정형 자연어 요구사항으로부터 UML 시퀀스 다이어그램 및 툰 이미지 생성 메커니즘 ..... 김현태, 공지훈, 박영식, 박찬솔, 이상호, 김영철 98
언어학적 의미 분석 기반 요구 공학을 통한 만화(Toon) 이미지 생성 ..... 김장환, 공지훈, 장우성, 이근상, 김기두, 김영철 100
선박무선통신 음성인식 기술 개발 연구.....김광일, 유상록, 문일주 102
SAINT 기반의 소프트웨어 결함 예측..... 스리만 모하파트라, 김지영, 류덕산 104
원자력 안전 소프트웨어 대상 신뢰도 평가 도구..... Lingjun Liu, 최우영, 지은경, 류덕산 106
모바일 환경에서 실시간 족부 불균형 판별을 위한 딥러닝 앙상블 시스템 설계 및 구현.....정혜선, 김태구, 조용훈, 신기훈, 신채림, 이수경, 백윤주 108
AIS데이터를 이용한 대형선망어선의 조업 해역 분석..... 송은아, 정은주, 김광일 112
ChatGPT 활용에 대한 고찰: 다양한 도메인에 적용되는 프롬프트 엔지니어링 전략 분석..... Ivan Stanislavov Ivanov, 송지영 115
분산 식별자를 사용한 머신 러닝 데이터 수집 과정의 개선.....한윤경, 고한경, 이주희, 서중원, 조성우, 박수용 122

학부생 논문

SAND: 거대 언어 모델을 이용한 소프트웨어 요구사항 명세서에서의 비원자 문장 탐지 및 수정..... 박상준, 이선구, 백종문 126

Explainable AI 기법을 활용한 다중 오믹스 데이터 기반 COVID-19 중증도 예측 모델..... 장호중, 주정진, 김준구, 조겨리134

오픈소스 시뮬레이터 기반 자율주행 구현 및 검증.....윤수한, 기석철 140

KNN 기반의 가짜 리뷰 계정 분류 모델 및 시스템 개발..... 한철현, 권세빈, 박상근 146

가속도 센서를 이용한 CNN 기반의 스마트폰 터치 에러 감소 기법 개발..... 김은호, 박상근 152

스마트 시티 실현을 위한 성북구 도로의 혼잡시간과 혼잡시간이 아닐 때의 교통사고 비교 분석..... 정가은, 이다영, 강종구 158

스마트 시티 실현을 위한 전국 졸음운전 다발 구역과 졸음 쉼터 위치 비교 분석..... 이채원, 강종구 165

모빌리티 데이터를 활용한 MTGNN, AGCRN 모델의 하이퍼파라미터 최적화를 위한 예비 연구..... 양하늘, 김효은, 정민서, 강종구 171

ARM-Net 기반의 소프트웨어 결함 예측..... 이정화, 주은정, 류덕산 179

산업체 논문

EAGLE: 고가용 AI모델 서빙을 위한 마이크로 서비스 아키텍쳐 기반 API 게이트웨이 연구..... 박진우, 이원영, 이상정, 이동훈, 윤형화, 최호영 183

API Management를 위한LLM 기반 Test Case 자동 생성 기법..... 정하늘, 백두산, 전하영, 박성준 190

SDV 개발을 위한 차량 제어 소프트웨어 리팩토링..... 구태완, 김백준, 성병준, 조광현, 이승준, 손정호 198

로우코드개발플랫폼과 RPA, ML을 융합한 Intelligent Hyper Automation..... 박병용 206

소프트 랜딩 스케일러: 웹 트래픽 폭주에 대한 효율적 대응을 위한 적응적 자원 할당 전략

강병욱, 고인영
한국과학기술원
bw.kang@kaist.ac.kr, iko@webeng.kaist.ac.kr

Soft Landing Scaler: Adaptive Resource Allocation Strategies for Efficient Handling of Web Traffic Bursts

Byungwook Kang, Inyoung Ko
Korea Advanced Institute of Science and Technology

Abstract

This research paper addresses managing sudden web traffic increases in cloud computing, emphasizing effective resource allocation. Traffic surges can overwhelm servers, leading to service instability and dissatisfaction. The proposed solution, the Soft Landing Scaler (SLS), dynamically adjusts resources based on traffic fluctuations using a Kubernetes-based architecture. SLS is designed for optimal resource efficiency and adaptability, maintaining satisfactory response times. The study analyzes traffic patterns such as Sharp Increase then Exponential Decrease (SIED), and Sharp Increase then Linear Decrease (SILD), demonstrating SLS's performance. Results show improved resource efficiency and user response times, highlighting SLS's effectiveness in handling diverse traffic surges. The study contributes to the field by presenting an adaptive scaling system focused on response times, using real-world traffic data, and emphasizing buffer resources and scaling size limits in downscaling strategies.

1. Introduction

In the modern age of digital technology, web services have become an indispensable aspect of our daily lives by providing platforms for education, entertainment, and commerce. However, these services often face the challenge of web traffic surges: sudden increases in user access that can overwhelm servers and disrupt their availability. Such surges can be triggered by various events, such as the scheduled opening of ticket sales leading to the paralysis of ticketing websites, the commencement of EBS online classes during the COVID-19 pandemic resulting in website crashes, or a spike in search queries on portal sites coinciding with quiz shows. These instances illustrate how significant events can lead to service disruption and user dissatisfaction.

Managing these surges is crucial for maintaining service reliability, user experience, and cost-effective resource allocation. The complexity of this challenge is exemplified in the work of Shan et al. (2023), who discussed the difficulties in resource allocation within Kubernetes environments, particularly under unexpected resource-request spikes [1]. These real-world examples underscore the importance of effectively handling traffic surges to ensure continuous service availability and performance.

1.1. Research Problem

The main concern of this study is the inefficient handling of web traffic spikes within cloud services. The traditional approach to this problem has been to maintain a buffer of excess resources, leading to a scenario where resources are often over-provisioned, incurring unnecessary costs, and resulting in wastage through underutilization. Conversely, underprovisioning is fraught with its own perils, primarily the risk of service disruptions and consequent user dissatisfaction, which can tarnish the service provider's reputation. This study seeks to address the absence of a nuanced mechanism that can dynamically scale resources in alignment with the actual demand, thereby marrying service quality with cost efficiency. The importance of such dynamic resource scaling, particularly within versatile yet demanding Kubernetes platforms, was underscored in the research conducted by Zhou et al. (2023), which drew attention to the imperative for real-time resource adjustments to cater to the ebb and flow of user demands [2].

1.2. Research Objectives and Methodology

The principal objective of our study is to devise and refine an resource-scaling system called the Soft Landing Scaler (SLS), which is specifically tailored for Kubernetes ecosystems. The envisioned functionality of the SLS is to proactively upscale resources in the face of impending traffic surges and, equally importantly, to downscale them in a judicious manner once demand subsides. This system leverages a suite of real-time metrics, including transaction volumes, response times, and resource utilization, to make informed and responsive scaling decisions. Our methodological framework encompasses a meticulous analysis of traffic patterns derived from real-life scenarios, replication of these patterns within a controlled testbed, and comparative assessment of the efficacy of SLS relative to conventional scaling paradigms. This investigative trajectory is in harmony with the adaptive resource-allocation strategies delineated by Bekcheva et al. (2018), who advocated for a dynamic and responsive resource-management approach in the face of fluctuating workload demands [3].

1.3. Scope and Structure of the Paper

This paper focuses on the development and evaluation of SLS in a cloud computing context, specifically within Kubernetes-managed environments. The study is structured as follows: Following the introduction, we review related work in the field to provide context and background. We then detail the design and implementation of SLS, followed by an extensive evaluation of its performance using real-world traffic patterns. Finally, we discuss the implications of our findings, the limitations of the current study, and potential avenues for future research.

2. Related Work

Our research on the effective management of sudden increases in web traffic within cloud computing environments using the Soft Landing Scaler (SLS) is grounded in a wealth of prior studies.

The work in [4] serves as a foundation for our understanding of resource allocation within cloud services. This study explores a variety of adaptive management strategies while delving into the complexities of service workflows, thereby providing valuable insights into the nuanced demands of cloud computing environments.

Similarly, the work in [5] is of critical importance to our own endeavors. Focusing on the challenges posed by sudden traffic spikes, this research offers essential strategies for dynamic resource management, which is a key aspect of our SLS system. The emphasis on agility and adaptability in resource allocation resonates with our approach.

The exploration of Quality of Service (QoS) in cloud environments, as presented in [6], closely aligns with our goal of maintaining service stability during traffic surges. This study's focus on balancing resource allocation with service quality provides a framework for the SLS system's emphasis on response time and user satisfaction.

The economic aspect of cloud resource management, as discussed in [7], provides further context for this research. This study's insights into cost-effective resource allocation strategies underpin our approach to optimize resource efficiency within the SLS system, ensuring that scalability does not come at an unsustainable cost.

Finally, the work in [8] complements the use of real-world traffic data for resource management. By highlighting the importance of data-driven decision making in resource allocation, this study reinforces the analytical foundations of our SLS system.

Taken together, these studies provide a comprehensive backdrop against which our research is situated. Our contribution with the SLS system lies in integrating these various dimensions—adaptive resource management, traffic surge handling, QoS maintenance, budgetary considerations, and data-driven approaches—into a cohesive system designed to address the unique challenges of cloud computing environments.

3. Approach

In this section, we introduce the Soft Landing Scaler (SLS), a system engineered to dynamically scale resources in Kubernetes environments. SLS integrates a sophisticated algorithm that optimizes resource allocation in response to web traffic fluctuations. We detail its design, including the scaling formula, traffic pattern analysis, buffer resource strategy, and performance metrics of resource units. This overview shows how SLS enhances resource management and ensures stability and efficiency in cloud-computing environments under varying traffic conditions.

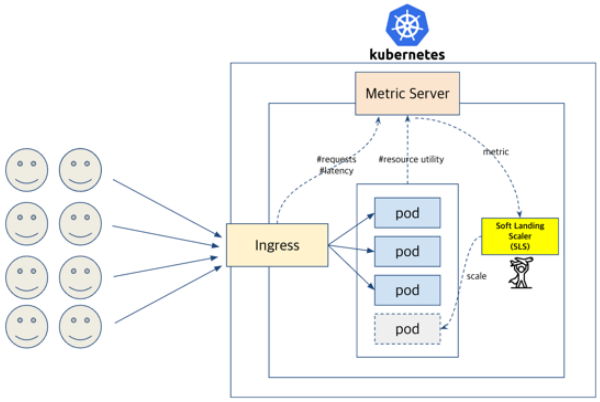


Figure 1. SLS architecture in Kubernetes system

Figure 1 presents the architectural blueprint of the Soft Landing Scaler (SLS) within a Kubernetes orchestrated environment. This schematic is fundamental to understanding the operational mechanics of SLS, which is engineered to

dynamically adjust the allocation of computational resources in response to fluctuating web traffic patterns.

At the forefront of the diagram is an ingress point that serves as the initial receptacle for incoming digital traffic from various user agents, illustrated by a series of smiley-face icons. These icons represent users whose requests are channeled through ingress to the underlying infrastructure.

Within the confines of the Kubernetes framework, indicated by the larger rectangle bearing the Kubernetes insignia, lies a component designated as the "Metric Server." This server plays a pivotal role in the gathering and analysis of performance metrics, such as the number of requests, response latency, resource utilization from ingress, and resource manager. It is intricately linked to a collection of smaller rectangles, each marked as "pod." These pods are discrete units of deployment within Kubernetes and are direct recipients of processed requests.

To the right, ensconced within the Kubernetes demarcation, is the "Soft Landing Scaler (SLS)" module. This module is tasked with a critical function of resource scaling. It is connected to the Metric Server via a dashed arrow adorned with the label "metric," indicating the transmission of analytical data that supports scaling actions.

The overall design of the diagram encapsulates a system of relentless monitoring and dynamic adjustment crafted to ensure the judicious deployment of resources in correspondence with the current demand. This system underscores the implementation of an efficient resource utilization strategy, which is pivotal for the maintenance of stability and performance within cloud computing frameworks, particularly under conditions of variable traffic influx.

3.1. System Design of Soft Landing Scaler (SLS)

The Soft Landing Scaler (SLS) is a system designed for dynamic resource scaling in Kubernetes environments. It operates on a set of predefined metrics, including the transaction volume, response time, and resource utilization. The formula for calculating the optimal number of pods is as follows:

$$P_t = \max\left(P_{transaction}(T_{t-1}), P_{latency}\left(L_{t-1}(R_{t-1})\right)\right) + P_{buffer}$$

P_t represents the required number of pods at time t, considering the highest value among the transaction-based, latency-based, and buffer pod needs. $P_{transaction}$ is derived from historical transactions, whereas $P_{latency}$ scales pods to maintain response times within acceptable limits, and P_{buffer} adds buffer pods to handle unexpected traffic spikes. This ensures that resources are scaled effectively without preemptive predictions, maintains performance, and minimizes waste during downscaling. This approach aligns with the comprehensive container resource management approach discussed by Rodriguez and Buyya (2018), emphasizing the importance of autoscaling based on current workloads in Kubernetes [9].

3.2. Traffic Pattern Research

Our research employed real-world traffic data sourced from web services, which were subsequently classified into distinct patterns, such as Sharp Increase then Exponential Decrease (SIED), Sharp Increase then Linear Decrease (SILD), and Sharp Increase, Linear Decrease, then Steady (SILDS). Traffic patterns were obtained from NAVER services to enhance the realism of our simulation. Traffic pattern models are used to evaluate the system's performance metrics, such as response time and resource utilization, to assess the scaling efficiency under various traffic scenarios.

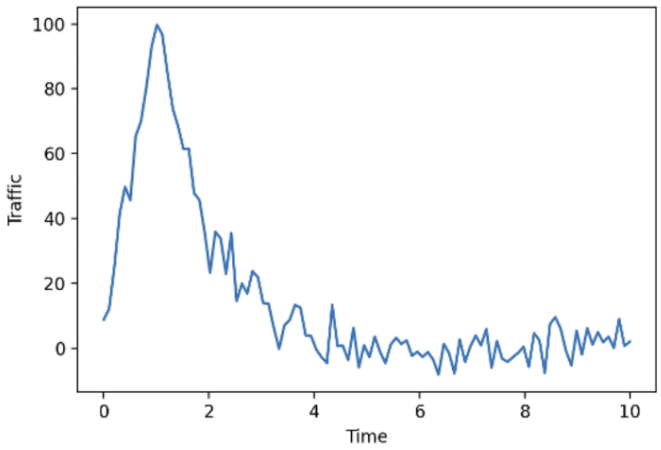


Figure 2. Sharp Increase then Exponential Decrease (SIED) pattern

The first pattern, termed as Sharp Increase then Exponential Decrease (SIED), is typically observed during large-scale online events or product launches. As depicted in Figure 2, this pattern is characterized by an initial surge in user interest, leading to a significant load on the servers. Following the peak of the event, user interest wanes exponentially rather than linearly, marking a rapid decline in traffic after the event.

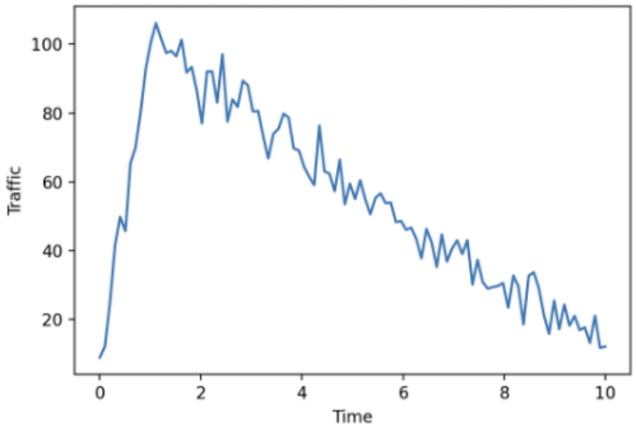


Figure 3. Sharp Increase then Linear Decrease (SILD) pattern

The second identified pattern, Sharp Increase then Linear Decrease (SILD), often emerges in response to the release of significant news articles or popular blog posts. This is illustrated in Figure 3, where the initial spike in user attention is very high, reflecting immediate engagement with new or compelling content. Over time, as shown by the traffic trend, user interest diminishes gradually, transitioning into a linear decline as the focus shifts to newer stories or alternative news.

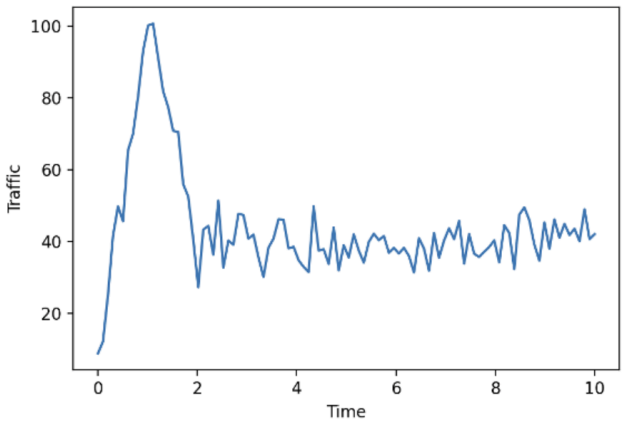


Figure 4. Sharp Increase, Linear Decrease, then Steady (SILDS) pattern

Our third pattern, Sharp Increase, Linear Decrease, then Steady (SILDS), is typically associated with viral campaigns on social media or events with a news element. As shown in Figure 4, there is an initial flocking of users to the content, creating a substantial influx that peaks early. Unlike the previous patterns, the SILDS pattern shows a slight decline over time but then levels off to a steady state of high traffic. This plateau suggests sustained interest over a longer period, likely because of the content's continued relevance or ongoing discussion.

These patterns, SIED, SILD, and SILDS, serve as critical test cases for assessing SLS efficacy. They represent realistic traffic scenarios that SLS must accommodate, ensuring that resource scaling is both responsive and efficient under diverse web traffic conditions. The SLS's ability to adapt to these patterns is crucial for maintaining service stability and performance at varying levels of user demand.

3.3. Need for Buffer Resource

The necessity of buffer resources was identified during the pattern analysis. Following the onset of a surge, traffic exhibits highly irregular behavior, necessitating the use of buffers to cope with such volatility. Buffer resources refer to the additional resources deployed in anticipation of unexpected scenarios. These resources serve to absorb sudden traffic increases and prevent system overloads. Consequently, buffer resources are essential for enhancing the operational stability, ensuring service reliability, and guaranteeing user satisfaction, as discussed in the context of network autoscaling by Serracanta et al. (2021) [10].

3.4 Optimum Measurement of Single Resource Unit

Efficient resource utilization in services necessitates measuring the optimal capacity of each resource unit, which is crucial for deriving the constants in our scaling formula. The performance of a resource unit is evaluated using three key metrics: Transactions Per Second (TPS), Response Time (RT), and Requests Per Second (RPS). A high TPS indicates a robust processing capacity, whereas a low RT reflects a quick response, enhancing the user experience. Although RPS is similar to TPS in representing the server load, it specifically measures the user request volume. Optimal resource performance is identified when increases in TPS and RPS do not significantly impact RT, indicating a balanced capacity and demand.

In this study, we focused on a single Kubernetes pod as a resource unit, drawing on insights from Zhao and Uta (2022) regarding dynamic CPU allocation for serverless functions in Kubernetes [11]. This approach allows for detailed scaling and resource management, enabling us to adjust resources effectively in response to fluctuating traffic demand. By analyzing individual pod performance, we ensure efficient resource use, optimal service delivery, and the determination of the necessary buffer resources to handle traffic volatility and enhance service stability and reliability.

4. Evaluation

In this section, we present a comprehensive evaluation of the Soft Landing Scaler (SLS), a system designed to improve efficiency in cloud computing environments, particularly under the challenging conditions of web traffic surges. The evaluation was structured to rigorously assess the performance and efficiency of SLS by comparing it with traditional scaling methods. This comparison is pivotal for demonstrating the innovative capabilities of SLS in managing resource allocation dynamically and efficiently. Our evaluation methodology is meticulously designed to mirror real-world scenarios, ensuring that the findings are not only theoretically sound, but also practically applicable. We delve into the specifics of our simulation environment, experimental procedures, and key metrics used for evaluation, providing a thorough understanding of how SLS operates under various traffic conditions and its potential impact on cloud computing resource management.

4.1. Simulation Environment

We conducted simulations in a controlled cloud environment using the Kubernetes clusters. These simulations replicated real-

world traffic conditions, allowing us to rigorously test SLS under various traffic scenarios. To benchmark its effectiveness, we compared the performance of SLS with that of traditional scaling methods, such as the Horizontal Pod Autoscaler (HPA) of Kubernetes to benchmark its effectiveness.

4.2. Experimental Procedure

In our study, we initiated a detailed experimental procedure by setting up a Kubernetes environment and implementing SLS. Utilizing Python-based tools, such as Locust, for traffic generation and monitoring systems, such as Prometheus and Grafana, we simulated varied traffic patterns and collected performance metrics. This comprehensive approach allowed us to conduct a real-time analysis of SLS performance compared with traditional scaling models, yielding significant insights into system operations and stress testing under diverse conditions.

4.3. Evaluation Metrics

The evaluation of SLS was conducted with two primary objectives: to evaluate cost and latency.

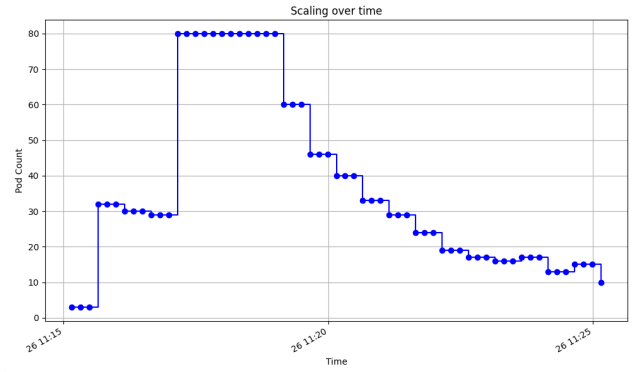


Figure 5. Scaling the Number of Pods in Response to Scaling Operation

$$Cost = \sum_{t=1}^{n} f_{resource}(t)$$

To quantify cost-effectiveness, we expressed it as the total amount of pod usage over time, as shown in Figure 5. It is a fundamental premise that the lower the total pod usage, the higher the cost-effectiveness of the scaling strategy. This is because fewer pods means less resource utilization, leading to lower operational costs.

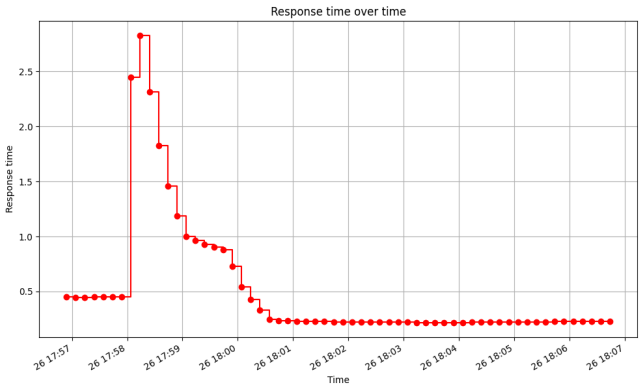


Figure 6. Average Response Time during Scaling Operation

$$Latency = \sum_{t=1}^{n} f_{response}(t)$$

In contrast, the response time was analyzed to guarantee that the SLS could maintain superior service quality during high traffic surges, which are critical times when customer satisfaction can be compromised. Figure 6 shows the response time that users experience when scaling is in progress. A lower total delay time directly correlates with a more responsive system, which is critical for maintaining a seamless user experience.

5. Result & Discussion

In this section, we delineate the outcomes of the empirical evaluation of the Soft Landing Scaler (SLS) as juxtaposed with traditional resource-scaling paradigms within Kubernetes-managed environments. Central to this discourse is the assessment of cost efficiency, response time maintenance, scalability, adaptability, and judicious implementation of scaling size limits. These facets are critically analyzed to determine the efficacy of SLS in striking an optimal balance between resource utilization and operational performance.

5.1. Cost Efficiency

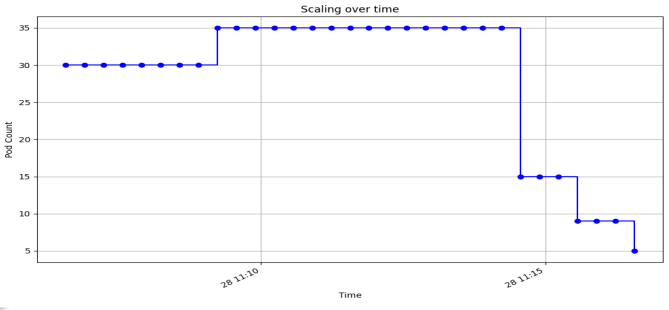


Figure 7. HPA's Scaling the Number of Pods in Response to SIED pattern

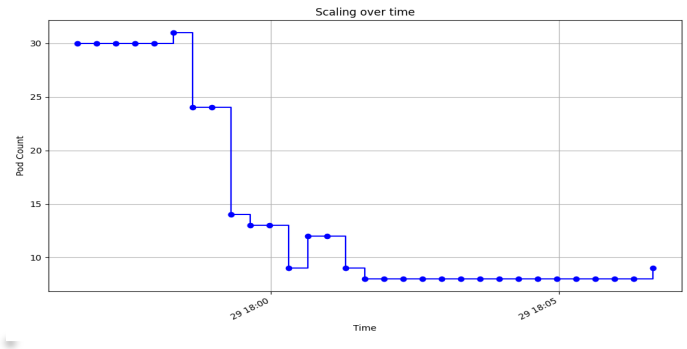


Figure 8. SLS's Scaling the Number of Pods in Response to SIED pattern

For the SIED traffic pattern, SLS achieved a 50% reduction in resource usage compared to the baseline, the Kubernetes Horizontal Pod Autoscaler (HPA). This significant decrease highlights SLS's ability to dynamically tailor resources to current demands, thus avoiding wasteful expenditure. Although there is a 15% increase in response time with SLS, it remains within the acceptable threshold, ensuring the end-user's service quality is unaffected.

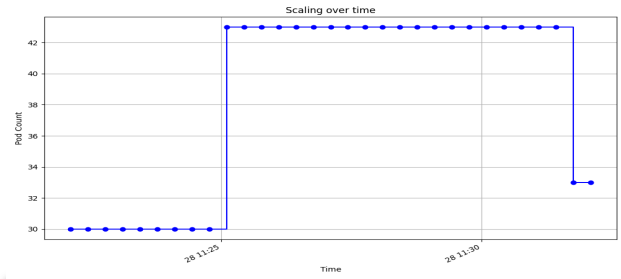


Figure 9. HPA's Scaling the Number of Pods in Response to SILD pattern

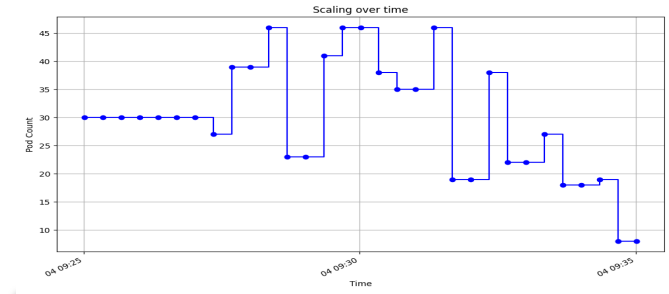


Figure 10. SLS's Scaling the Number of Pods in Response to SILD pattern

In the case of the SILD traffic pattern, SLS demonstrated its cost efficiency with a 23% reduction in resource usage over the baseline established by the Kubernetes Horizontal Pod Autoscaler (HPA). Alongside this reduction, a 25% increase in the response time was observed. However, SLS managed to maintain the latency within levels that did not adversely impact the user experience, thus presenting a favorable balance between cost efficiency and performance.

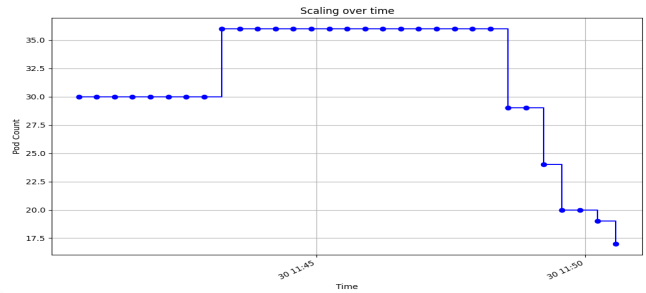


Figure 11. HPA's Scaling the Number of Pods in Response to SILDS pattern

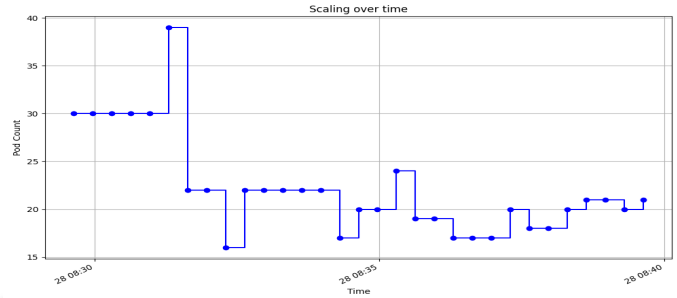


Figure 12. SLS's Scaling the Number of Pods in Response to SILDS pattern

When examining the SILDS pattern, SLS presented 33% cost saving. While there was also a 25% increase in response time, the SLS generally maintained the performance within an acceptable range. However, owing to the high volatility of the SILDS pattern, there were moments when the response time peaked above the threshold, indicating a need for scale volume limitations with SLS to manage such irregularities.

Occasionally, the response time increased rapidly, a situation attributed to the excessive reduction of resources due to the absence of restrictions on scaling size volatility. This particular issue of scaling size volatility and its implications for response time are discussed in more detail in the subsequent section of our analysis.

In each traffic pattern, SIED, SILD, and SILDS—SLS have proven to be efficient alternatives to the baseline HPA, consistently reducing resource costs. The trade-off between resource savings and response time increases is clear, but the overall performance remains within a range that does not compromise user experience. These findings validate the efficacy of SLS in enhancing resource efficiency, while maintaining a satisfactory response time.

The inverse relationship between resource cost and response time is evident across different traffic patterns, suggesting that while SLS prioritizes resource efficiency, it does so with keen awareness of the importance of maintaining service quality. The occasional spikes in latency, specifically in the SILDS pattern, will be further discussed in the following section, including

potential solutions to mitigate such issues.

In conclusion, SLS showed improved resource efficiency compared with traditional scaling methods, while maintaining performance at a level supportive of a positive user experience. In scenarios modeled after the SIED, SILD, and SILDS traffic patterns, SLS demonstrated a 23-50% reduction in resource usage. This efficiency gain indicates the SLS's capability to dynamically allocate resources according to real-time demand, thereby reducing unnecessary resource consumption and the associated costs.

Table 1. Resource usage comparison between Kubernetes HPA and SLS

Pattern	Kubernetes HPA (baseline)	SLS	Result
SIED	883	449	50% decreased
SILD	1196	912	23% decreased
SILDS	974	645	33% decreased

5.2. Adaptiveness

The adaptiveness of SLS were tested under varying traffic conditions. The system effectively handled abrupt traffic spikes and gradually increased, demonstrating its adaptability to a range of traffic dynamics. Compared with static scaling methods, SLS's adaptive approach allows for more precise resource allocation, especially in unpredictable traffic scenarios.

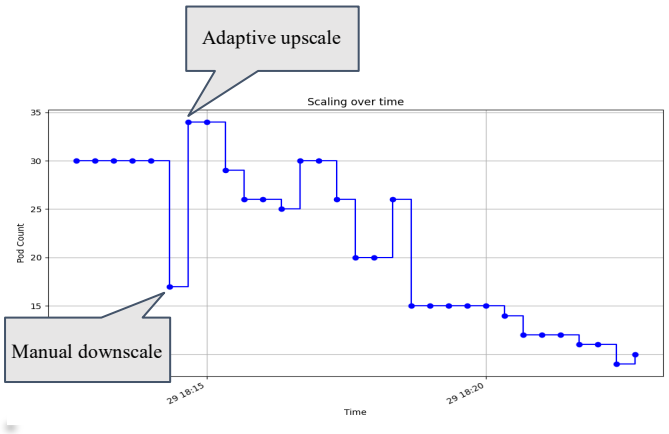


Figure 13. SLS's upscaling immediately after manual downscaling.

Furthermore, it was noted that even when the number of pods was manually reduced using the Kubernetes command-line interface (CLI) during a period of rapid traffic increase, SLS demonstrated a high degree of resilience. It quickly adapts and scales to an appropriate number of pods to handle surges. This responsiveness not only emphasizes the system's ability to cope with human intervention, but also its capability to maintain service continuity and performance in the face of unexpected changes, ensuring that resource allocation remains both efficient and sufficient.

5.3. Effectiveness of Scaling Size Limit

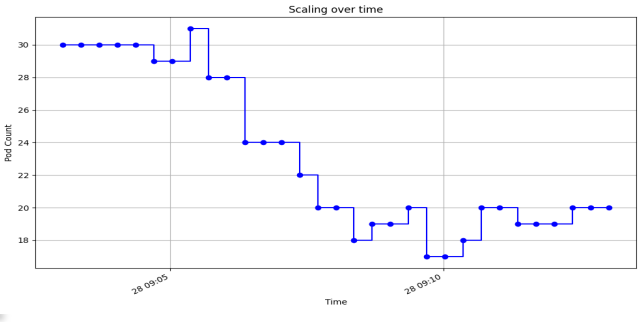


Figure 14. SLS's Scaling the Number of Pods in Response to SILDS pattern with scaling size limit

Without scaling size restrictions in place, as depicted in Figure 12, which shows scaling in response to the SILDS pattern, the system's response time exhibited rapid and intermittent fluctuations that surpassed the threshold after scaling operations began. This volatility is indicative of the scaler's overly sensitive reaction to temporary decreases in response time, resulting in an excessive reduction in resources. Consequently, if a surge in traffic follows this resource reduction, the system may encounter a volume of requests that exceed its capacity, causing response times to escalate beyond the critical point.

Conversely, when scaling size limitations are applied, as shown in Figure 14, these swift fluctuations in response time are mitigated, eliminating spikes that previously breached the threshold. The scaling graph under these conditions illustrates that the volatility of resource volume scaling is reduced compared with the unrestricted scenario, contributing to more stable service delivery.

Performance metrics indicate that implementing scaling-size caps results in a 10% increase in resource costs and a 2% decrease in response time, which are comparable outcomes. This implies that with scaling size constraints, resources cannot be reduced abruptly but must be scaled down incrementally, inevitably leading to higher resource costs. However, by limiting the extent of resource fluctuation, the system has the advantage of maintaining a consistent response time level, even amidst rapid traffic increases.

These findings suggest that imposing limits on scaling size is a strategy that can enhance system stability and augment user experience, underscoring the importance of a balanced approach to resource scaling.

6. Conclusion

Our research culminates in the empirical validation of the Soft Landing Scaler (SLS), a system which has demonstrated remarkable efficacy in managing the erratic nature of web traffic

surges. Through the application of SLS across a series of simulated traffic patterns that closely emulate real-world conditions, namely, the SIED, SILD, and SILDS patterns, the system consistently outperformed traditional resource scaling methods. The findings are robust, highlighting a significant decrease in resource utilization, ranging from 20% to 50%. This reduction was achieved without compromising the response times, which were maintained within an acceptable range even during peak loads. Additionally, SLS has exhibited a level of cost efficiency that positions it as a potentially transformative solution for resource management within cloud-based services.

The results of this study indicate a broader shift towards more intelligent and responsive web service platforms. The SLS's dynamic scaling approach, which is both proactive and reactive, not only allows for real-time adaptation to traffic demands but also paves the way for more sustainable cloud service practices. By reducing resource consumption and ensuring high availability during demand spikes, SLS aligns with the evolving needs of the digital landscape, where user expectations and environmental considerations are of paramount importance.

6.1. Implications of the Findings

The ramifications of our study's findings are significant for cloud-based web services. The dynamic resource management capabilities of SLS demonstrate a substantial advancement towards achieving greater operational sustainability and efficiency within cloud environments. Such a system can lead to profound cost savings, particularly for services that frequently encounter fluctuating web traffic. Additionally, the ability of the SLS to uphold service quality by maintaining response times within desirable limits is a critical factor in enhancing user experience and satisfaction, a key metric for the success of any web service.

These findings also emphasize the role of SLS in addressing some of the intrinsic challenges faced by cloud services. The balance between resource provisioning and cost management is delicate, and the SLS offers a nuanced solution that minimizes waste while ensuring service reliability. This has potential implications not only for the financial bottom line of service providers, but also for the broader goal of creating environmentally conscious, green computing practices within the industry.

6.2. Limitations and Future Work

While the outcomes of this study are promising, it is imperative to recognize the limitations inherent in any research. The primary limitation of this study is its reliance on simulated traffic patterns, which, despite being derived from real-world data, may not capture all complex variables of live web traffic. Future research should therefore include the deployment of SLS in actual service environments to comprehensively evaluate its effectiveness under real-world conditions. This would allow for a more detailed understanding of SLS performance and potential areas for improvement.

Furthermore, the incorporation of advanced machine learning techniques to refine SLS's traffic prediction capabilities could significantly enhance its performance. By accurately forecasting traffic surges, SLS can preemptively scale resources with greater precision, thereby improving its efficiency and overall user experience. Future work in this area could lead to considerable advancements in the field of resource scaling and cloud computing.

6.3. Contribution to the Field

This study enriches the field of cloud computing and web services using an innovative resource-scaling strategy exemplified by the Soft Landing Scaler (SLS) framework. What sets the SLS apart is its proficiency in pre-empting web traffic surges, providing a robust solution for service providers to prepare for and adeptly manage periods of high demand. The model is particularly advantageous in scenarios where traffic spikes are predictable, allowing for preemptive scaling that ensures resource availability while maintaining efficiency. This foresight, coupled with SLS's adaptability, makes it an indispensable tool in modern digital environments, where even predictable traffic patterns can present significant operational challenges.

7. Acknowledgement

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2020-0-01795) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

8. References

[1] Shan, C., Wu, C., Xia, Y., Guo, Z., Liu, D., & Zhang, J. “Adaptive Resource Allocation for Workflow Containerization on Kubernetes.”

[2] Zhou, Z., Zhang, C., Ma, L., Gu, J., Qian, H., Wen, Q., Sun, L., Li, P., & Tang, Z. “AHPA: Adaptive Horizontal Pod Autoscaling Systems on Alibaba Cloud Container Service for Kubernetes.”

[3] Bekcheva, M., Fliess, M., Join, C., Moradi, A., & Mounier, H. “Improving resource elasticity in cloud computing thanks to model-free control.”

[4] Yi Wei, M. Brian Blake, Iman Saleh, “Adaptive Resource Management for Service Workflows in Cloud Environments.”

[5] Heng Lu, Haopeng Chen, Sixiang Ma, Wenyun Dai, Pu Xing, “Dynamic Virtual Resource Management in Clouds Coping with

Traffic Burst.”
[6] Baldev Singh, “Resource Provisioning in Cloud Environment for Enhanced Quality of Service.”
[7] Qian Zhu, Gagan Agrawal, “Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments.”
[8] Rodrigo N. Calheiros, Rajiv Ranjan, Rajkumar Buyya, “Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments.”
[9] M. A. Rodriguez and R. Buyya, "Containers Orchestration with Cost-Efficient Autoscaling in Cloud Computing Environments."
[10] B. Serracanta, J. Paillisse, A. Cabellos, A. Claiborne, A. Rodriguez-Natal, D. Ward, and F. Maino, "Wide Area Network Autoscaling for Cloud Applications."
[11] Y. Zhao and A. Uta, "Tiny Autoscalers for Tiny Workloads: Dynamic CPU Allocation for Serverless Functions."

확률적 모델 체킹을 이용한 깃허브 기반의 프로젝트 참여자 기여 분석

조수희, 장지영, 박소희, 권기현
경기대학교 일반대학원 SW안전보안학과
{sujo, gzerosa, sosop, khkwon}@kyonggi.ac.kr

Participant Contribution Analysis of GitHub Project using Probabilistic Model Checking

Suhee Jo, Jiyoung Chang, Sohee Park, Gihwon Kwon
Dept. of Software Safety and Cyber Security, Kyonggi University

요 약

본 논문은 이산시간 마코프 체인을 사용하여 GitHub 기반 프로젝트에서 개발팀의 협업을 분석한다. 우리는 모델링의 검증과 프로젝트 관리를 위한 개선점을 확인하고자 확률적 시제 논리를 이용하여 질의를 명세하고 모델 체킹을 수행한다. 이를 통해 프로젝트 참여자의 병합률, 참여 수준, 개발 주기, 평균 커밋에 대한 정량적 수치를 계산하였고 그 결과 프로젝트 관리자와 각 개인 기여자의 특징에 대해 분석할 수 있었다. 이와 같이 제안된 방법이 상위 수준의 기여자를 식별하고 프로젝트에 대한 지속적인 참여를 장려하는데 도움이 될 수 있을 것으로 기대한다.

1. 서 론

소프트웨어 프로젝트의 생산성과 품질을 향상시키기 위해서는 팀 내에서 원활한 협업이 이루어져야 한다 [1]. 그러나 협업 활동을 식별하고 수준을 측정하는 것은 어려운 문제이다. 이를 해결하기 위해서는 전반적인 프로젝트 개발 활동의 프로세스와 개발자들 간의 관계를 파악한 프로젝트 관리가 필요하다.

본 논문에서는 협업 활동을 분석하기 위해 DTMC(Discrete Time Markov Chain) [2] 모델과 pCTL (probabilistic Computation Tree Logic) [3]을 이용하여 개발 활동의 프로세스를 모델링하고 이를 모델 체킹(Model Checking) [4] 하는 방안을 제안한다. 여기서 우리는 깃허브 저장소에 대해 팀 내의 기여도와 활동의 관계를 파악하는 사례 연구를 제공한다.

2. 제안 연구

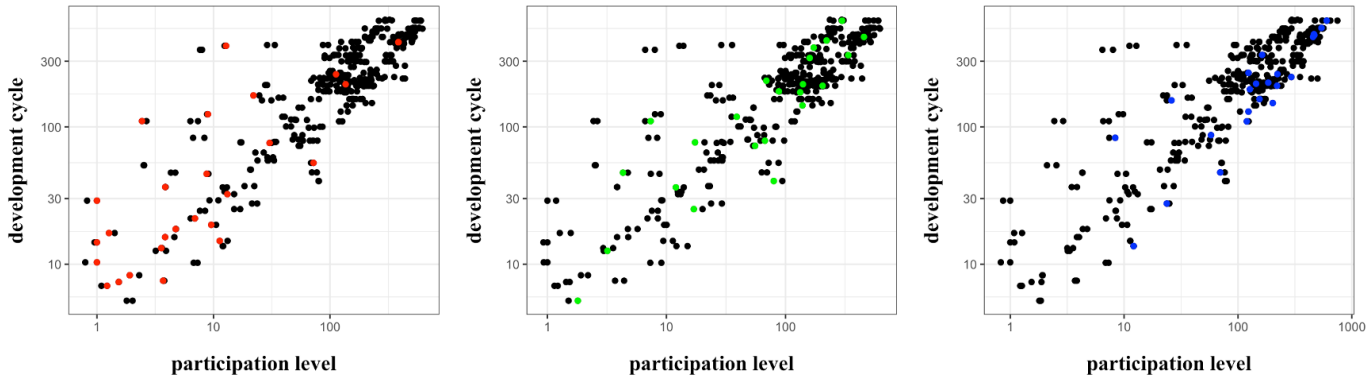
데이터 수집. 팀 내부에서의 협업 네트워크를 분석하기 위해 소규모 팀을 대상으로 데이터를 수집하였다. 자세한 기준은 다음과 같다: 1) 2018년도 이후에 생성된 저장소. 2) 3~10명의 기여자. 3) 최소 100개 이상의 커밋 로그. 4) star 100개 이상. 본 연구에서는 산출된 저장소 중 38개의 저장소를 샘플링하여 사용한다. 이를 통해, 총 331명의 참여자와 총 20,000여개의 활동 이벤트 로그를 추출하였다. 이때 참여자란 저장소에 기여하기 위한 활동을 적어도 한번 이상 수행한 사람을 의미한다.

깃허브 모델링. 본 연구에서 사용된 DTMC의 상태는 다음의 3-튜플로 정의된다:{participant, location, action}. 여기서 participant는 프로젝트에 참여한 개인을 의미하며 익명성을 위해 프로젝트에 참여한 순서대로 p1부터 pn까지 할당된다. location은 작업이 수행된 브랜치의 위치를 나타낸다. 기본(default) 브랜치를 의미하는 "master"와 그 이외의 다른 브랜치를 의미하는 "branch"로 구성된다. action은 다음과 같이 6가지 유형의 개발 활동을 포함한다: {create, commit, pullRequest, rejected, accepted, merge}. 표 1에서는 본 논문에서 제공하는 pCTL 기반의 4가지 질의(Query) 명제를 정의한다. 표 1의 나열된 질의 명제들에 대한 해석은 다음과 같다:

- **Q1**: 장기적으로 각 팀원이 병합을 수행할 확률이다. 이를 '**병합률** (merge rate)'이라 부른다.
- **Q2**: 참여자에게 도달하기 위한 평균 스텝이다. 이를 '**참여 수준** (participation level)'이라 부른다. 질의 결과값이 작을수록 참여 수준은 높아진다.
- **Q3**: 참여자가 "pullRequest"를 수행한 후 새로운 브랜치를 생성하는 데 걸리는 시간이다. 이는 하나의 작업을 종료하고 새로운 작업을 수행하는 개발 주기로 간주한다. 이를 '**개발 주기**(development cycle)'라 부른다.
- **Q4**: 풀 리퀘스트에 포함된 커밋의 평균 수이다. 이를 '**평균 커밋** (average commit)'이라 부른다.

표 1. 질의 명세 목록

	Query Specification
Q1	$S =? [participant \ \& \ "merge"]$
Q2	$R\{"r\_steps"\} =? [F \ participant]$
Q3	$filter(avg, R\{"r\_steps"\}$ $=? [F \ participant \ \& \ "create"], participant \ \& \ "pullRequest")$
Q4	$filter(avg, R\{"r\_participant\_branch\_commit"\}$ $=? [F \ participant \ \& \ "pullRequest"], participant \ \& \ "create")$



(a) p1: 빨간 포인트 (b) p2: 초록 포인트 (c) p3: 파란 포인트

그림 1. 참여 수준과 개발 주기의 상관관계 분석

3. 사례 연구

프로젝트 관리자에 대한 분석. 깃허브에서 풀 리퀘스트를 병합(merge)하는 것은 해당 저장소에 대해 권한이 있는 사람만이 가능하다. 본 논문에서는 이 특징을 이용하여 프로젝트 관리의 권한에 따른 변화를 살펴본다. 먼저, 병합에 대한 경험은 Q1을 통해 얻는다. 모델 체킹 결과, 38개의 저장소 중 15개에서 단 한명만이 병합을 수행한 것으로 나타났다. 또한, 각 저장소에서 병합률이 가장 높은 사람들에 대해 평균을 구한 결과는 78.22%, 중앙값은 91.53%로 산출되었다. 이러한 값은 대부분의 프로젝트가 한 명의 팀원을 통해 관리된다는 것을 알 수 있다. 흥미로운 점은, 63.16%의 비율로 p1이 각 저장소의 가장 높은 병합률을 기록하였다. 이는 프로젝트를 제일 처음에 시작한 참여자가 프로젝트 관리자 역할을 할 가능성이 높다는 것을 보여준다.

참여자 개인의 기여도에 대한 분석. 그림 1은 참여 수준(Q2)과 개발 주기(Q3)의 관계를 보여준다. 이를 통해 참여 수준이 낮을수록 개발 주기가 증가하는 경향이 있음을 드러낸다. 프로젝트에 참여한 순서에 따른 변화를 분석하기 위해 그래프 상에 p1, p2, p3의 위치를 서로 다른 색(순서대로 빨강, 초록, 파랑)의 포인트로 표시하였다. 그 결과, p3는 p1에 비해 참여 수준이 낮고 개발 주기가 긴 것으로 나타나며, 프로젝트에 늦게 참여할수록 참여 수준이 감소하는 경향이 있음을 알 수 있다. 즉, 개발 속도가 느려지고 잠재적으로 프로젝트 포기로 이어질 수 있는 가능성이 있기 때문에 장기적인 프로젝트 지속을 위한 노력이 필요하다. 주목할 점은 분석 대상 팀의 구체적인 특성에 따라 참여 수준과 개발 주기의 관계에 대한 해석이 달라질 수 있다는 점이다. 예를 들어, 짧은 개발 주기와 낮은 참여 수준은 참여자가 기능 개발을 완료한 후 바로 다음 기능 개발 작업에 착수했다는 것을 의미하지만 기능 개발에 있어 오랜 시간이 걸려 어려움을 겪고 있다는 것을 나타낼 수 있다. 참여 수준과 개발 주기의 상관 관계 모니터링은 프로젝트의 지속 가능성 평가 지표를 제공하여 다양한 분석이 가능하고, 프로젝트를 정상 궤도에 유지하기 위한 추가적인 지원이나 자원이 필요할 수 있는 영역을 식별하는 데 도움이 된다.

풀 리퀘스트에 포함된 커밋의 수는 종종 풀 리퀘스트의 병합 결정에 영향을 미치기 때문에 평균 커밋(Q4)은 개발 프로세스에서 품질과 효율성을 평가하는 데 유용할 수 있다. 모델 체킹 결과는 평균 약 3.54로, 각 풀 리퀘스트는 평균적으로 3개 정도의 커밋을 포함한다는 것을 의미한다.

4. 결 론

본 연구에서는 오픈 소스 프로젝트의 개발 활동을 모델링하고 관리하기 위해 DTMC 기반의 접근 방식을 제안하였다. 개인의 활동 패턴을 식별하고 프로젝트 관리의 개선을 위한 요소들을 관찰하기 위해 4가지의 질의를 이용하였다. 우리는 이를 통해 프로젝트에 제일 먼저 기여한 참여자가 관리자 역할을 맡아 혼자 프로젝트를 관리할 가능성이 더 높다는 것을 발견했다. 또한, 원활한 협업을 위해 늦게 참여하는 사람들의 기여를 독려할 필요가 있음을 파악할 수 있었다.

현재 연구는 소규모 프로젝트를 중심으로 수행되었다. 추후에는 대규모 프로젝트에 대해 적용하여 소규모 프로젝트의 차이와 공통점을 분석하여 규모에 따른 프로젝트 관리 접근 방법에 대해 연구하고자 한다. 이러한 연구 방향을 통해 오픈 소스 프로젝트의 지속 가능성과 효율성을 높이는 방안을 모색할 것이다.

사사의 글

이 논문은 과학기술정보통신부 및 정보통신기술 진흥센터의 고안전 SW 개발을 위한 안전 분석 및 검증 도구 기술 개발 사업의 연구 결과로 수행되었음 (No. 2021-0-00122)

참고 문헌

[1] J. Hahn, J. Y. Moon and C. Zhang, "Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties", Information Systems Research, vol. 19, no. 3, pp. 360-391, 2008.

[2] W. Ching, X. Huang, M. K. Ng and T. K. Siu, "Markov chains: Models, algorithms and applications", Springer Publishing Company, 2013.

[3] F. Ciesinski and M. Größer, "On probabilistic computation tree logic," Validation of Stochastic Systems: A Guide to Current Research, pp. 147-188, 2004.

[4] M. Kwiatkowska, G. Norman and D. Parker, "Probabilistic model checking: Advances and applications", in Formal System Verification, Cham: Springer International Publishing, pp. 73-121, 2018.

STPA, FTA 및 FMEA 분석 기법을 연계한 소형 항공기 충돌 회피 소프트웨어 안전 요구사항 도출

이종원[1(O)]· 이의천[1,2]· 김태환[2]· 이선아[1,2,†]
경상국립대학교 공과대학 항공우주 및 소프트웨어공학부[1]
경상국립대학교 공과대학 AI융합공학과[2]
ranger3264@naver.com, rndnjswk123@naver.com, up_qut@naver.com, saleese@gnu.ac.kr

Safety Requirement Extraction for Small Aircraft Collision Avoidance Software using STPA, FTA and FMEA

Jongwon Lee[1(O)], Uicheon Lee[1,2], Taehwan Kim[2], Seonah Lee[1,2,†]
[1]Department of Aerospace and Software Eng., Gyeongsang National University
[2]Department of AI Integrative Eng., Gyeongsang National University

요 약

최근 항공 모빌리티는 소형화되고, 소프트웨어 집약적 시스템으로 변화하고 있다. 이에 항공기 안전성 평가 프로세스인 ARP4761 중심의 안전성 분석 기법도 소프트웨어를 중심으로 변화할 필요가 있다. 본 논문에서는 STPA, FTA 및 FMEA를 연계한 안전성 분석 방법을 제안하며, 소형 항공기 충돌 회피 시스템의 사례 연구를 통해 본 방법의 소프트웨어 분석에 대한 효과를 확인하고, 안전 요구사항을 도출하였다.

1. 서 론

ARP4761[1]은 전체 시스템 계층에서 소프트웨어와 관련된 안전 요구사항을 할당하기까지 상당히 복잡한 절차를 거치며 컴포넌트 중심의 분석이 수행되기에, 소프트웨어 집약적 시스템에 적용시 누락될 수 있는 위험들이 존재한다. 우리는 소형 항공용 시스템의 주요 소프트웨어 컴포넌트 개발을 위한 위험 분석 및 안전 요구사항 도출 방법으로, STPA(Systems Theoretic Process Analysis), FTA(Fault Tree Analysis) 및 FMEA(Failure Modes and Effects Analysis)를 활용하여 연계하고자 한다.

2. 제안 방법

본 논문은 먼저 STPA로 시스템 계층의 제어 관계 기반 추상화를 통해 분석범위 축소 및 컴포넌트의 상호작용에서 발생하는 위험 분석을 수행한다. 이후 FTA로 원인-결과 관계를 직관적으로 표현 및 원인을 세분화한다. 마지막으로 FMEA로 원인의 영향과 위험도를 파악하여 우선순위를 적용하고, 이를 바탕으로 안전 요구사항을 도출한다.

2.1. STPA

STPA[2][3]는 하향식 분석으로, 사고 및 위험을 정의 후 위험이 발생하지 않도록 하는 안전 제약사항을 정의한다.

이후 제어 구조(Control Structure)를 작성하여 각 컴포넌트 간의 제어 관계를 파악한다. 소프트웨어 및 복잡한 시스템은 이 과정에서 추상화를 통한 단순화가 진행되기에, 소프트웨어를 유의미한 단위로 분석할 수 있다.

2.2. FTA

FTA[4][5]는 결함(Fault)을 유발하는 원인들의 논리적 조합으로 결함 트리(Fault Tree)를 그려 분석하는 하향식 분석이다. FTA는 'chain of events' 모델을 채택하여 원인-결과 관계를 직관적으로 보여줄 수 있으며, 사고 원인들에 대해 계층적으로 분석한 결과를 포함하여 연쇄적 연결 관계를 도출할 수 있다.

2.3. FMEA

FMEA[6][7]는 시스템을 구성하는 하위 시스템 혹은 기능의 고장 상태(Failure Mode)가 시스템에 미치는 영향(Effect)을 도출하는 상향식 분석 방법이다.

FMEA는 STPA, FTA와는 달리 각 고장에 대해 위험도(Severity)를 분석하기에 해결해야할 문제들에 대한 우선순위를 결정할 수 있어, 다른 하향식 분석과 상호보완적이다.

2.4. STPA, FTA 및 FMEA 연계 분석

연계 분석 과정은 그림 1과 같이, 하향식 분석 기법인 STPA, FTA를 우선 수행 후 상향식 분석 기법인 FMEA를 수행한다.

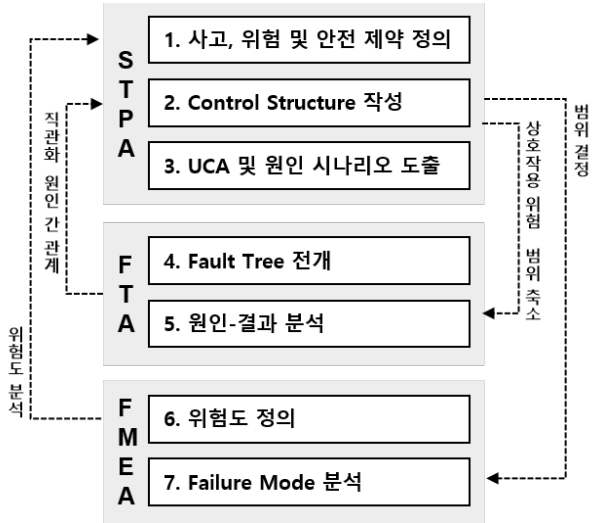


그림 1. STPA, FTA 및 FMEA 연계 방법

STPA는 제어 관계 기반의 추상화를 통해 무형의 소프트웨

어를 유의미한 형태로 나타내어 분석 범위를 축소하고, 상호 작용 위험을 비롯한 다양한 위험의 원인을 분석할 수 있다. 이러한 사고, 위험 및 원인들은 FTA를 통해 트리 형태로 원인-결과 관계를 가시화하고, 위험 및 원인들의 연쇄적 연결 관계를 분석한다. 이후 원인들은 FMEA의 고장 상태가 되어, 영향을 분석하고, 정량적인 위험도를 통해 안전 우선순위를 도출하게 된다. 이러한 결과물을 토대로 안전 요구사항의 명세가 가능하다.

3. 사례 연구

본 논문에서는 소형 항공기의 개발 상황에서, 소프트웨어 시스템인, 충돌 회피 시스템(Collision Avoidance System, CAS)의 안전 요구사항 도출하기 위한 위험 분석 수행에 대해 보고한다.

3.1. STPA 적용

STPA를 통해 소프트웨어 시스템을 추상화하고, 제어 기반의 위험 및 다양한 위험을 분석할 수 있다.

3.1.1. 사고, 위험 및 안전 제약 정의

항공기의 충돌 회피 수행이 감시-분석-계획-수행의 연속적 4단계를 거쳐 수행되는 모델을 전제하였으며, 그에 따라 항공기 수준에서 위험 및 위험에 대응되는 안전 제약사항을 정의하였다.

3.1.2. 제어 구조 정의

조종사-CAS-항공기로 추상화 이후 CAS 내부에 대해서 4단계 모델을 토대로 상세화를 진행하여 안전 제약사항을 모두 반영한 제어 구조를 도출하였다.

3.1.3. UCA 및 원인 시나리오 도출

제어 구조의 모든 UCA(Unsafe Control Action)를 도출하였으며, 크게 수행 모델(Process Model), 제어 알고리즘(Control Algorithm) 및 피드백(Feedback)으로 인한 원인 시나리오들이 식별되었다.

3.2. FTA 적용

결함 트리로 원인 간 계층 및 연쇄 관계를 도출하여 위험분석하고, 사고의 원인-결과 관계를 직관적으로 파악할 수 있다.

3.2.1. 결함 트리 전개

UCA의 원인 시나리오는 결함 트리의 기초 사건이 되어 전개하였다. 이에 기초 사건이 사고에 이르기까지의 원인-결과 관계를 직관적으로 확인할 수 있다.

3.2.2. 원인-결과 분석

본 과정을 통해, STPA에서는 분석 관점 차이로 인해 표현되지 않았던 이벤트 연쇄적 관계가 분석되었다.

3.3. FMEA 적용

고장 상태의 영향 및 위험도 분석을 수행하고, 위험도에 따른 안전 우선순위를 도출할 수 있다.

3.3.1. 위험도 정의

본 사례 연구는 STPA의 제어 관계를 기반으로 수행하였으므로. 시스템에 영향을 미치는 제어 수준에 따라 위험도를 정의하였다.

3.3.2. 고장 상태 분석

FTA의 각 기초 사건들은 고장 상태로 설정되며, 그에 대한 CAS의 각 기능 실패로 인한 영향을 분석하고, 이를 통해 이전 단계에서 정의한 위험도를 부여하였다.

3.4. 안전 요구사항 도출

본 연계 과정을 통한 결과물을 바탕으로 사고의 원인으로 인한 위험이 발생하지 않게 하거나, 발생 시 대응할 수 있는 기능 등 안전을 위해 시스템에 요구되는 우선순위가 있는 안전요구사항을 도출할 수 있다.

4. 결론

본 논문은 STPA, FTA 및 FMEA를 연계하여 항공기 소프트웨어의 안전성을 분석하는 방법을 최초로 제안하였다. STPA는 소프트웨어를 분석에 유의미한 단위로 추상화 및 상호 작용 위험과 원인 도출이 가능했다. 연이어 FTA는 각 원인의 계층화 및 연쇄 관계 도출로 STPA를 보완했으며, FMEA는 각 원인에 대한 위험도 분석이 가능했다.

안전성 기법들의 이러한 연계 방법은 시스템의 안전성 분석 범위를 줄이면서도 전통적인 안전성 분석 기법의 세분화된 절차를 충족하여, 최근 소프트웨어 집약적 시스템으로 변화되는 항공 모빌리티의 안전성 분석 및 안전 요구사항 도출을 용이하게 하리라 기대된다.

본 논문이 제안한 연계 방법은 다양한 유형의 실제 시스템 및 소프트웨어 시스템에 대한 사례 연구가 향후 수행되어, 방법론 검증 및 효율성 향상에 대한 연구가 수행될 필요가 있다.

Reference

[1] “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. ARP 4761.” SAE International (1996)

[2] Ishimatsu, Takuto, et al. "Hazard analysis of complex spacecraft using systems-theoretic process analysis." Journal of spacecraft and rockets 51.2 (2014): 509-522.

[3] Leveson, Nancy G., and John P. Thomas. “STPA Handbook.” MIT (2018)

[4] Kabir, Sohag. "An overview of fault tree analysis and its application in model based dependability analysis." Expert Systems with Applications 77 (2017): 114-135.

[5] Yazdi, Mohammad, et al. "Fault tree analysis improvements: a bibliometric analysis and literature review." Quality and Reliability Engineering International (2023).

[6] Filz, Marc-André, et al. "Data-driven failure mode and effect analysis (FMEA) to enhance maintenance planning." Computers in Industry 129 (2021): 103451.

[7] Talwar, Palak. "Software failure mode and effects analysis." Intelligent Human Systems Integration 2020: Proceedings of the 3rd International Conference on Intelligent Human Systems Integration (IHSI 2020): Integrating People and Intelligent Systems, February 19-21, 2020, Modena, Italy. Springer International Publishing,2020.

오토인코더 기반 조인트 상태 진단을 통한 협동 로봇의 작업 수행 성능 및 건전성 평가 기법

최민서 [1(O)], 김진세 [1], 이정원 [1,2]
아주대학교 AI 융합네트워크학과 [1], 아주대학교 전자공학과 [2]
minseo24@ajou.ac.kr, jinsae913@gmail.com, jungwony@ajou.ac.kr

Autoencoder-Based Joint State Diagnostics for Collaborative Robot Task Performance and Health Assessment Method

Minseo Choi[1(O)], Jinse Kim[1], Jung-Won Lee[1,2]
Department of AI Convergence Network, Ajou University[1]
Department of Electrical and Computer Engineering, Ajou University[2]

요 약

협동 로봇은 다양한 작업의 유연한 수행과 여러 객체와의 협업이 가능한 특성으로 인해 산업 분야의 수요가 증가하고 있다. 하지만, 협동 로봇은 높은 작업 복잡도 및 동적 환경 변화로 인해 일반적인 PHM(Prognostics and Health Management) 기술을 적용하기에는 한계가 존재한다. 따라서, 본 논문은 협동 로봇의 PHM 요구사항을 체계적으로 분석 및 구체화하는 과정을 통해 조인트 센서 데이터 기반의 건전성 평가 기법을 제안한다. 제안 기법은 오토인코더 모델 기반의 조인트 상태 레벨 평가 단계와 종합된 평가 결과 기반의 작업 수행 성능 레벨 도출 단계로 구성되어, 디바이스 통합 상태 및 건전성을 평가할 수 있다. 본 기법의 검증을 위해 협동 로봇의 카 매트 접착 작업 수행 시 과적재된 하중에 의해 발생한 건전성 저하를 조인트별 토크 데이터를 통해 도출하는 실험을 진행하였다. 실험 결과, 다양한 작업에 대해 효과적인 이상성 기준을 도출하여 조인트의 상태를 진단함과 동시에, 과도한 부하에 따라 협동 로봇에 발생한 25.00% 이상의 건전성 저하를 평균 약 72.22%의 정확도로 도출하였다.

1. 서 론

제조 현장에서 산업용 로봇의 사용은 빠른 생산 속도와 생산 비용 절감 등 다양한 측면의 이익을 가져온다[1]. 그 중, 협동 로봇(Collaborative Robot)은 인간과 작업 환경을 공유하며 제조 및 생산 프로세스의 혁신을 촉진하고 있다[2,3]. 협동 로봇은 충돌 감지와 같은 위험 관리 모듈이 탑재되어 있어 인간과 협력 또는 로봇 간의 협력을 지원한다[4]. 또한, 협동 로봇은 작업자의 프로그래밍에 의한 직관적인 조작 및 전환이 가능한 특성을 바탕으로, 픽앤플레이스(Pick-and-Place), 용접(Welding), 접착(Gluing) 등 다양한 작업을 유연하게 수행할 수 있어 효율적인 작업 환경 조성에 기여한다[5,6].

협동 로봇에 대한 산업적 수요 증가에 따라 효과적인 예지보전 및 유지보수 기술에 대한 관심이 증가하고 있다[7,8]. 대표적으로 건전성 예측 및 관리 기술(Prognostics and Health Management, PHM)은 모션 센서, 토크 센서, 전류 센서 등 설비 기기에 탑재된 다양한 센서로부터 추출되는 센서 데이터를 통해 이상(Anomaly)을 감지하고, 사전에 고장을 예측 및 진단하는 기술이다. 효과적인 PHM 기술은 고장에 의해 발생하는 기기 교체와 시스템의 다운타임 그리고 정기적인 기기 예방 보수에 따른 비용적 손실을 절감시켜 작업 환경의 안전성을 향상시킬 수 있다[9,10]. PHM 기술은 IEEE 표준에 의해 물리, 데이터, 확률 등에 기반한 다양한 방법론으로 구체화되어 산업용 설비의 예지보전에 효과적으로 사용되고 있다[11,12,13]. 최근에는 Industry 4.0과 인공지능의 발전에 따라 데이터 기반의 방법론과 딥러닝을 결합한 PHM 기술의 수요가 증가하고 있다[14,15].

대부분의 PHM 기술은 반복 작업을 수행하는 기기의 유지보수를 다뤄왔기 때문에 고정된 작업을 수행하는 산업용 기기에는 효과적인 적용이 가능하지만, 다양한 작업을 수행하는 협동 로봇에 적용하기에는 한계가 있다. 협동 로봇은 수행하는 작업 프로그램에 따라 가해지

[1] 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2023R1A2C1006332).

는 하중, 작업의 경로, 작업에 대한 각 조인트의 사용 여부 등이 상이하며, 그에 따라 추출되는 센서 데이터의 특성과 패턴이 다르게 도출된다. 즉, 협동 로봇의 건전성 평가를 위해서는 물리적 및 기능적 제약 없이 실시간 데이터를 효과적으로 측정하고 분석할 수 있는 효과적인 건전성 평가 기법이 요구된다.

따라서, 본 논문은 협동 로봇의 건전성 평가를 위한 PHM 기술 개발 요구사항을 체계적으로 분석하여, 작업 단위 건전성 평가를 위한 오토인코더(AE) 기반 조인트 세부 상태 평가 및 작업 수행 성능 진단 기법을 제안한다. 제안하는 기법은 협동 로봇의 작업 간 추출되는 조인트의 내부 센서 데이터를 활용하여, 작업 수행 성능을 반영하는 건전성 평가를 지원한다. 이는 다양한 작업에 대해 모듈화 된 기법의 적용을 통해 작업 별 세부 상태를 진단 및 평가함으로써, 정밀하고 효과적인 협동 로봇의 건전성 평가가 가능하게 한다.

제안하는 기법의 효용성을 검증하기 위해 협동 로봇의 접착 작업 시나리오를 구축하여 실험에 적용하였으며, 검증 결과 최소 25.00%에서 최대 58.82%의 건전성 저하 추이를 보임으로써 효용성을 검증하였다.

본 연구의 주요 학문적 기여를 요약하면 다음과 같다.

- 본 논문은 협동 로봇의 건전성 평가를 위한 PHM 요구사항을 구체화한 건전성 평가 모델을 통해 작업 성능과 건전성을 효과적으로 평가한다.
- 본 논문은 협동 로봇 조인트 단위의 세부적인 이상 특성을 고려한 건전성 평가 기법을 제공함으로써, 협동 로봇의 물리적인 특성을 반영한 신뢰성 있는 건전성 평가를 지원한다.
- 본 논문의 제안 기법을 통해 세부적인 레벨 단위의 건전성 평가가 가능하며, 기법의 작업 별 모듈화를 통해 정밀하고 효과적인 협동 로봇의 건전성 평가를 가능하게 한다.

2. 관련 연구

본 절에서는 산업용 기기에 주로 적용되는 물리 기반의 PHM 방법론과 데이터 기반의 PHM 방법론의 관련 연구를 분석하고 서술한다.

2.1 물리 기반의 PHM 방법론

물리 기반의 방법론은 실제 기기 및 부품의 위치 및 상관 관계를 물리적인 모델로 정의하고, 이를 통해 손상, 성능 저하 등의 상태를 진단하고 미래 상태를 예측 및 평가하는 PHM 기술이다. 대표적으로 [16]은 가스 터빈 엔진(Gas Turbine Engines)의 성능 저하 모니터링을 위한 물리 기반 방법론을 제안한다. 해당 연구는 엔진의 작동 조건과 사이클 매개변수와의 관계를 설명하는 열역학 모델을 기반으로 GTE의 성능 저하를 모니터링한다. 또 다른 연구로 [17]은 회전 요소 베어링(Rolling Element Bearings)의 상태 진단을 위해 결함의 크기와 상태 변화를 물리적으로 모델링하여 베어링의 상태 저하를 파악하고 고장을 확률적으로 계산하여 예측한다. 이와 같이 물리 기반의 방법론은 기기의 작동 원리에 기반한 신뢰성 있는 건전성 평가를 제공한다.

물리 기반의 방법론의 구현을 위한 정확한 모델링은 해당 도메인에 대한 깊은 전문 지식과 기기가 작동하는 환경에 대한 제약 조건에 대한 정확한 파악을 요구한다. 하지만, 협동 로봇의 예지보전은 지속적으로 변화하는 작업 조건과 환경에 빠르게 적응하는 PHM 기술이 요구되며, 물리 기반의 방법론은 고정된 물리 모델을 통해 예지보전을 수행하기 때문에 협동 로봇의 다양한 환경과 동적인 작업 변화에 효과적으로 대응하기 어렵다는 한계를 지닌다.

2.2 데이터 기반의 PHM 방법론

데이터 기반의 PHM 방법론은 수집된 경험적 데이터를 기반으로 기기의 상태를 예측 및 평가하는 PHM 기술로, 물리적 모델링 과정 없이 데이터를 기반으로 모델링을 수행하기 때문에 복잡한 기기 환경에 대해서도 쉽게 적용이 가능하다. 데이터 기반의 PHM 방법론은 크게 통계 기반의 방법론과 머신 러닝 기반의 방법론으로 나눌 수 있다.

2.1.1 통계 기반의 PHM 방법론

통계 기반 방법은 기기로부터 추출된 데이터를 통계적으로 분석하고, 분석된 결과를 기반으로 모델을 구축하여 건전성을 평가한다. 예시로, [18]은 회전 기계의 상태 모니터링을 위해 추출된 정상 데이터를 주파수 스펙트럼으로 변환하고, 이 스펙트럼을 통해 그래프 모델링을 수행한다. 이후, 새로운 스펙트럼 데이터를 모델의 입력 데이터로 주입하여 회전 기계의 상태를 진단한다. 이와 같은 연구는 적용 가능한 데이터 타입을 제한하지 않아 다양한 환경에 적용이 가능하다. 하지만, 다양하고 복잡한 작업 환경에서 유연성 있게 작업을 수행하는 협동 로봇의 가동 환경에서 수집되는 센서 데이터는 일반적으로 고차원이며 복잡한 패턴의 양상을 보여 통계 기반의 방법론을 적용하는 것은 한계가 존재한다.

2.1.2 머신 러닝 기반의 PHM 방법론

머신 러닝 기반 방법은 기기로부터 추출된 데이터를 학습하여 머신 러닝 모델을 구축하고, 해당 모델을 기반으로 새로운 데이터를 입력 데이터로 주입하여 건전성을 평가하는 방법이다. 머신 러닝 기반의 방법은 신속한 건전성 평가를 기대할 수 있으며, 데이터 특성 및 패턴의 효과적인 추출에 의해 정밀한 건전성 평가가 가능하다. 예시로, [19]는 진동 센서를 활용한 딥러닝 기반의 회전 부품의 잔여 수명(Remaining Useful Life) 예측을 위한 DBN-FNN(Feedforward Neural Network) 알고리즘을 제안한다. 제안 기법은 DBN(Deep Belief Network)를 통해 진동 데이터의 특성을 효과적으로 학습하였으며, 우수한 잔여 수명 예측 성능을 보였다. 또 다른 연구로 [20]은 양방향으로 센서 데이터의 특징 및 패턴을 효과적으로 학습하여 잔여 수명을 예측하는 BiLSTM(Bidirectional Long Short-Term Memory) 네트워

크 기반의 기법을 제안한다. 해당 기법은 기존 잔여 수명 예측을 위해 많이 적용되는 CNN, LSTM, SVR 등에 비해 효과적으로 잔여 수명을 예측하는 결과를 보였다.

그러나, 머신 러닝 기반 PHM 방법론은 기기에서 추출 가능한 센서 데이터가 부족한 경우 부정확한 모델링에 의해 기기의 건전성 평가가 어려울 수 있으며, 데이터 라벨링을 위한 성능 저하의 판단 기준의 설정이 중요한 요소로 작용한다. 또한, 일반적인 협동 로봇의 가동 환경에서는 결함 발생의 빈도가 정상 대비 현저히 적기 때문에 정상 데이터에 비해 매우 적은 수의 결함 데이터가 수집되어 학습 데이터 불균형 문제가 발생하며, 다양한 작업 환경에서 작업을 수행하기 때문에 일관된 성능 저하 기준을 정의하는 것이 매우 어렵다. 따라서, 단일 머신 러닝 모델을 구축하여 협동 로봇의 건전성을 진단하기에는 한계가 존재한다.

이러한 문제를 해결하기 위해, 본 논문은 협동 로봇의 건전성 평가를 위한 요구사항을 체계적으로 분석하여, 요구사항의 구체화 과정에서 요구되는 다관점의 방법론을 종합한 PHM 기법을 제안한다.

3. 협동 로봇의 PHM 요구사항 분석

3.1 PHM 기술 개발을 위한 요구사항 도출

본 절에서는 기존의 PHM 방법론을 기반으로 물리 기반, 통계 기반 그리고 머신 러닝 기반 총 세 가지의 관점으로 PHM 기술의 요구사항을 도출하였다.

3.1.1 물리 기반 PHM 요구사항

물리 기반 PHM 기술 요구사항은 물리 모델 기반의 PHM 기술 개발을 위해 고려되어야 하는 요구사항으로, '견고성(Robustness)', '추적성(Tracking)', '통합성(Integrality)', '비용 효율성(Cost)'의 4가지 측면에서 표 1과 같이 정의된다.

표1. 물리 기반 PHM 요구사항

측면	정의
견고성 (P-Ro)	물리 모델은 변화하는 운용 환경에 대해 안정적으로 진단 가능해야 함
추적성 (P-Tr)	물리 모델은 물리적 구성 요소의 상태를 지속적으로 관찰해야 함
통합성 (P-In)	물리 모델은 추적 대상에 대한 통합적인 분석이 가능해야 함
비용 효율성 (P-Co)	물리 모델은 목표 운용 도달 하에 시간 및 금전적 비용을 최소화해야 함

3.1.2 통계 기반 PHM 요구사항

통계 기반 PHM 기술 요구사항은 통계 모델 기반의 PHM 기술 개발을 위해 고려되어야 하는 요구사항으로, '데이터 운용성(Operation on data)', '보안성(Security)', '유연성(Flexibility)', '식별성(Identification)'의 4가지 측면에서 표2와 같이 정의된다.

표2. 통계 기반 PHM 요구사항

측면	정의
데이터 운용성 (S-Op)	통계 모델은 수집되는 데이터를 처리하고 관리할 수 있어야 함
보안성 (S-Se)	통계 모델은 수집된 데이터를 외부 접근으로부터 보호하고 시스템의 신뢰성을 보장해야 함
유연성 (S-Fl)	통계 모델은 다양한 운용 환경에 유연하게 적용 가능해야 함
식별성 (S-Id)	통계 모델은 특정 이상치에 대해 효과적인 예외 처리가 가능해야 함

3.1.3 머신 러닝 기반 PHM 요구사항

머신 러닝 기반 PHM 기술 요구사항은 머신 러닝 기반의 PHM 기술 개발을 위해 고려되어야 하는 요구사항으로, '정확성(Performance)', '실시간성(Synchronization)', '측정 가능성(Measurability)', '확장성(Scalability)'의 4가지 측면에서 표 3과 같이 정의된다.

표3. 머신 러닝 기반 PHM 요구사항

측면	정의
정확성 (M-Pe)	머신 러닝 모델은 운용 목표에 대한 높은 정확도를 보장할 수 있도록 개발되어야 함
실시간성 (M-Sy)	머신 러닝 모델은 운용 목표를 달성하기 위한 실시간성을 유지해야 함
측정 가능성 (M-Me)	머신 러닝 모델은 목표 운용을 위해 적용된 환경의 변화를 감지하고 측정 가능해야 함
확장성 (M-Sc)	머신 러닝 모델은 적용 도메인의 물리적/기능적 제한 없이 확장 가능해야 함

3.2 건전성 평가를 위한 PHM 기술 개발 요구사항 분석

3.1절에서 도출된 요구사항을 기반으로 협동 로봇의 건전성 평가 관점에서 우선순위로 작용하는 요구사항을 분석하였다. 협동 로봇의 건전성 평가를 위해 우선적으로 고려되어야 하는 세부 모델 요구사항은 다음과 같다.

- **물리 모델 기반 건전성 평가 요구사항 분석 : 견고성, 추적성, 통합성**

  협동 로봇의 건전성 저하는 내•외부 부품의 마모, 결함 등으로 인한 상태 저하를 의미한다. 이를 방지하기 위해서는 지속적인 부품의 상태 관찰과 전반적인 상태 평가를 위한 각 관찰 대상의 통합적 분석이 요구된다. 이에 따라, 물리 모델 기반 건전성 평가에는 '견고성(P-Ro)', '추적성(P-Tr)' 그리고 '통합성(P-In)'이 핵심 요구사항으로 적용된다.

- **통계 모델 기반 건전성 평가 요구사항 분석 : 유연성, 식별성**

  협동 로봇의 유기적 구조에 의해 작업 수행 중 다양한 형태의 데이터가 수집된다. 이와 같은 다양성

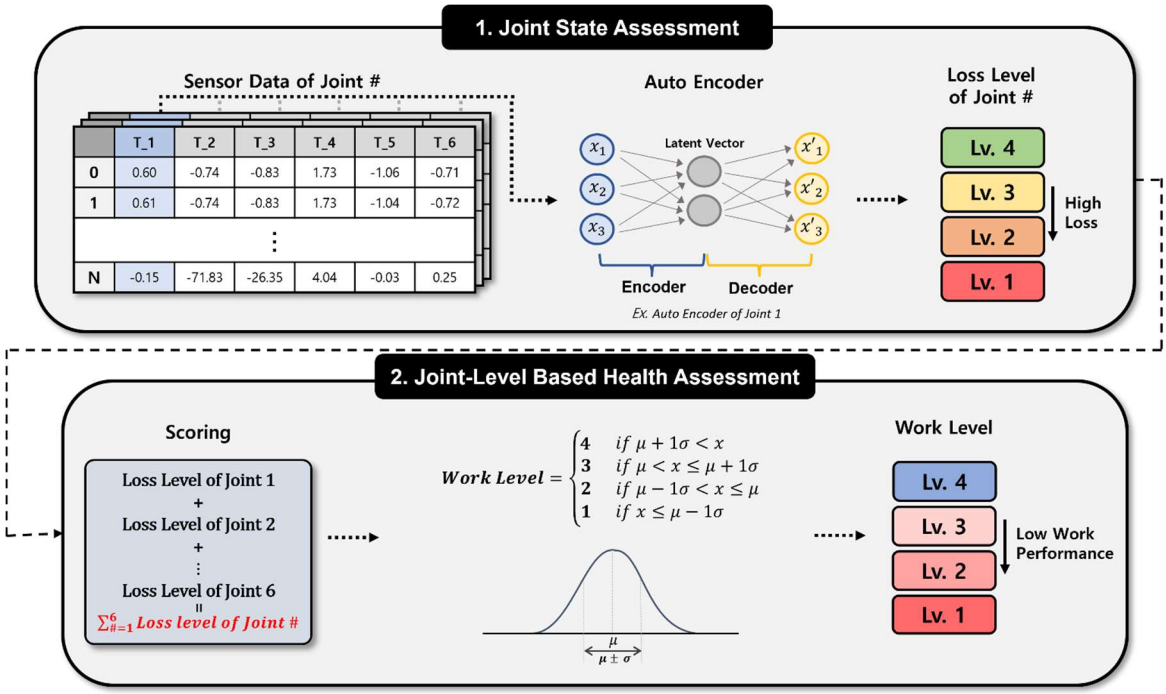


그림 1. 조인트 센서 데이터 기반 협동 로봇의 건전성 평가 기법

을 고려할 때, 통계 모델은 협동 로봇의 상태 평가를 위해 유연하게 적용 가능해야 한다. 특히, 정상적인 작업 상태에서 발생할 수 있는 소수의 이상치에 대해 효과적으로 예외처리 할 수 있어야 한다. 이를 위해, '유연성(S-Fl)'과 '식별성(S-Id)'이 핵심 요구사항으로 적용된다.

- **머신 러닝 모델 기반 건전성 평가 요구사항 분석: 정확성, 실시간성, 측정 가능성**
  협동 로봇의 건전성 저하는 작업 성능 저하 및 잠재적 사고를 야기할 수 있으며, 부정확한 상태 진단은 심각한 비용 손실을 초래할 수 있다. 따라서, 조속하고 정확한 상태 진단이 요구되며, 이를 위해 '정확성(M-Pe)', '실시간성(M-Sy)' 그리고 '측정 가능성'(M-Me)'이 핵심 요구사항으로 적용된다.

4. 조인트 상태 진단 기반 협동 로봇 건전성 평가 기법

본 장에서는 협동 로봇의 건전성 평가를 위한 PHM 기술 개발 요구사항 기반의 건전성 평가 기법을 제안한다. 해당 기법은 협동 로봇의 작업 성능 평가를 기반으로 하며, 각 조인트 단위의 성능 수준을 고려하여 협동 로봇의 작업 상태를 평가한다.

4.1 요구사항 기반 협동 로봇의 건전성 평가 기법

3장에서 정의한 협동 로봇의 건전성 평가 기법의 요구사항을 구체화한다. 제안하는 협동 로봇의 건전성 평가 기법은 단일 방법론이 아닌 물리, 머신 러닝, 통계 기반 방법론의 세부 기법을 종합한 형태로 구현되며, 각 방법론의 요구사항 구체화 내용은 표 4와 같다.

표4. 요구사항 기반 협동 로봇의 건전성 평가 기법

	요구사항	구체화 방법
물리	지속적인 상태 관찰 및 통합 분석	조인트 센서 데이터 수집 + 조인트 레벨 통합
머신 러닝	변화 감지를 통한 높은 정확도의 실시간 상태 진단	오토인코더 기반 조인트 상태 진단
통계	유연한 통계 기준 적용 및 예외처리	조인트 성능 레벨화 + 작업 성능 레벨화 + 건전성 평가

4.2 센서 데이터 기반 조인트 상태 진단 및 평가

협동 로봇의 작업 성능 평가 기반 건전성 평가를 위한 각 조인트의 상태 레벨 분석 방법을 제안한다. 협동 로봇은 오랜 작동 및 과부하로 인해 건전성 저하가 발생하더라도 각 조인트가 목표 동작을 독립적으로 수행하기 때문에, 조인트 별 동작 수행 성능 및 건전성 저하 정도가 상이할 수 있다. 따라서, 각 조인트에 대한 독립적인 분석을 통해 수행 작업의 성능을 세밀하고 정확하게 평가할 수 있으며, 이를 위해 각 조인트에 대한 레벨 분석을 수행한다.

조인트 레벨을 정의하는 단계는 협동 로봇의 작업 성능 기반 건전성 저하 평가를 위한 사전 단계로, 그림 1의 1번 프로세스와 같이 각 조인트의 정상 상태의 작업 동작과 건전성이 저하되었을 때의 동작을 오토인코더 손실 값으로 비교하여 조인트의 레벨을 결정한다. 해당 기법은 각 조인트 단위로 레벨을 정의하기 때문에 사용 로봇의 축 개수와 대응되는 수의 오토인코더를 구축하

며, 각 조인트 별로 레벨 1부터 레벨 4까지 총 4개의 레벨로 정의한다. 레벨 4는 협동 로봇에 건전성 저하가 없는 정상 상태의 협동 로봇의 조인트 상태를 의미하며, 레벨 1은 가장 심각하게 건전성이 저하되었을 때의 상태를 의미한다. 과대 대표와 과소 대표에 의해 특정 레벨에 편향된 레벨 평가가 수행되지 않도록 데이터의 분포를 기반으로 각 레벨에 동일한 데이터의 수가 포함되도록 레벨 구간을 정의하였으며, 그림 2는 조인트 레벨 구간 정의 예시이다. 해당 그림의 데이터 포인트는 레벨 분석을 위해 사용한 각 데이터를 오토인코더 모델에 입력하였을 때 도출되는 손실 값을 의미한다.

- X : 조인트 레벨 구간 정의에 사용되는 데이터셋
- Y: 조인트 레벨 구간 정의를 위한 데이터셋의 오토인코더 모델에 대한 테스트 손실값

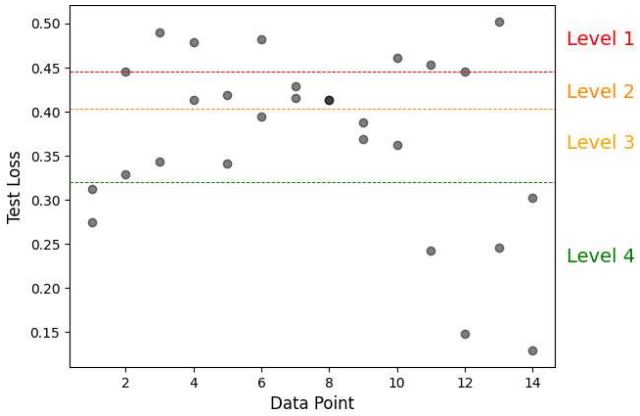


그림 2. 균등 데이터 포인트 기반 레벨 구간 도출

4.3 조인트 상태 레벨 기반 협동 로봇의 건전성 평가

본 절에서는 4.1절에서 정의된 조인트 레벨을 기반으로 협동 로봇의 작업 레벨을 정의하고, 작업 별 수행 가능한 상태 정보를 반영하는 건전성 평가 방법을 제안한다. 해당 기법은 그림 1의 2번 프로세스와 같이, 조인트 레벨을 기반으로 작업 레벨을 단계적으로 정의하며, 각 조인트의 상태 레벨을 합산하는 방식으로 작업 레벨 수치를 계산한다. 해당 과정은 아래의 수식과 같다.

$$Joint\ Integration\ Level = \sum_{\#=1}^{number\ of\ Joint} (Loss\ Level\ of\ Joint\ \#) \quad \text{(수식 1)}$$

앞서 4.1절에서 언급한 것과 같이 레벨 4는 가장 이상적인 동작 수행을 나타내는 레벨이다. 즉, 합산한 값이 큰 수일수록 더 좋은 작업 성능을 나타내며, 이는 협동 로봇이 상대적으로 건전한 상태로 동작하고 있음을 의미한다. 도출된 결과를 기반으로 (수식 2)와 같이 작업 평가 레벨을 정의한다. 평균 조인트 레벨 총합의 평균(μ)을 기준으로 수식을 정의하였으며, μ는 정상 상태의 작업 성능과 건전성 저하 상태의 작업 성능의 일반적인 수준을 대표하는 지표로 사용되었다. 또한, 표준편차(σ)는 작업의 완성도에 대한 변동성을 고려하기 위해 사용하였으며, 시나리오의 조인트 레벨 합계가 평균(μ)을 중심으로 어느 정도 분산되어 있는지 정량화하여 세분화된 작업 성능 레벨을 설정할 수 있도록 수식을 정의하였다. 각 레벨의 수식 정의와 x, μ 그리고 σ의 의미는 다음과 같다.

- x : 각 시나리오에 대한 조인트 레벨의 합계
- μ : 각 프로그램 내 조인트 레벨 구간 정의에 적용되는 시나리오의 조인트 레벨 총합의 평균
- σ : 각 프로그램 내 조인트 레벨 구간 정의에 적용되는 시나리오의 조인트 레벨 총합의 표준편차

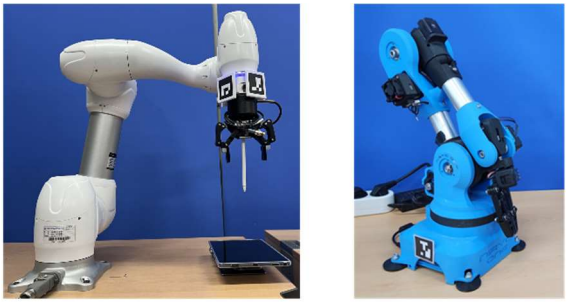
$$work\ Level = \begin{cases} 4 & if \quad \mu + 1\sigma < x \\ 3 & if \quad \mu < x \leq \mu + 1\sigma \\ 2 & if \quad \mu - 1\sigma < x \leq \mu \\ 1 & if \quad x \leq \mu - 1\sigma \end{cases} \quad \text{(수식 2)}$$

5. 실험 설정

본 절은 실험 결과 분석에 앞서, 본 실험을 수행한 실험 환경, 데이터셋 그리고 학습 모델에 대해 설명한다.

5.1 실험 환경

실험에 사용된 협동 로봇은 두산 로보틱스의 M0609 모델로, 6개의 축을 갖는 로봇팔(Robot Arm) 구조를 갖는다. 해당 로봇은 최대 900mm의 작업 반경을 가지며, 최대 하중은 6Kg이다. 본 실험은 해당 로봇의 동작 시나리오 간에 조인트의 내부 센서 데이터를 수집하였다.



(a) Doosan Robotics M0609 (b) Niryo One

그림 3. 실험에 사용된 협동 로봇

5.2 실험 데이터

본 절에서는 실험에서 사용한 데이터셋 구축 방법에 대해 설명한다. M0609 협동 로봇은 산업 환경에 적용되도록 설계되어 내구성이 매우 높아, 직접 하중을 가하여 협동 로봇의 건전성 저하를 야기하는 것은 매우 어렵다. 따라서, 본 논문의 실험은 [21]에서 수집한 실제 건전성 저하 데이터를 활용한다. [21]에서는 그림 3의 (b) Niryo One 협동 로봇에 직접 하중을 가하여 건전성 저하 데이터를 수집하였다. 해당 논문은 협동 로봇의 카메트 생산을 위한 접착 작업의 다양한 시나리오를 구성하였으며, 터치 패널을 이용한 가상의 작업 환경을 구축하였다. 해당 논문은 각 작업 프로그램 당 4개의 Phase로 나누어서 센서 데이터를 수집하였다. Phase 1(w/o load)은 하중을 가하지 않은 정상 상태의 협동 로봇에서 수집한 데이터이며, Phase 2(Normal)는 정상 범위 내의 하

중을 가한 상태의 협동 로봇에서 수집한 데이터이다. Phase 3(Overloaded)는 정상 범위를 초과하는 하중을 가한 상태의 협동 로봇에서 수집한 데이터이며 마지막으로 Phase 4(Degraded due to Overload)는 Phase 3의 과도한 부하로 인해 건전성이 저하된 협동 로봇에 대해 하중을 제거한 조건에서 수집한 데이터이다. 하중이 가해진 상태인 Phase 2와 Phase 3은 협동 로봇의 건전성 저하를 유발시키기 위한 실험 과정으로, 디바이스의 건전성 저하의 영향과 물리적인 하중의 영향이 동시에 작용한다. 협동 로봇의 건전성 저하 분석하기 위한 조건은 Phase 1과 Phase 4로, 동일하게 하중이 부여되지 않은 조건에서 과도한 작업(Phase 2, Phase 3)의 전후 데이터 패턴을 비교하여 건전성 저하의 분석이 가능하다. 이에 대한 예시로 그림 4는 소형 협동 로봇인 Niryo One의 실제 전류 데이터에 대한 통계 기반의 분석을 통해 건전성 저하를 평가한 결과이다. 이는 소형 협동 로봇의 건전성 저하는 통계 기반 방법론으로 충분히 분석 가능함을 보여주며, 본 논문에서는 스마트팩토리의 PHM 실적용 관점에서 기법을 검증하기 위해 크기가 크고 복잡한 구조를 지닌 M0609 협동 로봇의 테스트베드를 활용하였다.

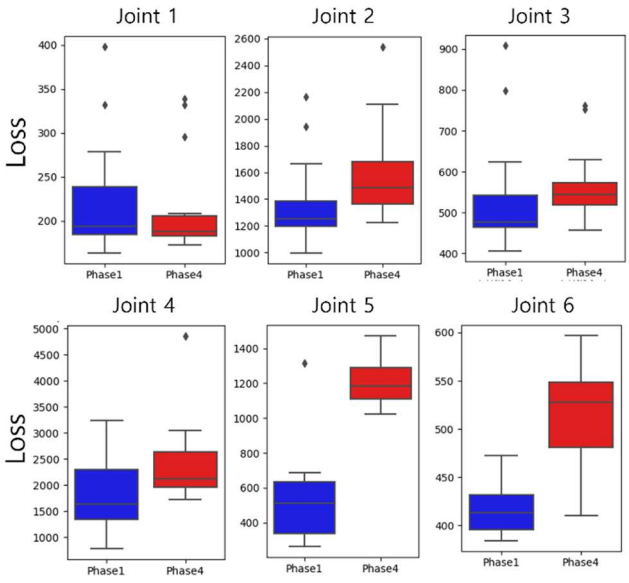


그림 4. Niryo One 데이터 기반 오토인코더 적용 결과

구체적으로는 Niryo One의 작업 간 수집된 Phase 1과 Phase 4 데이터를 추출하고 좌표 스케일을 조정하여 M0609 협동 로봇에 좌표 기반의 정상/건전성 저하 상태의 시나리오를 주입하였으며, 시나리오에 따라 생성되는 토크 데이터를 수집하였다. [21]의 실험 시나리오 중 세 가지의 작업 시나리오를 선택하였으며, 프로그램 1, 프로그램 2 그리고 프로그램 3에 대하여 정상 상태에서 실험한 Phase 1의 시나리오 40개와 건전성이 저하된 상태 Phase 4의 시나리오 20개를 데이터를 수집하였다. 각 작업 시나리오의 형태는 그림 5와 같으며, 각 목적별 시나리오 데이터의 사용 개수는 표 5와 같다.

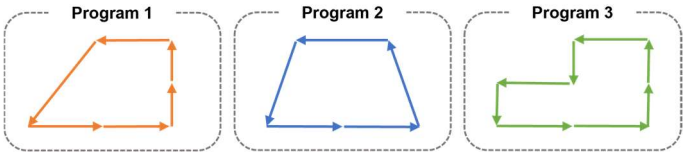


그림 5. 접착 작업 시나리오

표5. 실험 데이터셋 구성

	Learning Auto-Encoder	Level-Wise	Test Case
Phase 1	20	14	5
Phase 4	0	14	5

5.3 학습 모델

비지도 학습 모델인 오토인코더를 사용하여 정상 상태 로봇의 각 조인트 토크 데이터를 학습한다. 실험에서 사용한 오토인코더 모델의 구조는 표 6과 같다.

표6. 실험 모델 세부 구조

Layer	Output Shape	Parameter #
Input Layer	(None, 6000)	0
Dense	(None, 128)	768128
Dense	(None, 64)	8256
Dense	(None, 32)	2080
Dense	(None, 64)	2112
Dense	(None, 128)	8320
Dense	(None, 6000)	774000
Total Parameter		1,562,896

6. 실험 결과 및 분석

본 장에서는 제안하는 기법을 적용하여 각 프로그램의 작업 수행 능력을 평가하고 각 작업에 대한 작업 레벨을 결정하여 건전성을 평가한다. 제안 기법의 적용 결과 및 성능에 대한 일관성을 검증하기 위해 각 프로그램 별 3번의 random sampling을 적용하여 제안 기법의 효용성을 입증한다.

6.1 Euclidean Distance 기반 실험 검증 기준 설정

본 절에서는 Phase 1과 Phase 4 조건의 작업 결과 패턴에 기반한 실험 결과 검증 기준 설정에 대해 설명한다. Phase 1의 그래프는 정상 작업 조건에서의 전체 작업 결과 패턴을 의미하며, Phase 4의 그래프는 건전성이 저하된 조건에서의 전체 작업 결과 패턴을 의미한다. 해당 그래프의 각 축은 M0609 협동 로봇에 이식된 동작의 X와 Y 좌표값이며, 표 7은 각 프로그램의 Phase 별 Euclidean Distance(ED)의 평균과 표준편차를 의미한다. Phase 1 상태에서 특정 시나리오의 ED 합계가 해당 Phase 1의 평균±표준편차 범위 내에 있다면, 이는 해당 시나리오가 높은 작업 완성도를 가지는 것으로 고려할 수 있다. 그러나,

표8. 프로그램 별 조인트 상태 레벨화에 적용된 시나리오의 평가 결과

Program	Sample	μ / σ	Phase #	Level 1	Level 2	Level 3	Level 4	Integration Score	Degradation %
Program 1	Sample 1	15	Phase 1	1	2	7	4	3.0	36.67%
		4.55	Phase 4	3	9	2	0	1.9	
	Sample 2	15	Phase 1	1	2	7	4	3.0	36.67%
		4.57	Phase 4	5	5	4	0	1.9	
	Sample 3	15	Phase 1	3	1	7	3	2.7	22.22%
		3.88	Phase 4	3	7	3	1	2.1	
Program 2	Sample 1	15	Phase 1	3	4	6	1	2.4	16.67%
		3.57	Phase 4	3	9	1	1	2.0	
	Sample 2	15	Phase 1	0	1	7	6	3.4	52.94%
		4.31	Phase 4	6	7	1	0	1.6	
	Sample 3	15	Phase 1	0	1	10	3	3.1	45.16%
		3.96	Phase 4	5	8	1	0	1.7	
Program 3	Sample 1	15	Phase 1	0	2	9	3	3.1	48.39%
		4.56	Phase 4	7	5	2	0	1.6	
	Sample 2	15	Phase 1	0	5	7	2	2.4	20.83%
		4.65	Phase 4	7	1	6	0	1.9	
	Sample 3	15	Phase 1	0	4	5	5	3.1	38.71%
		3.83	Phase 4	3	10	1	0	1.9	

Phase 4에서는 건전성 저하 상태로 인해 작업 동작에 변동성이 크기 때문에 평균과 표준편차를 기준으로 작업의 완성도를 평가하는 것은 어려움이 있어, Phase 1과 비교하여 큰 차이가 발생한 경우를 낮은 작업 완성도를 나타내는 지표로 설정하였다. 작업 완성도 평가를 위해 표 7의 결과에 (수식 3)을 적용함으로써 실험 결과의 검증 기준을 설정하였다.

표7. 프로그램 별 ED 기반 실험 검증 기준

Program	Phase	ED 평균	ED 표준편차
Program 1	Phase 1	1021.34	92.61
	Phase 4	1031.76	67.23
Program 2	Phase 1	985.12	13.08
	Phase 4	1029.15	111.84
Program 3	Phase 1	1087.71	20.89
	Phase 4	1031.23	110.28

$$Average\ of\ ED - Std\ of\ ED \le ED\ of\ the\ Test\ Scenario \le Average\ of\ ED + Std\ of\ ED \quad (수식\ 3)$$

6.2 프로그램 별 조인트 상태 레벨 도출 및 분석

표 8은 각 조인트의 상태 레벨화를 위해 사용된 Phase 1과 Phase 4에 속하는 각 14개의 시나리오에 대한 평가 결과이다. 표의 평균 μ와 표준편차σ는 (수식2)의 변수로 계산된 각 프로그램 내 시나리오 별 조인트 레벨 총합의 평균과 각 프로그램 내 시나리오 별 조인트 레벨 총합의 표준 편차를 의미하며, 해당 값들은 정수 값을 갖는 작업 레벨을 도출하기 위해 반올림하여 적용하였다. Integration Score는 건전성 저하의 통계 분석을 위한 척도로 각 샘플의 Phase 별 작업 레벨의 평균적인 통합 점수를 의미하며, 해당 점수의 감소 비율을 통해 협동 로봇의 건전성 저하 수준을 도출할 수 있다. Integration Score 유도 수식은 (수식 4)와 같다.

$$Integration\ Score = \frac{\sum_{Level=1}^{4} Level * Count\ of\ Level\ Scenarios}{Count\ of\ Phase\#'s\ Scenarios} \quad (수식\ 4)$$

실험 결과, Phase 1 데이터는 주로 작업 레벨 4와 작업 레벨3에 집중해서 분포되어 있고, Phase 4 데이터는 주로 작업 레벨 1과 작업 레벨 2에 집중해서 분포되는 경향을 보였다. 그러나, 각 프로그램의 분포 결과에서 Phase 1이 작업 레벨 1 또는 작업 레벨 2에, 그리고 Phase 4가 작업 레벨 3 또는 4에 분포하는 결과가 일부 관찰된다. 이는 건전성 저하가 항상 낮은 작업 성능으로 귀결되는 것은 아니며, 정상 상태의 협동 로봇의 경우에도 특정 가동 조건에 따라 낮은 작업 성능을 보일 수 있음을 보여준다. 따라서 건전성 저하의 판단 기준은 작업 Phase 내에서 고성능/저성능의 작업 결과가 나타나는 빈도를 고려해야 하며, 제안하는 기법을 통해 통계적으로 이를 반영한 분석 결과를 도출하였다.

표9. 프로그램 별 테스트 시나리오의 평가 결과

Program	Sample	μ / σ	Phase #	Level 1	Level 2	Level 3	Level 4	Integration Score	Degradation %
Program 1	Sample 1	15	Phase 1	0	1	2	2	3.2	37.50%
		4.55	Phase 4	1	3	1	0	2	
	Sample 2	15	Phase 1	0	0	2	3	3.6	27.78%
		4.57	Phase 4	0	2	3	0	2.6	
	Sample 3	15	Phase 1	0	0	4	1	3.2	31.25%
		3.38	Phase 4	1	2	2	0	2.2	
Program 2	Sample 1	15	Phase 1	0	0	4	1	3.2	50.00%
		3.57	Phase 4	2	3	0	0	1.6	
	Sample 2	15	Phase 1	0	0	4	1	3.2	37.50%
		4.31	Phase 4	2	1	2	0	2	
	Sample 3	15	Phase 1	0	0	3	2	3.4	58.82%
		3.96	Phase 4	3	2	0	0	1.4	
Program 3	Sample 1	15	Phase 1	0	1	3	1	2.4	25.00%
		4.56	Phase 4	2	2	1	0	1.8	
	Sample 2	15	Phase 1	0	3	2	0	2.4	58.33%
		4.65	Phase 4	5	0	0	0	1	
	Sample 3	15	Phase 1	0	0	4	1	3.2	43.75%
		3.83	Phase 4	2	2	1	0	1.8	

분석 결과, 프로그램 1의 샘플 1과 2는 Integration Score가 3.0에서 1.9로 각각 36.67%의 건전성 저하가 발생했음을 확인할 수 있으며, 샘플 3의 경우에는 Integration Score가 2.7에서 2.1로 도출된 것을 보아 22.22%의 건전성 저하가 발생하였음을 확인할 수 있다. 프로그램 2, 3의 경우에도 모든 샘플에 대해 건전성이 저하되는 양상을 확인할 수 있었으며, 이는 제안 기법이 다양한 경우의 데이터셋에 대해 효과적인 건전성 평가를 달성할 수 있음을 시사한다.

6.3 테스트 데이터 기반 제안 기법 검증

본 절에서는 테스트 데이터에 대한 적용 결과를 통해 제안 기법의 효용성을 검증한다. 각 프로그램의 샘플 별 건전성 평가 결과는 표 9와 같으며, 모든 프로그램의 전체 샘플에 걸쳐 일관되게 건전성 저하가 발생한 양상을 확인할 수 있다. 각 프로그램의 샘플을 정의한 검증 기준과 비교함으로써, 제안 기법을 통해 도출된 레벨 수치의 타당성을 확인하였다.

프로그램 1의 Phase 1 분석 결과, ED의 평균은 1021.32, 표준편차는 92.61로 도출되었다. 이는 Phase 4의 표준편차인 67.61과 비교하였을 때, 매우 큰 변동성을 갖는 작업임을 의미한다. 프로그램 1의 높은 작업 완성도의 범위는 (수식 3)에 기반하여 928.73에서 1113.95로 설정되며, 해당 범위 내에서 총 10개의 테스트 데이터 중 샘플 1은 5개, 샘플 2와 3은 각각 8개, 7개의 데이터가 검증 기준에 부합하였다. 이는 프로그램 1이 평균적으로 66.67%의 정확도로 작업 상태를 평가할 수 있음을 의미한다. 프로그램 2의 Phase 1 분석 결과, ED의 평균은 985.12, 표준편차는 13.08로 프로그램 1과 달리 일관된 작업이 수행되었으며, 프로그램 2의 높은 작업 완성도 범위는 972.04에서 998.2로 설정되었다. 총 10개의 테스트 데이터에 대해 샘플 1은 10개, 샘플 2와 3은 각각 7개, 10의 데이터가 검증 기준에 부합하였으며, 평균 90.00%의 높은 정확도로 작업 상태 평가가 수행되었다. 마지막으로, 프로그램 3의 Phase 1 분석 결과, ED의 평균은 1087.71, 표준 편차는 20.89로 프로그램 2와 같이 일관된 작업이 수행되었음을 확인할 수 있다. 프로그램 3의 높은 작업 완성도 범위는 1066.82에서 1108.6으로 샘플 1에 대해 총 7개의 샘플이 검증 기준에 부합했으며, 샘플 2와 3에 대해서는 각각 7개, 4개로 평균 60.00%의 정확도를 보였다.

제안 기법의 정확도가 낮은 작업에 대한 분석 결과, 그림 6의 프로그램 3의 Phase 1 첫 번째 시나리오와 같이 특정 시나리오에서 검증 기준의 범위 경계와 근접한 ED 값을 갖는 테스트 시나리오의 경우, 작업 결과 패턴은 높은 작업 성능을 보이는 패턴 양상과 비슷하며, 검증 기준의 경계 범위에 의해 기준에 부합하지 못하였음을 확인할 수 있다. 또한, 프로그램 1과 같이 변동성이 큰 Phase 1 작업의 경우 크게 설정되는 기준 경계 범위에 의해 Phase 별 명확한 검증이 어려우며, 이는 ED의 평균과 표준편차에 기반한 검증 기준 설정이 특정 시나리오들을 검증하는 데에 적합하지 않을 수 있음을 시사한다. 따라서, 표준편차의 비율을 통한 마진을 부여하는

등의 추가 검증 방식이 요구된다.

하지만, 평균적인 실험 결과는 평균 72.22%의 정확도를 보였으며, 분류 문제가 아닌 수치 기반의 레벨 평가 방식임을 고려했을 때 제안 기법이 협동 로봇의 작업 성능을 평가하는 데에 효용성을 지님을 보여준다. 또한, 표 9의 각 프로그램 별 실험 결과는 Phase 1에 비해 Phase 4 상태에서 수집된 데이터가 공통적으로 낮은 작업 레벨을 갖는 경향성을 보였다. 이를 Integration Score로 분석한 결과, 모든 샘플 케이스에서 협동 로봇의 건전성 저하를 확인하였으며, 본 기법이 건전성 저하를 탐지하는 데에 효과적임을 확인하였다.

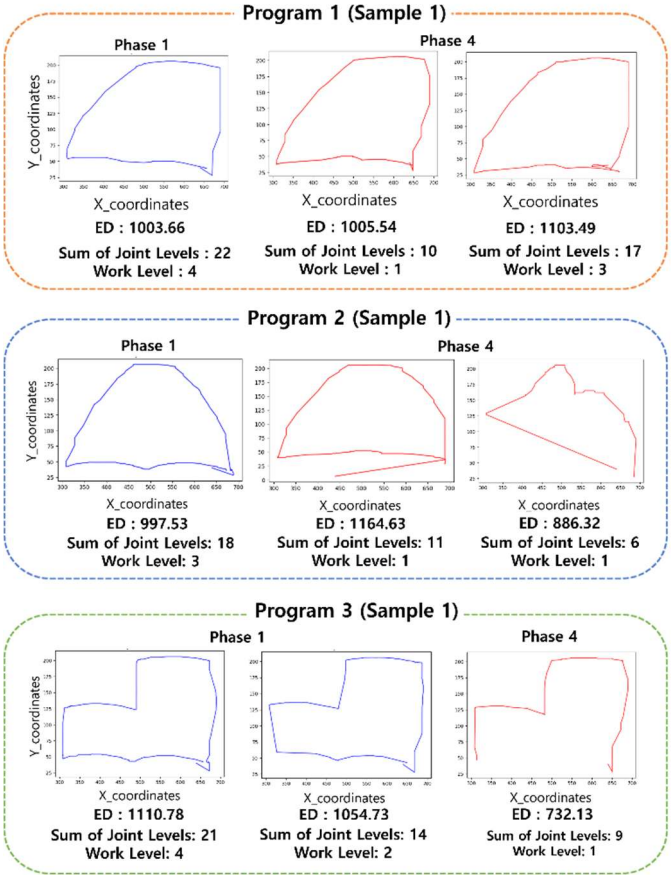


그림 6. 일부 테스트 시나리오의 세부 평가 결과

6.4 위협 요소 분석

본 논문에서는 협동 로봇의 작업 성능 및 건전성 저하 평가를 위해 각 작업 시나리오의 작업 레벨을 분석하고 이를 기반으로 건전성 저하 비율을 도출하였다. 하지만, 제안하는 기법과 진행한 실험 기반의 평가 과정에서 다음과 같은 잠재적 위협 요소가 존재할 수 있다.

6.4.1 데이터셋의 복잡도 및 크기 증가에 따른 기법의 적용 가능성

본 논문에서는 제안하는 협동 로봇의 작업 성능 및 건전성 저하 평가를 위해 각 프로그램 별 Phase 1 상태에서 수집된 39개의 토크 데이터와 Phase 4 상태에서 수집된 19개의 토크 데이터를 활용하여 실험을 진행하였다. 해당 데이터들은 6000 ticks로 구성되었으며, 각 프로그램 별 random sampling을 통해 협동 로봇의 작업 성능 및 건전성 저하에 대한 평가를 효과적으로 수행할 수 있음을 확인함으로써 실제 협동 로봇에 적용함에 있어서 일반화 될 수 있는 가능성을 보여준다. 그러나, 데이터셋의 복잡성과 크기가 증가함에 따라, 제안된 기법의 적용 가능성에 대해 추가적인 평가가 요구된다.

6.4.2 작업 성능 평가에 대한 세부적인 접근의 필요성

본 논문에서는 특정 작업에 대한 협동 로봇의 조인트 관여도를 동일하게 분석하여 작업 성능을 평가하였다. 하지만, 기법을 적용하는 협동 로봇의 물리적인 구조에 따라 평가 결과가 다르게 도출될 수 있다. 따라서, 협동 로봇의 물리적인 구조를 반영한 수식의 모델링 또는 작업의 세부 모션 단위의 조인트의 관여도를 기반으로 하는 기법으로의 확장 및 검증이 요구된다.

6.4.3 실제 환경에 대한 적용 가능성

본 논문에서는 Phase 1과 Phase 4 상태에서 수집된 작업 데이터에 레벨을 부여하여 협동 로봇의 건전성 저하를 확인하였다. 하지만, 실제 작업 환경에서는 본 연구에서와 같이 명확한 Phase 구분을 통한 Integration Score 분석이 직접적으로 적용되지 않을 여지가 있다. 본 논문과 같이 데이터 수집 주기를 구분하여 Integration Score를 분석한다면 특정 수집 주기에서 발생한 건전성 저하 양상을 감지할 수 있지만, 효과적인 기법의 적용을 위해서는 적절한 데이터 수집 및 샘플링 주기에 대한 설정이 요구된다.

7. 결론 및 향후 연구

본 연구는 협동 로봇의 건전성 저하 평가를 위한 요구사항을 체계적으로 분석하고 구체화하는 과정을 통해 조인트 센서 데이터 기반의 작업 성능 평가 기법을 제안한다. 해당 기법은 조인트의 물리적 특성을 고려하여 신뢰성 높은 건전성 평가를 가능하게 하며, 각 작업에 적용하여 모듈화함으로써 협동 로봇의 전반적인 건전성 평가에 기여할 수 있다. 제안 기법의 효용성 검증을 위해 협동 로봇의 접착 작업 간 부하에 의해 발생한 건전성 저하를 토크 데이터 기반으로 평가하였으며, 최소 25.00%에서 최대 58.82%의 건전성 저하 추이를 약 72.22%의 정확도로 확인하였다. 이는 제안 기법을 통해 실제 환경에서 협동 로봇의 세부적인 작업 성능 저하를 식별하고 평가할 수 있으며, 평가를 위한 효과적인 도구로서 적용될 수 있음을 보여준다. 향후 연구에서는 작업의 세부적인 모션 단위의 조인트의 관여도를 분석하여 건전성 평가의 정밀도를 더욱 향상시킴으로써 보다 효과적이고 실용적인 평가 기법으로 확장할 예정이다.

참고 문헌

[1] Ying Hong, Zhenzhong Sun, Xiaohong Zou, Jianyu Long, “Multi-joint Industrial Robot Fault Identification using Deep Sparse Auto-Encoder Network with Attitude Data”, 2020 Prognostics and Health Management Conference (PHM-

Besançon), pp.176-179, 2020
[2] Carole S. Franklin a, Elena G. Dominguez b, Jeff D. Fryman c, Mark L. Lewandowski d, “Collaborative robotics: New era of human-robot cooperation in the workplace, Journal of Safety Research”, vol.74, pp.153-160, 2020
[3] Bhanoday Reddy Vemula, Marcus Ramteen, Giacomo Spampinato, Björn Fagerström, “Human-robot impact model: For safety assessment of collaborative robot design”, 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), pp.236-242, 2017
[4] Shirine El Zaatari, Mohamed Marei, Weidong Li, Zahid Usman, “Cobot programming for collaborative industrial tasks: An overview”, Robotics and Autonomous Systems, vol.166, pp.162-180, 2019
[5] Casper Schou, Rasmus Skovgaard Andersen, Dimitrios Chrysostomou, Simon Bøgh, Ole Madsen, “Skill-based instruction of collaborative robots in industrial settings”, Robotics and Computer-Integrated Manufacturing, vol, 53, pp. 72-80, 2018
[6] Xingyu Yang, Zhengxue Zhou, Leihui Li, Xuping Zhang, “Collaborative robot dynamics with physical human-robot interaction and parameter identification with PINN”, Mechanism and Machine Theory, vol. 89, pp.1-17, 2023
[7] Adalberto Polenghi, Laura Cattaneo, Marco Macchi, “A framework for fault detection and diagnostics of articulated collaborative robots based on hybrid series modelling of Artificial Intelligence algorithms”, Journal of Intelligent Manufacturing, pp.1-19, 2023
[8] Andrea Raviola , Roberto Guida, Antonio Carlo Bertolino , Andrea De Martin, Stefano Mauro, Massimo Sorli, “A Comprehensive Multibody Model of a Collaborative Robot to Support Model-Based Health Management”, Robotics, vol. 12, no.3, pp.1-22, 2023
[9] Guixiu Qiao, Craig Schlenoff, Brian A. Weiss, “Quick positional health assessment for industrial robot prognostics and health management (PHM)”, 2017 IEEE International Conference on Robotics and Automation (ICRA), pp.1815-1820, 2017
[10] Qiaoqian Zhou, Yuanchao Wang, Jianming Xu, “A Summary of Health Prognostics Methods for Industrial Robots”, 2019 Prognostics and System Health Management Conference (PHM-Qingdao), pp.1-6, 2019
[11] John W. Sheppard, Mark A. Kaufman, Timothy J. Wilmering, “IEEE standards for prognostics and health management”, 2008 IEEE AUTOTESTCON, pp. 97-103, 2008
[12] Enrico Zio, “Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice”, Reliability Engineering & System Safety, vol.218, pp. 1-16, 2022
[13] Seokgoo Kim; Nam Ho Kim; Joo-Ho Choi, “A Study Toward Appropriate Architecture of System-Level Prognostics: Physics-Based and Data-Driven Approaches”, IEEE Access, vol.9, pp.157960-157972, 2021
[14] Moussa Hamadache, Joon Ha Jung, Jungho Park, Byeng D. Youn, “A comprehensive review of artificial intelligence-based approaches for rolling element bearing PHM: shallow and deep learning”, JMST Advances, vol.1, pp.125-151, 2019
[15] Carlos Ferreira, Gil Gonçalves, “Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods”, Journal of Manufacturing Systems, vol. 63, pp. 550-562, 2022
[16] Houman Hanachi, Jie Liu, Avisekh Banerjee, Ying Chen, Ashok Koul, “A Physics-Based Modeling Approach for Performance Monitoring in Gas Turbine Engines”, IEEE Transactions on Reliability, vol. 64, no.1, pp.197-205, 2015
[17] Mehdi Behzad, Hesam Addin Arghan, Abbas Rohani Bastami, Ming J. Zuo, “Prognostics of rolling element bearings with the combination of paris law and reliability method”, 2017 Prognostics and System Health Management Conference (PHM-Harbin), pp.1-6, 2017
[18] Teng Wang, Guoliang Lu, Peng Yan, “A Novel Statistical Time-Frequency Analysis for Rotating Machine Condition Monitoring”, IEEE Transactions on Industrial Electronics, vol. 67, no.1, pp.531-541, 2020
[19] Jason Deutsch, David He, “Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components”, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no.1, pp.11-20, 2018
[20] Jiujian Wang, Guilin Wen, Shaopu Yang, Yongqiang Liu, “Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network”, 2018 Prognostics and System Health Management Conference (PHM-Chongqing), pp.1037-1042, 2018
[21] Ye-Seul Park, Dong-Yeon Yoo, Jung-Won Lee, “Programmable Motion-Fault Detection for a Collaborative Robot”, IEEE Access, vol.9, pp.133123-133142, 2021

소프트웨어 결함 예측을 위한 XAI 기반 특징 공학

아비섹 차우다리, 류덕산, 최선오

전북대학교

{abhishek.chaudhary, duksan.ryu, suno7}@jbnu.ac.kr

XAI-based Feature Engineering for Software Defect Prediction

Abhishek Chaudhary, Duksan Ryu, Sunoh Choi

Jeonbuk National University

Abstract

Software Defect Prediction (SDP) is an essential part of software development which is an inevitable factor. Any defect in software can cause harm(s) hence, it becomes essential to remove such defects within a minimal time. Machine Learning (ML) models have made defect prediction practice much easier, but those models do not explain the reason behind the prediction which makes it difficult to focus on the direction of optimization. For that, eXplainable Artificial Intelligence (XAI) is an emerging tool for developers which helps in visualization, giving insight into the reason for such prediction. In this research, XAI is used for model optimization in SDP. The objective is to use XAI to explain traditional ML models (Logistic Regression (LR) and Decision Tree Classifiers (DT)) and understand the feature importance through SHapely Additive eXplanations (SHAP) value visualization. After initial prediction, through the SHAP method, the reasons behind the predictions were visualized in the form of feature contribution values to analyze their importance in prediction. After analyzing the feature's importance, model optimization was done to increase its performance or efficiency by selecting high-contributing features. In this research, to understand the feature importance, the highest and lowest contributing features were removed to compare the change in outcomes. The result shows that removing the highest contributing feature has a negative impact and vice versa on model performances which were compared by AUC and G-mean performance metrics. Hence, by using SHAP, the optimization of the model's performance is possible by understanding feature importance and applying feature engineering techniques.

1. Introduction

In the contemporary landscape of software development, the continual evolution of technology has facilitated the creation of intricate and sophisticated software systems. As the complexity of these systems grows, so does the challenge of ensuring their reliability and functionality [1]. A crucial aspect of software quality assurance is the early identification and remediation of defects, which can significantly impact the performance and user experience of applications. In this context, the utilization of machine learning models for software defect prediction has become increasingly prevalent [2]. Traditional machine learning models, including logistic regression and decision tree classifiers, have proven effective in predicting software defects based on historical data. However, their complex structures often lead to "black box" models, where the internal workings are challenging to interpret [3]. This lack of transparency poses a significant obstacle, especially in safety-critical domains where understanding the rationale behind a model's predictions is paramount. The advent of XAI seeks to address this challenge by providing mechanisms to interpret and explain the decisions made by complex machine learning models [4]. Among various XAI techniques, the SHAP method stands out for its ability to attribute a value to each feature's contribution to a particular prediction. This research embarks on an exploration of how SHAP can be integrated into the feature engineering process of logistic regression and decision tree classifier models for software defect prediction [5].

A key motivation for employing SHAP in this context is to bridge the interpretability gap that often exists between machine learning models and human understanding [6]. By dissecting the impact of each feature on model predictions, SHAP empowers software developers, project managers, and quality assurance professionals to make informed decisions based on a deeper comprehension of the model's reasoning [7]. Logistic Regression and Decision Tree classifiers models were used to predict software defects and the results obtained were quite satisfiable. Still, these models could be optimized after understanding the reason behind such predictions. The

SHAP method was used to visualize the prediction results through feature-contributing values. A deep understanding of the feature contributing was helpful for feature engineering. The contribution varies, which means there were high and low contributing features for the prediction. For analysis purposes, the highest and lowest contributing features were removed to find the impact on prediction. After removing the highest contributing features, the performance of the models was decreased and vice versa. Through such results, we can summarize that a proper understanding of the features for feature engineering is essential which can leverage the performance of the model.

This research aims not only to improve the predictive performance of machine learning models for software defect prediction but also to contribute valuable insights into the application of XAI in the software engineering domain. Through a detailed analysis of SHAP's interpretability enhancements, this study endeavors to provide actionable knowledge for practitioners while advancing the broader adoption of transparent and explainable machine learning models in software defect prediction. In summary, the integration of SHAP with LR and DT models for software defect prediction represents a significant step towards creating more transparent and trustworthy machine learning models in the realm of software engineering [8]. This research contributes to the ongoing discourse on XAI applications, with the potential to influence best practices in software defect prediction and beyond.

There were many researchers conducted to analyze the feature's contribution values but none of them focused on model optimization. So, we proposed a novel approach to use XAI in SDP.

2. Background and Motivation

XAI is an emerging term among developers and stakeholders [9]. It is used to describe an AI model which means, its impact and potential biases. It also makes the model transparent making it easier to understand the reason behind the prediction and importance of the features. Moreover, it also helps to characterize model accuracy, fairness, and outcome in prediction. Most organizations are evolving and adopting data-driven technologies which require AI. Such organizations must make sure that AI in production is working properly, building trust and confidence. In such cases, XAI helps them to choose the AI model based on their business needs.

In many cases, XAI has been applied to have transparent production and development but only in a few cases they are adopted to predict the defect. In cases of software, it is crucial to identify the defects or bugs in the early stage or predict them to mitigate their impact [10]. XAI has proven its promising effects in giving insight into the reason behind prediction, so it becomes a valuable tool to use in defect prediction.

During the defect prediction, XAI can help stakeholders or developers understand the behavior of the model and its features [11]. The proper analysis of such behavior can give deep insights and can be essential in scaling. This can be done through visual investigation models with an interactive chart which generates feature attributes [12]. Such charts can display positive and negative values of the model.

The motivation comes from the SHAP methods which helps to visualize the feature's contribution values for the prediction. Since every feature does not play a significant role in prediction. It becomes essential to choose only those features which will be beneficial to predict the defects. Such selection of features helps to increase computational efficiency, minimize training time, and make it easier for model interpretability and support in model optimization. For this research, the main objective is to focus on model optimization through appropriate selection of features with the help of SHAP.

Then applying proper technique can help to optimize the ML model's performance with higher efficiency and mitigate the problems in earlier stages of the products in upcoming days.

3. Related Work

Begum et. al. [13] used real datasets of National Aeronautics and Space Administration. For software fault diagnosis, they used thirteen Machine Learning models, (i.e., Random Forest Regression, Linear Regression, Naïve Bayes, Decision Tree Classifier, Logistic Regression, KNeighbors Classifier, AdaBoost, Gradient Boosting Classifier, Gradient Boosting Regression, XGBR Regressor, XGBoost Classifier, Extra Trees Classifier and Support Vectors Machine). Then they compared each model to find the best model. As per the result they found that XGBR outperformed in case of AUC, Mean Square Error (MSE) and R2 score. They also used XAI (i.e., Local Interpretable Model (LIME), and SHapley Additive exPlanations (SHAP)) to determine software fault features. The result showed that average true positive by LIME were greater than that of SHAP, which indicated LIME can afford the greatest

impact on the model outcomes to identify features that are the most significant reasons for software defects. This research concluded the method to identify features which could be significant for software defects, but our research focuses on selection of appropriate features removing low contributing features to increase the model performance.

Gezici et. al. [14] focused on the explainability of Gradient Boosting (GB) classifier used for software defect prediction (SDP). They used post-hoc model-agnostic methods called "Explain Like I am a 5-year-old" (ELI5), on the dataset provided by NASA. Specifically, they used SHAP, ELI5 and LIME to explain local instances, and SHAP to explain both local and global explanation. The results suggest a post-hoc and model-agnostic way to quantify Explainability and indicate that all three methods used in this study performed consistent results with each other while explaining the GB model. They focus on selection of XAI technique to explain the local and global explanation. In contrast to their research, we not only focus on explanation but also analyzing the feature importance and understanding the feature importance and their contributions.

Shin et. al. [15] to identify the applicability of XAI and its trustworthiness used LIME and BreakDown which are two state-of-the-art model-agnostic technique to explain the prediction results of bug prediction models. The experiments show these tools can generate promising results and the generated explanations can assist developers understand the prediction results. They investigate the consistency and reliability of model-agnostic technique-based explanation generation approaches. The results show that both LIME and BreakDown generate inconsistent explanations under different software defect prediction settings for the same test instances, which makes them unreliable for explanation generation. Thus, more research in explainable software defect prediction towards achieving consistent and reliable explanation generation is necessary. They found the inconsistent explanation during the explanation but in our case, we found inconsistent explanation after the removal of certain features where the contribution behavior and values of features has changed, explained more in RQ2.

Chmielowski et. al. [16] evaluated the use of explainable artificial intelligence (XAI) in processes related to software development and bug classification based on bug reports created by either software testers or software users. The research was conducted on two different datasets and the task was to classify issues based on crash, memory, performance, and security. Studies on XAI-related algorithms show that there are no major differences in the results of the algorithms used when comparing them with others. XAI does not provide degradation of accuracy so users can easily obtain results before experts verify and can be applied to production easily. The focus of this research is to classify the software bugs, but our focus is to understand the features and optimize the model.

There were several research works and applications of XAI in SDP but none of them have applied it find the notable features to optimize the model after analyzing the reason behind the prediction using XAI.

4. Proposed Methods

This research is focused on the SHAP method for the visualization of each feature's contribution for the prediction. The procedures are:

1. Selection of database
2. Define the ML prediction models.
    a. Logistic Regression
    b. Decision Tree Classifier
3. Initialize the SHAP explainers.
    a. Linear Explainer
    b. Tree Explainer
4. Compute the SHAP values for each feature.
    a. SHAP Input: prediction result
    b. SHAP Output: feature contribution values
5. Summarize and visualize SHAP values for interpretation.
6. Apply Feature Engineering technique.
7. Compare the values with previous results.

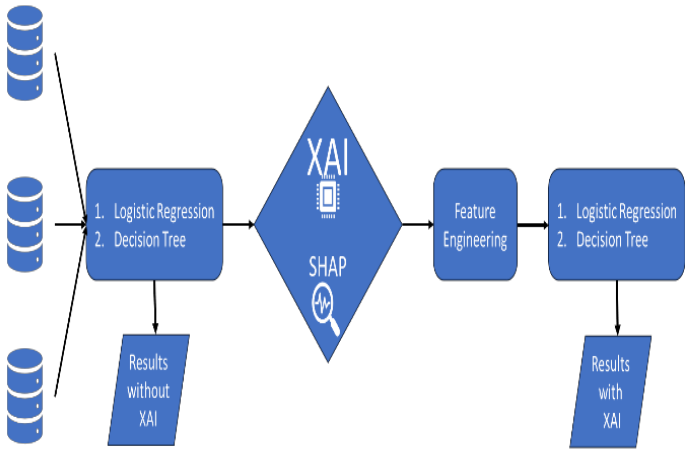


FIGURE 1: PROPOSED APPROACH

The first step is to select the dataset. The AUDI automotive dataset was chosen for research with three different software

projects. The use of different projects might make the outcome more reliable and support deep analysis of the model's behavior based on the projects used. Then, LR and DT classification models were applied to get initial results on each dataset. Next, the SHAP technique was applied to each prediction to understand the reason behind the prediction. Linear explainer and Tree explainer were used to visualize the feature contributions. For deep analysis of the feature importance and their contributions, the highest and lowest contributing features were dropped, also to find their impact on prediction. Then again, LR and DT methods were applied. The results obtained before and after using SHAP techniques were compared to find its effect on model optimization.

5. Experimental Setup

5.1 Dataset

Table 1: AUDI Dataset

SN	Projects	# of metrics	# of instances	# of buggy instances
1.	PROJECT_A	10	1908	77 (4.03%)
2.	PROJECT_K	10	2515	112 (4.45%)
3.	PROJECT_L	10	2891	61 (2.10%)

In Table 1, the details of the three different projects (PROJECT_A, PROJECT_K, PROJECT_L) selected for research purposes are presented. These datasets are from the domain of Automotive and is available for download at [17]. The reason for the selection of three different projects is to perform more reliable research and find how the models perform under automotive industry.

The features of all three projects were the same, which are listed in Table 2.

Table 2: Feature description

SN	Features	Description	Reference
1	author	Author name	[18]
2	sloc	Executable code	[18]
3	McCab	McCabe Cyclomatic	[19]
4	Hv	Halsted Volume	[18]
5	Hd	Halsted Difficulty	[18]
6	He	Halsted Effort	[18]
7	loc_add	Line of Code added	[18]
8	loc_remove	Line of Code removed	[18]
9	nfunction	Number of functions	[18]
10	bug	Change metric	[18]

5.2 Machine Learning Models

5.2.1. Logistic Regression (LR)

LR is the data analysis technique used to find the relationship between data factors. The relationship found is then used to predict the value of any one factor based on the other factors. LR is an important technique in the field of artificial intelligence and machine learning (AI/ML). LR is used because of its some key features like simplicity, speed, flexibility, and visibility [20].

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}}$$

The sigmoid function is referred to as an activation function for logistic regression and is defined as:

e = base of natural logarithms, x = input value, b0 = bias or intercept term and b1 = coefficient for input x

5.2.2. Decision Tree Classifier (DT)

Decision Tree is an ML algorithm that can be used for both regression and classification tasks. DT is easy to understand, interpret, and implement, making it an ideal choice for beginners in the field of machine learning. DT is a non-parametric supervised learning algorithm. It is also a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility [21].

5.3 XAI Techniques

5.3.1. SHapely Additive exPlanations (SHAP)

SHAP is a XAI technique which is model agnostic that helps to explain the output of any ML models. It uses a game theoretic approach that measures each features contributions to find prediction or outcomes. SHAP helps to visualize the contribution values known as SHAP values of each feature in final prediction. Such visualization helps to understand the feature impact on the model's prediction [22].

There are different SHAP explainers to explain different models. For this research, Tree and Linear Explainer were selected to explain DT and LR respectively in terms of feature contribution.

5.4 Evaluation Metric

5.4.1. Area Under Curve (AUC)

AUC evaluates the ability of a classification model to discriminate between positive and negative instances across various classification thresholds. A higher AUC indicates better discrimination ability of the model. The maximum value

is 1 which means, the closer to the value 1, the better the model [23].

5.4.2. G-Mean

G-Mean is the performance metric which is calculated based on the confusion metrics which includes values such as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) [24].

The following formula is used to calculate it,

$$G - Mean = \sqrt{\frac{TP}{TP + FN} \; x \; \frac{TN}{TN + FP}}$$

5.5 Research Questions

Since every feature has its own importance, they contribute differently to different predictions. It becomes essential to understand the features before using them for a specific task. Through these Research questions, we want to find out which features have the highest and lowest contribution for the defect prediction purpose and select appropriate features for SDP. The research was supported and conducted based on the following research questions.

RQ1. What is the effect on performance after removal of Highest and Lowest contributing features?

From RQ1, we want to find the effect on the prediction results after removing the highest and lowest contributing features. This is because an enormous number of low contributing features might increase training time and decrease model performance.

RQ2. What is the effect on other features after removal of high and low contributing features?

From RQ2, we want to find the behavior of other existing features like change in their contribution values and change in efficiency of the model.

RQ3. What things should be considered while applying XAI?

From RQ2, we can analyze the importance of certain features. This will help to understand and focus on those features who have contributed which will mark them important and prevent them from elimination. Elimination of such features may cause negative impact on prediction.

6. Experimental Results

The purpose of this experiment is to find whether XAI can be used for model optimization or not. Hence, at first the LR and DT models were applied. The purpose of those models is to find the defects in terms of AUC and G-mean as performance metrics. The obtained performances are listed in the following Table 3.

Table 3: AUC and G-Mean obtained after applying LR Model

SN	Dataset	AUC	G-mean
1.	PROJECT_A	0.421	0.693
2.	PROJECT_K	0.366	0.798
3.	PROJECT_L	0.545	0.312

After the LR model, SHAP approach was applied to find the contribution values of each feature.

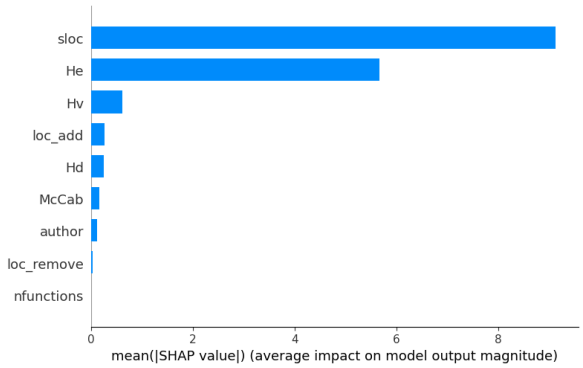


FIGURE 2. THE FEATURE CONTRIBUTION VALUES FROM THE HIGHEST TO THE LOWEST OF PROJECT_A.

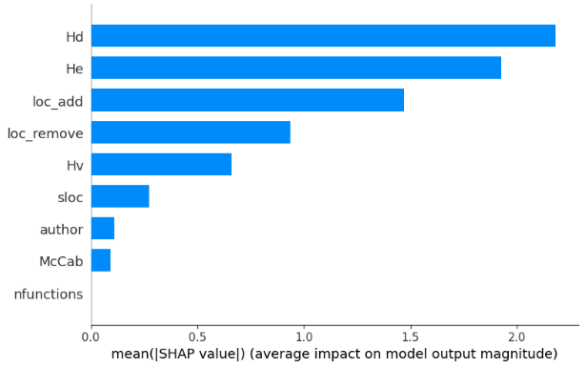


FIGURE 3. THE FEATURE CONTRIBUTION VALUES OF EACH FEATURES FROM HIGHEST TO LOWEST OF PROJECT_K

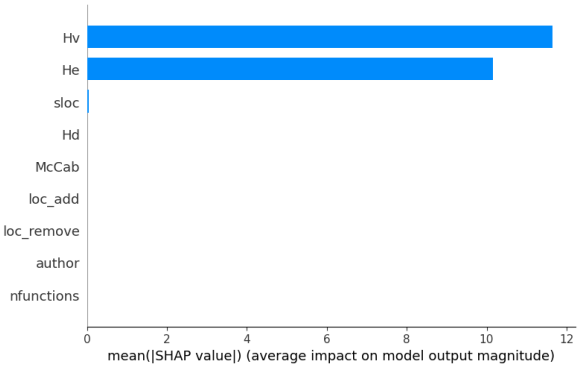


FIGURE 4. THE FEATURE CONTRIBUTION VALUES OF EACH FEATURES FROM HIGHEST TO LOWEST OF PROJECT_L

The visualization of each feature through the SHAP values gave insight into the contribution of each feature. Such visualization and understanding of the contribution are very essential for feature engineering which lead us toward the Research Question 1.

RQ1. What is the effect on performance after removal of Highest and Lowest contributing features?

1. Logistic Regression

Figures 2, 3 and 4, show the features contributing from highest to lowest. The highest and lowest contributing features were removed to see the effect on performance.

Initially, the highest contributing features were selected from PROJECT_A (i.e., sloc, He), PROJECT_K (i.e., Hd, He) and PROJECT_L (i.e., Hv, He) which were analyzed from Figure 2, 3, and 4, respectively. Those features were removed to find an effect on the prediction of each project.

After selecting and removing those highest contributing features, the LR model was again applied. The results obtained are in the following Table 4.

Table 4: Results obtained after removing Highest contributing features.

Dataset	AUC		G-Mean	
	Before	After	Before	After
PROJECT_A	**0.421**	0.381	**0.693**	0.620
PROJECT_K	**0.366**	0.341	**0.798**	0.597
PROJECT_L	0.545	**0.645**	0.312	0.312

The result obtained shown in Table 4, in the case of PROJECT_A and PROJECT_K shows that the performance of both AUC and G-mean has decreased after dropping highest contributing values. Meanwhile, in PROJECT_L the performance of AUC has increased but the G-mean remained unchanged.

Similarly, the lowest contributing features were also selected after analyzing the highest feature contribution removed chart. The lowest contributing features were selected from PROJECT_A (i.e., loc_add, nfunctions), PROJECT_K (i.e., loc_add, nfunctions) and PROJECT_L (i.e., loc_add, loc_remove) which were analyzed form Figure 5, 6 and 7, respectively.

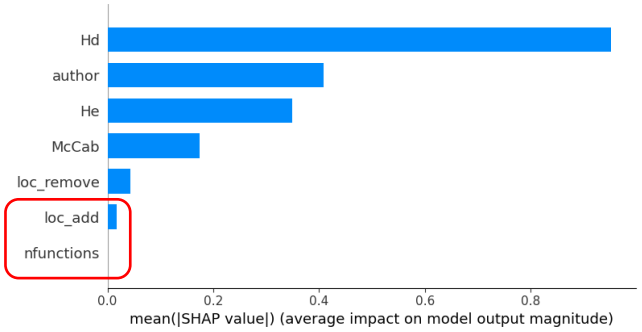


FIGURE 5. ALL CONTRIBUTING FEATURES AFTER HIGHEST FEATURES WERE DROPPED FROM PROJECT_A

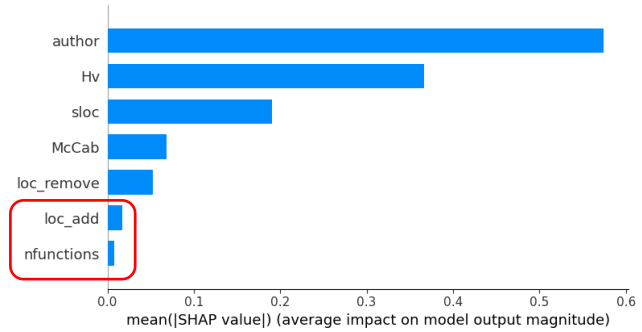


FIGURE 6. ALL CONTRIBUTING FEATURES AFTER HIGHEST FEATURES WERE DROPPED PROJECT_K

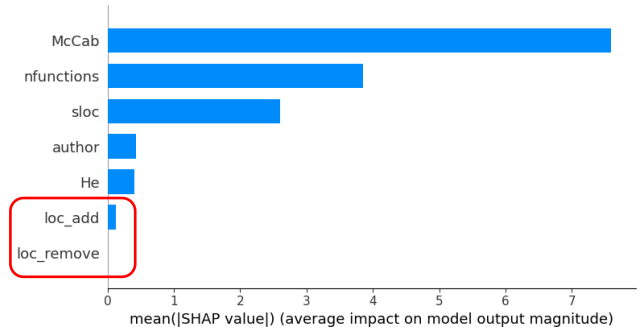


FIGURE 7. ALL CONTRIBUTING FEATURES AFTER HIGHEST FEATURES WERE DROPPED PROJECT_L

The lowest contributing features selected were removed from each project. Then again, LR model was re-applied to find the

effect on prediction. The obtained results are illustrated in the following Table 5.

Table 5: AUC and G-mean obtained after removing lowest contributing features.

Dataset	AUC		G-mean	
	Before	After	Before	After
PROJECT_A	0.421	**0.712**	0.693	**0.983**
PROJECT_K	0.366	**0.816**	**0.798**	0.691
PROJECT_L	0.245	0.245	0.312	0.312

From Table 5, only PROJECT_A shows the improvement in both performance metrics after removing the lowest contributing features. In the case of PROJECT_K, the performance of AUC has increased but the G-mean has decreased. Meanwhile, PROJECT_L has no change after removing the lowest contributing features. In Figure 4, it shows that two features have a high contribution in comparison to others. When the lowest contributing features were removed, the highest contributing features were added back. The result was the same with and without removing the lowest contributing features.

2. Decision Tree Classifier

Similar techniques to LR were applied in the case of DT to find the impact of effects on prediction after removing highest and lowest contributing features. The results obtained are illustrated in Table 6.

Table 6: AUC obtained after removing highest and lowest contributing features.

Datasets	AUC Before XAI	AUC After XAI	
		Removed Highest Contributors	Removed Lowest Contributors
PROJECT_A	0.997	0.995	**0.998**
PROJECT_K	0.998	0.879	**0.998**
PROJECT_L	**0.930**	0.900	0.910

From Table 6, the result shows that the performance of PROJECT_A and PROJECT_K has been increased when removing the lowest contributing features. But PROJECT_L has shown better performance without applying XAI.

Table 7: G-MEAN obtained after removing highest and lowest contributing features.

Datasets	G-mean	G-mean After XAI	
	Before XAI	Removed Highest Contributors	Removed Lowest Contributors
PROJECT_A	0.977	0.963	**0.997**
PROJECT_K	0.888	0.883	**0.907**
PROJECT_L	0.959	**0.965**	0.948

Table 7 shows that PROJECT_A and PROJECT_K have increased their performance after removing some low contributing features. But in the case of PROJECT_L, G-mean has increased when removing the highest contributing features which directs toward more proper and deep analysis of features and their contribution.

RQ1 Answer:

The removal of both the highest and lowest contributing features shows a direct impact on the performance of the model. The result obtained after applying feature engineering on two different ML models has the same outcome. It was found that in most cases the performance of models increased when the lowest contributing features were removed and vice versa.

RQ2. What is the effect on other features after removal of high and low contributing features?

In both models, some uncertain behaviors were seen after removing some contributing and non-contributing features. That uncertain behavior could be seen in all projects. The illustrations of the LR model in PROJACT_A and PROKECT_K are presented in Figure 8 and Figure 9.

Figures 8 and 9 are a combination of three different states of prediction for comparison purposes. The first figure is before removing any features contributing features, the second figure is after removing the highest contributing features and the third figure is after removing the lowest contributing features.

In Figure 8, after removing the highest contributing and lowest contributing features, features showed a random pattern of their contribution. Some features have increased, and some have decreased their contribution to the prediction. Such uncertain behavior may make complicated feature importance analysis. In the second chart of Figure 8, the mean SHAP value was dropped by ten times while in the third chart, it was increased by one and half times.

In Figure 9, a similar pattern can be noticed in the case of PROJET_K where features have changed their contribution values as well as after removing the lowest contributing feature the contributing values of top features have increased. In the

second chart of Figure 9, the mean SHAP value was dropped by almost three times while in the third chart, it was doubled.

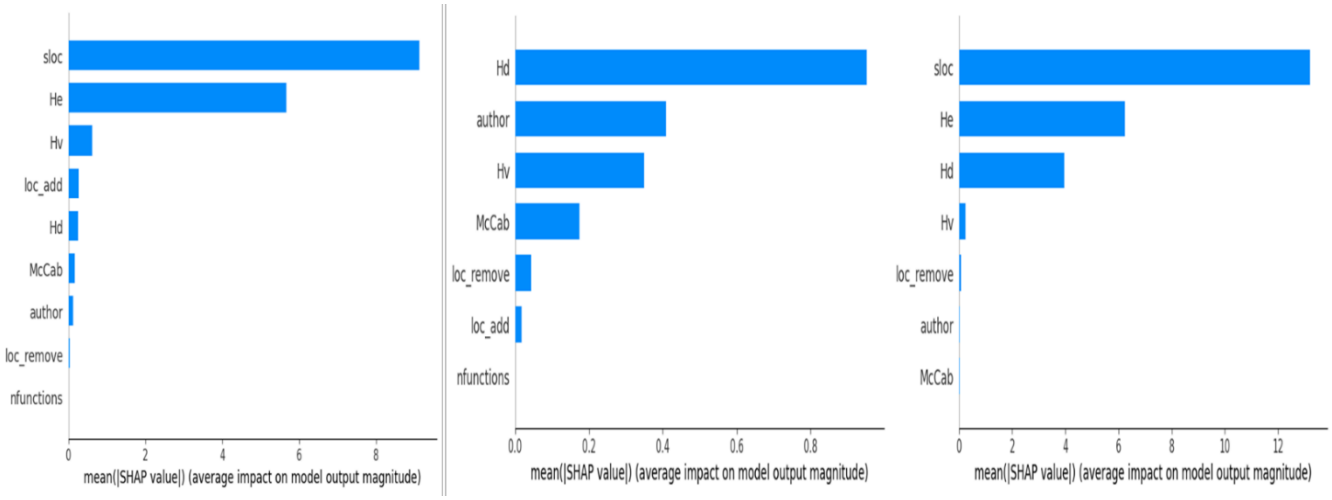


FIGURE 8. FEATURE BEHAVIOR ANALYSIS AFTER REMOVING HIGHEST AND LOWEST CONTRIBUTING FEATURES IN PROJECT_A

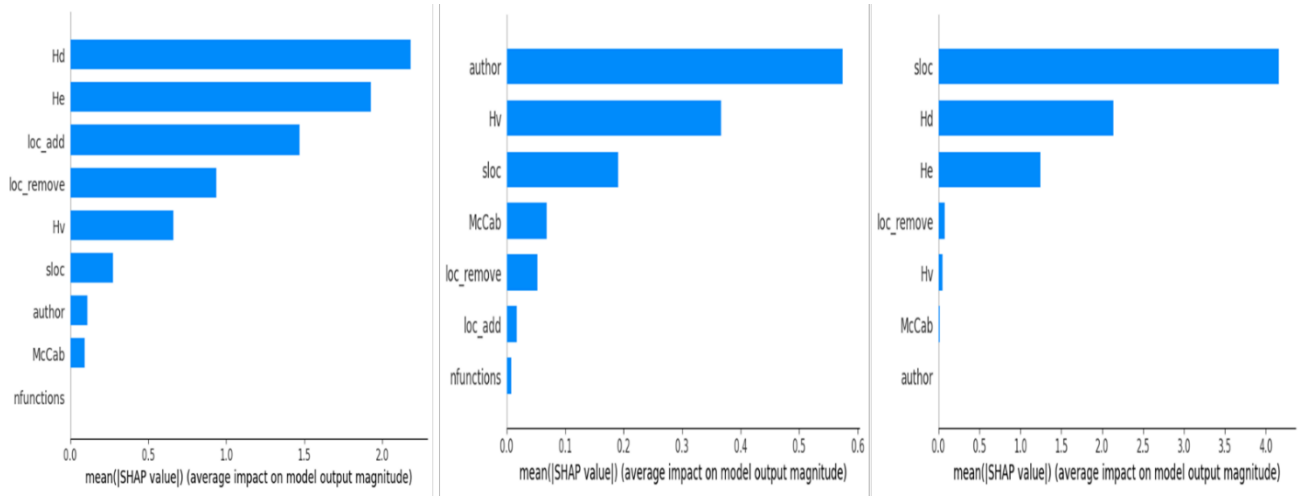


FIGURE 9. FEATURE BEHAVIOR ANALYSIS AFTER REMOVING HIGHEST AND LOWEST CONTRIBUTING FEATURES IN PROJECT_L

RQ2 Answer:

After removing the highest and lowest contributing features, it was found that the features changed their contribution values, and the mean SHAP contribution values of features increased after removing the lowest contributing features. This increase in mean SHAP value has also led to an increase in performance which is a positive indication toward model optimization. And those uncertain behaviors can give a new direction to study.

RQ3. What are the things that should be considered while applying XAI?

According to the RQ2 conclusion, some uncertain behaviors of features were noticed regarding their uncertain change in their contribution values. Such uncertain behavior shows the necessity of proper analysis of the features before applying a feature engineering approach in any dataset, project, or algorithm. The importance of features differs based on the prediction requirement so, based on the prediction requirement certain features can be kept or removed after understanding their contribution values for the prediction using XAI.

Hence, it is important to understand each feature's importance and contribution before using them for specific tasks.

7. Threats to Validity

7.1 External Validity

Our model demonstrates superior performance on defect prediction tasks, and we could apply XAI for model optimization based on the feature importance analysis. However, it may not be generalized to other tasks and may require careful adaptation to different domains. The dataset and the projects we trained our model were limited but in future, the size and features of the dataset may increase which will surely make the analysis process more complicated.

7.2 Construct Validity

The experiment expects to optimize the model performance through feature engineering using XAI. We were able to meet our expectations in most cases but not in all cases. For those cases, we consider them as future works. SHAP technique cannot explain all models due to their complex structure. So proposed technique might not be generalized for all approaches.

8. Conclusion

In conclusion, our research on Explainable Artificial Intelligence techniques, specifically SHapley Additive eXplanations within the context of Logistic Regression and Decision Tree Classifiers for Software Defect Prediction has presented valuable insights with the novel approach for model optimization. By harnessing SHAP's interpretability capabilities, we have not only obtained the transparency of these models but also gained a deeper understanding of the influential factors or features contributing to software defect prediction. With the help of feature contribution values, we analyzed the importance of the features and were able to apply feature engineering technique. After feature engineering technique, we were able to optimize ML model's performance. The optimized performances were compared using AUC and G-MEAN. This research also highlights the significance of XAI in enhancing model interpretability, thereby fostering trust, and facilitating more informed decision-making in software engineering. The findings underscore the potential for XAI to play a vital role in addressing the inherent opacity of ML models, contributing to the advancement of reliable and interpretable software defect prediction methodologies.

9. Future Work

This research has proposed a novel approach for feature engineering by applying XAI's SHAP technique. Still there are some areas where improvement can be made which are considered as future work that may be helpful to other researchers in working in the same or related fields.

The future work to be considered are as follow:

- Use XAI to understand the change in contributing behavior of the features for more robust prediction.
- Use XAI to compare other state-of-the-art Deep Learning methods to understand other methods in more detail.

Acknowledgment

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2023-00237159) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2022R1I1A3069233).

References

[1] J. Pan, "Software Reliability," Carnegie Mellon University, 1999. [Online]. Available: https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/. [Accessed 18 11 2023].

[2] M. Assim, Q. Obeidat and M. Hammad, "Software Defects Prediction using Machine Learning Algorithms," *IEEE Xplore,* pp. 1-6, 20220.

[3] D. Tsang, "White Box vs. Black Box Algorithms in Machine Learning," 19 07 2019. [Online]. Available: https://www.activestate.com/blog/white-box-vs-black-box-algorithms-in-machine-learning/. [Accessed 20 11 2023].

[4] C. Rudin and J. Radin, "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition," 22 11 2019. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/f9kuryi8/release/8. [Accessed 19 11 2023].

[5] H. Patel, "What is Feature Engineering — Importance, Tools and Techniques for Machine Learning," 30 08 2021. [Online]. Available: https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10. [Accessed 21 11 2023].

[6] V. Trevisan, "Using SHAP Values to Explain How Your Machine Learning Model Works," 18 01 2022. [Online]. Available: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137. [Accessed 22 11 2023].

[7] C. Tantithamthavorn and J. Jiarpakdee, Explainable AI for Software Engineering, Monash University, 2021.

[8] B. John, "How to Use SHAP Values to Optimize and Debug ML Models," 28 08 2023. [Online]. Available: https://neptune.ai/blog/shap-values. [Accessed 28 11 2023].

[9] H. Heaton and S. W. Fung, "Explainable AI via learning to optimize," Scientific Reports, 2023.

[10] S. Dechand, "Rule of Ten: How To Cut Your Development Costs," [Online]. Available: https://www.code-intelligence.com/blog/rule-of-ten. [Accessed 24 11 2023].

[11] H. Y. G. F. Z. S. M. L. Y. L. Zijie Huang, "Aligning XAI explanations with software developers' expectations: A case study with code smell prioritization," *ScienceDirect,* vol. 238 Part A, no. 0957-4174, 2023.

[12] W. On, "Visualizing AI," 24 03 2020. [Online]. Available: https://towardsdatascience.com/visualizing-ai-8fad4ea70b87. [Accessed 25 11 2023].

[13] M. Begum, M. H. Shuvo, I. Ashraf, A. A. Mamun, J. Uddin and M. A. Samad, "Software Defects Identification: Results Using Machine Learning and Explainable Artificial Intelligence Techniques," *IEEE Access ,* vol. 11, pp. 132750-132765, 2023.

[14] B. Gezici and A. K. Tarhan, "Explainable AI for Software Defect Prediction with Gradient Boosting Classifier," *IEEE,* 2022.

[15] R. A. J. N. J. W. S. W. Jiho Shin, "Explainable Software Defect Prediction: Are We There Yet?," *Arxiv,* 2021.

[16] Ł. Chmielowski, M. Kucharzak and R. Burduk, "APPLICATION OF EXPLAINABLE ARTIFICIAL INTELLIGENCE IN SOFTWARE BUG CLASSIFICATION," *Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska,* 2023.

[17] H. Altinger, ""Dataset on automotive software repository,"," [Online]. Available: http://www.ist.tugraz.at/_attach/Publish/AltingerHarald/MSR_2015_dataset_automotive.zip. [Accessed 10 12 2023].

[18] H. Altinger, S. Siegl, Y. Dajsuren and F. Wotawa, "A Novel Industry Grade Dataset for Fault Prediction Based on Model-Driven Developed Automotive Embedded Software," in *12th Working Conference on Mining Software Repositories*, Florence, Italy, 2015.

[19] T. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering,* Vols. SE-2, no. 4, pp. 308-320, 1976.

[20] V. Kanade, "What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices," spiceworks , 18 04 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/. [Accessed 15 11 2023].

[21] A. Saini, "Decision Tree Algorithm – A Complete Guide," Analytics Vidhya, 13 09 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/. [Accessed 14 11 2023].

[22] A. A. Awan, "An Introduction to SHAP Values and Machine Learning Interpretability," datacamp, 06 2023. [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability. [Accessed 16 11 2023].

[23] Dun Aengus, "About ROC, AUC Curve: Receiver Operating Characteristic, Area Under the Curve," 25 10 2020. [Online]. Available: https://nicola-ml.tistory.com/30. [Accessed 01 12 2023].

[24] Rfriend, "R, Python analysis and programming friend (by R Friend)," 29 1 2023. [Online]. Available: https://rfriend.tistory.com/774. [Accessed 03 12 2023].

모바일 에지 컴퓨팅환경에서의 서비스 품질 연구 고찰

김지영, 남준성, 이재혁, 류덕산*
전북대학교 소프트웨어공학과
{jiyoung_kim, nam48677, messi3000, duksan.ryu}@jbnu.ac.kr

Service quality research review in mobile edge computing environment

Jiyoung Kim, Junseong Nam, Jaehyuk Lee, Duksan Ryu*
Department of Software Engineering, Jeonbuk National University

요 약

모바일 에지 컴퓨팅(Mobile Edge Computing, MEC)은 기존의 중앙 집중 클라우드가 아닌 네트워크 가장자리(Edge)에서 연산을 수행해 성능 향상과 지연 감소를 이루는 기술이다. MEC의 적용을 촉진하기 위해서 MEC 환경에서 QoS(Quality of Service) 관련 연구 현황을 파악할 필요가 있다. 본 연구에서는 사용자에게 최적의 서비스를 제공하기 위한 MEC 관련 연구 현황과 연구 분야에 대해 분석하고, 특징에 따라 분류 후 유형별로 어떤 특징이 있는지 분석한다. 분석 결과, 연구는 크게 보안, QoS 모니터링, 에지 서버 위치 선정 세 가지로 구분된다. 보안 분야에서는 보안기법을 적용하기 위한 데이터 전처리 적용 및 보안과 QoS를 동시에 고려하였고, QoS 모니터링 분야는 데이터 종속성 고려, 협업필터링 기반의 서비스 추천 방법이 적용되었다. 에지 서버 위치 선정 분야는 사용자 패턴을 사전에 예측한 연구, 다중 목적 최적화 문제 해결 기법이 활용되었다. 본 연구를 통해 후속 연구 시 MEC 환경에서 QoS 향상을 위한 기존 기법들을 쉽게 파악해 새로운 QoS 향상 연구를 수행할 수 있을 것으로 기대한다.

1. 서 론

모바일 에지 컴퓨팅 (Mobile Edge Computing, MEC)은 모바일 클라이언트 근처와 모바일 클라이언트와 클라우드 서버 사이의 에지 서버에 클라우드 리소스를 배포하여 모바일 클라우드 컴퓨팅(Mobile Cloud Computing, MCC)에 권한을 부여해 서비스를 제공하는 새로운 기술로 최종 사용자에게 빠르고 강력한 컴퓨팅, 전력 효율성, 저장 공간, 이동성, 위치, 그리고 맥락 인식 지원을 제공한다[1]. 그리고 이는 사용자에게 고품질의 QoS(서비스 품질, Quality of Service)를 제공하는데 기여한다.

MEC 관련 연구들을 정리한 연구는 존재하지만 MEC 환경에서 이동성을 고려한 QoS 향상 연구들의 현황을 분석한 연구는 부재하기 때문에 이 연구들의 현황과 향후 연구 방향을 파악할 필요가 있다. 따라서 본 논문은 MEC 관련 연구 중 사용자의 이동성을 고려한 QoS 향상을 위한 연구들에 대해 분석한다. 분석 결과로, QoS 모니터링, 에지 서비스 위치 선정 추천, 그리고 보안성(개인 정보 보호)으로 분류 후 각 연구 별 특징에 대해 설명한다. 마지막으로 진행된 연구들의 한계점과 이를 개선하기 위해 진행될 수 있는 연구 방향을 제안한다.

2. 배경 및 관련 연구

MEC 환경에는 다양한 애플리케이션 및 애플리케이션 시나리오에 적용 가능한 기술들이 있다. 증강 현실[3], 콘텐츠 전달 및 캐싱[2], 건강관리[4], 비디오 분석[5], 모바일 빅데이터 분석[6], 차량 연결[7], 스마트 그리드[2], 무선 센서 및 액추에이터 네트워크[2], 스마트 건물 제어[2], 소프트웨어 정의 네트워킹[8], 그리고 해양 모니터링[9]은 모두 MEC 환경에서 수행될 수 있으며 MEC의 장점이 유용하게 사용될 수 있다[2]. 또한 최근 연구들은 MEC의 몇 가지 특징인 계산 오프로딩, 낮은 지연 시간, 저장 공간, 그리고 전력 효율성을 다루고 있다[2].

이전의 관련 논문은 MEC가 활용될 수 있는 시나리오에 관련된 연구 및 MEC의 특징에 따른 연구들을 개괄적으로 정리하였다[2]. 그러나 MEC 환경에서 이동성을 고려한 사용자의 QoS를 향상시키기 위한 연구들에 대해 정리한 논문은 부재하다.

3. 연구 방법 및 설정

RQ1. 모바일 에지 컴퓨팅 환경에서 이동성을 고려한 QoS 연구는 어떤 유형으로 분류할 수 있는가?

RQ2. 모바일 에지 컴퓨팅 환경에서 이동성을 고려한 QoS 연구의 유형별 특징은 무엇인가?

이와 함께 MEC 환경에서 이동성을 고려한 연구들의 공통점과 향후 집중해야 할 분야에 대해 기술하고자 한다.

4. 연구 결과

4.1 RQ1. 모바일 에지 컴퓨팅 환경에서 이동성을 고려한 QoS 연구는 어떤 유형으로 분류할 수 있는가?

MEC 연구는 다양한 분야에서 진행되며, 사용자 중심의 서비스 최적화에 주목하고 있다. 이는 성능, 효율, 가용성, 사용자 맞춤 경험을 제공하여 사용자 만족도를 높이는 것을 목표로 하고 있다. 본 연구는 '구글 스칼라'를 활용하여 'MEC', 'QoS', 그리고 'Mobility' 키워드의 검색 결과를 바탕으로 최신 연구 동향을 분석하였다.

MEC 환경에서 서비스 제공에 있어 신뢰성은 중요한 요소이다. 사용자는 저지연성과 안전한 서비스를 기대하며, 이를 위한 연구가 지속적으로 수행되고 있다. 이러한 연구는 보안, QoS 모니터링, 에지 서버 위치 선정 등 세부 분야에서 진행된다.

보안 분야에서는 차등 개인정보 보호, 위치정보 강화, 민감정보 형태 변환 등의 연구가 주요하게 이루어졌다[10, 11, 12]. QoS 모니터링 분야에서는 가우시안 모니터링, 협업 필터링 예측, 자동 인코더, 다중 QoS 예측 등의 연구가 주로 진행되었다[13, 14, 15, 16]. 에지 서버 위치 선정 분야에서는 위치 정보 강화, 민감 정보의 형태 변환 등의 연구가 주로 수행되었다[17, 18].

MEC 환경에서 사용자에게 최적의 서비스를 제공하기

* 교신저자임

위해서는 에지 서버의 위치 선정은 사용 사례 및 서비스 특성에 따라 최적의 위치를 선정하는 것이 중요하다.

4.2 RQ2. 모바일 에지 컴퓨팅 환경에서 이동성을 고려한 QoS 연구의 유형별 특징은 무엇인가?

연구현황은 보안, QoS 모니터링, 그리고 에지 서버 위치 선정 3가지로 분류할 수 있다.

보안 분야에서는 보안 기술 적용을 위한 데이터 전처리가 필요하며, 보안 기법과 QoS를 동시에 고려하는 접근법이 주로 사용된다. 보안 알고리즘을 실행하기 위한 에지 장치의 효율성이 낮은 편이고, 보안 강도를 높이는 경우 QoS 저해 요소가 발생할 수 있다. 따라서, 이동성을 기반으로 QoS와 보안을 모두 고려하는 연구가 필요하다.

QoS 모니터링 분야에서는 데이터 종속성에 대한 문제를 고려하고, 협업 필터링을 활용한 서비스 추천 방식으로 연구가 진행되고 있다. 데이터 누락으로 인한 QoS 감소를 방지하기 위한 데이터 추정치 사전 계산이 이루어진다. 그러나 제공되는 데이터양에 따라 결과값에 오차가 발생하는 문제가 있다.

에지 서버 위치 선정 분야에서는 사용자의 이동 패턴을 예측하고, 다중 목적 제약 최적화 문제를 해결하는 방식을 주로 적용하고 있다. 그러나, 각각의 에지 서버의 성능에 대한 고려가 부족한 상태이다.

MEC 환경에 대한 연구에서 MEC 환경을 재현한 데이터셋은 아직까지 없다. 따라서, 다양한 데이터 퓨전 방법 및 응답시간과 처리량 이 외의 성능 평가 기준도 필요하다.

5. 위협요소

MEC 환경에서 다양한 연구분야들 간의 연결고리를 찾고, 서로 보완하여 일관성을 유지하는 결과가 도출되어야 한다. 본 연구에서 선정된 논문은 특정 시각과 접근 방식에 기반하므로, 이 분석결과가 보편적으로 적용되는 분류를 제공하기는 어렵다.

6. 결론 및 향후방안

MEC 분야 연구는 **4.2**에서 확인할 수 있듯이 보안, QoS 모니터링, 에지 서버 위치 선정의 세 분야에서 이루어지고 있으며, 몇몇의 방법을 활용하고 있다. MEC 환경 재현을 위해 데이터 퓨전 방법을 적용하고 있지만, QoS 성능 평가 기준이 응답시간과 처리량에 한정되어 있어, 이에 대한 개선이 필요하다. 이는 데이터셋 부족으로 인한 문제로, 다양한 데이터셋 확보가 필요하다. 또한, 사용자 서비스의 복잡도, 에지 서버의 성능 복잡도 등 다양한 변수를 고려한 최적화 연구와 신뢰성 제공을 위한 보안과 QoS 품질의 균형을 유지하는 연구가 필요하다.

감사의 글

이 논문은 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터지원사업 (IITP-2024-2020-0-01795)과 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과임.(2023RIS-008)

참고 문헌

[1]Yi et al, “A survey of fog computing: Concepts, applications and issues,” in Proc. Workshop Mobile Big Data, Hangzhou, China, 2015, pp. 37–42.

[2]Abbas et al, "Mobile Edge Computing: A Survey," in *IEEE Internet of Things Journal*, vol. 5, no. 1 , Feb. 2018.

[3]Dastjerdi et al, “Fog computing: Principles, architectures, and applications,” in Internet of Things: Principles and Paradigms. San Mateo, CA, USA: Morgan Kaufmann, 2016, ch. 4, pp. 61–75.

[4]Stantchev et al, “Smart items, fog and cloud computing as enablers of servitization in healthcare,” Sensors Transducers, vol. 185, no. 2, pp. 121–128, 2015.

[5]Hong et al, “Mobile fog: A programming model for large-scale applications on the Internet of Things,” in Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput. (MCC), Hong Kong, 2013, pp. 15–20.

[6]Taherkordi et al, “From IoT big data to IoT big services,” in Proc. Symp. Appl. Comput. (SAC), Marrakech, Morocco, 2017, pp. 485–491

[7]Datta et al, “Fog computing architecture to enable consumer centric Internet of Things services,” in Proc. Int. Symp. Consum. Electron. (ISCE), Madrid, Spain, 2015.

[8] R. Jain and S. Paul, “Network virtualization and software defined networking for cloud computing: A survey,” IEEE Commun. Mag., vol. 51, no. 11, pp. 24–31, Nov. 2013.

[9] E. Ahmed and M. H. Rehmani, “Mobile edge computing: Opportunities, solutions, and challenges,” Future Gener. Comput. Syst., vol. 70, pp. 59–63, May 2017.

[10]ZHANG et al. “Privacy-preserving quality prediction for edge-based IoT services,” Future Generation Computer Systems, 2021, 114: 336-348.

[11]JIN et al. “Mobility-aware and privacy-protecting qos optimization in mobile edge networks,” in IEEE Transactions on Mobile Computing, 2022.

[12]JIN et al. “Privacy-aware forecasting of quality of service in mobile edge computing,” in IEEE Transactions on Services Computing, 2021.

[13]ZHANG et al. ”Mobility and dependence-aware QoS monitoring in mobile edge computing,” in IEEE Transactions on Cloud Computing, 2021, 9.3: 1143-1157.

[14]WANG et al. “QoS prediction for service recommendations in mobile edge computing,” in Journal of Parallel and Distributed Computing, 2019, 127: 134-144.

[15]YIN et al. “QoS Prediction for Mobile Edge Service Recommendation With Auto-Encoder,” in IEEE, 2019.

[16]LIU et al. “Context-aware multi-QoS prediction for services in mobile edge computing,” In: 2019 IEEE international conference on services computing (SCC). IEEE, 2019. p. 72-79.

[17]ZHANG et al. “Mobility-aware personalized service recommendation in mobile edge computing,” EURASIP Journal on Wireless Communications and Networking, 2021.

[18]WANG et al. “Edge server placement in mobile edge computing,” Journal of Parallel and Distributed Computing, 2019, 127: 160-168.

사전 훈련된 기계 학습 모델의 효과적인 조합을 위한 공유 모델 허브 분석

Arogya Kharel[O], 고인영

한국과학기술원 전산학부

{akharel, iko}@kaist.ac.kr

Analyzing Model Hubs for Effective Composition of Pre-trained Machine Learning Models

Arogya Kharel[O], In-Young Ko

School of Computing, Korea Advanced Institute of Science and Technology

요 약

Deep Neural Network (DNN) models have become prevalent and are increasingly adopted as components in software systems. Designing and training these DNNs from scratch is not trivial as they require substantial expertise and resources. Consequently, developers often reuse Pre-Trained Models (PTMs) organized in model hubs to alleviate these concerns. However, challenges arise when there is an absence of PTMs that match a developer's specific requirements. In this paper, we explore the concept of PTM composition and investigate whether PTM composition can fulfill application requirements without fine-tuning or creating a new DNN. We present the current challenges in PTM composition through our case study and identify the shortcomings of existing model hubs. By drawing parallels between PTM composition and web service composition, we highlight the essential technologies required for successful PTM composition and discuss potential solutions to these issues.

1. Introduction

Deep Neural Networks (DNNs) are now widely used in various applications. However, designing and training these DNNs are not trivial tasks as they require substantial expertise and resources. Consequently, model hubs like Hugging Face that host Pre-Trained Models (PTMs) have surged in popularity, helping developers reuse existing DNNs rather than creating new DNNs from scratch.

While PTMs are useful, developers might need to fine-tune them or create new DNNs if no PTMs meet their specific needs. Both approaches require suitable training datasets, which can be difficult to obtain [1]. This raises questions about the possibility of reusing and combining existing PTMs for specific applications, the adequacy of model hubs in supporting PTM composition, and the potential for developing new functionalities and services through PTM reuse.

This paper introduces PTM composition, aggregating PTMs to create new functionalities. We explore its feasibility, particularly the challenges in using existing model hubs, and present a case study with examples to illustrate common obstacles like PTM discovery, selection, and heterogeneity. Finally, we suggest strategies to overcome these challenges in future work.

2. Related Works

Web Service Composition (WSC) integrates various Web services to create sophisticated services [2]. The key similarity between PTM composition and WSC lies in their core principle of integrating multiple, specialized components to create a composite system capable of addressing more complex tasks. In contrast, in PTM composition these components are PTMs, while in WSC they are Web services.

3. Composition of Pre-trained Models

A PTM is a saved DNN that has been trained on large datasets, which can then be used as-is or adapted for different tasks [3]. This practice, known as PTM reuse, aligns with software engineering and service computing principles of reuse. PTM composition refers to the strategic aggregation of pre-existing PTMs each originally trained for specific tasks or datasets. The aim is to address more complex or varied tasks than any single PTM could handle independently.

Figure 1 illustrates the composition process with current model hubs. The developer searches for the necessary PTMs. The suitable candidate PTMs can be downloaded via Hub APIs. Next, the developer creates proper interfaces for the constituent PTMs to interact with each other. Lastly, the final composite system is deployed.

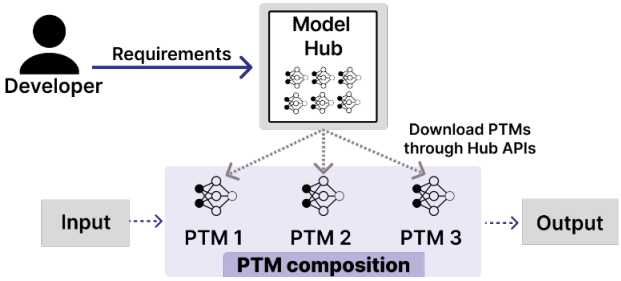


Figure 1. PTM composition process with current model hubs.

The common workflow in public model hubs such as Hugging Face or PyTorch Hub, involves users submitting PTMs, which are then accessible via hub APIs [4]. However, this user-driven model poses challenges in standardizing and assuring quality, impacting PTM reliability and effectiveness. Significant issues include the lack of standardized naming conventions, inadequate search filters for specific requirements, and the absence of comparison information. Additionally, many hubs don't provide detailed information on PTM inputs and outputs, complicating PTM selection, and integration.

4. Case Study

We conducted a case study highlighting the challenges associated with composing PTMs. The study focused on software identifying ambulances in low-resolution images. Figure 2 illustrates the overall composition of this case study. The developer utilized a classification PTM to label "ambulance" as the object detection PTM was unable to do so. The incorporation of an upscaler markedly improved the classification accuracy of the classification model from 37.5% to 96.9%. Currently, a developer faces significant challenges, including the difficulty in discovering, selecting, and integrating PTMs from model hubs with varying documentation and input requirements.

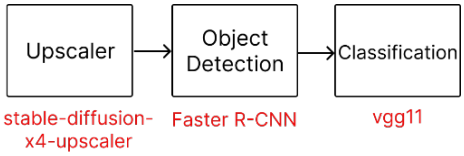


Figure 2. The overall PTM composition.

5. ML Repository for PTM Composition

WSC and PTM composition share similarities in their overall lifecycles. These lifecycles encompass four key phases: *definition*, *selection*, *deployment*, and *execution* [2]. By drawing parallels from WSC, we propose a PTM repository to address PTM composition challenges, offering improvements over current model hubs. This repository uses Large Language Models (LLMs) for natural language processing in the *definition* phase, enabling developers to specify requirements easily. In the *selection* phase, it considers various factors like deployment systems and QoS metrics, utilizing crowdsourced data for accurate model selection. Post-selection, the repository encapsulates PTMs with custom wrappers for compatibility, facilitating smooth data and process flow between models. These wrappers and the repository's APIs aid in seamless integration, *deployment*, and *execution*, streamlining the PTM composition process and overcoming current challenges.

6. Conclusion

We introduce and explore PTM composition, highlighting its potential and challenges, as evidenced by our case study. We propose a future vision of a model hub for more effective PTM composition by drawing parallels with WSC to identify key technologies for PTM integration.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2020-0-01795) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

[1] Liang, Weixin, et al. "Advances, challenges and opportunities in creating data for trustworthy AI." Nature Machine Intelligence 4.8 (2022): 669-677.

[2] Sheng, Quan Z., et al. "Web services composition: A decade's overview." Information Sciences 280 (2014): 218-238.

[3] Han, Xu, et al. "Pre-trained models: Past, present and future." AI Open 2 (2021): 225-250.

[4] Jiang, Wenxin, et al. "An empirical study of artifacts and security risks in the pre-trained model supply chain." ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses. 2022.

STPA를 활용한 협업 가상물리시스템의 안전성 테스트 케이스 생성

허윤아○, 유준범
건국대학교 컴퓨터공학과
hyoona1202@naver.com, jbyoo@konkuk.ac.kr

Generation of Safety Test Cases for Cooperative Cyber-Physical Systems Using STPA

Yoona Heo○, Junbeom Yoo
Department of Computer Science and Engineering, Konkuk University

요 약

가상물리시스템 (Cyber-Physical System, CPS)은 연산 (computation)부, 제어 (control)부, 그리고 통신 (communication)부가 통합되어 동작하는 실시간 시스템으로, 일반적으로 안전성이 중요한 시스템이다. 많은 경우 서로 다른 CPS들이 공동의 목표를 달성하기 위하여 협업 (cooperate)하는데, 이러한 상황에서는 다양한 시스템이 협업할 때 나타날 수 있는 잠재적 위험 (hazard) 또한 고려되어야 한다. 따라서 개별적인 CPS 뿐 아니라 협업하는 CPS들 전체의 안전성을 확보하기 위해 노력할 필요가 있다. 본 연구에서는 안전성을 확보하기 위한 방안 중 하나인 안전성 테스팅을 위하여 위험 분석 기법인 Systems-Theoretic Process Analysis (STPA)를 활용하여 안전성을 검증하기 위한 테스트 케이스를 생성하는 프로세스를 제안한다. 또한, guideword를 제안하여 STPA 결과를 활용해 안전성 테스트 케이스를 작성함에 있어 도움을 줄 수 있도록 한다. 추가로, 기존에 개발한 테스트베드 시스템을 사례 연구 대상 시스템으로 활용하여 안전성 테스트 케이스를 생성하고, 생성한 테스트 케이스를 통해 안전성 테스팅까지 진행한 결과를 통해 제안하는 프로세스를 분석한다.

1. 서론

가상물리시스템 (Cyber-Physical System, CPS)은 연산 (computation)부, 제어 (control)부, 그리고 통신 (communication)부가 통합되어 동작하는 실시간 시스템 **[1, 2]**으로, 일반적으로 시스템에 문제가 발생할 경우 인명 피해, 재산 손실 등의 손실을 미칠 수 있는, 안전성 (safety)이 중요한 시스템이다. 많은 경우 서로 다른 CPS들이 공동의 목표를 달성하기 위하여 협업 (cooperate)한다. 시스템 이론에서는 시스템의 각 구성 요소 (component) 수준에서는 발견되지 않았던 특성들이 시스템 수준에서는 나타날 수 있다고 이야기하며, 이러한 특성들을 emergent property라고 부른다 **[3]**. 이와 마찬가지로, CPS가 공동의 목표를 가지고 서로 연계되어 동작, 즉 협업할 때, 개별 CPS에서는 확인할 수 없었던 emergent property가 나타날 수 있다. 이러한 Emergent property 중 하나인 safety는 개별 CPS 뿐 아니라 협업하는 여러 CPS에서도 중요하게 다뤄져야 한다. 다양한 CPS가 협업하면서, 개별 CPS에서는 나타나지 않았던 잠재적인 위험 (hazard)들이 등장할 수 있다. 따라서 개별적인 CPS 뿐 아니라 협업하는 CPS들 전체의 안전성을 확보하기 위해 노력할 필요가 있다.

일반적으로, 시스템 안전 표준인 미 국방부 (Department of Defense)의 MIL-STD-882E **[4]**, 기능 안전성 (functional safety) 표준인 IEC 61508 **[5]**, ISO 26262 **[6]**, 기능 안전성에 대한 직접적인 언급은 없으나 항공 분야에서의 기능 안전성 표준으로 간주되는 DO-178C **[7]** 등을 살펴보면, 안전 관련 (safety-related) 시스템 혹은 안전 중요 (safety-critical) 시스템의 안전성은 보장되어야 하며, 이를 위하여 위험 분석 (hazard analysis, HA), 리스크 평가 (risk assessment), 안전 요구사항 정의 등의 활동을 수행하여 안전성을 보장할 수 있어야 한다고 말한다. 이 외에도 시스템의 안전성을 확보하기 위한 활동에는 안전성에 대한 검증 (verification)도 포함될 수 있다 **[8]**.

본 연구에서는 안전성을 확보하기 위한 방안으로 safety verification 방법 중 하나인 안전성 테스팅 (safety testing)을 위하여 위험 분석 기법인 Systems-Theoretic Process Analysis (STPA) **[9]**의 결과를 활용하여 안전성 테스트 케이스 (test case, TC)를 수동으로 생성하는 프로세스를 제안한다. 이 때 품질 특성 (Quality Attribute, QA) 요구사항을 시나리오 형태로 나타내는 Quality Attribute Scenario (QAS) **[10]**의 형태를 빌려 안전성 테스트 케이스를 생성한다. 또한, TC 생성에 있어 도움을 줄 수 있는 guideword를 제안하여 STPA 결과를 활용해 safety TC를 생성함에 있어 도움을 줄 수 있도록 한다. 추가로, 사례 연구를 통해 사례 연구 대상 시스템의 safety TC를 생성

하고, 생성한 TC를 활용하여 safety testing을 진행하고 그 결과를 살펴본다.

본 논문의 구성은 다음과 같다. 2장에서는 앞서 언급된 STPA와 QAS에 대한 배경지식을 설명하고, 3장에서는 관련 연구에 대하여 설명한다. 다음으로 4장에서는 제안하는 안전성 테스트 케이스 생성 프로세스에 대한 내용을 설명하고, 5장에서는 사례 연구 대상 시스템과 프로세스 적용 결과에 대하여 설명하고 논의한다. 마지막으로 6장에서는 결론과 향후 연구에 대해 기술한다.

2. 배경지식

2.1 Systems-Theoretic Process Analysis (STPA)

Systems-Theoretic Process Analysis (STPA)는 HA 기법의 한 종류로, 시스템 이론에 기반한 causality model인 Systems-Theoretic Accident Model and Process (STAMP)에서 새롭게 식별된 원인 요소 (causal factor)를 포함하기 위해 제안된 기법이다 **[9]**. **[11]**에 따르면, STPA는 총 네 단계로 구성된다.

첫 번째 단계는 '분석 목적 정의' 단계이다. 이 단계에서는 시스템 수준에서 발생 가능한 loss (accident)와 각 loss를 발생시키는 원인이 되는 시스템 수준에서의 hazard를 식별한다. 이 과정에서 분석 대상 시스템과 그 경계 또한 식별한다. 또한 hazard를, 더 나아가서는 시스템의 loss를 막기 위한 시스템 수준의 제약 사항까지도 식별한다.

두 번째 단계에서는 control structure를 모델링한다. Control structure는 하나 이상의 control-feedback loop로 구성된 시스템의 계층 관계를 나타내는 추상적 모델이다. 각 control-feedback loop는 control의 주체가 되는 controller, control의 대상이 되는 controlled process, controller가 controlled process를 제어하기 위해 제공하는 control action (CA), 그리고 controlled process가 controller에게 제공하는 feedback으로 구성된다. 이 외에도 controller에서 controlled process를 제어하기 위해 필요한 actuator와 controller가 controlled process의 feedback을 입력 받기 위한 sensor가 있다. Controller는 의사 결정을 내리기 위해 내부적으로 control algorithm과 process model을 가지며, 이 중 process model은 controller가 가지는 내부적인 믿음으로, 시스템 혹은 환경에 대한 정보나 제어하는 controlled process의 상태 등을 포함할 수 있다.

세 번째 단계에서는 두 번째 단계에서 식별한 control structure에서의 CA가 특정 상황 또는 환경에서 hazard를 초래할 때, 즉 unsafe control action (UCA)이 될 때 이를 식별한다. 어떤 상황 또는 환경에서 CA가 제공되는지에 따라 안전한지 아닌지가 결정되므로, UCA를 식별할 때는 context를 명시해야 한다. UCA에는 네 가지 유형이 있고, 각각은 다음과 같다: (1) CA를 제공하지 않는 것이 hazard를 유발한다; (2) CA를 제공하는 것이 hazard를 유발한다; (3) 잠재적으로 안전한 CA를 제공하나, 너무 일찍, 너무 늦게, 혹은 잘못된 순서로 CA를 제공한다; (4) (지속적으로 제공되는 CA의 경우) 너무 오래 지속되거나 너무 빨리 중지된다.

마지막 단계에서는 UCA의 발생 원인을 나타내는 loss (혹은 causal) scenario를 식별한다. Loss scenario는 크게 두 유형으로 나뉜다. 각 유형에서는 'UCA가 발생하는' 원인과 'CA가 잘못 실행되거나 실행되지 않아서 hazard가 발생하는' 원인에 집중한다. UCA가 발생하도록 하는 원인에는 네 가지가 있다. (물리적 controller일 때) controller의 고장, 부적절한 control algorithm, 다른 controller에서 제공받은 UCA, 그리고 부적절한 process model이 이에 해당한다. 다음으로 CA가 잘못 실행되거나 실행되지 않아서 hazard가 발생하는 원인으로는 actuator를 포함하는 control path에서의 문제와 실제로 control에 따라 동작하는 controlled process의 문제가 있다. Loss scenario 식별 시에는 각 항목에 대한 구체적인 causal factor를 기술하고, 이것이 발생하지 않도록 하는 후속 조치에 활용할 수 있도록 한다. 이렇게 도출된 loss scenario를 통해 시스템의 안전성을 확보하기 위한 추가 요구사항이 도출될 수 있으며, 시스템의 아키텍처를 도출하거나 테스트 케이스를 정의하는 등의 활동이 후속될 수 있다.

2.2 Quality Attribute Scenario (QAS)

Quality Attribute (QA)는 시스템을 둘러싼 이해관계자들의 요구 (needs) 중 가치를 창출할 수 있으며 측정 가능한 (measurable) 비기능적 속성을 의미하며, 그 예시로는 안전성 외에도 신뢰성 (reliability), 확장성 (scalability), 사용성 (usability) 등이 있다 **[10, 12, 13]**. 이런 QA에 대한 요구사항을 시나리오의 형태로 구체화한 것을 품질 특성 시나리오 (Quality Attribute Scenario, QAS) **[10]**라고 부른다. QA는 일반적으로 SW의 설계 단계에서 architecture를 설정하는 데에 영향을 미치는 가장 큰 요인으로, QAS를 통해 SW가 가져야 할 QA 관점의 요구사항을 나타낸다. QAS는 총 여섯 가지의 요소로 구성되며, 각각의 이름과 정의는 다음과 같다:

- Source: 사람, 다른 시스템 등 stimulus를 제공하는 개체
- Stimulus: source에 의해서 artifact에게 주어지는 이벤트
- Artifact: stimulus가 도착하는 대상 (시스템의 전체 또는 일부, 혹은 시스템의 집단)

- Response: stimulus가 도착한 이후 artifact의 동작
- Response measure: 테스트를 위한, response에 대한 측정 가능한 값
- Environment: 시스템의 state 등 시나리오가 발생하는 환경의 집합

QAS는 크게 어느 한 시스템에 특정되지 않는 general scenario와 시스템에 특정되는 concrete scenario로 구분한다. 몇몇 QA의 general scenario는 **[10]**에서 확인할 수 있다. 특정 시스템에 대하여 제공된 general scenario를 유용하게 사용하기 위해서는 해당 시스템에 적용 가능하도록 general scenario에서 제공하는 값을 수정해야 한다. Concrete scenario는 이를 위하여 general scenario에서 제공하는 값들을 시스템에 적용 가능하도록 수정하여 작성한, 실제 시스템에 대한 QAS이다. 본 연구에서는 STPA 결과를 활용하여 safety에 대한 general scenario를 설정하고, 테스트베드에 적용하여 concrete scenario를 식별하는 예시를 보인다.

3. 관련 연구

3.1 협업/협력 가상물리시스템의 안전성

[14]에서는 협력하는 (collaborative) CPS의 가변성 (variability)을 고려하여 기존의 여러 HA 기법 중 세 가지 기법, 이벤트 트리 분석 (Event Tree Analysis, ETA) **[15]**, 결함 트리 분석 (Fault Tree Analysis, FTA) **[16]**, 고장 모드 및 효과 분석 (Failure Mode and Effect Analysis, FMEA) **[17]**을 확장하는 방법을 제안하였다. **[14]**에서는 또한 여러 HA 기법을 통해 도출된 결함을 추적하기 위하여 variability를 고려한 결함 추적 그래프 (fault traceability graph, FTG)를 개발하였다. 추가로, 확장된 각 HA의 모델링과 결함 추적 그래프의 생성을 지원하는 도구까지 개발하였다. 이를 통해, ISO 21448 **[18]**에서 언급되는 known-safe 영역을 최대화하는 동시에 unknown-unsafe 영역을 최소화하고자 하였다.

[19]에서는 무선 통신을 이용한 Safe Cooperating CPS (SafeCOP) 프로젝트 **[20]**의 일부로 진행된 군집 주행 시스템에 대하여 진행한 연구를 나타냈다. 해당 연구에서는 안전성을 보장하기 위하여 설계 단계에서는 Goal Structuring Notation (GSN) **[21]**을 활용한 safety case를 사용하고, 런타임에는 런타임 관리자를 통하여 지속적으로 시스템의 동작에 대한 명세에 해당하는 contract의 위반을 확인하고자 하였다. **[19]**에 직접적으로 언급되어 있지는 않으나, 같은 프로젝트를 다루고 있는 **[20]**을 통해 contract의 여러 subset은 STAMP **[9]**를 활용해 도출한 것으로 이해할 수 있다.

3.2 System of Systems의 안전성

System of Systems (SoS) **[22]**는 기존의 시스템들이 각 구성 요소가 합쳐져 하나의 시스템으로써 동작할 수 있었던 것처럼, 공동의 목표를 달성하기 위해 여러 시스템들이 합쳐져 하나의 시스템으로써 동작하는 것을 의미한다. 그런 의미에서 협업하는 여러 CPS를 하나의 더 큰 시스템으로 바라본다면, SoS와 유사한 형태를 가진다고 볼 수 있다.

[23]에서는, ISO 26262에서 요구되는 위험 분석 및 리스크 평가 (HARA) 방법을 사용하여 SoS의 hazard를 식별하는 프로세스를 제안하고 채석장 자동화 환경을 사례 연구로 활용하였다. 채석장에서 작동되는 자율 주행 기계의 impact matrix를 생성, 제안한 프로세스를 통해 분석하여 잠재적 사고와 hazard를 분석하였다.

[24]에서는 HA 기법인 Hazard and Operability Study (HAZOP) **[25]**와 FTA를 활용하여 전기 채석장 내에서의 서로 다른 기계 사이의 emergent behavior를 다루는 것에 초점을 맞췄다. 우선적으로 HAZOP을 적용하여 hazard와 잠재적 영향을 식별한 후, 결과물을 FTA에서의 top event로 설정한 후 FTA를 적용하여 원치 않는 이벤트의 발생을 예방할 수 있는 방법을 도출하였다.

본 연구와 기존 연구의 공통점은, 궁극적으로 더 안전한 협업하는 CPS를 만들고자 한다는 것이다. 단, 그것을 달성하기 위한 접근 방법에는 차이가 있다. 앞서 3.1절과 3.2절에서 언급한 관련 연구들의 경우, 대부분 HA를 적용하여 hazard의 발생을 예방하는 데에 초점을 맞추었고, HA 기법의 적용 이후 안전성을 검증하는 내용을 다루는 연구는 찾아보기 어려웠다. 단, 기존에 단일 CPS의 안전성 검증에 대한 연구 **[26, 27]**는 존재한다. **[26]**의 경우 인공 신경망과 정형 기법 (formal method)을 활용해 검증을 진행했다는 점에서 테스팅을 목표로 하는 본 연구와는 다소 접근이 다르다. **[27]**은 자율운항 선박 (autonomous ship) 시스템을 대상으로 하여 safety verification, 그 중에서도 safety testing을 위한 test scenario의 작성에 대한 내용을 포함하며, 본 연구에서처럼 STPA를 작성하는 과정에서 활용하였다.

Safety testing은 safety verification 중 동적 분석 (dynamic analysis)의 한 방법으로, safety verification에는 이외에도 정적 분석 (static analysis)인 정형 검증 (formal verification)과 SW 수준에서 수행하는 FTA (SFTA) 등이 있다 **[8]**. HA는 safety verification을 수행할 때 시스템의 hazard와 safety-critical한 구성요소 또는 변수 등에 대한 기본적인 내용들을 제공해줄 수 있기 때문에, 사전에 수행되는 것이 권장된다. 또한, safety verification은 HA의 후속 활동이므로, 테스팅의 형태가 아니더라도 safety-critical 시스

템의 안전성의 검증은 시스템 검증 단계에서 수행되어야 한다. **[11]**에서는 STPA 수행 이후에 결과물인 loss scenario를 활용해 TC를 생성하고 테스트 계획을 수립할 수 있다고 설명한다. 본 논문에서는 검증 단계에서 시스템의 안전성을 검증하기 위해 활용 가능한 safety TC를 생성하는 방법을 제안하고자 한다.

4. STPA를 활용한 안전성 테스트 케이스 생성 프로세스

본 연구에서 제안하고자 하는 주된 내용은 'STPA를 활용하여 safety TC를 생성하는 프로세스'이다. 이때 1장에서 언급한 것처럼 QAS의 형태로 safety TC를 생성하며, 이를 위하여 STPA 결과를 활용할 수 있는 safety general scenario를 제시한다. 협업 CPS의 분석에서 STPA를 사용하는 이유는, STPA에서는 기존의 다른 HA 기법과는 달리 컴포넌트 간의 상호작용으로 인해 발생 가능한 hazard를 포함하기 때문이다. 이것을 협업 시스템 수준에서 적용한다면, STPA를 통해 시스템 간의 협업 시에 발생하는 상호작용을 다룰 수 있다.

제안하고자 하는 프로세스에 대한 기본 아이디어, 즉 STPA의 결과를 활용하고 QAS를 작성하여 안전성을 확보하는 데에 활용하는 것은 저자의 학위논문 **[28]**에서 처음 제시하였다. 해당 논문에서는 STPA의 결과를 활용하여 QAS를 도출하면, 이를 협업 CPS의 요구사항이나 설계 명세 등을 안전성 측면에서 보완하기 위해 활용할 수 있다고 주장하였다. 이를 위해 먼저 협업하는 각각의 CPS를 전체 CPS의 구성 시스템으로 간주하여 각 구성 시스템들에 대해 별도로 STPA를 수행하고, 이후 협업으로 인해 추가되는 내용들을 고려한 전체 CPS에 대한 STPA를 추가로 진행하였다. QAS 식별을 위한 프로세스도 제안하였는데, 당시에는 단순히 STPA의 결과물인 UCA와 loss scenario를 어떻게 활용하여 QAS를 식별할 수 있는지에 대한 과정을 언급하였다. 또한, QAS의 각 항목으로 선택할 수 있는 STPA 결과 값들을 제안하고, 제안한 값들을 활용하여 QAS를 식별하는 예시를 보였다. 이때 QAS의 식별을 돕기 위한 guideword도 함께 제시하였는데, 당시에 제시한 guideword는 'without failure, 'until (operation), 'within (time)', 'after (operation)'의 네 가지로, QAS의 response measure로 활용할 수 있다고 설명하였다. 이후 추가 연구를 진행함으로써 기존의 아이디어를 발전시키고 제안하는 프로세스를 다듬어 본 논문을 통해 제안한다. 제안하는 프로세스를 통해 STPA의 결과물과 제공되는 guideword를 활용하여 QAS의 형태로 safety TC를 생성할 수 있으며, 생성한 TC를 통해 safety testing을 진행할 수 있다.

본 논문에서 제안하는 Safety TC를 생성하는 프로세스는 다음과 같다. 먼저, TC 생성에 앞서 일반적인 STPA 프로세스를 따라 대상 시스템을 분석한다. 이후, STPA 결과물을 활용하여 safety TC를 생성하는 프로세스를 따른다. Safety TC를 생성하는 프로세스는 2.1절에서 언급된 STPA의 프로세스와 함께 다음의 <그림 1>에서 확인할 수 있다.

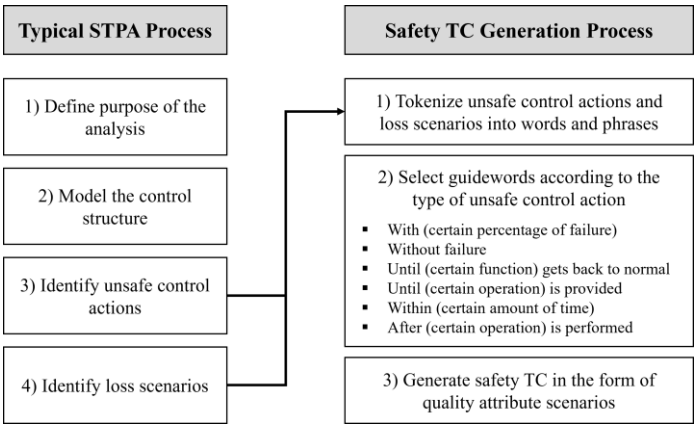


그림 1 STPA를 활용한 안전성 테스트 케이스 생성 프로세스

먼저, STPA의 결과물인 UCA와 loss scenario를 필요에 따라 단어 혹은 절의 형태로 토큰화 (tokenize)한다. 이후, <그림 1>에 나타낸 guideword 중 적절한 것을 선택한다. 선택한 guideword는 response measure로 활용된다. 마지막으로, 토큰화한 STPA 결과물과 선택한 guideword를 활용하여 safety TC를 생성한다. 제안하는 프로세스를 따라 생성한 safety TC는 STPA의 결과로써 도출된 각각의 loss scenario에 대응되어야 하고, 그 형태는 QAS의 형태를 따른다.

<그림 1> 우측의 Safety TC Generation Process의 두 번째 단계에서 제시된 guideword들은 앞서 설명한 **[28]**에서 제시하였던 guideword에서 추가 연구를 거쳐 작성되었으며, 추후 연구를 통해 추가될 수 있다. STPA 결과를 활용하여 QAS의 형태로 safety TC를 생성할 때 사용 가능한 값, 즉 safety general scenario는 <표 1>에서 확인할 수 있다.

5. 사례 연구

5.1 테스트베드 시스템

본 논문에서 제안한 approach를 적용하는 대상은 기존에 개발한 테스트베드 시스템으로, 간단한 지능형 도로교통시스템 (Intelligent Transportation System) **[29]**의 예시로써 활용된다. 해당 시스템은 내부에 직교하는 도로를 포함하여 총 다섯 개의 교차로를 가지는 루프 형태의

표 1 STPA 결과를 활용하여 QAS의 형태로 safety TC를 생성할 때 활용 가능한 safety general scenario

항목	사용 가능한 값
Source	▪ Operator ▪ Sensor ▪ I/O device ▪ Controller ▪ (internal/external) Data source
Stimulus	Source가 제공하는 데이터 혹은 제어명령
Environment	([10]에서 제공하는 safety general scenario) ▪ In normal operation ▪ In degraded operation ▪ In manual operation ▪ In recovery mode
Artifact	▪ Controller ▪ Controlled process ▪ I/O device
Response	토큰화한 STPA 결과 중 동작에 대한 부분의 내용에 반대되는 내용 또는 아래에 기술된 것처럼 [10]에서 제공하는 safety general scenario의 response 항목에 대한 값 중에서 선택한다. Unsafe state를 인식하고 다음 중 하나 이상을 따른다: ▪ Avoid the unsafe state ▪ Recover ▪ Continue in degraded or safe mode ▪ Shut down ▪ Switch to manual operation ▪ Switch to a backup system ▪ Notify appropriate entities (people or systems) ▪ Log the unsafe state (and the response to it)
Response measure	식별하는 사람의 판단에 따른 적절한 guideword 사용

가상의 도로 상에서 동작하며, 내부에는 통신 모듈을 통해 동작하는 두 종류의 간단한 CPS가 협업하여 '도로교통시스템을 외부 환경에 맞춰 적응형으로 제어한다'는 공동의 목표를 달성하기 위해 존재한다. 각 시스템의 이름은 '교통정보시스템'과 '신호체계관제시스템'으로 설정하였다. 테스트베드 시스템이 동작하는 가상의 도로 환경을 포함하는 전체 시스템의 모습은 <그림 2>를 통해 나타냈다.

교통정보시스템은 각 교차로에 배치되어 도로의 환경

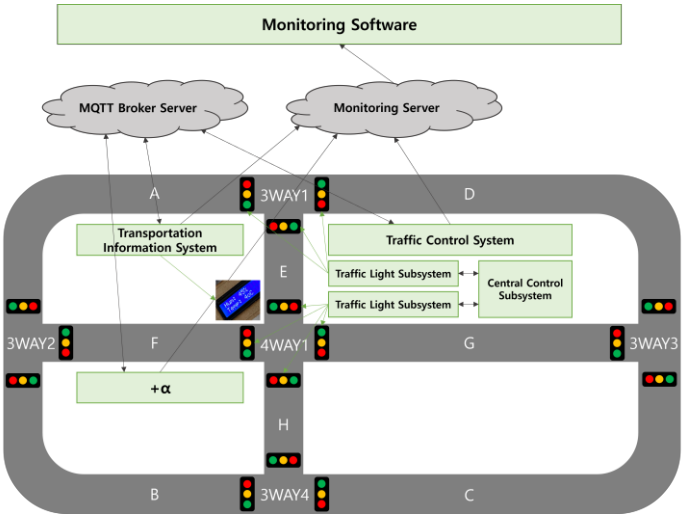


그림 2 테스트베드 시스템의 구조

정보를 수집하고, 도로 상에 화재가 발생하거나 도로가 침수되는 등의 긴급 상황이 발생하면 이를 알린다. 이외에도 우회 도로를 안내하거나, 정체 상황을 알릴 수 있고, 긴급 상황이 발생하지 않으면 날씨 정보를 디스플레이 모듈에 출력한다. 총 5개의 서로 다른 인스턴스가 각 교차로에 배치되며, 개별 시스템처럼 상호작용하며 동작한다.

신호체계관제시스템은 두 종류의 서브시스템으로 구성되어 있으며, 각각은 중앙통제 서브시스템과 신호등 서브시스템으로 설정하였다. 중앙통제 서브시스템은 신호등 서브시스템을 필요에 따라 제어하기 위한 시스템으로, 관리자의 입력 혹은 외부 시스템의 입력에 따라 신호등 서브시스템을 제어한다. 추후 다른 시스템이 테스트베드 시스템에 편입될 수 있으나, 여기서의 외부 시스템은 현재까지는 교통정보시스템만 해당된다. 신호등 서브시스템은 중앙통제 서브시스템으로부터 별도의 명령이 입력되지 않는 경우 내부 규칙을 따라 신호등의 점등 및 소등을 제어하여 교통 흐름을 통제한다. 신호등 서브시스템은 교통정보시스템과 마찬가지로 가상의 도로의 서로 다른 5개 교차로에 배치된다. 각 교차로마다 서로 다른 인스턴스가 배치되므로, 동일한 요구사항을 가지는 시스템이지만 서로 다른 시스템처럼 개별적으로 중앙통제 서브시스템과 상호작용하며 동작한다.

각 시스템은 아두이노 보드를 중심으로 통신 모듈, 센서 모듈, 디스플레이 모듈 등 동작에 필요한 모듈이 물리적으로 연결되어 있으며, 아두이노 보드에 탑재된 SW 컨트롤러를 통해 각 모듈을 제어한다. 이외에도 두 시스템의 정보 교환을 위한 MQTT (Message Queuing Telemetry Transport) 프로토콜 [30]을 따르는 브로커 서버를 포함하며, 관리자의 모니터링을 위한 모니터링 서버와 SW로 구성된 모니터링 시스템도 추가로 구현되어 있다.

해당 시스템은 [28]에서도 사례 연구 대상으로 활용하

였으며, 당시의 연구 결과를 활용하여 작성되었으므로 본 논문에서 사용하는 시스템의 명칭이나 내부 구성에 있어 다소 차이는 있으나, 본질적으로는 동일한 목적을 가진 시스템이다. 단, 다음 5.2절에서 후술되는 STPA의 적용이나 safety TC의 생성에 관한 내용에는 차이가 있다.

5.2 안전성 테스트 케이스의 생성

제안한 프로세스를 테스트베드 시스템에 적용하기에 앞서, STPA 프로세스를 따라 테스트베드 시스템을 분석하였다. STPA를 수행하여 도출된 결과인 UCA 중 일부는 다음과 같다. 예시로 나타낸 UCA는 관리자의 개입에 의한 시스템의 동작과 서로 다른 종류의 두 CPS의 협업을 위한 상호작용을 확인할 수 있는 내용을 포함한다. 즉, 서로 다른 CPS 간의 협업을 고려하지 않았다면 드러나지 않았을 UCA를 다루고 있다.

- UCA-13: 관리자가 교통정보시스템이 MQTT 서버를 통해 긴급 정보를 제공하지 않는 경우에 중앙통제 서브시스템에 수동 제어 명령을 제공하지 않는다.
- UCA-19: 중앙통제 서브시스템의 SW 컨트롤러가 교통정보시스템으로부터 MQTT 서버를 통해 긴급 정보가 제공된 경우에도 RF (Radio Frequency) 모듈에 제어 명령을 제공하지 않는다.
- UCA-30: 신호등 서브시스템의 SW 컨트롤러가 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받은 경우에도 수동 신호등 제어 명령을 제공하지 않는다.

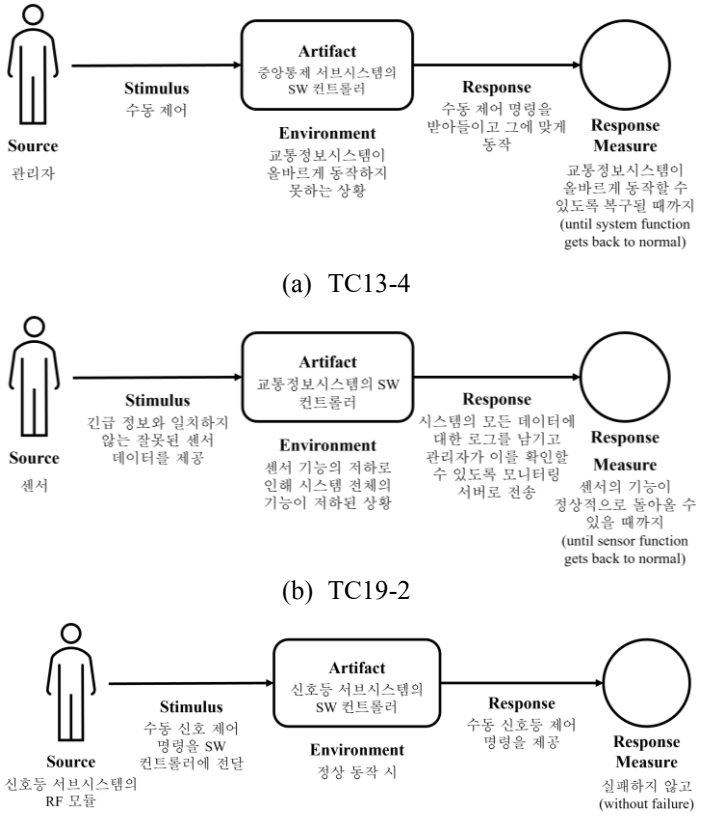
위 예시 중 UCA-13은 관리자가 개입하여 시스템을 제어하는 상황을 다루고 있다. 관련된 hazard는 ‘WiFi 통신 오류’와 ‘제어 명령의 누락’이며, 이 hazard는 인명 피해, 도로 인프라 및 시스템 인프라의 손실 또는 손상이라는 loss를 초래할 수 있다. UCA-19는 교통정보시스템과 신호체계관제시스템의 중앙통제 서브시스템 사이의 상호작용을 다루고 있으며, 이와 관련된 hazard는 ‘시스템의 제어권 상실’이다. UCA-30은 신호체계관제시스템 내부의 두 서브시스템의 상호작용을 다루고 있으며, 이에 관련된 hazard는 ‘RF 통신 오류’와 ‘신호체계 오작동’이다. UCA-19와 UCA-30에 관련된 세 가지 hazard도 위와 동일한 loss를 초래할 수 있다. 각각에 관련된 loss scenario 중 일부는 다음과 같다.

- LS13-4: 교통정보시스템이 올바르게 동작하지 못하는 경우에도, 중앙통제 서브시스템의 SW 컨트롤러가 관리자로부터 올바른 수동 제어 명령을 입력 받았으나 이를 처리하여 전달하는 동작을 수행하지 못하였다.
- LS19-2: 도로 상에 긴급 상황이 발생했음에도 해당 상황에 대한 데이터를 제공받지 못했기 때문에 교통정보시스템이 긴급 정보를 제공하지 않아서 중앙통제 서브시스템이 잘못된 process model 값을 가지게 되었다. 긴급 상황에 대한 데이터를 교통정보시스템이 제공받지 못한 것은 센서 기능의 저하로 인해 정상적인 데이터를 수집하지 못했기 때문이다.
- LS30-1: 신호등 서브시스템의 SW 컨트롤러가 잘못된 control algorithm을 가지고 있어서 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받았으나 수동 신호등 제어 명령을 제공하는 데에 실패하였다.

위의 loss scenario는 STPA handbook **[11]**에 제시된 causal factor의 예시를 따라 분석되었다. 이 중 LS19-2의 경우 교통정보시스템으로부터 잘못된 정보를 제공받아 중앙통제 서브시스템이 잘못된 process model을 가지게 되었기 때문에 UCA-19가 발생했다고 분석하였다. 따라서 LS19-2는 교통정보시스템에 발생한 문제를 중점적으로 다룬다. 경우에 따라서는 동일한 loss scenario 유형에서도 여러 가지 causal factor가 발견될 수 있으므로, 위의 넘버링은 하나의 UCA에 대해 여러 가지 loss scenario가 도출될 수 있음을 보이는 수단 정도로 이해해야 한다. 제안한 프로세스를 따라 위의 loss scenario에 대한 각 예시에 대응하는 QAS 형태의 safety TC를 생성한 예시는 다음과 같다.

- TC13-4: 관리자는 교통정보시스템이 올바르게 동작하지 못하는 경우 중앙통제 서브시스템을 수동 제어한다. 그러면 중앙통제 서브시스템의 SW 컨트롤러는 교통정보시스템이 올바르게 동작할 수 있도록 복구될 때까지 (until system function gets back to normal) 관리자가 제공한 수동 제어 명령을 받아들이고 그에 맞게 동작해야 한다.
- TC19-2: 센서 기능의 저하로 인해 시스템 전체의 기능이 저하된 상황에서 센서는 긴급 정보와 일치하지 않는 잘못된 센서 데이터를 제공한다. 그러면 교통정보시스템의 SW 컨트롤러는 센서의 기능이 정상적으로 돌아올 수 있을 때까지 (until sensor function gets back to normal) 시스템의 모든 데이터에 대한 로그를 남기고 관리자가 이를 확인할 수 있도록 모니터링 서버로 전송해야 한다.
- TC30-1: 정상 동작 시에, 신호등 서브시스템의 RF 모듈은 수동 신호 제어 명령을 SW 컨트롤러에 전달한다. 그러면 신호등 서브시스템의 SW 컨트롤러는 실패하지 않고 (without failure) 수동 신호등 제어 명령을 제공해야 한다.

각 safety TC의 구성을 살펴보면 <표 1>에 나타낸 각 항목에 대해 '사용 가능한 값'이 활용되었음을 알 수 있다. 이를 직관적으로 확인하기 위해 QAS를 나타내기 위한 다이어그램을 활용하면 <그림 3>과 같이 나타낼 수 있다. TC13-4는 (a)로, TC19-2는 (b)로, TC30-1은 (c)로 나타냈다.



(a) TC13-4

(b) TC19-2

(c) TC30-1

그림 3 QAS의 형태로 나타낸 safety TC의 예시

5.3 평가 및 논의

테스트베드 시스템에 제안한 프로세스를 적용하여 총 43 개의 UCA, 110 개의 loss scenario를 식별하고, 각 loss scenario에 상응하는 safety TC를 110 개 생성하였다. 생성한 safety TC를 활용해 테스팅을 진행한 결과, 테스트베드 시스템은 생성한 모든 safety TC를 통과하였다. 단, STPA 결과에 의존하여 safety TC를 생성하고 테스팅을 수행하였으므로, STPA의 수행 정도에 따라 식별되는 UCA, loss scenario, 그리고 그에 따른 safety TC의 개수와 통과율이 달라질 수 있음을 인지해야 한다. 그리고 만일 safety TC를 통과하지 못했다면, 어떤 원인으로 인해 통과하지 못했는지를 분석하고 해당 원인이 이후 시스템의 안전성을 위협할 수 있다고 판단될 경우 시스템의 요구사항을 수정하고 이에 맞게 시스템을 개선하는 과정을 거치는 것이 필요하다.

제안한 프로세스의 장점은 다음과 같다. 첫 번째는, 체계적인 HA 기법으로 알려진 STPA의 결과를 활용하고 안전성과 같은 시스템의 Quality Attribute에 대한 요구사항을 체계적으로 기술하기 위한 QAS의 형태를 빌려 safety TC를 생성하기 때문에 보다 체계적인 접근이 가능하다는 점이다. 두 번째는, STPA를 활용하기 때문에 loss scenario에 포함된 causal factor를 추후 시스템에 행해져야 할 점검에 대한 항목으로 고려할 수 있다는 점이다. 예를 들어, CPS를 제어하는 SW 컨트롤러가 올바르게 동작하는지에 대한 TC를 생성했을 때, 이에 관련된 loss scenario의 causal factor가 SW 컨트롤러가 잘못된 control algorithm을 가지고 있는 경우를 나타낼 수 있다. 만일 해당 TC를 통과하지 못했다면, control algorithm이 잘못되었는지 점검하는 등의 조치를 취할 수 있다. 마지막으로 여러 표준 **[5, 6]**에서 강조하는 기능 안전성을 포함하여 제어 관점에서의 안전성 (control safety), 물리적 동작의 안전성 (physical safety) 등 안전성의 다양한 측면 **[31]**을 고려할 수 있다는 장점이 있다.

단, 본 논문에서 제안한 프로세스의 가장 주요한 한계점은, 제안한 프로세스를 통해서 생성하는 safety TC가 다소 전형적일 수 있다는 것이다. 다르게 말하자면, 매우 드물게 발생하는 실패 (failure)의 경우에는 다루지 못할 수 있다는 것이다. 제안한 프로세스 외적으로는, 사례 연구를 진행하는 테스트베드 시스템의 상호작용의 형태와 그 동작이 단순하다는 한계점이 있다. 또한, 테스트베드 시스템이 지니는 특성 중 동일한 시스템의 여러 인스턴스가 존재한다는 특성에 대하여 다루지 못했다는 한계점이 있다.

단, TC가 전형적이라고 해서 그 TC가 의미 없는 TC가 되는 것은 아니다. 의미 없어 보이는 TC라고 해도, 해당 TC가 다루는 부분에 대한 테스팅을 진행하고 통과 여부를 확인하는 것은 필요한 과정이다. 또한, 테스팅을 통해서 모든 경우를 완벽하게 다루는 것은 거의 불가능에 가깝다. 이것은 테스팅의 근본적인 한계점이므로, 더 철저한 검증이 필요하다면 테스팅 이외의 검증의 수단을 추가로 마련하여 이를 보완할 수 있도록 하는 것이 필요하다.

6. 결론 및 향후 연구

본 논문에서는 STPA 수행 결과를 활용하여 QAS의 형태로 시스템의 안전성을 테스트하기 위한 safety TC를 식별하는 방법을 제안하고 사례 연구를 수행하였다. 사례 연구는 기존에 개발한 테스트베드 시스템을 대상으로 수행되었으며, 제안한 approach의 적용을 통하여 safety TC를 식별하고, safety testing까지 수행하여 결과를 살펴보았다. 또한, Safety testing의 수행 결과를 통해 제안한

approach의 적용 효과를 살펴보았다.

추후 사례 연구를 진행하는 테스트베드 시스템의 상호작용에 대한 복잡도를 증가시키고 기능적 단순함을 극복하기 위하여 기존의 테스트베드 시스템에 존재하는 CPS들과는 다른, 새로운 시스템을 추가할 계획이다. 또한, 테스트베드 시스템처럼 한 시스템의 여러 인스턴스가 존재하고 상호작용하는 경우에 대하여 다룰 수 있도록 제안한 프로세스를 발전시키고자 한다. 이를 통해 앞서 제시한 guideword 이외에 추가적인 guideword를 제시하는 데에도 도움이 될 것이라고 예상된다. 뿐만 아니라, STPA 이외에 다른 HA 기법의 추가적인 적용 또한 고려 중이다.

Acknowledgment

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2021R1F1A1047246). 또한 저자의 석사학위 논문 **[28]**에 기반하였음.

참고문헌

[1] R. Baheti, H. Gill, “Cyber-Physical Systems,” The Impact of Control Technology, vol. 12, no. 1, pp. 161-166, 2011.

[2] L. Zhang, Y. P. Fallah, R. Jihene, “Cyber-Physical Systems: Computation, Communication, and Control,” International Journal of Distributed Sensor Networks, vol. 9, no. 2, 2013.

[3] P. B. Checkland, “Systems Thinking, Systems Practice,” John Wiley, 1999.

[4] Department of Defense (DOD), MIL-STD-882E: System Safety, 2012.

[5] International Electrotechnical Commission (IEC), IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, 2010.

[6] International Organization of Standardization (ISO), ISO 26262: Road Vehicles – Functional Safety, 2011.

[7] Radio Technical Commission for Aeronautics (RTCA), DO-178C: Software Considerations in Airborne Systems and Equipment Certification, 1992.

[8] N. G. Leveson, “Safeware: System Safety and Computers,” Addison-Wesley, 1995.

[9] N. G. Leveson, “Engineering a Safer World: Systems Thinking Applied to Safety,” MIT Press, 2016.

[10] L. Bass, P. Clements, R. Kazman, “Software Architecture in Practice,” Fourth Edition, Addison-Wesley Professional, 2021.

[11] N. G. Leveson, J. P. Thomas, “STPA Handbook,” 2018, [Online]. Available: http://psas.scripts.mit.edu/home/materials/

[12] N. Rozanski, E. Woods, “Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives,” Second Edition, Addison-Wesley Professional, 2011.

[13] International Organization for Standardization (ISO), ISO/IEC 25010: Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models, 2011.

[14] N. Ali, M. Hussain, J-E. Hong, “Analyzing Safety of Collaborative Cyber-Physical Systems Considering Variability,” IEEE Access, vol. 8, 2020.

[15] M. Cepin, “Event Tree Analysis,” Assessment of Power System Reliability: Methods and Applications, pp. 89-99, 2011.

[16] C. A. Ericson II, “Fault Tree Analysis,” System Safety Conference, vol. 1, pp. 1-9, 1999.

[17] D. J. Reifer, “Software Failure Modes and Effects Analysis,” IEEE Transactions on reliability, vol. R-28, no. 3, pp. 247–249, 1979.

[18] International Organization of Standardization (ISO), ISO 21448: Road vehicles – Safety of the Intended Functionality, 2019.

[19] S. Medawar, D. Scholle, I. Sljivo, “Cooperative Safety Critical CPS Platooning in SafeCOP,” 2017 6th Mediterranean Conference on Embedded Computing (MECO), pp. 1-5, 2017.

[20] P. Pop, et al., “The SafeCOP ECSEL Project: Safe Cooperating Cyber-Physical Systems Using Wireless Communication,” 2016 Euromicro Conference on Digital System Design (DSD), pp. 532-538, 2016.

[21] T. Kelly, R. Weaver, “The Goal Structuring Notation – A Safety Argument Notation,” Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, vol. 6, 2004.

[22] R. L. Ackoff, “Towards a System of Systems Concepts,” Management Science, vol. 17, no. 11, pp. 661-671, 1971.

[23] S. Baumgart, J. Fröberg, S. Punnekkat, “Analyzing Hazards in System-of-Systems: Described in a Quarry Site Automation Context,” 2017 Annual IEEE International Systems Conference (SysCon), pp. 1-8, 2017.

[24] F. U. Muram, M. A. Javed, S. Punnekkat, “System of Systems Hazard Analysis Using HAZOP and FTA for Advanced Quarry Production,” 2019 4th International Conference on System Reliability and Safety (ICSRS), pp. 394-401, 2019.

[25] T. C. McKelvey, “How to Improve the Effectiveness of Hazard and Operability Analysis,” IEEE Transactions on Reliability, vol. 37, no. 2, pp. 167-170, 1988.

[26] H-D. Tran, et al., “Safety Verification of Cyber-Physical Systems with Reinforcement Learning Control,” ACM Transactions on Embedded Computing Systems (TECS), vol. 18, no. 5s, pp. 1-22, 2019.

[27] B. Rokseth, O. I. Haugen, I. B. Utne, “Safety Verification

for Autonomous Ships," MATEC Web of Conferences, vol. 273, pp. 02002, 2019.
[28] Yoona Heo, "A Study on Identification of Quality Attribute Scenario for Safety of Cooperating Cyber-Physical Systems Using Systems-Theoretic Process Analysis," Master's thesis, Konkuk University, 2023.
[29] G. Dimitrakopoulos, P. Demestichas, "Intelligent Transportation Systems," IEEE Vehicular Technology Magazine, vol. 5, no. 1, pp. 77-84, 2010.
[30] R. A. Light, "Mosquitto: Server and Client Implementation of the MQTT Protocol," Journal of Open Source Software, vol. 2, no. 13, pp. 265, 2017.
[31] X. Lyu, Y. Ding, S-H. Yang, "Safety and Security Risk Assessment in Cyber-Physical Systems," IET Cyber-Physical Systems: Theory & Applications, vol. 4, no. 3, pp. 221-232, 2019.

하이브리드 안전 분석을 통한 시각 기반 자율 주행 시스템의 적대적 견고성 향상

후세인 만주루*, 이브라힘 아메드, 김경민, 상정위, 홍장의

충북대학교 소프트웨어지능공학연구실

Improving Adversarial Robustness of Vision-based Autonomous Driving Systems Through Hybrid Safety Analysis

Manzoor Hussain*, Ahmed D. M. Ibrahum, Kyeong-Min Kim, Zhengyu Shang, and Jang-Eui Hong

Software Intelligence Engineering Lab, Department of Computer Science

Chungbuk National University, Cheongju, Republic of Korea

Email: {manzoorhussain, Ahmed_ibrahum, jetilous, shangzhengyu, jehong}@chungbuk.ac.kr

Abstract

In safety-critical systems, particularly Autonomous Vehicles (AVs), concern is rising over the potential consequences stemming from erroneous decisions made by Deep Learning (DL) models due to adversarial attacks. The inherent inaccuracies in these models pose a substantial threat to human well-being, with the potential for severe outcomes such as injuries or fatalities. Relying only on AI (i.e., ensemble modeling for robust prediction) or traditional safety analysis techniques such as Fault Tree Analysis (FTA) is insufficient for safety critical systems like AVs. Therefore, this study proposed a hybrid safety analysis technique to improve the adversarial robustness of AV by combining two safety analysis methods: the FTA and simulation-based safety analysis technique. In our approach, we first perform FTA to meticulously examine the probability of failure resulting from adversarial attacks on the perception system. Secondly, a simulation-based safety analysis technique is proposed to validate and verify the outcome of safety analysis using a high-fidelity self-driving car simulator. Thirdly, we validated and verified the effectiveness of the proposed hybrid safety analysis by presenting a case study on an end-to-end Autonomous Driving Systems (ADSs) model. Finally, by incorporating the outcomes of hybrid safety analysis and adversarial training, we improved the adversarial robustness of the target ADSs model. The experiment showed that the proposed hybrid safety analysis technique improved the adversarial robustness of the target model after adversarial training. This research endeavors to enhance the resilience of AV systems by analyzing vulnerabilities introduced by adversarial attacks, ultimately contributing to the safety and reliability of autonomous driving technology.

Keywords: Autonomous Vehicles, Robustness, Safety Analysis, Fault Tree Analysis, Adversarial Attacks

1. Introduction

The integration of autonomous vehicles (AVs) into modern transportation systems underscores the need for a comprehensive assessment of their reliability and susceptibility to adversarial attacks. The increasing reliance on deep learning models in safety-critical applications, as initially highlighted in studies by Szegedy et al. [1], raises concerns about potential erroneous decisions and vulnerabilities to adversarial manipulations. This study proposed a hybrid approach to address these challenges to improve the adversarial robustness of deep learning models used in AVs. The proposed approach integrates two safety analysis methods: Fault Tree Analysis (FTA) and simulation-based safety analysis.

Our assumption is based on the hypothesis that safety analysis of safety-critical systems like AVs is insufficient by incorporating only conventional safety analysis techniques such as FTA or AI (e.g., ensemble-based robust prediction). Conventional safety analysis

techniques cannot capture the faults in the architecture of AI models due to their intrinsic complexity. Similarly, the AI-based safety analysis also does not reasonably mitigate all vulnerabilities associated with AI models. For example, when an adversary attacks a model, verifying the validity of the safety of the output produced by the model becomes challenging. Thus, it is imperative to mitigate this issue by combining both conventional and AI-based safety analysis methods, such as simulation-based safety analysis techniques, to improve the robustness of the models in AVs. By combining the conventional and AI-based safety analysis techniques (i.e., simulation-based safety analysis), the inherent limitation of conventional safety analysis techniques can be mitigated via simulation-based safety analysis techniques, and conventional techniques can minimize the limitation of AI-based safety analysis techniques.

Another reason to propose the hybrid approach for improving the robustness is that the AVs demand a systematic and integrative approach to identifying and mitigating potential hazards due to Level 3 autonomous driving (i.e., human driver and AI models both are involved in driving tasks). Therefore, safety analysis plays a pivotal role in investigating potential failure modes and recommending failure mitigation plans such as safety guards. The final outcome of our hybrid safety analysis approach is to recommend a plan for failure mitigation. Therefore, based on the outcomes of the safety analysis, our approach also recommends a failure mitigation plan, such as adversarial training, input transformation, etc., to improve adversarial robustness. However, this article chooses adversarial training as a failure mitigation plan to improve adversarial robustness. The overall approach is illustrated in Figure 1. We made the following core contributions in this article:

1. We proposed a hybrid safety analysis approach by combining FTA and Simulation-based safety analysis techniques.
2. We presented a case study on an end-to-end Autonomous Driving System (ADS) model, showing that the proposed method effectively improved adversarial robustness.
3. A detailed, comprehensive qualitative and quantitative analysis shows that the suggested mitigation plan effectively improved adversarial robustness.

The remainder of the article is organized as follows: Section 2 provides the literature review on safety analysis techniques. Section 3 describes the proposed methodology, and Section 4 presents the experimental results. Finally, Section 5 concludes this article.

2. Related Work

Various researchers have explored the application of conventional safety analysis to different domains. For example, Bein et al. [2] investigated the challenges of ensuring safety and reliability in autonomous driving systems. They used the probabilistic Failure Modes and Effects Analysis (FMEA) and employed a closed-loop simulation strategy to analyze the safety. Schönemann et al. [3] proposed an FTA-based approach for obtaining safety requirements in compliance with the ISO 26262 standard and applied to fully automated valet parking. Utilizing Hazard Analysis and Risk Assessment (HARA), their method offers a comprehensive framework for systematically deriving safety requirements from safety goals. Chen et al. [4] introduced a taxonomy of takeover scenarios for automated driving using FTA to identify potential failure. Chen et al. [5] highlighted the critical role of safety in road vehicles, particularly in the context of rapidly developing driver assistance and automated

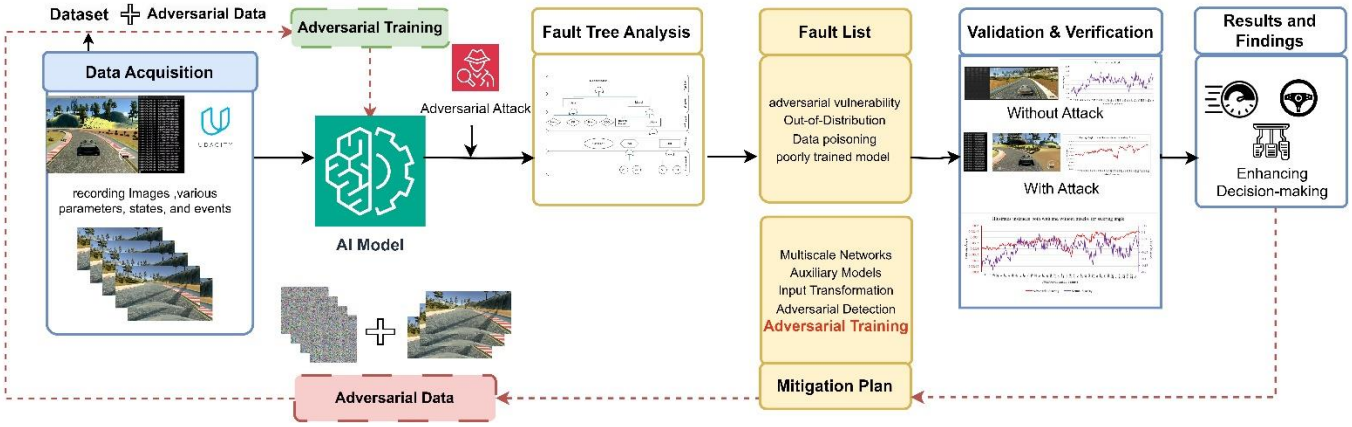


Figure 1. A hybrid safety analysis approach to improve adversarial robustness of ADSs model.

driving systems. The study uses FTA to identify safety issues related to these systems. It introduces Causal Scenario Analysis (CSA) as a methodology based on possibility theory and a fuzzy relation model. Yan et al. [6] used the failure mode effects and criticality analysis (FMECA) to assess the reliability of the automated guided vehicle system and evaluated mission reliability using the Petri net (PN) method.

Gheraibia et al. [7] tackled the challenges associated with modeling and evaluating complex safety-critical systems by proposing an innovative approach that integrates FTA, machine learning, and real-time operational data. Pedroza et al. [8] proposed a safe-by-design development process to overcome challenges associated with autonomy, correctness, and the prevention of catastrophic risks. Other researchers proposed adversarial training to mitigate the risk associated with the ADS model [9]. In contrast, some researchers used the ensemble approach for robust prediction [10].

3. Proposed Approach

When dealing with the vulnerability of deep learning models in complex systems like AVs, it becomes challenging to ensure safety using only conventional or AI-based safety analysis techniques. However, safety analysis is paramount to ensure safety in AVs as it helps to identify vulnerabilities and provide recommendations for safety guards [11]. Therefore, we present a hybrid safety analysis technique to improve AVs' robustness and safety, focusing on adversarial attacks. Our approach consists of four modules: data acquisition, FTA, simulation-based safety analysis, and mitigation plan to robustify the model against adversarial vulnerabilities. In the following, we explain each module of the proposed approach.

3.1. Data Acquisition

We collect the data under normal conditions (i.e., without adversarial attacks) to train the target model. We choose NVIDIA's AVs model as a case study in this study [12]. Unlike the modular AV model, the end-to-end models directly map the raw image pixel to control parameters such as steering and braking. We first collected the dataset under normal conditions to train the model and then validated its performance in challenging driving environments in the Udacity self-driving car simulator [13]. The AV models can be trained either via imitation learning or reinforcement learning. However, we trained the target model in this study via imitation learning. During the validation phase, we recorded the final output of the model, such as the

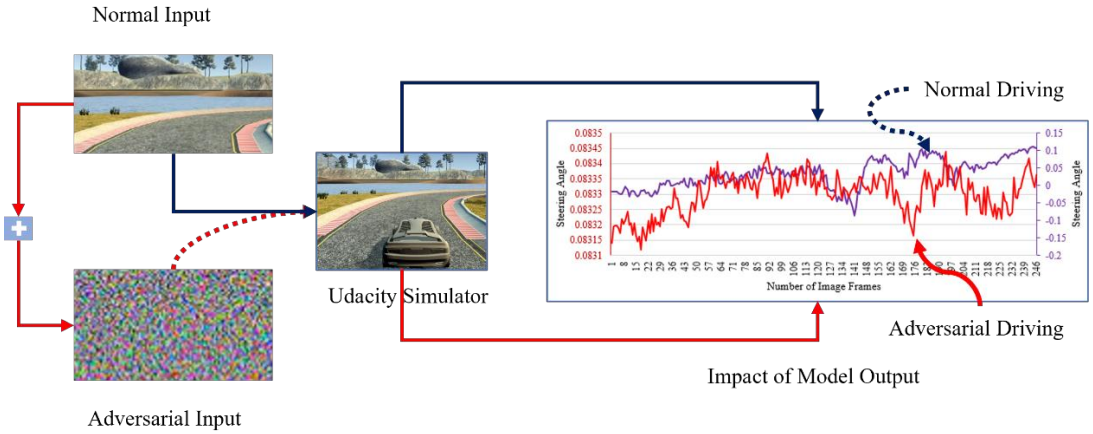


Figure 2. Simulation-based safety analysis to verify the adversarial robustness of the target model under adversarial attack.

steering angles, to measure the performance. We choose the steering angle to measure the performance as it is the final output of the model, and any external influence, such as adversarial attacks, can directly impact the steering angle. Also, it is easy to measure the deviation in the output when the AV model faces adversarial attacks or other vulnerabilities.

3.2. Fault Tree Analysis

FTA is a technique that is used to analyze and identify potential causes of systems failure [4]. It is performed before the system's deployment during the designing and planning phase to assess the various combinations of faults that could lead to system failure. Thus, our approach extensively analyzes the potential risk associated with the AV model, focusing on model vulnerabilities such as adversarial attacks. The outcome of the FTA is then fed to the simulation-based safety analysis module, which is an AI-based safety analysis technique using a high-fidelity simulator. The simulation-based safety analysis module aims to verify whether or not the potential hazards identified by the FTA are valid. Thus, each scenario is simulated in a realistic simulator and recorded in the simulation logs. For example, during the FTA, we found that adversarial attacks could significantly cause the AV collision. Thus, by simulating the adversarial attacks on the trained NVIDIA's AVs model (i.e., mentioned in Section 3.1) in the simulator, it was confirmed that such an attack causes significant deviation in steering angle, which, as a result, caused AVs collision.

3.3. Simulation-based Safety Analysis

The third step in our approach is to verify the correctness of the identified risks by FTA using the simulation-based safety analysis technique. To achieve this, we incorporated the Udacity self-driving car simulator. Since our focus is to improve the adversarial robustness of the AV model, we simulated an adversarial attack scenario in the Udacity self-driving car simulator. The target model was attacked using various Blackbox attacks, such as decision-based/boundary attacks.

Table 1. Mitigation plan for adversarial vulnerabilities.

Vulnerabilities	Method for Improving NN Robustness
Natural	Multiscale Networks [14]
	Feature Aggregating [15]
	Adversarial Logit Pairing [16]
	Runtime Out-of-Distribution Detection [17]
Adversarial	Adversarial Training [18]
	Input Transformation [19]
	Adversarial Detection [18]
	Auxiliary Models [18]

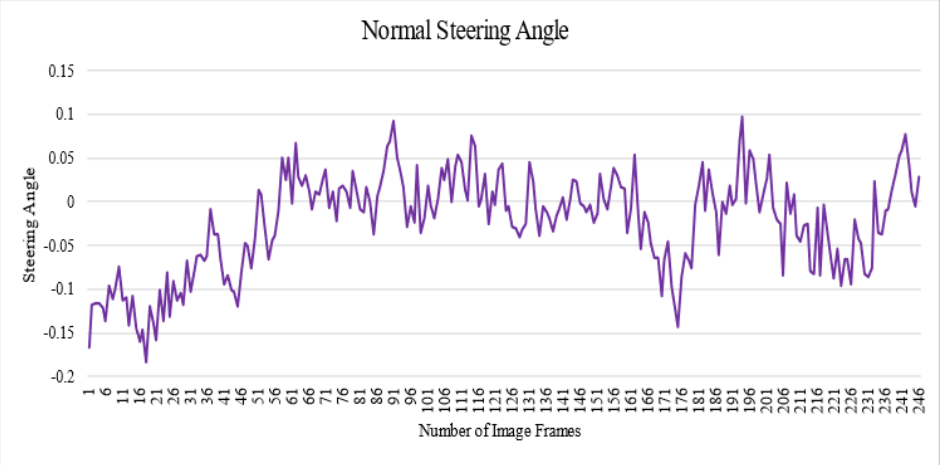


Figure 3. An illustration of Normal Steering Angle.

During the FTA, we found the adversarial attack could be a potential risk for AV collision. Therefore, we extensively simulated various adversarial attacks. Among the adversarial attacks, we found that the decision-based/boundary attack [20] strongly degraded the prediction accuracy of the AVs model. Figure 2 illustrates the simulation-based safety analysis of the AVs model.

3.4. Adversarial Training to Improve Robustness

The outcome of the hybrid safety analysis is the potential mitigation plan for each fault. These recommended mitigation plans are based on the historical data applied to mitigate specific faults. For example, in history, previous methods used redundancy approaches to mitigate the sensor and actuator faults by adding extra sensors [21]. Similarly, various researchers used ensemble approaches for robust prediction to mitigate uncertainties and inaccuracies in machine learning models [22]. Thus, we also prepared a list of potential mitigation plans for each fault based on historical data. Since we focused on improving the adversarial robustness of the target model in this article, we collected the most common adversarial attack defense methods as mitigation plans. Table 1 presents the mitigation plan for adversarial vulnerabilities.

Various researchers [23] have shown that adversarial training is the most effective and common technique to improve adversarial robustness. Therefore, our proposed approach adopted adversarial training as a mitigation plan to minimize the effect of adversarial attacks.

4. Experimental Results

4.1. Case Study-Safety Analysis of End-to-end AV Model

We employed a self-driving car as a case study that uses NVIDIA's end-to-end model to validate the proposed approach. This model maps the input image directly into the control parameters. The target model consists of nine layers, including a normalization layer and five convolutional layers. The first layer of the network performs image normalization. While on the other hand, the convolutional layers are designed to perform feature extraction. The Udacity self-driving car simulator has two modes: The training mode is used to collect datasets, while the autonomous mode is used to test the trained model. We collected the dataset and trained the model with imitation learning technique to follow the center of the lane

without collision in the first phase. The outcome of the trained model under the normal scenario is presented in Figure 3.

4.2. Safety Analysis using FTA.

Since FTA is the most commonly used technique to identify potential hazards associated with safety-critical systems, we also performed the safety analysis by incorporating the FTA to identify potential risks related to AVs' potential collision, focusing more on adversaries and vulnerabilities. The FTA emphasizes exploring the potential adversaries and vulnerabilities associated with AV models. Figure 4 illustrates the FTA of AVs model collision. This FTA aims to investigate the potential faults responsible for the "*ADS Collision*" event. The proposed FTA illustrates the potential risks that may cause the final outcome, i.e., "*ADS Collision,*" with more focus on adversarial vulnerabilities.

From Figure 4, let us analyze the potential risk associated with the top event, i.e., "ADS Collision." Based on the historical data and domain knowledge, we found that the common potential causes of "*ADS Collision*" are *software failure, hardware failure, sensor failure, human factors, and ADS model failure*. Other than the "*ADS Model Failure*," all other faults can be mitigated using conventional techniques such as redundancy or graceful degradation techniques. However, considering the intermediate event *"ADS Model Failure*," we cannot directly apply these conventional techniques to mitigate the risk associated with this intermediate event. Thus, we need other

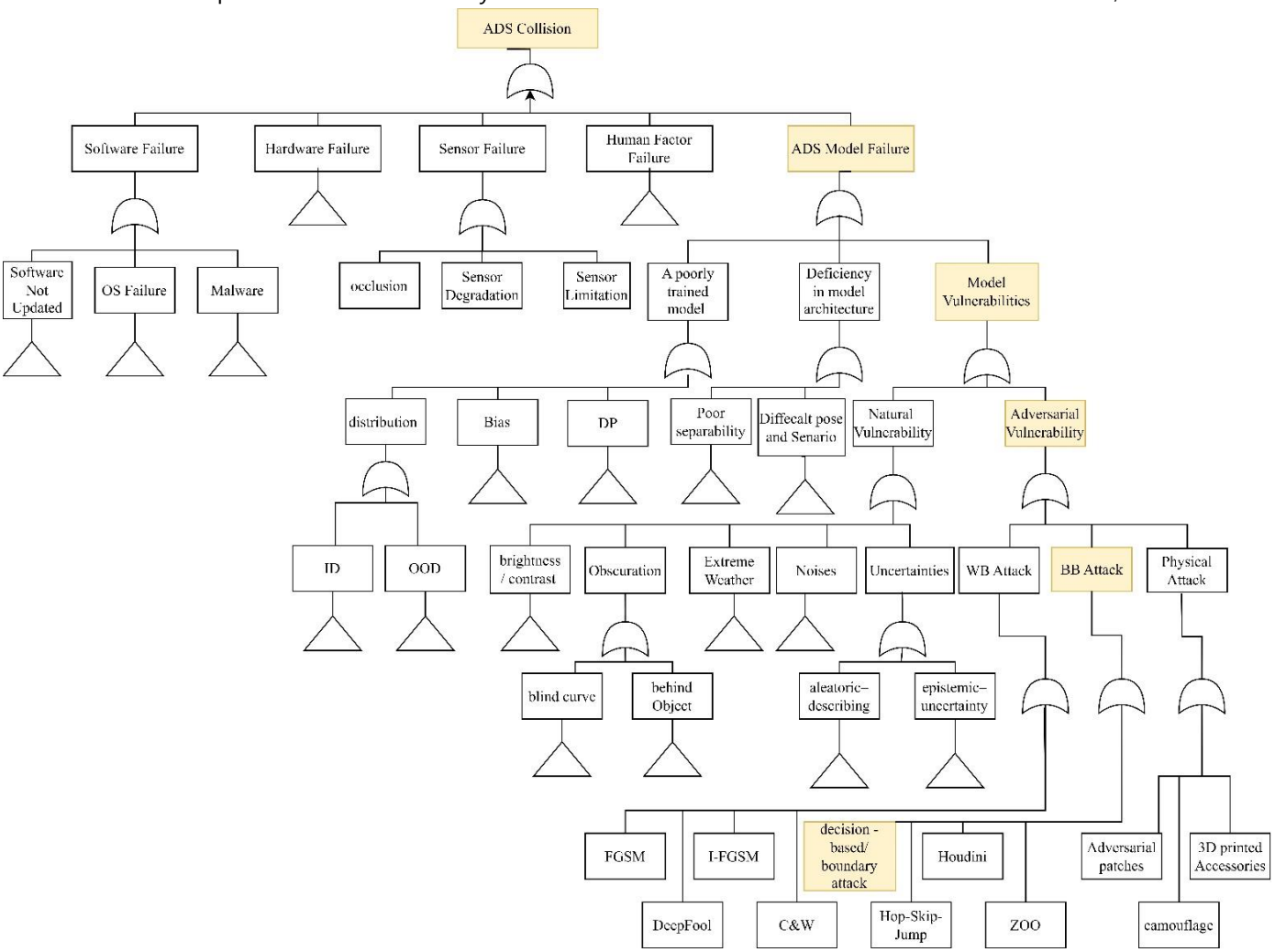


Figure 4. Fault Tree Analysis of ADS Collision event under adversarial attack.

techniques, such as adversarial training or an ensemble approach. However, we need to determine the root cause of this specific fault. Upon deeper investigation and historical events, we found that even sophisticated ADS models are vulnerable to adversarial attacks. Thus, we further investigated various adversarial attacks such as white box attacks, black box attacks, and physical attacks. However, studies have shown that the white box attacks are hard to implement in the ADSs domain, as accessing the models at runtime is almost impossible. Therefore, we further investigated other types, such as the black box and physical (i.e., patch attacks). Among these two attacks, black box attacks are easy to implement, and the attacker does not necessarily need access to the target models. Consequently, our investigation found that black box-type attacks are more hazardous than other types. We then simulated the different types of black box attacks, such as decision-based/boundary attacks, Hop-Skip-jump attacks, etc. During the simulation, the decision-based/boundary attacks were found to be more impactful and caused a serious deviation in the final output of the target model.

4.3. Simulation-based Safety Analysis

The FTA can only describe the potential vulnerabilities that might cause the ADS collision event. However, it can not verify the actual impact of the basic event on the top event. For example, one of the root causes of the ADS Collision was potential Adversarial attacks by malicious attackers. Upon deeper investigation, the FTA (i.e., Figure 4) suggests that one possible root cause of the ADSs Collision may be a "Decision-based/boundary Attack," a black box-type adversarial attack. FTA can only provide information about the potential fault; however, it cannot guarantee that the discovered fault is the actual cause of the "*ADS Collision*" event. To verify it, we need to simulate the various black box attacks, including "*Decision-based/ Boundary Attacks,*" using the Udacity Self-driving car simulator with the experimental setup illustrated in Figure 2. In this setting, we deployed the trained target model in the autonomous mode in the Udacity simulator and attacked it with the "*Decision/based Boundary attack*." We recorded the model outputs under the *Decision-based/Boundary attack*. Surprisingly, we found significant deviations and strong discrepancies in the final output (i.e., Steering angle). Figure 5 illustrates the impact of the decision-based/boundary attack on the target model. We can see that in the same setting, the deviation in the steering for each frame is huge. The extensive experimental results suggest a 0.108 mean deviation was recorded when the model was under *decision-based/boundary attack*. Note that the normalized output of the NVIDIA's models ranges between [-1,1]. Considering this normalized value, a mean deviation of

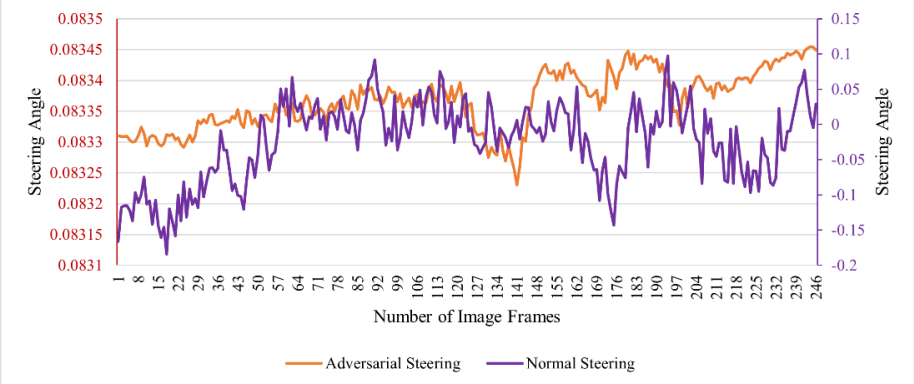


Figure 5. Impact of decision-based/boundary attack on the target model.

0.108 degrees in the steering is catastrophic for the safety of the AVs. Therefore, we found that the root cause of the top event suggested by FTA was valid and validated in the simulation environment. However, another key contribution of this article is to mitigate these issues. Therefore, following domain knowledge and the historical mitigation plan (i.e., Table 1), we also applied adversarial training as a mitigation plan to reduce the risk of the *decision-based/boundary attack* and other black box attacks. In the following, we explain the mitigation plan for adversarial attacks to improve the adversarial robustness.

4.4. Improving adversarial robustness via Adversarial Training,

Adversarial training is a technique to improve the robustness and generalizability of AI models. It is effective in defending against adversarial attacks and provides resilience to the target model. Our hybrid safety analysis method recommended the most commonly used risk mitigation plans. Among the recommended plans, we found the best and most common practice to achieve adversarial robustness is to train the target model adversarially. Therefore, data collected during the simulation-based safety analysis when the target model was under attack was used to train model mixing with the original data. Training the model with the adversarial samples generated during the adversarial attack using the decision-based/boundary attack, the model becomes resistant to these attacks. We analyzed the model's adversarial robustness by measuring the steering angle's mean deviation. Compared to the scenario, when the model was not trained with adversarial training, the mean deviation in the steering angle was recorded as less than 0.01.

On the other hand, the mean deviation in the steering angle was recorded as greater than 0.108 without adversarial training. This huge difference between the model trained with adversarial training vs. without adversarial training shows that the mitigation plan significantly improved the adversarial robustness of the target model. It also verifies the validity of the hybrid safety analysis and risk mitigation plan.

5. Conclusion

We proposed a hybrid safety analysis technique to improve the adversarial robustness of the ADSs. By combining conventional and AI-based safety analysis techniques, we addressed the limitations of both AI-based and conventional safety analysis methods. The limitation of the conventional safety analysis was tackled with AI and the lack of AI in ensuring the safety of safety-critical systems like ADSs was tackled using conventional techniques. Thus, the proposed hybrid approach improved the overall safety and robustness of ADSs. The case study on ADSs showed that our approach significantly improved adversarial robustness and overall systems safety.

In the future, we aim to combine other safety analysis techniques, such as failure mode effect analysis, with simulation-based safety analysis techniques to investigate the failure mode, causal factor, and runtime safety recommendations for safety critical systems in the ADS domain.

Acknowledgment

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (RS-2023-00237203).

Reference

[1] C. Szegedy *et al.*, "Intriguing properties of neural networks," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, Dec. 2013, [Online]. Available: http://arxiv.org/abs/1312.6199.

[2] T. Bein *et al.*, "Verification and validation of automated driving systems utilizing probabilistic FMEA and simulation approaches," *Transp. Res. Procedia*, vol. 72, pp. 470–477, Jan. 2023, doi: 10.1016/J.TRPRO.2023.11.429.

[3] H. W. Valerij Schönemann, "Fault Tree-Based Derivation of Safety Requirements for Automated Driving on the Example of Cooperative Valet Parking," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[4] K. T. Chen, H. Y. W. Chen, A. Bisantz, S. Shen, and E. Sahin, "Where Failures May Occur in Automated Driving: A Fault Tree Analysis Approach," *J. Cogn. Eng. Decis. Mak.*, 2023, doi: 10.1177/15553434221116254.

[5] M. Chen, A. Knapp, M. Pohl, and K. Dletmayer, "Taming Functional Deficiencies of Automated Driving Systems: A Methodology Framework toward Safety Validation," 2018, doi: 10.1109/IVS.2018.8500679.

[6] R. Yan, L. M. Jackson, and S. J. Dunnett, "Automated guided vehicle mission reliability modelling using a combined fault tree and Petri net approach," *Int. J. Adv. Manuf. Technol.*, vol. 92, no. 5–8, pp. 1825–1837, Sep. 2017, doi: 10.1007/S00170-017-0175-7/METRICS.

[7] Y. Gheraibia, S. Kabir, K. Aslansefat, I. Sorokos, and Y. Papadopoulos, "Safety + ai: A novel approach to update safety models using artificial intelligence," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2941566.

[8] G. Pedroza and M. Adedjouma, "Safe-by-design development method for artificial intelligent based systems," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2019, vol. 2019-July, pp. 391–397, doi: 10.18293/SEKE2019-094.

[9] A. Muhammad and S. H. Bae, "A Survey on Efficient Methods for Adversarial Robustness," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3216291.

[10] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, May 2017, [Online]. Available: http://arxiv.org/abs/1705.07204.

[11] N. Ali, M. Hussain, and J.-E. Hong, "Analyzing Safety of Collaborative Cyber-Physical Systems Considering Variability," *IEEE Access*, 2020, doi: 10.1109/access.2020.3021460.

[12] M. Bojarski *et al.*, "End to End Learning for Self-Driving Cars," 2016, Accessed: Jan. 01, 2023. [Online]. Available: http://arxiv.org/abs/1604.07316.

[13] Udacity, "A self-driving car simulator built with Unity," 2017. https://github.com/udacity/self-driving-car-sim (accessed May 16, 2021).

[14] A. Gutfraind, L. A. Meyers, and I. Safro, "Multiscale Network Generation," *2015 18th Int. Conf. Inf. Fusion, Fusion 2015*, pp. 158–165, Jul. 2012, [Online]. Available: http://arxiv.org/abs/1207.4266.

[15] S. Sun, X. Yue, X. Qi, W. Ouyang, V. Prisacariu, and P. Torr, "Aggregation with Feature Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 507–516, doi: 10.1109/ICCV48922.2021.00057.

[16] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial Logit Pairing," Mar. 2018, Accessed: Jan. 03, 2024. [Online]. Available: http://arxiv.org/abs/1803.06373.

[17] T. Belkhouja, Y. Yan, and J. R. Doppa, "Out-of-distribution Detection in Time-series Domain: A Novel Seasonal Ratio Scoring Approach," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 1, pp. 1–24, Jul. 2024, doi: 10.1145/3630633.

[18] G. R. MacHado, E. Silva, and R. R. Goldschmidt, "Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective," *ACM Computing Surveys*, vol. 55, no. 1. 2021, doi: 10.1145/3485133.

[19] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification," *IEEE Access*, vol. 10. pp. 102266–102291, 2022, doi: 10.1109/ACCESS.2022.3208131.

[20] W. Brendel, J. Rauber, and M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.04248.

[21] R. Grepl, M. Matejasko, M. Bastl, and F. Zouhar, "Design of a fault tolerant redundant control for electro mechanical drive system," 2015, doi: 10.1109/IConAC.2015.7313984.

[22] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *36th International Conference on Machine Learning, ICML 2019*, 2019, vol. 2019-June, pp. 8759–8771.

[23] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent Advances in Adversarial Training for Adversarial Robustness," in *IJCAI International Joint Conference on Artificial Intelligence*, 2021, pp. 4312–4321, doi: 10.24963/ijcai.2021/591.

드럼 용기 라벨 육안 검사 AI pipeline 설계 및 구축

구민○ 허의남
경희대학교 소프트웨어융합학과
crom9401@khu.ac.kr johnhuh@khu.ac.kr

Drum container label visual inspection AI pipeline design and implementation

요 약

전통적인 드럼 용기 라벨 육안 검사 방식의 비효율성과 불완전성을 해결하기 위해, 인공지능(AI) 기반 자동화 기술의 적용 가능성을 분석하고, A사의 Cloud 기반 어플리케이션을 활용한 육안 검사 자동화를 위한 AI pipeline 설계와 구축 방법을 연구하여 적용하고 결과를 검증하는 것이 본 연구의 목적이다.

1. 서론

드럼 용기는 의약품, 화학 물질 등을 보관하는 중요한 저장 용기로, 제품의 품질과 안전을 위해 정확한 라벨링이 필수적이다. 화학 제조 업체 S사는 다양한 제품을 여러 고객에서 공급하며, 이 과정에서 제품과 고객 요구에 맞는 정보를 담은 라벨을 부착 후 작업자에 의한 육안 검사를 통해 품질을 관리한다.

[그림 1 S사 드럼 용기 라벨 육안 검사 절차]

[그림 1]과 같이 작업자에 의한 육안 검사는 노동 집약적이며 오류 발생 가능성이 높아 품질 문제를 야기할 수 있다. 이를 해결하기 위해, AI와 컴퓨터 비전 기술을 활용한 자동화 시스템의 필요성이 커지고 있다. 또한, AI나 IT 개발 전문성이 부족한 전통적인 제조 회사에서는 클라우드 기반 어플리케이션을 적용하는 것이 효과적이며, 이러한 접근은 확장성과 비용 효율성, 실시간 분석을 제공하여 효율적인 AI 기반 라벨 검사 시스템의 개발과 검증을 촉진할 수 있을 것으로 기대한다.

본 연구에서는 작업자에 의해 수행되는 드럼 용기 라벨 육안 검사의 정확성과 효율성을 높이기 위해 A사의 클라우드 기반 어플리케이션을 사용하여 AI pipeline을 설계하고 적용 및 검증하는 것을 목표로 한다.

이를 위해, 작업자에 의해 수행되는 육안 검사 프로세스를 평가하고 한계를 이해하며, 정확성과 효율성 측면에서 수동 검사를 뛰어넘는 것을 목표로 육안 검사 프로세스의 자동화 가능성을 평가 한다. 또한, 라벨 정보를 정확하게 인식하고 분석할 수 있는 AI 모델 개발을 위해 딥 러닝 기반의 이미지 검출 및 분류 기술 동향에 대하여 살펴 본다. 그리고, AI pipeline 개발을 위해 A사의 클라우드 서비스를 탐색하고, 유사 사례를 발굴하여 연구한다. 이를 통해 실제 제조 환경에서 AI pipeline을 테스트하여 실제 라벨 육안 검사의 정확성과 효율성을 평가하고 다양한 라벨 유형에 대한 AI pipeline의 확장성, 적응성과 다양한 산업에 적용 가능성을 조사 한다.

이를 통해, AI 기반 검사 시스템의 생산 과정에서 품질 관리 개선, 비용 절감, 자원 효율성 향상 가능성을 검증하고 AI를 산업에 통합하는 것에 대한 넓은 이해가 될 수 있도록 한다.

2. 본론

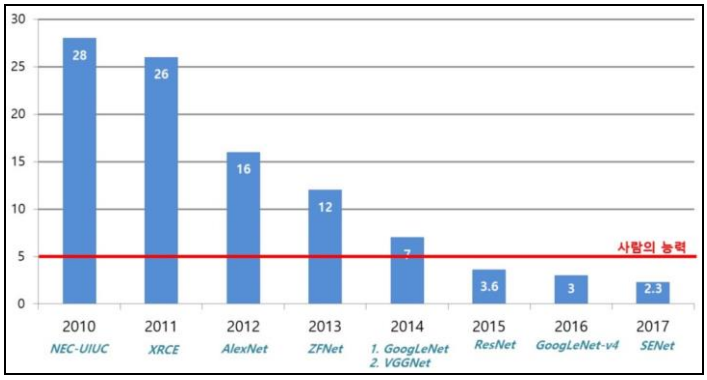
2.1. 선행 연구

HCI(Human Computer Interface) 분야에서 딥 러닝의 역사는 1950년대 다투어서 회의와 학습 가능한 신경망 모델의 출현으로 거슬러 올라 간다. 특히, GPU의 발전으로 딥 러닝은 더욱 실용적이고 효율적으로

변모했다. 딥 러닝은 데이터를 통한 자동 기능 학습으로 전환함으로써 객체 인식에서 인간의 개입을 최소화하고 있다.

CNN는 2012년부터 이미지 분류와 객체 감지에서 딥 러닝의 핵심 기술로 부상했다. ImageNet와 같은 데이터 세트에서 CNN은 뛰어난 성능을 보였다. 이미지 분류는 이미지 내의 객체를 분류하는 것이며, 객체 감지는 이미지 내에서 여러 객체를 찾고 식별하는 것을 의미한다. 또한, ImageNet은 이미지 분류의 주요 벤치마크로 활용되며, 성능을 Top-1 및 Top-5 오류율로 측정한다[1][2][3].

딥 러닝 분야의 주요 모델을 살펴 볼 때, AlexNet은 데이터 증강과 드롭 아웃을 사용하여 ILSVRC 2012에서 15.2%의 Top-5 오류 율을 기록했다[4]. VGG 아키텍처는 간단한 구조로 유명하며, VGG16과 VGG19는 뛰어난 성능을 보였지만 메모리 사용량이 많은 단점이 있다[5][6]. GoogLeNet은 Inception 모듈을 사용하여 효율적으로 다양한 특징을 추출하며, ILSVRC 201에서 6.7%의 Top-5 오류 율을 달성했다[7]. DenseNet은 장기적인 종속성을 포착하고 정보 재사용에 탁월하지만, 대규모 데이터 세트에서 훈련이 어렵고 하이퍼 파라미터 선택에 민감한 단점이 있다[8]. 2017년 제안된 MobileNet은 모바일과 임베디드 환경을 위해 설계된 효율적인 딥 러닝 모델로, 깊이별과 점별 컨볼루션을 사용해 복잡성과 매개 변수를 최소화한다[9]. NASNet은 2018년 Google brain에 의해 제안되었다.



[그림 2 ILSVRC competition]

ILSVRC에서 2012년 딥 러닝, 특히 CNN 기반의 AlexNet 도입으로 인해 인식 오류 율이 크게 감소하고 있는 것을 [그림 2]를 통해 확인할 수 있다. 이러한 발전을 2015년 일반적인 인간의 정확도 오류 율인 5%를 넘어섰고, 2017년 SENet을 통해 오류 율이 인간의 정확도 반 이상을 넘어서는 2.3%로 낮아지면서 컴퓨터 비전 분야에서 딥 러닝의 실질적이고 빠른 발전을 보여 주었다[9][10][11].

딥 러닝 기반의 이미지 검출 및 분류 기술 동향 다음으로 본 연구에 직접 적용할 A사 Cloud 기반의 어플리케이션들의 주요 기능에 대하여 살펴 보았다.

Amplify는 A사에서 지원하는 Cloud Native 기반의 모바일, 웹 앱 개발과 배포를 지원하는 프레임워크이다. API Gateway는 개발자가 API를 생성, 게시, 유리 관리와 모니터링 및 보안 유지를 지원하는 관리 서비스이다. Lambda는 서버 리스 컴퓨팅 기반의 코드 서비스이며, S3(Simple Storage Service)는 가변적인 오브젝트 기반 스토리지 서비스를 지원한다. SageMaker는 기계 학습 모델을 빌드, 학습 및 배포할 수 있는 관리 서비스이며, Lookout for Vision은 산업 환경에서 생산 단위나 장비의 외관 결함 등을 감지할 수 있는 기계학습 서비스이다. Recognition은 딥 러닝 기반의 이미지, 동영상 분석 기능을 제공하고, CodeCommit은 A사에서 제공하는 Git-based 버전 관리 서비스이다.

본 연구의 AI pipeline에 적용될 A사의 각 어플리케이션들은 개별적으로 사용하는 것보다 함께 적용할 때 더욱 효과적이다.

S사와 같은 전통적인 화학 제조회사에서 이미지 분류와 검출 분야에 A사의 어플리케이션을 활용한 사례는 찾기 어려웠으나, 수백만 명의 회원과 6,000여 개의 패션 브랜드를 보유한 온라인 패션 플랫폼 M사의 활용 사례를 찾을 수 있었다.

[그림 3 M사 상품 및 스타일 후기]

[그림 3]과 같이, M사는 상품을 구매한 고객이 상품 이미지나 착용한 스타일 후기 이미지를 후기 게시판에 올리면, 포인트를 적립하는 시스템을 운영 중이다. 사람에 의해 수행되는 후기 이미지 검토를 자동화하기 위해 M사는 A사의 SageMaker, CodeCommit, S3 등을 사용하여 자동화된 이미지 검사 모델을 개발 하였다. RESNet 기반의 Sagemaker를 사용하여 이미지를 분류하고, S3 서비스에 이를 저장하는 자동화된 이미지 검사 모델이다. 이 모델은 TensorFlow 기반의 MobileNet 모델과 비교했을 때, 0.88보다 더 높은 F1 score 0.98을 달성했다. M사는 이러한 자동화 적용을 통해 작업자에 의해 수작업으로 진행하던 사용자 리뷰의 정확도 76%에서 98%까지 자동 처리 방식으로 개선 하였다. 또한, 담당자 별로 다르게 해석할 수 있는 이미지 후기를 자동화 구현으로 통해 일관된 리뷰 정책을 구현함으로써, 소비자 신뢰도 개선할 수 있었다[12][13][14].

2.2 육안 검사 대상 선정 및 AI pipeline 설계

본 연구에서는 S사의 제조 현장 중 드럼 용기를 가장 많이 취급하는 제조팀을 대상으로 드럼 용기 라벨의 육안 검사 과정에 AI pipeline을 설계하고 적용 하였다. [그림 4]는 적용 대상이 되는 6개 제품별 이미지이다. 각 제품은 고객별 주문 수량에 따라 일별로 차이는 있으나, 일 단위로 제품별 50 ~ 200개를 드럼 용기로 납품하고 있다.

제품	이미지	제품	이미지
HXX_1.0_일반		HXX_4.0_자동	
HXX_1.0_자동		HXX_4.0_수출	
HXX_4.0_일반		HXX_R2	

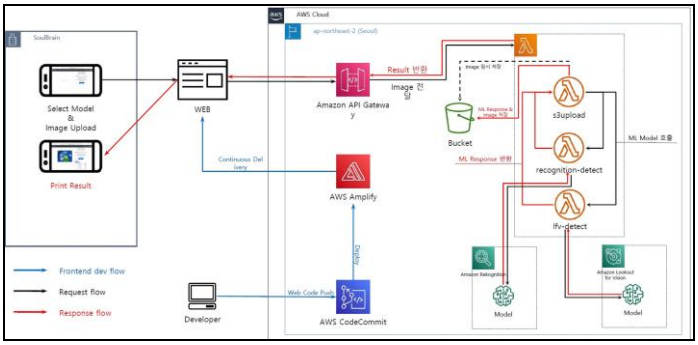
[그림 4 AI pipeline 적용 대상 제품]

이들 제품의 라벨 정보는 드럼 용기의 상단과 측면에 각각 부착된다. 본 연구에서는 실제 제품을 생산하고 있는 제조 현장을 대상으로 함에 따라, 납품 차질을 최소화하기 위해 드럼 용기 상단에 부착된 라벨 정보만을 대상으로 하였다.

작업자는 용기에 라벨을 부착 후 휴대전화 등으로 촬영하여, 검수용 PC에 사진을 업로드 한다. 검수용 PC가 라벨 부착 작업장과 거리가 있기 때문에, 모든 드럼 용기 라벨을 촬영 후 일괄 업로드 하는 방식으로 진행된다. 이후 품질관리팀의 검수 담당자는 업로드 된 이미지를 검수용 PC에서 라벨의 위치, Key-code 등 정해진 합격 조건에 부합하는지 육안으로 전수 검사 한다. 또한, 드럼 용기 라벨의 정확도는 품질 문제와 직결된 중요한 용소임에, 제품이 고객사로 최종 출하되기 전 제조 현장의 출하실에서 검수용 PC로 추가적인 육안 전수 검사를 실시하고 있다.

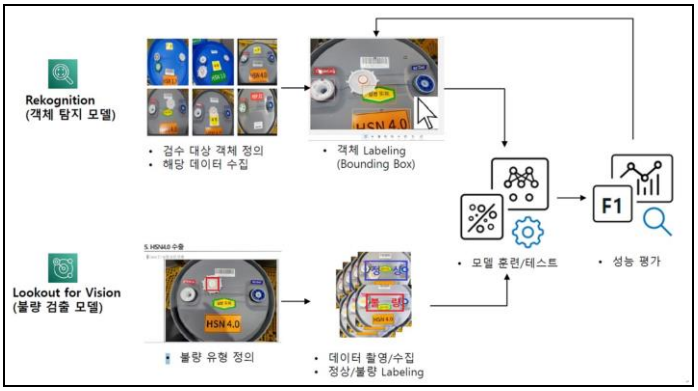
[그림 5]는 본 연구를 위해 설계한 AI pipeline Architecture로, 3주의 기간에 걸쳐 요구되는 기능에 따라 A사의 클라우드 기반 어플리케이션을 구성하여 AI pipeline을 설계하였다.

현장 작업자는 앞서 [그림 1]에서 설명한 기존의 육안 검사 방식 외에도, 본 연구를 위해 별도의 추가 휴대전화를 사용하여 드럼 용기 상단 라벨을 촬영한다. 이후 휴대전화로 개발한 웹 사이트에 접속 후 제품을 선택하고 이미지를 전송한다. 촬영 방식에 있어, 사람에 의한 촬영이 아닌, 고정된 설비에 카메라를 설치하여 촬영하는 것이 정확성을 높일 수 있으나, 실제 제조 현장의 변경 점을 최소화하기 위해 부득이하게 작업자에 의한 수작업으로 이미지를 촬영 하였다.



[그림 5 AI pipeline Architecture]

네트워크 환경에 따라 다르지만, 품질관리팀과 제조 현장 출하실의 검수 담당자는 생산 라인의 작업자가 이미지 전송 후 2~3초 후에 웹 사이트를 통해 검수 결과를 확인할 수 있다.



[그림 6 데이터 훈련/테스트]

[그림 6]과 같이, 모델 성능을 위한 데이터 세트 구성과 라벨링, 훈련 및 테스트, 평가 과정을 1개월 동안 반복적으로 수행 하였다.

라벨 검사에 대한 불합격 판독 알고리즘은 다음의 4가지 경우로 정의 한다. Lookout for Vision에서 Anomaly로 판단하는 경우, 지정한 해당 제품 모델이 탐지해야 하는 라벨의 개수만큼 라벨을 탐지하지 못하는 경우이다. 그리고, 지정한 해당 제품 모델이 탐지해야 할 라벨 외의 라벨을 탐지하는 경우와 각 라벨 중 Confidence가 60% 이하인 경우로 하였다. 마지막 조건, Confidence 60% 이하로 설정한 것은 라벨 검사의 품질 불량과 직결된 중요성을 감안하여 보수적으로 기준을 설정 하였다.

3. 결론 및 향후 연구

3.1. 결론

[표 1 성능 지표 결과]와 같이, 선정한 6개 제품의 AI pipeline을 통한 검사 결과의 전체 평균 정확도는 92.5%의 결과를 확인 하였다. 6개 제품 중 HXX_4.0_일반 제품 라벨에서 FN 오류가 가장 높은

것을 확인할 수 있었다.

제품	Accuracy	Precision	Recall	F1 score
HXX_1.0_일반	96.9%	100.0%	85.7%	92.3%
HXX_1.0_자동	96.2%	100.0%	83.3%	90.9%
HXX_4.0_일반	80.8%	100.0%	16.7%	28.6%
HXX_4.0_자동	84.4%	76.9%	83.3%	80.0%
HXX_4.0_수출	96.8%	91.7%	100.0%	95.7%
HXX_RS	100.0%	100.0%	100.0%	100.0%
합계	92.5%	90.7%	81.3%	85.7%

[표 1 성능 결과 지표]

이는 작업자가 촬영한 이미지의 회전, 각도, 조명 변화로 인해 발생한 것으로 학습 데이터와 다른 유형의 이미지에서 인식도와 정확도가 낮아지는 것을 확인할 수 있었다. 또한, 형태가 유사한 Key-code가 주변 배경과 과적합되어 잘못 분류되는 오류도 발견하였다.

3.2. 향후 연구

S사에서 제품 용기 라벨의 정확도는 매우 중요한 품질 요소로, 이를 위해 품질관리팀과 출하실에서 이중으로 육안 검사를 진행하고 있다. 지난 5년 동안 라벨 오류로 인한 품질 문제는 발생하지 않았지만, 본 연구를 통해 육안 검사의 AI pipeline을 통한 자동화 가능성과 검수 Lead time 감소를 확인 하였다. 또한, 일반적으로 수개월 개발 기간이 소요되는 기존의 방식이 아닌, A사 클라우드 기반 어플리케이션을 활용해 짧은 기간에 유의미한 결과를 얻을 수 있었다.

하지만, 드럼 용기 상부에 부착된 라벨 외 측면에 부착된 라벨에 대한 적용을 통한 실질적인 자동화 구현에 대한 추가적인 연구가 필요하다. AI 기술을 통해 보다 높은 정확도 향상을 위해 모니터링과 충분한 학습 데이터 확보, 제품 규격 변경 대응에 대한 추가 연구도 필요하다.

무엇보다, 본 연구의 방법으로 진행한 작업자에 의한 라벨 이미지 촬영 방식이 아닌, 변경 점을 최소화할 수 있도록 더욱 통제된 환경에서의 이미지 수집을 통한 추가적인 연구와 검증이 필요하다.

참고 문헌

[1] 고광은,and 심귀보. "딥러닝을 이용한 객체 인식 및 검출 기술 동향." 제어로봇시스템학회지 23.3 (2017): 17-24.

[2] 김봉모. "딥 러닝 기반의 이미지 분류 기술 동향." 정보와 통신 35.12 (2018): 8-14.

[3] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-
Fei. Novel dataset for fine-grained image categorization. In First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, June 2011.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[8] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. In IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[10] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le. Learning Transferable Architectures for Scalable Image Recognition. In CVPR, 2018.

[11] E. Real, A. Aggarwal, Y. Huang, and Q. V Le. Regularized evolution for image classifier architecture search. arXiv preprint arXiv:1802.01548, 2018.

[12] https://aws.amazon.com/ko/products/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc&awsf.re%3AInvent=*all&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all

[13] https://aws.amazon.com/ko/what-is-aws/

[14] https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html

한국어 음성으로부터의 조음기관 시각화를 통한 마비말장애 환자의 심각도 측정

주윤지 [O1], Rodrigo Picinini Méxas[2], 심윤섭 [1], 박운상[*2]

[1] 서강대학교 인공지능학과

[2] 서강대학교 컴퓨터공학과

yungie222@sogang.ac.kr , ropimex@sogang.ac.kr , yoonseop@sogang.ac.kr , unsangpark@sogang.ac.kr

Severity Assessment of Dysarthria of Patients with Speech Disorders through Visualization of Articulatory Mechanisms from Korean Speech

Yunji Chu[O1], Rodrigo Picinini Méxas[2], Yoonseop Shim[1], Unsang Park[*2]

[1]Department of Artificial Intelligence, Sogang University

[2]Deparment of Computer Science and Engineering, Sogang University

요 약

본 논문은 한국어 음성 데이터로부터의 조음기관 시각화를 통한 마비말장애 환자의 발화 심각도를 측정하는 새로운 방법을 제안한다. 기존 파이썬 라이브러리인 Vocal Tract Lab을 한국어 음성에 용이하도록 변형하여 시각화를 진행하였다. 시각화에 사용되는 파라미터를 활용해 정상인과 환자의 발화 시 혀 위치 차이를 L2 distance로 측정하여 발화 심각도를 정량적으로 평가하였다. 이 실험을 통해, 본 연구에서 제시하는 시각화 데이터가 마비말장애의 진단과 치료 방향성 제시에 의미 있는 역할을 수행할 것으로 기대해본다.

1. 서 론

마비말장애는 신경근계 장애의 일종으로, 언어 및 음성 생성에 필요한 근육의 제어와 조절에 영향을 미친다. 이러한 장애는 일상생활에서 의사소통의 중대한 장애물이 되며, 환자들의 생활 품질과 사회적 참여에 부정적인 영향을 줄 수 있다. 따라서, 마비말장애의 정확한 평가와 심각도 측정은 적절한 치료와 재활 프로그램 개발에 중요한 역할을 한다.

본 연구에서는 한국어 음성으로부터의 조음기관 시각화를 통해 마비말장애 환자의 심각도를 측정하는 새로운 접근 방식을 제안한다. 기존에는 마비말장애 환자의 심각도를 파악하기 위해 언어치료사가 직접 발화음성의 호흡, 명료도 및 여러 측정기준에 의거하여 판단하고 있다. 하지만, 본 연구에서 제시하는 조음기관 시각화는 음성 생성 조음기관의 동작을 시각적으로 나타내는 것을 의미하며, 이는 음성 장애에 대한 평가와 이해에 보조적으로 큰 도움을 줄 것으로 기대된다. 우리는 한국어로서의 특성을 고려하여, 한국어 음성 데이터를 활용하여 조음기관의 동작을 시각화하고 이를 통해 마비말장애 환자의 심각도를 측정하고자 한다.

본 논문에서는 한국어 음성을 시각화하기 위해 Vocal Tract Lab[1] (이후 VTL)을 활용하였다. 이를 정량적으로 평가하기 위해서 L2 distance를 이용해 각 환자들의 발화 심각도를 음성 데이터셋 분류에 맞춰 0, 1, 2의 총 3단계로 나누어 비교해보았다.

2. 관련연구

본 연구와 가장 흡사한 연구는 장애가 있는 환자의 음성 데이터셋을 활용하여 시각화한 연구이지만, 이전 관련 연구에서는 찾아볼 수 없었다. 음성 시각화와 관련하여 가장 연관 있다고 판단한 것은 새로운 언어를 배울 때 화자의 잘못된 발음과 관련된 연구였다.

논문 [2]에서 저자는 언어치료와 외국어 발음을 돕기 위한 연구목적으로 발음할 때의 혀의 위치를 추적하여 이를 초음파 이미지를 통해 시퀀스를 분석하였다. 논문 [3]에서는, 다층 신경망을 활용하여 Magnetic Resonance Imaging (이후 MRI) 데이터로부터 학습자의 음성을 성대의 좌표로 변환한다. 이후, 이를 활용하여 CG 애니메이션을 만들고, 특정 발음이 실제로 어떻게 발음되어야 하는지와 학습자는 어떻게 발음하고 있는지를 비교 분석한다.

논문 [4]에서 저자는 비원어민의 음성 입력만을 사용하여, 비원어민의 발음에서 정확한 음소와 부정확한 음소를 정확하게 분리하고자 한다. 이를 위해, 정확하게 발음된 음성과 이와 대응되는 성대 이미지와의 관계를 feature inversion을 통해 정립한다.

이러한 선행 연구들은 음성 입력을 발음 기관 이미지로 변환하여 시각화된 정보를 통해 마비말장애 환자의 심각도를 측정하고자 하는 목적에 있어 그 가능성의 근거가 된다.

3. 실험방법

3.1 연구대상

표 1 환자 정보

환자	심각도	나이	성별
No.1	0	70대	F
No.2	1	50대	M
No.3	2	50대	M

우리는 이화여자대학교 병원과의 협력을 통해 '가을' 문단을 발화한 심각도 0의 277명, 심각도 1의 60명, 심각도 2의 20명의 비공개 발화 음성 데이터셋을 수집할 수 있었다. 위의 표 1은 우리가 가진 음성 데이터셋 중에서 시각화를 했을 때 가장 표준적으로 심각도를 대표하는 환자를 뽑아 그 환자들의 정보를 나타낸 것이다. 연구를 진행할 때, 정상인 발화를 기준으로 하여 다음 환자들과 비교를 하는 식으로 연구를 진행하였다.

3.2 데이터셋

본 연구에서 사용한 데이터 셋은 '가을(김향희, 1996)'문단 중 일부이다. '가을' 문단은 발화 속도, 발화 명료도 등 음성과 발화의 평가에 널리 사용될 수 있도록 고안된 문단으로, 다양한 음소들이 고르게 분포되어 있는 평가 자료이다. 이는 총 210음절로 구성되어 있으며, 낭독 시 대략 50초 안팎의 시간이 소요된다. 발화 문단의 내용은 아래 표 2에 제시해 두었다.

표 2 발화 문단 내용

'가을(김향희, 1996)' 문단의 일부
우리나라의 가을은 참으로 아름답다. 무엇보다도 산에 오를 땐 더욱 더 그 빼어난 아름다움이 느껴진다. 쓰다듬어진 듯한 완만함과 깎아 놓은 듯한 뾰족함이 어우러진 산등성이를 따라 오르다 보면 절로 감탄을 금할 수가 없게 된다. 붉은색, 푸른색, 노란색 등의 여러 가지 색깔이 어우러져 타는 듯한 감동을 주며 나아가 신비롭기까지 하다. 숲 속에 누워서 하늘을 바라보라. 쌍쌍이 짝지어 있는 듯한 흰 구름, 높고 파란 하늘을 쳐다보고 있노라면 과연 옛부터 가을을 천고마비의 계절이라 일컫는 이유를 알게 될 것만 같다.

4. 실험내용

4.1 VTL 유효성 검증

이 실험이 성립하기 위해서는, VTL을 활용하여 만든 이미지의 유효성이 먼저 검증되어야 한다. 이를 검증하기 위해 우리는 기본 모음들(a, e, i, o, u)이 VTL에서 어떻게 생성되는지 확인하고, 그 이미지들 위에 모음 사각도를 덧씌워 보았다.

모음 사각도란 모음이 어떻게 발음되는지, 또한 발음을 할 때 혀가 상대적으로 어디에 위치하는지 이해하기 위해 만들어진 다이어그램을 의미한다[5]. 이 다이어그램은 그림 1에서 확인할 수 있다.

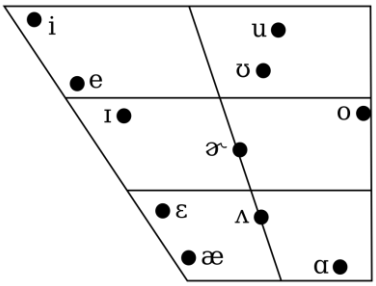


그림 1 모음 사각도 [6]

모음 사각도를 사용하면 발화자가 올바르게 발음하는지에 대한 여부를 확인할 수 있다. 혀가 다이어그램에서 위치해야 할 곳에서 벗어나 있다면, 이는 제대로 발음되지 않고 있다고 간주할 수 있다. 그림 2는 VTL 생성 이미지에 모음 사각도에서 제시하는 기본 모음들이 이미지의 어느 위치에 나타내는지 나타낸 그림이다.

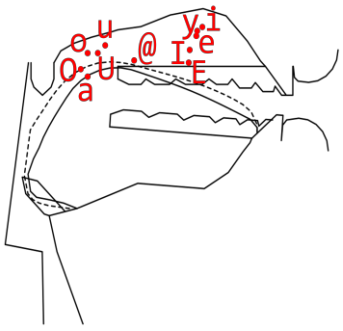


그림 2 모음의 위치를 표기한 VTL 모음사각도

4.2 음성의 시각화

4.2.1 한국어 음성의 SAMPA로의 변환

4.1절에서 제시한 유효성 검증을 통하여, VTL은 의미 있는 시각화 자료를 제공하는 것을 알 수 있다. 따라서, 우리는 VTL을 사용하여 정상인과 비정상인의 발화 음성을 시각화 하여 비교하는 실험을 진행하였다.

VTL은 'Speech Assessment Methods Phonetic Alphabet (이후 SAMPA)'를 입력값으로 하여 svg 파일을 생성한다. 이를 활용하여 그림 3과 같은 과정을 거쳐 발화 음성을 SAMPA로 변환하는 작업을 진행하였다.

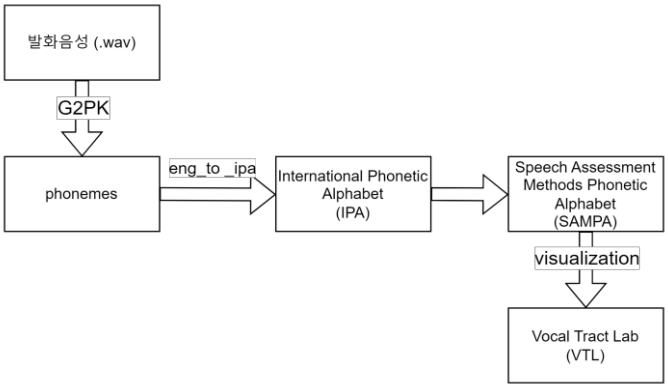


그림 3 발화음성의 시각화 순서도

입력값으로 사용된 가을 문단 발화 음성을 먼저 한국어 음소로 변환하였다. 변환에는 오픈소스 라이브러리인 g2pK 모델을 사용하였다. g2p란 graphemes to phonemes의 약자로, 자소를 음소로 변환하는 모델을 의미한다. 그 중, 우리가 사용한 g2pK 모델은 한국어의 자소를 음소로 변환하는 모델이다. 이를 활용하여 손쉽게 한국어 문단을 음소로 변환할 수 있었다. 이어서, 음소를 Unidecode 라이브러리를 활용하여 영어로 변환하고, 이를 SAMPA로 변경하기 위해 IPA로 먼저 변환을 진행하였다. IPA로의 변환은 제시한 그림 3에서 확인할 수 있듯이, 오픈소스 라이브러리인 eng_to_ipa를 활용하였다. 결과값으로 얻은 IPA는 Unicode mapping을 통해 SAMPA로 변환해 주었다. 이 변환과정을 표로 제시하면 아래 표 3과 같다.

표 3 한국어 문단을 SAMPA로 변환한 예시

한국어 문단	우리나라 우리나라의 가을은 참으로 아름다운 아름답다
한국어 음소	우리나라 우리나라의 가으른 차므로 아름다우 나름답따
영어로 변환	u ri na ra u ri na ra yi ga eu reun ca meu ro a reum da u na reum dab dda
IPA	['ju'], ['ri*'], ['nɑ'], ['rɑ'], ['ju'], ['ri*'], ['nɑ'], ['rɑ'], ['ji'], ['gɑ', ''ʤi'eɪ', ''ʤɔrʤə'], ['eu*'], ['reun*'], ['kɑ', 'kə', ''si'eɪ'], ['meu*'], ['roʊ'], ['eɪ', 'ə'], [''riəm'], ['dɑ', ''di'eɪ'], ['ju'], ['nɑ'], [''riəm'], ['dæb'], ['dda*']
SAMPA	['ju', 'ri*', 'nA', 'rA', 'ju', 'ri*', 'nA', 'rA', 'ji', 'gA"ʤi"eI"ʤOrʤ@', 'eu*', 'reun*', 'kAk@"si"eI', 'meu*', 'roU', 'eI@', '"ri@m', 'dA"di"eI', 'ju', 'nA', '"ri@m', 'd{b', 'dda*']

4.2.2 VTL을 활용한 시각화

4.2.1절에서 얻은 SAMPA 정보를 활용하여 VTL에서 시각화를 진행한 결과 그림 4와 같은 결과를 얻을 수 있었다. 이 그림에서는 혀의 위치가 돋보이지만, 실제로 VTL은 구강구조를 포함한 19가지의 파라미터를 활용하여 시각화를 진행한다. 그림 4에서 빨간색은 심각도 0, 초록색은 심각도 1, 노란색은 심각도 2의 환자의 혀 위치를 의미한다.

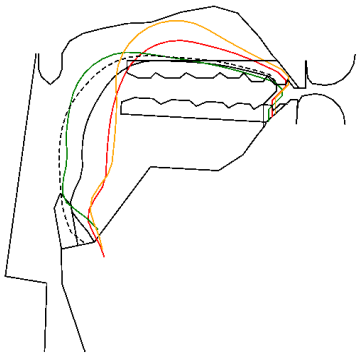


그림 4 VTL을 통해 생성한 이미지에 혀를 합성한 그림

4.3 평가지표(L2 distances)

마비말장애 환자의 발화 음성의 심각도를 정량적으로 평가하기 위해, 우리는 이상적인 혀의 위치와 환자의 음성에서 얻은 혀의 위치를 비교하여 거리를 계산하고자 했다. 이를 위해 고안한 방법은 L2 distance이다.

L2 distance[7]란 두 점 사이의 거리를 계산하기 위한 방법 중 하나로 유클리드 거리라고도 알려져 있는 방식이다. 점의 집합이 식 (1)과 같이 주어졌을 때, 식 (2)와 같이 L2 distance를 계산할 수 있다.

$$x = (x_1, x_2, ..., x_n), y = (y_1, y_2, ..., y_n) \quad (1)$$

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \quad (2)$$

그림 5 VTL에서의 정상 혀와 심각도별 혀 위치의 차이

그림 5는 왼쪽부터 심각도 0, 1, 2로 정상인과 마비말장애를 가진 환자의 음성으로부터 VTL을 통해 얻어낸 발음 기관 이미지를 각각 나타낸 것이다.

5. 실험결과

5.1. 실험 목적 및 방식

VTL을 통해 생성된 이미지 내의 혀를 나타내는 곡선 사이의 L2 distance와 마비말장애 환자의 심각도의

상관관계를 확인하고자 한다. 이를 위해 VTL을 통해 기준 문장을 발화하였을 때의 발음 기관을 먼저 시각화한 후, 각 심각도에 해당하는 마비말장애 환자의 실제 발화 음성으로부터 발음 기관을 시각화한 것과 비교를 진행하였다. 비교를 위하여 시각화 결과 이미지를 이용하여 L2 distance를 계산하여 정상 발화 시의 혀 위치와 환자의 발화 시의 혀 위치의 차이를 측정하였다.

5.2. 실험결과 정량화

정상인과 각 심각도별 환자와의 혀 위치의 L2 distance를 pixel 단위로 구하였다. 4.2.2절의 그림 4에서 혀를 나타내는 곡선은 VTL을 활용할 때 생성되는 svg파일 내 37개의 점을 잇는 직선으로 이루어져 있다. 따라서, 혀를 나타내는 곡선을 그리는 모든 점들을 통해 심각도 0의 277명, 심각도 1의 60명, 심각도 2의 20명에 대한 곡선 사이의 L2 distance의 값을 구하여 이들의 평균과 표준편차를 표 4와 같이 제시하였다.

표 4 심각도별 L2 distance의 평균과 표준편차

심각도	0	1	2
평균	10.86	16.22	21.83
표준편차	5.67	2.54	4.00

혀를 나타내는 곡선 간 차이의 평균은 심각도가 증가할수록 증가하는 것을 확인하였다. 그러나 표준편차의 경우 각 심각도 내에서의 L2 distance 값의 범위가 얼마나 넓은지를 나타내고 있기 때문에 심각도 1에 비하여 심각도 0, 2의 경우에는 평균 L2 distance가 해당 심각도의 특성을 대표한다고 판단하기 어렵다. 즉, 실제 심각도 0인 환자에 대하여 혀 위치에 대한 L2 distance가 심각도 1의 평균값과 더 가까울 확률이 작지 않다고 해석할 수 있다.

이를 그래프로 나타내면 6절에서 제시하는 그림 6과 같으며, 심각도별 정상 혀 곡선과 환자의 혀 곡선 사이의 L2 distance의 평균이 선형 상관관계를 가짐을 확인하였다. 즉, 평균적으로 L2 distance 값을 통하여 발화 음성의 심각도를 측정하는 것이 가능하다는 결론을 얻을 수 있었다.

6. 결론 및 향후계획

본 연구를 통하여 모음 사각도를 이용한 VTL의 정확성을 확인하였으며 L2 distance를 평가 지표로 하여 마비말장애 환자의 심각도에 대한 정량적 측정 가능성에 대하여 확인하였다. 이를 통해, 한국어 음성으로부터 마비말장애 환자의 심각도를 손쉽게 파악 가능한 알고리즘을 설계하기 위한 검증이 이루어졌다. 하지만, 실험 결과에 의하면, 심각도 1에 비해 심각도 0과 심각도 2의 L2 distance 표준편차가 크기 때문에 신뢰도에 있어 개선이 필요할 것으로 판단된다.

따라서, 심각도 평가에 대한 신뢰도 개선을 위해 향후 표준편차에 큰 영향을 주는 데이터에 대한 분석과 또 다른 선형 상관관계를 갖는 정량적 평가 지표를 고안하는 연구를 진행하려고 한다.

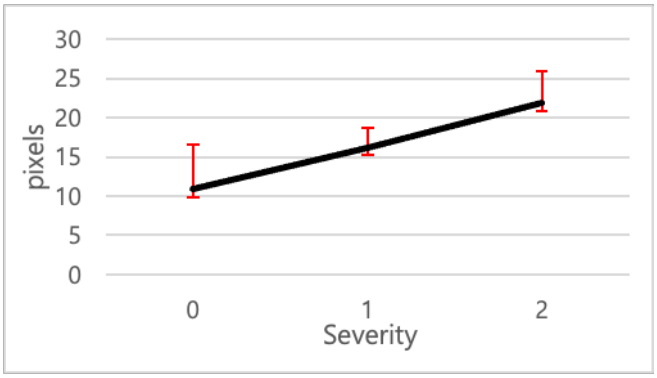


그림 6 Severity에 따른 L2 distance Pixel 그래프

Acknowledgement

이 논문은 2024년도 정부 (과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2022-0-00621, 대화 기반 설명가능성을 멀티모달로 제공하는 인공지능 기술 개발)

7. 참고문헌

[1] P. Birkholz, "Modeling consonant-vowel coarticulation for articulatory speech synthesis" in *PloS one*, Vol, 8, No. 4, 2013, pp. e60603.

[2] M. H. Mozaffari, S. Guan, S. Wen, N. Wang and W.-S. Lee, "Guided Learning of Pronunciation by Visualizing Tongue Articulation in Ultrasound Image Sequences" in *2018 IEEE CIVEMSA*, 2018, pp. 1-5.

[3] T. Nitta, S. Manosavan, Y. Iribe, K. Katsurada, R. Hayashi and C. Zhu, “Pronunciation training by extracting articulatory movement from speech” in *Int. Symposium on Automatic Detection of Errors in Pronunciation Training*, 2012, p. 75.

[4] O. Engwall, “Pronunciation analysis by acoustic-to-articulatory feature inversion.” in *Int. Symposium on Automatic Detection of Errors in Pronunciation Training*, 2012, p. 79.

[5] F. Trujillo, The vowel chart, https://www.ugr.es/~ftsaez/fonetica/vowels.pdf, Retrieved May 30, 2023

[6] P. Ladefoged, "American English", *Handbook of the International Phonetic Association. Cambridge: Cambridge University Press*, 1999, pp. 41-44.

[7] D. Cohen, “Precalculus: A Problems-Oriented Approach”, *Cengage Learning 6th ed*, 2004, p. 698.

기저선 변동 잡음 제거와 딥러닝을 이용한 심전도 QT 간격 예측

이승준 [1O] 박진주 [1] 김유리 [2*] 양형정 [1*]

[1] 전남대학교 인공지능융합학과 [2] 전남대학교 병원

dltmdwns3462@naver.com, pjj7367@naver.com, geniusyul@gmail.com, hyungjeong@gmail.com

ECG QT interval Prediction Using Baseline Wander Removing and Deep Learning

Seungjun Lee[1O] Jinju Park[1] Yoo Ri Kim[2*] Hyung-Jeong Yang[1*]

[1] Department of Artificial Intelligence Convergence, Chonnam National University

[2] Division of Cardiology, Department of Internal Medicine, College of Medicine,

Chonnam National University

요 약

심전도는 심장에서 발생하는 전기적인 신호를 기록하며, 심장 질환 진단에 필요한 중요한 정보를 제공한다. 심전도에서 QT 간격은 QRS파의 시작에서 T파의 끝까지의 시간으로 계산되어 심장의 전기적 특성을 평가하는 데 사용되는 중요한 지표이며, 부정맥 환자 관련 질환을 판단하는데 활용된다. QT를 활용한 심전도 분석은 심장 전문가가 직접 신호를 보고 판단해야 하며, 이는 판단 오류와 같은 문제가 발생할 수 있다. 이를 해결하기 위해, 저주파 통과 필터와 연속 웨이블릿 변환을 이용해 심전도 측정 시 발생하는 기저선 변동 잡음을 제거하고, 시계열 특성을 고려하기 위해 CNN과 Bi-LSTM을 결합하여 QT 간격을 추정하는 모델을 제안하였다. 제안 모델은 QT Database에서 Accuracy 0.940, Precision 0.977, Recall 0.902, F1-Score 0.938의 성능을 보여주었다.

1. 서 론

현대 의학 분야에서 생체 신호 분석이 세계적으로 많은 관심을 받으며, 특히 심전도(Electrocardiogram, ECG) 분석이 중요한 연구 대상이다. 심전도는 심장에서 발생하는 전기적인 신호를 기록한 것으로 다양한 분야에서 활용되며, 대표적으로 심장질환의 유무를 진단하는 중요한 정보를 얻을 수 있다[1]. 심전도는 크게 P파, QRS파, T파로 구성되어 있으며, 심전도의 노이즈 제거나 부정맥 분류 알고리즘의 정확도 비교 등이 중요한 연구 주제로 다뤄진다. 특히 QT 간격은 심전도 분석의 중요한 항목 중 하나로 심실성 부정맥의 위험도를 평가하거나 심혈관 사망률과 심장 돌연사를 예측하는 데 활용된다[2].

심전도의 경우 심장 전문가가 직접 신호를 보고 판단한다. 이 과정에서 판단 오류가 발생하며, 심전도 측정 시 발생하는 잡음은 정확한 진단을 방해한다. Luo, S et al[3]은 심전도 필터링 방법으로 고주파 통과 필터, 기저선 변동 잡음 제거 및 저주파 통과 필터와 같은 심전도 필터링에 대한 다양한 방법을 제시한다. 특히, 저주파 통과 필터는 QRS 간격과 같은 저주파 범위의 결과 해석을 용이하게 한다.

본 연구에서는 이러한 문제점을 해결하기 위해 QT 간격 추정을 통해 심장 전문가들에게 참고용으로 제공하는 것을 목표로 한다. 이를 위해, 기저선 변동 잡음 제거를 위한 저주파 통과 필터와 연속 웨이블릿 변환을 활용하여, 심전도 측정 시 발생하는 여러 잡음을 제거한다. 또한, 심전도의 데이터 특징과 시계열 특징을 활용하기 위해 CNN과 Bi-LSTM을 결합한 모델을 제안한다.

2. 제안 방법

2.1 잡음 제거

본 연구에서는 원본 심전도 데이터에서 기저선 변동 잡음을 제거하고, 이후, 신호 변화를 분석하고 심전도 특성을 정밀하게 분석하기 위해 연속 웨이블릿 변환을 적용하는 방법을 사용한다. 이 과정에서, 샘플링을 250Hz에서 125Hz로 다운샘플링하고, 스케일을 62로 설정하여 계산 복잡성을 줄였다. 생성된 데이터는 1초당 125개의 샘플로 이루어진 학습 데이터로 활용되며, 데이터의 간격을 10으로 설정하여 많은 데이터를 확보할 수 있도록 하였다. 또한, 데이터 라벨링은 QT 간격을 1로, 그 외 부분을 0으로 설정하여 진행하였다.

2.2 제안 모델

본 연구에서는 그림 1과 같이 심전도 데이터의 QT 간격을 추정하기 위해 CNN과 Bi-LSTM을 결합한 모델을 제안하였다. CNN은 데이터의 지역적 특성을 효과적으로 추출하고, Bi-LSTM은 시간적 의존성을 갖는 심전도 데이터의 패턴을 정밀하게 학습한다. 모델의 입력은 (125, 62, 1) 크기이며, Convolution Layer, ReLU, Batch Normalization, MaxPooling,

*: 교신저자

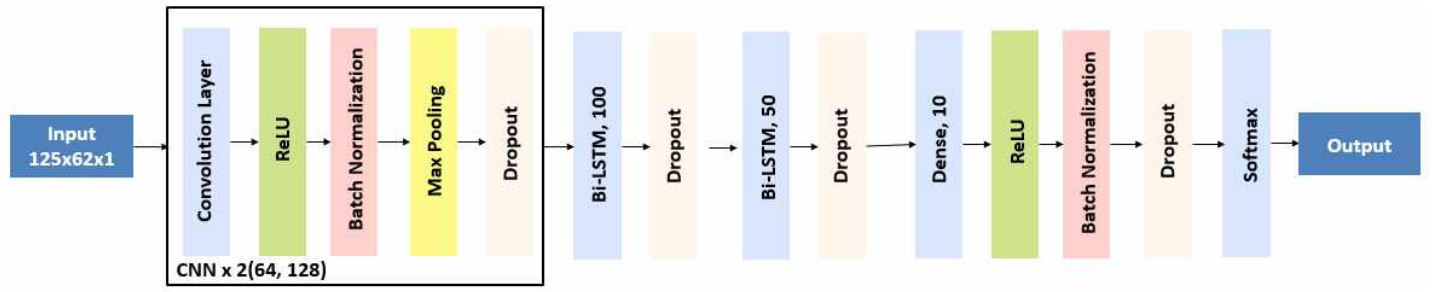


그림 1. 제안 모델 구조

그리고 Dropout으로 구성된 2개의 CNN 블록은 각각 64, 128개의 필터를 갖는다. 이어서 Bi-LSTM 블록은 100개와 50개의 유닛을 갖는 2개의 층으로 구성되어 있다. 마지막으로 Dense 층을 추가하여 Softmax를 통해 이진 분류를 수행하였다. 모델은 Categorical_crossentropy를 손실 함수로 사용하였으며, 최적화는 Adam 알고리즘을 수행하였다. 이 과정에서 과적합을 방지하기 위해 Validation loss를 기준으로 조기 종료 설정을 적용하였다.

3. 실험

3.1 데이터 셋과 전처리

본 연구에서 사용한 데이터셋은 QT Database[4]로, 105명의 환자를 15분동안 250Hz로 심전도를 측정한 데이터이다. 각 환자는 2개의 Lead와 P, QRS T파에 대한 주석 정보를 가지고 있다. 그러나 Lead 2는 소수의 환자만 가지고 있어 제외하였고, 주석이 일정하게 되어있지 않은 환자의 데이터도 제외하였다. 따라서 실제로 사용한 데이터는 전체 환자 중 63명이며 Lead 1 데이터이다.

같은 환자의 데이터가 훈련 세트와 검증 세트에 모두 포함되면 모델이 특정 환자의 데이터 패턴에 과적합될 수 있다. 이런 문제를 방지하고 모델의 일반화 성능을 향상시키기 위해, 환자 단위로 데이터셋을 분리하였다. 실험 데이터셋 및 샘플을 고려하여 Train 51명(15490 samples), Validation 6명(1564 samples), Test 7명(1682 samples)으로 약 8:1:1 비율로 구성하였다(표 1).

표 1. 실험 데이터셋

구분	환자(명)	데이터 수(sample)
Train	51	15490
Valid	6	1564
Test	7	1682

3.2 실험 결과

이진 분류 평가를 위해 Accuracy 와 F1-Score 를 평가지표로 선택하였다. 사용한 방법은 제안 모델에서 설명한 Bi-LSTM 을 2 개 사용한 방법과 100 개 유닛으로 한 개를 사용한 방법 그리고 이와 비슷한 순환 신경망인 GRU 를 사용해 같은 유닛 구성으로 실험을 진행하였다. 결과적으로, Bi-LSTM 2 개를 사용한 방법이 Accuracy 와 F1-score 에서 가장 높은 성능을 보였다.(표 2)

표 2. 실험 결과

Method	Accuracy	Precision	Recall	F1-Score
CNN+GRU (1 use)	0.926	0.954	0.900	0.925
CNN+GRU (2 use)	0.929	0.976	0.880	0.926
CNN+Bi-LSTM (1 use)	0.934	0.975	0.892	0.931
CNN+Bi-LSTM (2 use)	**0.940**	**0.977**	**0.902**	**0.938**

4. 결론

본 연구에서는 저주파 통과 필터와 연속 웨이블릿을 사용하여 기저선 변동 잡음을 제거 후, 시계열 특징을 활용하기 위해 CNN 과 Bi-LSTM 을 결합하여 QT 간격을 추정하는 모델을 제안하였다. 이를 통해 심전도 측정 시 발생하는 여러 잡음을 제거하고, 심전도의 시계열 특성을 고려하여 QT 간격 추정을 더 정확하게 수행하였다. 향후 연구에서는 본 연구에서 제안한 모델을 활용하여 심장질환 예후 예측 및 약물 유발성 심혈관계 질환 예측을 수행할 계획이다.

Acknowledgment

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2023-RS-2023-00256629)

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00208397).

참고 문헌

[1] Chauhan, S., Arora, A. S., & Kaul, A. A survey of emerging biometric modalities. Procedia Computer Science, 2, 213-218. 2010.

[2] Straus, S. M., Kors, J. A., De Bruin, M. L., van der Hooft, C. S., Hofman, A., Heeringa, J., ... & Witteman, J. C. Prolonged QTc interval and risk of sudden cardiac death in a population of older adults. Journal of the American College of Cardiology, 47(2), 362-367. 2006.

[3] Luo, S., & Johnston, P. A review of electrocardiogram filtering. Journal of electrocardiology, 43(6), 486-496. 2010.

[4] QT Database. PhysioNet. https://physionet.org/content/qtdb/1.0.0/. 1999

차량 횡 방향 제어를 위한 신경망 기반 차량 슬립 각도 예측*

김성현 ○, 유승한 [1], 강승우 [1]

한국기술교육대학교 컴퓨터공학과, 한국기술교육대학교 기계공학부, 한국기술교육대학교 컴퓨터공학부

seonghyeon.kim@misl.koreatech.ac.kr, shyoo@koreatech.ac.kr,
swkang@koreatech.ac.kr

Estimation of Vehicle Sideslip Angle Using Neural Networks for Vehicle Lateral Dynamics Control Applications

Seonghyeon Kim○, Seung-Han You[1], Seungwoo Kang[1]

Department of Computer Engineering, KOREATECH,
School of Mechanical Engineering, KOREATECH,
School of Computer Science and Engineering, KOREATECH

요 약

본 연구에서는 차량 상태 중 차량 슬립 각도와 Yaw rate를 추정하는 방법을 탐구한다. 차량에 장착되는 Active Control System(ACS)이 효과적으로 동작하기 위해서는 주행 상황이나 차량 상태를 정확히 획득하는 것이 중요하다. 그 중 차량 슬립 각도는 차량 횡 방향 거동 예측에 중요한 차량 상태이지만 비용 제약으로 인해 일반 차량에서는 센서를 통해 측정할 수 없다. 이에 따라 차량 슬립 각도 추정을 위해 동역학 기반 방법, 칼만 필터 기반 방법, Neural Network 기반 방법 등 다양한 방법들이 연구되었다. 본 연구에서는 2-DOF Single Track Model(Bicycle model)의 매개변수 중 하나인 타이어 코너링 강성을 Self-attention Neural Network로 추정해 모델 기반 방법과 Neural network 기반 방법을 융합하는 방법을 사용한다. 이 방법을 통해 Bicycle 모델의 추정 정확도를 Data-driven 학습으로 향상시킬 수 있고 Black Box인 End-to-end Neural Network 기반 방법과 달리 모델을 제어 관점에서 해석할 수 있다. 또한, 실제 차량 주행 환경과 유사한 Pure Prediction 시스템에서 발생할 수 있는 누적 오차에 대비하는 Mini Error Accumulated-loop 학습 방법을 제안한다. 실제 차량 주행 데이터에서 학습하고 Pure Prediction 환경에서 검증한 결과 기존 One-step-ahead Prediction 방법과 비교해 성능 향상을 보였다.

1. 서 론

자동차 소유 증가로 인해 교통사고 발생 빈도가 연간 증가하고 있다[1]. 이에 따라 운전자의 편의와 안전에 관한 관심이 높아지고 관련 시스템이 개발되어 차량에 장착되고 있다. 이러한 시스템을 ACS(Active Control System)이라고 하며 최근에는 카메라, 라이더, 레이더 등 환경 센서들이 차량에 장착되어 동작하는 ADAS(Advanced Driver Assistance Systems)가 널리 사용되고 있다 [2]. 효과적인 ADAS를 개발하기 위해서는 주행 상황이나 차량 상태를 정확히 획득해야 한다. 이러한 상태에는 차량 슬립 각도, yaw rate, 종방향 속도, 횡 방향 속도 등이 포함된다. 차량 상태는 주로 차량 내 장착된 센서를 통해 측정하지만 장착된 센서의 정확성 및 비용 제약, 측정 잡음 등으로 인해 일부 상태는 측정되지 못하거나 측정 성능이 정확도 요구 사항을 충족시키지 못할 수 있다. 이에 대한 해결책으로 차량 상태 추정기를 설계하여 일부 상태를 추정하는 방법들이 연구되고 있다.

우선 모델 기반 방법이 있는데 그 중에서는 동역학 기반 방법과 칼만 필터 기반 방법이 널리 연구되었다 [3-4]. 이러한 모델 기반 방법은 미리 매개변수를 식별하여 모델을 설계하는데, 예상하지 못한 변수로 인해 오차가 발생할 수 있다. 이에 따라 Neural Network를 통해 Data-driven으로 학습하고 비선형으로 추정하는 End-to-end Neural Network 방법들이 연구되었다 [5-6]. 하지만 데이터 입력이 어떤 과정을 거쳐 추정하였는지 확인할 수 없는 Black-box 방법이기 때문에 제어 해석이 불가능하고 다른 제어 변수들을 계산할 수 있는 확장성이 떨어진다. 이에 따라 Data-driven으로 학습할 수 있고 제어 관점에서 해석 가능한 융합 모델기반 방법이 연구되었다. 융합 모델 기반 방법은 모델 기반 접근 방식에 Neural Network를 융합하여 Data-driven으로 학습할 수 있고 모델에 대해 제어 해석이 가능하다.

동역학 모델 중 하나인 2-DOF Single Track Model(Bicycle model)에서는 타이어 횡 방향 힘과 타이어 슬립 각도 사이의 비선형적인 관계를 선형으로 가정하고 타이어 코너링 강성을 상수로 가정하여 타이어 횡 방향 힘과 타이어 슬립 각도 사이의 비례계수로 사용한다. 이는 Bicycle 모델의 주요한 오차 중 하나이다. 그래서 본 연구에서는 Bicycle 모델의

* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1A2C1004405).

1 교신 저자

매개변수 중 타이어 코너링 강성을 Neural Network로 추정해 성능을 향상시키는 융합 기반 방법을 사용한다.

차량 슬립 각도나 타이어 코너링 강성은 센서의 비용 및 정확도의 문제로 차량 주행 중 지속해서 정확한 측정값을 얻기 어려운 차량 상태이다. 하지만 대부분의 연구에서는 차량 상태를 지속해서 얻을 수 있다고 가정하고 차량 상태 센서 측정 값을 입력으로 다음 시점 차량 상태를 예측하는 One-step-ahead Prediction 시스템을 설계하고 미리 측정된 데이터를 입력하여 검증하고 있다. 본 연구에서는 차량 슬립 각도를 측정할 수 없다고 가정하고 모델의 이전 추정 값을 입력으로 다음 시점을 추정하는 Pure Prediction 시스템을 제안한다. 이러한 Pure Prediction 시스템에서는 오차가 포함된 추정값을 다시 입력으로 사용하므로 오차 누적이 발생할 수 있다. 그러므로 오차 누적을 대비해 누적 오차를 함께 학습할 수 있는 Neural Network 학습 방법을 제안한다.

2. 본 론

본 장에서는 시스템의 구조와 시스템을 학습하는 방법 그리고 시스템 검증을 위해 진행한 실험에 대해 설명한다.

2.1 시스템 구조

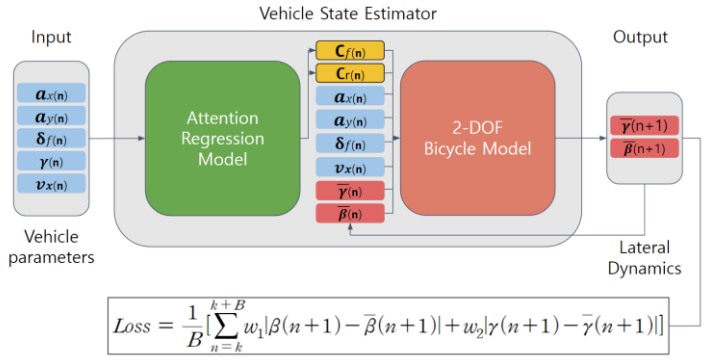


그림 1 Bicycle 모델, Neural network 융합 모델 구조

[그림 1]은 이 연구에서 제안하는 전체 모델 구조이다. 종 방향 가속도(a_x)[m/s²], 횡방향 가속도(a_y)[m/s²], 전륜 타이어 조향각(δ_f)[rad], Yaw Rate(γ)[rad/s], 종 방향 속도(v_x)[m/s]를 Attention Regression Model 입력으로 전륜 타이어 강성(C_f)과 후륜 타이어 강성(C_r) 값을 추정한다. 이렇게 비선형성을 보상한 전/후륜 타이어 코너링 강성과 전륜 조향각, 종 방향 속도, Yaw Rate, 차량 슬립 각도(β)[rad]를 2-DOF Bicycle Model에 대입해 다음 시점의 차량 슬립 각도와 Yaw Rate를 도출한다. 이 때, 입력되는 차량 슬립 각도와 Yaw Rate는 모델의 이전 시점의 출력 데이터이다. [그림 1] 구조에서 Attention Regression Model과 Bicycle 모델은 이 연구에서 제안하는 모델 외의 다른 Neural Network나 동역학 모델로도 대체할 수 있다.

2.1.1 Attention Regression Model

본 연구에서 제안하는 Attention Regression Model은 “Attention Is All You Need” [7]에서 제안한 Transformer 모델의 Encoder 부분을 변형하여 사용하고 Decoder 부분은 Linear Layer로 대체한 모델이다. Transformer의 Self-attention Layer는 자연어 처리 분야 외에 Regression을 위한 모델로 다양하게 변형되어 사용된다 [8-9]. 제안하는 시스템은 순서 정보가 없는 센서 데이터를 입력하므로 기존 Transformer와 달리 Positional Embedding은 사용하지 않고 각 입력 데이터 사이의 Attention Value를 구한다. 그리고 Decoder 부분은 Regressor 역할을 하는 Linear Layer를 추가해 전/후륜 타이어 코너링 강성을 추정한다.

2.2 Pure Prediction 시스템을 위한 학습 방법

본 연구에서는 Pure Prediction 상황을 가정하여 누적 오차에서 향상된 성능을 보일 수 있는 Neural network 학습 방법을 제안한다. 이전 시점의 모델 출력을 모델에 입력하고 그 과정에서 발생하는 누적 오차에 대응할 수 있어야 한다.

Mini Error Accumulated-loop 학습: 학습 시 데이터를 작은 Error Accumulated-loop 여러 개로 분리해 학습하는 방법이다. [그림 2]와 같이 모델 학습 시 n번째 Step마다 이전 시점의 모델 출력 대신 정답 값을 입력한다. 전체 학습 데이터를 n개의 크기로 이루어진 Mini Error Accumulated-loop 여러 개로 분리할 수 있어 적당한 누적 오차를 가진 데이터에 대해서 학습할 수 있다.

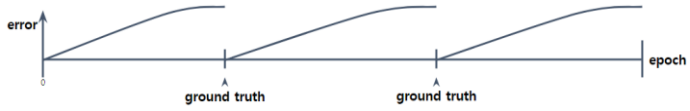


그림 2 Mini Error Accumulated-loop 누적 오차

2.3 실험

제안하는 시스템을 Mini Error Accumulated-loop 학습 방법으로 학습하고 Pure Prediction 검증 환경에서 성능을 평가한다. 다양한 Error Accumulated-loop의 크기로 학습한 결과를 비교한다.

2.3.1 데이터와 전처리

실제 차량이 정해진 시나리오를 여러 회 주행해서 획득한 데이터로 모델을 학습하고 검증한다. 일반 차량에서 획득 가능한 차량 상태와 전용 센서를 통해 Vehicle Sideslip Angle과 Yaw rate를 측정한다. Sampling Rate은 0.01초이고 전체 데이터는 172,536개이다. 총 5가지 시나리오에서 26회 시행했다. 주행 시나리오는 왼쪽 원 선회 7회, 오른쪽 원 선회

6회, Slalom 4회, J-turn 6회, Double Lane Change 3회이다. 각 입력 데이터 측정값의 분포 범위를 맞추기 위해 각 데이터의 최댓값을 나눠서 [-1, 1] 사이의 범위로 만들고 각 시나리오에서 한 회차씩을 Testing Dataset으로 사용하고 나머지를 Training Dataset으로 사용한다.

2.3.2 실험 환경

차량 슬립 각도와 Yaw Rate를 차량 주행 중 센서로 측정할 수 없다고 가정하고 모델의 이전 시점 추정값을 계속해서 모델에 입력하는 상황에서 시스템을 검증한다. 검증에 사용되는 Neural Network의 Hyper-parameter는 동일한 조건을 유지했다. Learning Rate는 1e-3으로 시작해 매 Epoch 마다 0.95씩 곱해 감소시켰다. Batch Size는 256으로 Batch Size 크기 비교 실험을 시행하여 가장 좋은 성능을 보이는 Batch Size로 선정하였다. 전체 Training Dataset에 대해 50 Epoch 학습을 수행하고 가장 Validation 성능이 높은 Weight를 저장하여 검증을 수행했다.

2.3.3 실험 결과

표 1 Mini Error Accumulated-loop 크기 비교 결과

	1	8	16	32	64	128	256
β MAE	0.01698	0.00956	0.00734	0.00721	0.00718	0.00717	0.00715
γ MAE	0.02804	0.02081	0.01919	0.01913	0.0194	0.01976	0.01986

[표 1]은 Mini Error Accumulated-loop 크기를 다르게 하여 학습하고 Pure Prediction 상황에서 검증한 차량 슬립 각도($\boldsymbol{\beta}$) MAE(Mean Absolute Error)와 Yaw Rate($\boldsymbol{\gamma}$) MAE(Mean Absolute Error) 결과이다. Error Accumulated-loop 크기가 1인 것은 One-step-ahead Prediction을 의미하고 8부터 256까지 Error Accumulated-loop 크기를 다양하게 하여 학습한 결과를 비교했다. 차량 슬립 각도는 Mini Error Accumulated-loop를 증가시켜 검증 환경과 더 가까운 환경에서 학습할수록 성능이 증가함을 알 수 있다. 반면에 Yaw Rate는 비슷한 추세를 보였지만 Error Accumulated-loop 크기 32에서 가장 좋은 성능을 보였다. Error Accumulated-loop 크기 32에서 256 사이의 Yaw Rate 결과는 매우 근소한 성능 차이이다.

3. 결 론

본 논문에서는 차량 슬립 각도와 Yaw Rate를 추정하는 Bicycle 모델과 Self-attention Neural Network를 융합한 시스템을 제안한다. 또한, Pure Prediction 상황에서 발생할 수 있는 누적 오차에 대비할 수 있는 Mini error accumulated-loop 학습 방법을 제안하였다. 실제 차량 주행 데이터에서 모델 학습하고 검증한 결과 효과적인 학습 방법임을 검증하였다. 추후에는 End-to-end Neural Network와 융합 모델과의 비교 실험을 통해 성능 검증을 수행할 계획이다.

참고 문헌

[1] Guo, H. Y., Cao, D. P., Chen, H., Lv, C., Wang, H. J,. Yang, S. Q. “Vehicle Dynamic State Estimation: State of the Art Schemes and Perspectives.” IEEE-CAA J. Autom. Sin. 418-431. 2018, May.

[2] Paden, B., C˘áp, M., Yong, S. Z., Yershov, D., Frazzoli, E. “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles.” IEEE Trans. Intell. Veh. 33-55. 2016, January.

[3] You, S.-H., Hahn, J.-O., & Lee, H. “New adaptive approaches to real-time estimation of vehicle sideslip angle.” Control Engineering Practice. Elsevier BV. 2009, December.

[4] Wang, Y., Geng, K., Xu, L., Ren, Y., Dong, H., & Yin, G. “Estimation of Sideslip Angle and Tire Cornering Stiffness Using Fuzzy Adaptive Robust Cubature Kalman Filter.” IEEE Transactions on Systems, Man, and Cybernetics: Systems. Institute of Electrical and Electronics Engineers (IEEE). 2022, March.

[5] Li, Y., Yin, G., Zhuang, W., Zhang, N., Wang, J., & Geng, K. “Compensating Delays and Noises in Motion Control of Autonomous Electric Vehicles by Using Deep Learning and Unscented Kalman Predictor.” IEEE Transactions on Systems, Man, and Cybernetics: Systems. Institute of Electrical and Electronics Engineers (IEEE). 2020, November.

[6] Kong, D., Wen, W., Zhao, R., Lv, Z., Liu, K., Liu, Y., & Gao, Z. “Vehicle Lateral Velocity Estimation Based on Long Short-Term Memory Network.” World Electric Vehicle Journal. MDPI AG. 2021, December 23.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. “Attention Is All You Need” (Version 7). arXiv. 2017.

[8] Sun, J., Shi, H., Zhu, J., Song, B., Tao, Y., & Tan, S. “Self-attention-based Multi-block regression fusion Neural Network for quality-related process monitoring.” Journal of the Taiwan Institute of Chemical Engineers. Elsevier BV. 2022, April.

[9] Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. “Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting.” Transactions in GIS. Wiley. 2020, June.

클러스터링 알고리즘에 대한 성능 비교 평가

문지원

애리조나주립대학교

hannah.jwmoon@gmail.com

Comparative Study on Performance of Clustering Algorithm

Moon Jiwon

Arizona State University

요 약

클러스터링 알고리즘은 방대한 데이터로부터 유의미한 지식을 추출하는데 중요한 방법으로 그 중요성이 확산되고 있다. 그러나, 많은 연구들이 기존 알고리즘을 개선하거나 새로운 알고리즘을 제시하는데 집중되어 있으며, 기존 알고리즘을 비교하여 적합한 알고리즘을 선택하는 가이드라인을 제시하는 연구는 제한적이었다. 본 연구는 다양한 클러스터링 알고리즘들의 성능을 비교하고, 이를 기반으로 알고리즘을 선택하는 가이드라인을 제시하고자 한다. 이를 위해, K-Means, DBSCAN, GMM, AC 알고리즘에 대해 3개의 데이터셋을 이용하여 SC, CHI, DBI와 수행시간을 측정하였다. 모든 클러스터링 알고리즘은 데이터 축약 방법을 적용하면 그렇지 않은 경우에 비해 성능이 향상되었다.

1. 서 론

개인이 데이터를 생산하고 소비하는 주체가 되면서, 어느때보다 다양하고 방대한 데이터가 빠르게 축적되는 빅데이터 시대에 살고 있다. 지난 수천년 동안 수집한 데이터에 숨겨져 있는 의미와 유용한 지식을 추출하여 가치로 연결시키려는 노력이 끊임없이 지속되어 왔다. 특히, 대량의 데이터가 축적되는 시대에 들어오면서, 방대한 데이터로부터 가치 있는 지식을 창출하는 데이터 분석 기술의 중요성이 날로 중요해지고 있다[1].

방대하게 축적된 데이터에 속성정보를 추가하는 것은 많은 시간과 비용이 든다. 속성정보를 갖고 있지 않은 데이터에서 통계적 규칙과 숨겨진 패턴을 파악하고 유사한 특징을 갖는 데이터들로 구분하여 의미 있게 활용하게 하는 방법이 클러스터링 알고리즘이다. 즉, 유사한 성격을 갖는 개체를 동일 그룹으로 묶는 클러스터링 알고리즘은 데이터를 분석할 때 인간의 주관적인 판단을 배제할 수 있어 분석 결과에 객관성과 신뢰성을 높여 주며, 데이터 마이닝, 패턴분류, 이미지 세분화 등 그 활용분야가 확산되고 있다[1]-[4].

클러스터링 알고리즘은 크게 계층적 클러스터링과 비계층적 클러스터링 방법으로 분류된다. 계층적 방법은 Bottom-Up 방법과 Top-Down 방법으로 분류되며, 비계층적 방법은 KMeans로 대표되는 중심기반 방법, DBSCAN으로 대표되는 밀도기반 방법과 Gaussian Mixture Model로 대표되는 분포기반 방법으로 구분된다. 다양한 클러스터링 알고리즘은 많은 연구로 인하여 선택의 폭이 넓으나, 적용하는 알고리즘에 따라 결과가 상이하게 나타난다[2]-[4]. 본 논문에서는 적합한 클러스터링 알고리즘 선택에 도움을 주고자 많이 사용되는 K-Means, DBSCAN(Density Based Spatial Clustering of Applications with Noise), GMM(Gaussian Mixture Model), AC(Agglomerative Clustering)의 성능을 비교하였다. 클러스터링 성능은 클러스터간 유사도를 측정하는 Silhouette Coefficient(SC), 클러스터간 및 클러스터내 분산 정도를 측정하는 Calinski-Harabasz Index(CHI), 각 클러스터와 가장 유사한 클러스터 간의 평균 유사성을 계산하는 Davies-Bouldin Index(DBI)와 클러스터를 계산하는데 소요되는 시간으로 비교하였다. 또한, 데이터셋이 클러스터링 알고리즘에 미치는 영향도를 분석하기 위하여, 기본 데이터셋과 UMAP (Uniform Manifold Approximation and Projection)과 PCA(Principle Components Analysis) 알고리즘으로 축약된 데이터셋을 이용하여 성능을 분석하였다.

결론적으로, 모든 알고리즘은 데이터 축약 기법을 적용하면 성능이 향상되는 것을 확인하였다. 본 논문은 서론에 이어 관련 연구를 기술하고, 3장에서 클러스터링 알고리즘과 성능 지표를 설명한다. 4장에서 실험 방법과 결과를 기술한다. 마지막으로, 5장에서 결론을 맺는다.

2. 관련연구

그 동안 클러스터링 알고리즘의 성능을 개선하거나 새로운 알고리즘을 제안하는 연구와 기존 알고리즘을 비교하는 연구들이 수행되어 왔다. 본 연구에서는, 기존 알고리즘을 비교 분석하는 연구에 대해 설명한다.

Rodriguez등은 9개 클러스터링 알고리즘(KMeans, EM, optics, clara, hierarchical, hcmodel, spectral,

subspace, dbscan) 중 어떤 방법이 적합한지 탐색하였다. 클래스 수, 클래스 간 간격 등을 조정하여 데이터 특성을 변형시키며 알고리즘의 민감도를 평가하였다. 그 결과 spectral 알고리즘이 전반적으로 우수했으며, 디폴트 구성이 항상 성능이 좋지 않으며, 매개변수를 랜덤하게 선택하는 것만으로도 성능이 향상되는 것을 발견하였다[2].

Shah은 6개의 서로 다른 데이터셋을 이용하여 6개 클러스터링 알고리즘(Canopy, EM, KMeans Farthest First, Hierarchical Cluster, Make density based Cluster)에 대해 소요시간, 생성된 클러스터 수, 클러스터 분포와 반복값을 측정했다. 그 결과, 짧은 수행 시간과 정확도가 높은 KMeans가 상대적으로 우수하였으며, EM 알고리즘이 가장 부정확하고 많은 시간이 소요되었다. 또한, 계층적 알고리즘은 데이터 크기에 민감하여 작은 데이터셋에는 적합하나, 대규모 데이터셋에는 부적합하였다. 이외의 알고리즘들은 비슷한 성능을 보였으며, 밀도 기반 클러스터링 알고리즘은 밀도가 다양한 경우 성능이 저하되었다[3].

Shahriar는 클러스터링 알고리즘을 수행하는 플랫폼의 처리 능력을 비교하기 위하여, KMeans와 DBSCAN 클러스터링 알고리즘을 Python, Matlab, R, Wolfram Mathematica 등 4개의 플랫폼에서 수행하여, 수행시간과 클러스터 결과의 정확도를 비교하였다. K-Means에 대해서는 Matlab, R, Python, Wolfram 순이었고, DBSCAN의 경우는 Matlab, Python, R, Wolfram 순으로 분석되었다. 클러스터 결과는 플랫폼간에 차이가 미비하였다[4].

3. 클러스터링 알고리즘

3.1 기본 모델

클러스터 분석은 속성 정보를 모르는 데이터를 물리적 또는 추상적으로 특성이 유사하거나 동질인 동일한 그룹으로 묶어 주는 비지도 학습 기반 데이터 분석 방법으로, 클러스터내 응집도와 클러스터간 분리도를 최대화하도록 데이터를 분석하는 방법이다. 클러스터링 알고리즘은 크게 계층적 클러스터링과 비계층적 클러스터링 방법으로 구분되며, 계층적 방법은 Bottom-Up 방법과 Top-Down 방법으로 구분된다. 또한, 비계층적 클러스터링 방법은 중심기반 방법, 밀도기반 방법 및 분포기반 방법으로 구분된다. 본 논문에서 고려한 클러스터링 알고리즘의 특성은 표 1과 같다.

클러스터 분석은 숨겨진 유의미한 데이터의 구조를 파악하거나 이해하기 위해 분석 초기 탐색적 분석 단계에서 아주 많이 활용된다. 최근에, 클러스터 분석을 통해 데이터를 세분화하여 데이터 묶음을 만든 후, 분류 방법으로 분석된 라벨이 적은 데이터셋을 이용하여 라벨이 없는 대량의 데이터셋을 분류하는 연구가 확산되고 있다. 예를 들어, 클러스터링 방법으로 이미지 특징을 세분화한 후, 라벨링 되어 있는 적은 이미지 데이터셋을 이용하여 방대한 이미지를 분류하는 준지도 학습 방법이 생성형 AI 기술의 부상과 더불어 새롭게 주목받고 있다. 따라서, 클러스터링 알고리즘의 성능이 클러스터 분석 후에 이루어지는 분류 정확도에 많은 영향을 미칠 수 있어, 적절한 클러스터링 알고리즘 선택이 중요해지고 있다.

표 1 클러스터링 알고리즘 장단점 비교

	K-Means	DBSCAN	GMM	AC
클러스터링 방법	중심기반 (중심점-데이터 사이 거리)	밀도 기반 (근접 데이터)	분포기반 (가우시안분포)	상위-하위기반 (근접 데이터)
클러스터 수	지정/ 제약	비지정/ 제약없음	지정/ 제약없음	지정/ 제약없음
초기화	O	X	O	X
데이터 밀도	X	O	X	O
고차원 데이터	O	O	X	O
클러스터 분포/모양	선형/ 원형	비선형/ 복잡	가우시안/ 복잡	선형,비선형/ 다양
Outlier	Weak	Robust	Weak	Robust
계산부하	적음	낮음	높음	높음

3.2 클러스터 평가 지표

클러스터링 알고리즘에 대한 평가는 클러스터가 의미있고 견고한지 판단하는데 중요하다. 클러스터 품질은 클러스터내 또는 클러스터간 변동이나 클러스터 유효성 지수 등을 이용하여 평가할 수 있다. 이러한 지수는 SC, CHI, DBI 등으로, 이들 지표를 이용하여 클러스터링 알고리즘과 매개변수 설정을 비교하고 데이터에 적합한 클러스터링 알고리즘을 선택할 수 있다. 또한, 이들은 클러스터링 결과의 유효성과 신뢰성을 보장하여 데이터 기반 의사 결정을 내리는데 도움을 줄 수 있다. 클러스터 성능 평가 지표는 표 2와 같다.

표 2 클러스터 성능 평가 지표

지수	세부 설명
SC	▪클러스터 분리 우수성 평가(클러스터내 응집도, 클러스터간 유사성) ▪1에 가까울수록 최적(0은 적정 클러스터 없음)
CHI	▪클러스터간 분산과 클러스터내 분산 비율 (클러스터내 데이터 밀집도, 클러스터 간 거리) ▪높은 점수가 최적(클러스터 수에 비례할 수 있음)
DBI	▪클러스터내 거리와 클러스터간 거리의 비율의 유사성 측정 ▪작은 값이 최적
Time	▪클러스터링 알고리즘을 수행하는데 소요된 시간

4. 실험 방법 및 성능 평가

4.1 데이터셋

캐글에 공개되어 있는 데이터셋 중에서, 클러스터 분석에 적합하며 element 수가 많은 데이터셋(credit card)과 적은 데이터셋(iris) 및 클러스터 분석에 적합하지 않은 데이터셋(wine)과 같이 3개 데이터셋을 이용하여 앞서 언급한 4개 클러스터링 알고리즘의 성능을 비교하였다. 0.8 이상이면 클러스터링 성향이

높다고 판단할 수 있는 Hopkins Score를 이용하여 데이터셋에 대한 클러스터 성향을 측정한 결과, Iris와 Credit Card 데이터셋은 클러스터에 적합한 반면, Wine 데이터셋은 다소 랜덤하게 분포되어 앞선 두 데이터셋에 비해 적합성이 떨어지는 것으로 판단되었다. 또한 데이터셋의 분산을 구해 본 결과 3개의 데이터셋은 비선형 특성을 갖는 것으로 판단되었다. 본 연구에서 사용한 데이터셋 특성은 표 3과 같다.

표 3 사용한 데이터셋 특성

	Element	Feature	Hopkins Score	특성
Iris	150	4	0.8270	균일, 비선형
Wine	178	13	0.6348	약간 균일, 비선형
Credit Card(CC)	8950	17	0.7938	균일, 비선형

4.2 성능 평가 방법

앞서 언급한 3개의 원본 데이터셋에 4개의 클러스터링 알고리즘(K-Means, DBSCAN, GMM, AC)을 적용하여 데이터셋 특성에 따른 클러스터링 알고리즘의 성능을 비교하였다. 추가적으로 UMAP과 PCA를 적용하여 축약된 2개의 데이터셋을 추가로 적용하여 데이터 축약 방법이 클러스터 결과에 미치는 영향도 분석하였다. 본 연구에서는 ①K-N(KMeans-Normal), ②K-U(KMeans-UMAP), ③K-P(KMeans-PCA), ④D-N(DBSCAN-Normal), ⑤D-U(DBSCAN-UMAP), ⑥D-P(DBSCAN-PCA), ⑦G-N(GMM-Normal), ⑧G-U(GMM-UMAP), ⑨G-P(GMM-PCA), ⑩A-N(AC-Normal), ⑪A-U(AC-UMAP), ⑫A-P(AC-PCA) 등 12개의 조합에 대해 SC, CHI, DBI, 수행시간을 측정하였다.

4.3 실험결과

클러스터링 알고리즘의 성능을 평가한 결과는 그림 1과 같다. 데이터 차원을 축약할 경우, 노이즈가 줄어들고 데이터 패턴이 명확해져 모든 클러스터링 알고리즘의 성능이 데이터를 차원을 축약하지 않은 경우에 비해 향상되었다. 특히, 사용한 3개의 데이터셋이 비선형 특성을 갖고 있어, 모든 알고리즘에서 UMAP 축약을 적용한 경우 성능이 가장 좋았다. 단, CC 데이터셋은 UMAP 축약 후 데이터 밀도가 떨어지거나 노이즈로 인해 데이터 구조가 왜곡될 수 있어 DBSCAN의 경우 오히려 성능이 하락하였다. KMeans는 데이터셋의 클러스터링 성향이 크면 데이터 축약 방법이 전반적인 성능 향상에 미치는 영향이 크지 않았으나, 클러스터링 성향이 좋지 않은 경우에는 데이터셋의 분포를 고려한 데이터 축약으로 성능이 향상되는 것을 알 수 있었다. 또한, GMM, AC의 경우 데이터 축약을 적용하면 성능 향상을 볼 수 있었다. 특히 실험에 사용한 데이터셋의 비선형 특성으로 인해 UMAP을 적용한 경우 좋은 성능 나타냈다. 그리고, Iris와 같이 데이터의 클러스터링 특성이 크고 데이터셋이 크지 않을 경우, 알고리즘이나 데이터 축소에 의한 성능 차이는 크지 않았다. AC는 데이터셋의 크기가 큰 경우보다 적은 경우에 좋은 성능을 보였다. KMeans나 DBSCAN은 데이터셋의 element 수에 수행시간이 많은 영향을 받지 않으나, GMM이나 AC는 데이터셋 크기에 따라 수행시간이 많이 소요되는 것으로 나타났다.

결론적으로, 잘 정의된 클러스터를 얻기 위해서는, PCA와 UMAP 등을 적용하여 데이터 차원을 축소한 후 클러스터링 알고리즘을 적용하는 것이 우수한 성능을 얻을 확률이 높은 것을 알 수 있었다 특히, 데이터셋 형태가 비선형일 경우 UMAP을 적용하는 것이 성능 향상에 도움이 될 것이다. 또한, 데이터셋이 큰 경우는 DBSCAN을 우선적으로 적용하는 것도 좋은 접근이다.

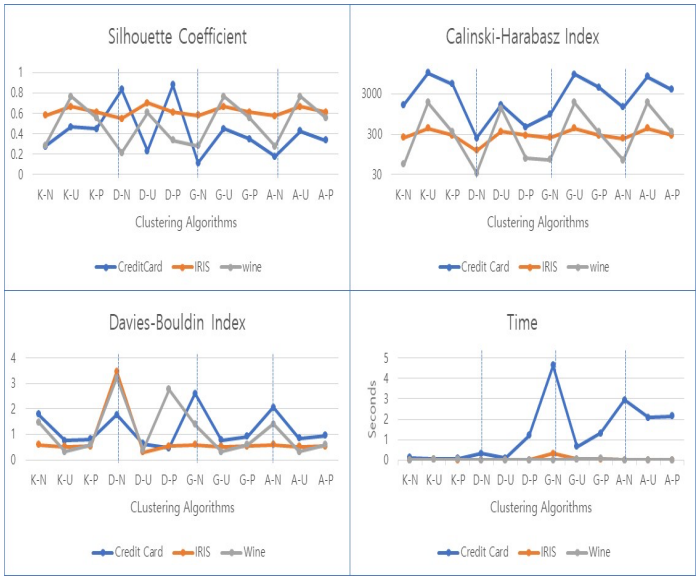


그림 1 클러스터 성능 비교 결과

5. 결론

본 연구에서는 KMeans, DBSCAN, GMM, AC 4개의 클러스터링 알고리즘의 성능을 3개의 다른 데이터셋을 이용하여 비교하였다. 알고리즘 성능은 SC, CHI, DBI와 수행시간을 측정하여 비교하였다. 결론적으로, 우수한 성능을 얻기 위해서는 PCA와 UMAP을 적용하여 데이터 차원을 축소한 후 클러스터링 알고리즘을 적용하는 방법을 제안하며, 클러스터 특성이 좋으면 우선적으로 DBSCAN 알고리즘을 적용해보는 것도 좋다.

향후에는 클러스터링 알고리즘의 더 나은 이해와 분석을 위하여 다른 클러스터링 알고리즘과 상이한 클러스터 성향을 갖는 데이터셋을 이용하여 성능 비교를 확장하고, 매개 변수를 다변화시켜 매개 변수에 의한 영향도 분석할 계획이다. 또한, 클러스터링 알고리즘 수행에 필요한 초기값이 알고리즘 성능에 미치는 영향도 분석하고자 한다.

참고문헌

[1] K. Bindra, A. Mishra, “A Detailed Study of Clustering Algorithms,” 6th International Conference on Reliability, Infocom Technologies and Optimization, pp. 370-376, Sep. 20-22, 2017

[2] Rodriguez MZ, Comin CH, Casanova D, Bruno OM, Amancio DR, Costa LdF, et al, “Clustering algorithms: A comparative approach,”, PLOS, January 15, 2019

[3] S. H. Shah, M. J. Iqbal, M. Bakhsh and A. Iqbal, “Analysis of Different Clustering Algorithms for Accurate Knowledge Extraction from Popular DataSets,” Information Sciences Letters, No. 1, pp. 21- 31, 2020

[4] N. Shahriar, S. M. A. Al Faisal, Md. M. Pinjor, Md. A. S. Zobayer Rafi, A. R. Sarkar, “Comparative Performance Analysis of K-Means and DBSCAN Clustering algorithms on various platforms,” 22nd International Conference on Computer and Information Technology, pp. 18-20 Dec. 2019

문제 기술서에서 자연어 처리와 구조화를 이용한 요구사항 자동화

백영윤 ○ 박용범

단국대학교

youngyunbaek@dankook.ac.kr, ybpark@dankook.ac.kr

Utilizing Natural Language Processing and Structuring for Requirement Automation in Problem Descriptions

Young yun Baek ○ Yong Bum Park

Dankook University

요 약

요구 분석이 중요하지만, 요구사항 분석은 여전히 어렵고 복잡하며 분석하는 사람마다 각각의 분석 내용을 만들어 낸다. 또한 자연어 요구사항 분석은 자동화되지 않고 시간과 노력이 많이 사용되며 오류 발생의 문제가 있다. 따라서 부족한 요구사항 분석과 자동화되지 않은 요구사항 분석 시스템은 소프트웨어 개발에서 많은 문제를 만들어 낸다. 이러한 문제를 해결하기 위해서 본 논문에서는 요구사항 문제 기술서에서 ChatGPT 기반의 요구사항 구조화 분석 프로그램을 만들고 자연어 처리와 거시구조 개념을 적용하여서 요구사항 분석을 가능하게 하고 이를 온톨로지로 구조화하여서 유즈케이스 다이어그램과 유즈케이스 스펙을 만들었다. 이를 통해 자연어 요구사항 문서 분석을 가능하게 하고 자동화된 시스템을 제공하여서 요구사항 분석 결과를 분석할 수 있는 구조화 방법을 제안한다.

1. 서 론

요구 분석 방법은 소프트웨어 개발 프로세스에 있어서 사용자가 요청하는 요구사항을 효과적으로 얼마나 적용할 수 있는가에 대한 중요한 단계이다. 따라서 요구 분석 방법은 소프트웨어 개발에서 사용자가 요청하는 소프트웨어를 만드는데 있어서 중요하고 소프트웨어 개발 프로세스에서 중요한 과정이다. 하지만 요구사항 분석 방법에 있어서 여러 어려움이 존재하는데 문제 기술서가 자연어로 작성되어 있기 때문이다. 자연어로 작성된 문제 기술서는 일반적으로 정리된 요구사항이 아닌 문장 사이에 함축되어 있고 정형화되지 않은 상태로 작성된다. 이러한 문제는 이해관계에 따라서 문장을 다르게 이해할 수 있고 동일한 단어라도 다른 의미로 사용하고 이해할 수 있다. 이런 상황은 동일한 의미의 요구사항 문장을 분석하더라도 상황 따라 다른 결과의 결과물을 만들어 낼 수 있다. 즉 자연어로 작성된 문제 기술서를 보고 요구사항 분석을 하는 분석가의 능력과 분석가가 얼마큼의 노력과 시간으로 요구사항 분석을 하는가에 따라서 요구사항 분석 결과가 달라지고 각 분석가의 목적, 프로젝트 성격, 담당 부서, 비즈니스의 성격에 따라서 다른 분석 결과물을 나타낼 수 있다[1-8].

본 논문에서는 이러한 문제를 해결하기 위해서 문제 기술서 단계의 문서를 ChatGPT 기반의 프로그램을 통해서 유즈케이스 생성을 위한 답변 들을 추출하고 이 답변들을 자연어 처리를 통해 파악한다. 분석된 의미 내용을 기반으로 거시구조 개념을 적용한 전체 요구사항 텍스트, 요구사항 문장, 명제 단어로 도출하고 거시구조 개념을 사용한 요구사항 관계성을 확보한다. 이러한 거시구조 요소를 요구사항 작성을 위한 요소로 대치하여 구조화한다. 구조화된 데이터를 온톨로지로 구성하고 이 데이터를 통해서 사용자에게 유즈케이스 다이어그램과 유즈케이스 상세 설명을 분석할 수 있도록 작성해 준다.

이를 통해 자연어로 된 복잡한 문제 기술서에서 일관된 요구사항을 도출할 수 있으며, 사용자에게 ChatGPT를 통한 요구사항 분석 과정을 통해서 자동으로 요구사항을 도출할 수 있도록 제공함으로써 인적 작업을 해결하는 방법을 제안한다.

2. 관련 연구

Zhang et al[1]에서는 그래프 기반 순위 알고리즘을 기반으로 자동화된 요구사항 용어 추출 및 순위 프레임워크를 제안하였다. Wang et al[2]에서는 NLP에서 테스트 및 검증을 위한 모델 생성에서 물리적 환경과 사용자와의 상호 작용을 모델링(Modeling)하는

문제를 해결하기 위해서 NLP 기술과 모델 매핑(Mapping) 규칙을 사용하여서 모델 요소를 식별하였다. Limaylla-Lunarejo et al[3]에서는 소프트웨어 요구 사항을 기능적 및 비기능적 분류로 분류하기 위해 자연어 처리와 결합한 ML 알고리즘을 적용하여서 분류하였다. Güneş et al[4]에서는 자연어 처리 파이프라인과 휴리스틱을 사용하여 사용자 스토리에 의존하는 목표 모델을 자동으로 생성하고 시각화하였다. Hey et al[5]에서는 요구사항을 분류하기 위해 BERT의 미세 조정 메커니즘을 활용하는 접근 방식인 NoBERT를 제시하였다. Tiwari et al[6]에서는 엔터티 인식 NLP 기술을 사용하여서 텍스트로 작성된 요구사항 사양에서 유즈케이스 이름과 행위자 이름을 식별하는 접근 방식을 제안하였다. Deshpande et al[7]에서는 도메인 온톨로지와 라벨링 학습을 통해서 요구사항 종속성을 추출하였다. Wardhana et at[8]에서는 온톨로지를 사용하여 시스템 설계에서 의미론적 컨텍스트를 나타내는 방법을 제안하였다.

순차적 분석으로 요구사항 분류 및 엑터 행위 추출에는 강점이 있으나 이와 관련한 요구사항들의 관계성을 확보하는 데 단점이 있다. 이러한 단점은 요구사항 분석에서 한 문장에서의 분류와 추출은 가능하나 전체적인 요구사항의 문서에서 보았을 때는 전체 요구사항들의 관계와 구성을 알 수 없으므로 요구사항의 흐름을 파악하는 데는 단점이 있다. 계층형 분석의 경우 요구사항들의 상·하위 관계성을 파악함으로써 각각의 요구사항 간의 종속성과 계층 관계를 파악할 수 있는 장점이 있다. 하지만 계층형 분석만 수행하는 경우 단순한 요구사항 간의 비교만 가능하고 문장 내의 단어를 분석하지 못하기 때문에 각 단어의 관계를 파악하지 못한다는 단점이 있다. 또한 자동화를 지원하지 않는 논문들도 대다수이었는데, 자동화를 지원하지 않는 경우 사용자가 분석을 수행하면서 인적 오류가 발생할 수 있는 문제점이 있으며 요구사항을 분석함에 있어서도 시간적, 물리적 자원이 사용되기 때문에 요구사항 분석에 있어서 비효율적이다.

3. 거시구조 기반 요구사항 구조화 시스템

본 논문에서 제안하는 전체 요구사항 구조화 시스템을 간단하게 표현하면 그림 1과 같다. 제안하는 시스템은 요구사항 분석 단계에서 자연어로 작성된 요구사항 명세서를 분석하여서 사용자가 요구사항을 분석할 수 있는 모델이다.

기존의 요구사항 분석 모델의 경우 요구사항을 분석하기 위한 패턴 적용에 있어서 해당 도메인에 대한 전문적인 지식이 있어야만 분석하는 방법들이 대다수이기 때문에 해당 분야에 지식이 없는 경우 요구사항 분석 모델 사용에 있어서 제한이 있을 수 있다. 이러한 문제를 해결하기 위해서 ChatGPT의 생성형 인공지능을 활용하여서 사용자는 별도의 분석 없이 문제 기술서를 분석하여서 엑터와 유즈케이스를 분석할 수 있다. 이러한 ChatGPT의 생성형 인공지능은 사용자가 별도의 도메인 지식이 필요하지 않기 때문에 도메인 지식 습득을 위한 시간과 노력을 줄일 수 있고 또한 분석을 위해 해당 도메인에 경험이 많은 분석가를 통해서 분석하지 않아도 되므로 소프트웨어 프로세스에서 요구분석 작업에 걸리는 시간과 비용을 줄일 수 있다.

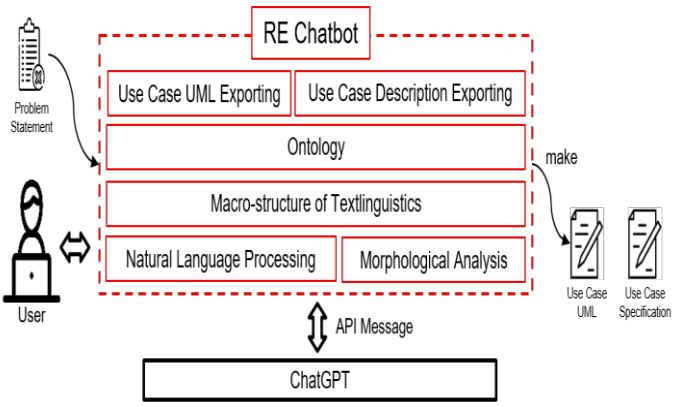


그림 1. Overview Macrostructure-based requirements structuring system

ChatGPT의 응답 메시지는 사용자가 바로 사용할 수 있지만 해당 메시지를 효과적으로 사용하기 위해서는 응답 메시지에서 필요한 내용을 찾고 분류하여서 데이터를 구성해야 한다. 따라서 특징적 요소를 추출하기 위해서는 형태소 분석과 자연어 처리를 통한 분석 과정이 필요하다.

예를 들어 “Allow students to view report cards”라는 문장이 있는 경우 형태소 분석을 통해서 “students”라는 엑터와 “view report cards”라는 행위로 분석할 수 있다. 또한 ChatGPT를 통해 명세서 내에서 존재하는 액터와 행위를 질문하여서 이중으로 확인하는 작업을 수행한다. 위와 같이 도출된 엑터와 행위는 거시 구조를 구성하는 요소가 된다.

이를 통해서 사용자는 ChatGPT의 응답으로 보내 준 메시지를 별도의 저장과 처리 과정 없이 특징적 요소를 추출할 수 있고 이를 통해서 ChatGPT가 응답 메시지로 보내 준 메시지에서만 언어 처리를 수행함으로 오류 발생의 가능성을 줄일 수 있다.

요구사항 구조화 모델의 구조화 방법은 텍스트 언어학의 거시구조 개념을 적용하여서 구성하였다. 거시구조는 문장 혹은 텍스트 전체를 기초로 하거나 큰 단위에 기초를 두고 거시적인 관점으로 연관성이 무엇인지 파악하는 구조이다. 즉, 텍스트는 단어와 단어 혹은 문장과 문장들이 결합하여 텍스트를 만들고, 이 텍스트들이 결합하여 상위의 텍스트를 만들어 내거나, 언어적 요소 또는 상황 등의 비언어적 단어들이

결합하여 텍스트를 만든다. 텍스트 언어학의 거시구조 개념을 사용하여 자연어로 된 텍스트에 있는 의미 간의 관계를 분석하여서 의미 단어들을 도출할 수 있다. 도출한 의미 단어들은 거시 명제를 도출하기 위한 필요 정보가 된다. 즉, 요구사항 분석 관점에서 거시구조 개념을 적용한다면 전체 요구사항 텍스트에서 요구사항 분석 요소를 위한 엑터와 행위인 의미 단어를 도출하는 과정이라고 할 수 있다. 이를 예로 들면 그림 2와 같은 과정을 수행한다.

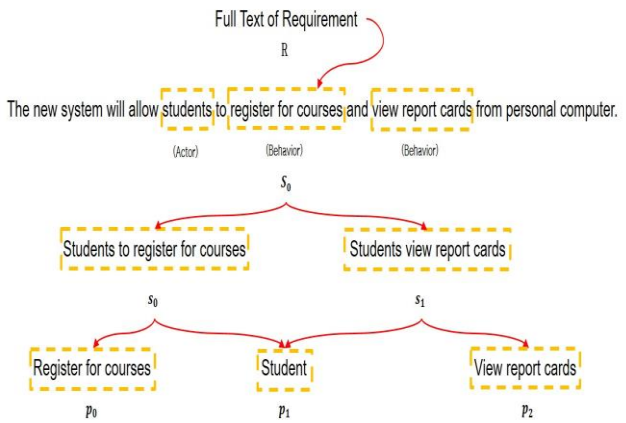


그림 2. Requirements analysis process using macrostructure concepts

이를 통해 분석된 엑터와 의미 단어를 통해서 관계성을 가지는 유즈케이스 다이어그램을 구성한 후, 사용자가 필요한 엑터와 유즈케이스를 선택하면 상세한 유즈케이스 설명을 제공하여서 분석 시간을 감소시키는 방법을 사용하고 있다. 즉, 이전 단계에서 생성된 엑터, 유즈케이스를 본 논문에서 구현하고 있는 자연어 처리 기술과 형태소 분석 알고리즘을 통해서 행위를 추출하고 이때, 추출된 행위와 엑터를 거시구조 개념을 통한 관계 분석을 통해서 관계성을 가지는 유즈케이스를 구성할 수 있다.

요구사항 구조화 모델을 통해 관계성을 가지는 유즈케이스를 구성한 후 해당 관계성을 효과적으로 사용하기 위해서 관계형 데이터 구조인 온톨로지를 사용하여 구성한다. 즉, 온톨로지 데이터 구성은 거시구조 개념에서 도출되는 거시 명제, 요구사항 문장, 대상, 대상의 행위나 기능, 행위나 기능의 특성, 하위 개념의 의미 명제를 가지고 구성하며 사용자는 관계성을 가지는 요구사항을 자동으로 구성할 수 있으며, 이미 요구사항 명세서 기반으로 도출된 엑터, 유즈케이스만 사용함으로써 일관성도 유지할 수 있다.

최종적으로 구조화된 온톨로지를 통해서 유즈케이스 다이어그램과 유즈케이스 설명을 포함하는 유즈케이스 스펙을 도출한다. 온톨로지로 도출된 UML 정보는 거시구조 개념 때문에 하나의 요구사항으로 다루어지고 각각의 요구사항 또한 관계성을 가지고 있으므로 온톨로지 구조가 요구사항 명세서에 대한 하나의 요구사항 내용을 구조화한다.

4. 요구사항 명세서 문서를 적용한 사례 연구

제안된 시스템을 검증하기 위해서 여러 요구사항 문서 중 대출 시스템에 관한 요구사항 명세서를 선택하여서 논문에서 제안하는 요구사항 구조화 시스템에 적용하여 시스템에 대한 검증을 진행하였다.

4.1 엑터 요청을 통한 명제 단어 구성

엑터 추출을 위하여 시스템 내에서 ChatGPT의 쿼리로 다음과 같은 쿼리를 작성한다. “Text에서 요구사항을 추출해서 알려줘”, “Text에서 엑터를 추출해서 알려줘”

위의 ChatGPT 응답을 통해 문장들은 필요한 정보만 사용할 수 있도록 자연어 처리를 통해 필요 없는 문장과 단어들을 제거하고 요구사항 분석을 위해 필요한 정보를 도출하여 리스트 형태로 저장하여 사용한다. 해당 정보들은 명제 단어 요소로써 사용된다.

4.2 요구사항 요청을 통한 요구사항 문장 구성

요구사항 추출을 위한 기본적인 내용을 추출한 뒤, 시스템 내에서 ChatGPT의 쿼리로 다음과 같은 쿼리를 작성한다. “Text, 엑터에서 행위를 추출해서 알려줘”, “Text, 엑터, 행위에서 유즈케이스를 추출하고 유즈케이스의 설명을 추출해서 리스트로 알려줘₩n 유즈케이스 추출:₩n유즈케이스 설명”

위의 ChatGPT 응답을 통해 문장들을 필요한 정보만 사용할 수 있도록 자연어 처리를 수행한다. 자연어 처리를 수행하면 요구사항 문장들을 구성하여 요구사항 텍스트 요소로써 사용된다. 이 요소들을 이전 단계에서 구성한 명제 단어와의 관계성을 구성하여서 요구사항을 구성하게 된다. 예를 들면 명제 단어 “Buyer”는 “Search for a Home” 이라는 요구사항 텍스트와 관계성을 가지게 되고 이를 통해서 요구사항으로써 구조화 할 수 있다.

4.3 요구사항 상세 설명 요청을 통한 요구사항 상세 설명 추출

상세 요구사항 추출을 위한 기본적인 내용을 추출한 뒤, 시스템 내에서 ChatGPT의 쿼리로 다음과 같은 쿼리를 작성한다. “Text에서 유즈 케이스가 Search for a Home이고 Active 엑터가 Buyer일 때 Basic flow와 Alternative flow를 알려줘”

위와 같은 과정을 통해서 ChatGPT를 통해 상세 요구사항 명세서를 도출 할 수 있으며 이는

유즈케이스의 상세 설명으로써 사용할 수 있다.

4.4 거시구조 구조화를 통한 유즈케이스 UML, 유즈케이스 Spec 생성

거시구조 구조화를 통한 온톨로지를 활용하여서 XML로 변환하였다. 또한 논문에서 제공하는 방법을 통해 분석된 요구사항 분석 결과를 하나의 문서로 정리하여 작성하였다. 이렇게 변환한 XML을 UML UI 시각화 프로그램을 통해서 요구사항 다이어그램으로 만들었고 각각의 엑터와 요구사항들의 관계성과 상세한 설명을 포함하는 텍스트 문서를 구성하여서 사용자에게 제공한다.

5. 결론

본 논문에서는 문제 기술서에 대해 ChatGPT를 통한 프롬프트 엔지니어링을 통해서 문장을 분석하고 이를 거시구조 개념을 통해 요구사항을 도출하고 별도의 프로그램을 이용하여 사용자의 선택을 통해서 관계성을 가지는 요구사항 항목을 제안하는 연구를 수행하였다.

ChatGPT의 질문 응답을 통해서 요구사항 분석을 위한 데이터를 추출할 수 있었으며, 문제 기술서에서 엑터와 행위를 추출하였다. 그리고 이 데이터를 거시구조 개념을 이용하여 구조화하고 온톨로지로 구성하여 요구사항 명세서에서 요구사항을 도출하는 방법을 연구하였다. 사용자는 별도로 제작한 프로그램을 활용하여서 문제 기술서를 프로그램에 입력하고 사용자의 메뉴 선택을 통해 요구사항 명세서 입력의 결과물로 유즈케이스 다이어그램과 유즈케이스 상세 스펙을 자동으로 분석할 수 있다.

ChatGPT를 통한 요구사항 분석 방법에서 ChatGPT에서 발생하는 문제점인 응답 메시지의 시행에 따른 다른 메시지가 전달되는 문제점과 메시지의 형식이 다른 부분에 대한 문제가 존재하고 ChatGPT에 의존하는 시스템이기 때문에 아직까지 시스템 안정성에 대한 우려가 있다.

또한 ChatGPT를 통해 얻어진 엑터, 요구사항, 요구사항 명세 등에 대해서 얼마나 요구사항 커버리지를 만족하는지와 생성된 내용에 대한 신뢰성 검증 또한 추가적인 연구를 통해 해결을 해야 할 문제점이다.

6. 참고 문헌

[1] Zhang, J., Chen, S., Hua, J., Niu, N., & Liu, C. (2022, August). Automatic terminology extraction and ranking for feature modeling. In 2022 IEEE 30th International Requirements Engineering Conference (RE) (pp. 51-63). IEEE.

[2] Wang, C., Hou, L., & Chen, X. (2022, August). Extracting Requirements Models from Natural-Language Document for Embedded Systems. In 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW) (pp. 18-21). IEEE.

[3] Limaylla-Lunarejo, M. I., Condori-Fernandez, N., & Luaces, M. R. (2022, August). Towards an automatic requirements classification in a new Spanish dataset. In 2022 IEEE 30th International Requirements Engineering Conference (RE) (pp. 270-271). IEEE.

[4] Güneş, T., & Aydemir, F. B. (2020, August). Automated goal model extraction from user stories using NLP. In 2020 IEEE 28th International Requirements Engineering Conference (RE) (pp. 382-387). IEEE.

[5] Hey, T., Keim, J., Koziolek, A., & Tichy, W. F. (2020, August). Norbert: Transfer learning for requirements classification. In 2020 IEEE 28th International Requirements Engineering Conference (RE) (pp. 169-179). IEEE.

[6] Tiwari, S., Rathore, S. S., Sagar, S., & Mirani, Y. (2020, August). Identifying use case elements from textual specification: A preliminary study. In 2020 IEEE 28th International Requirements Engineering Conference (RE) (pp. 410-411). IEEE.

[7] Deshpande, G., Motger, Q., Palomares, C., Kamra, I., Biesialska, K., Franch, X., ... & Ho, J. (2020, August). Requirements dependency extraction by integrating active learning with ontology-based retrieval. In 2020 IEEE 28th International Requirements Engineering Conference (RE) (pp. 78-89). IEEE.

[8] Wardhana, H., Ashari, A., & Sari, A. K. (2020). Transformation of sysml requirement diagram into owl ontologies. International Journal of Advanced Computer Science and Applications, 11(4).

오픈스택 기반 컨테이너 인프라 관리 방법 및 시스템

[1] 정수민 ○, [2] 박준석, [3] 염근혁*

부산대학교 [1] 정보융합공학과, [2] 지능물류빅데이터연구소, [3] 정보컴퓨터공학부

{sumin2708, pjs50, yeom*}@pusan.ac.kr

Method and System for Container Infrastructure Management based on Openstack

[1]Sumin Jeong○, [2]Joonseok Park, [3]Keunhyuk Yeom*

[1] Department of Information Convergence Engineering, Pusan National University

[2] Research Institute of intelligent Logistics Big data, Pusan National University

[3] School of Computer Science and Engineering, Pusan National University

요 약

클라우드 컴퓨팅을 활용하는 클라우드 서비스는 클라우드 네이티브(Cloud-Native) 패러다임이 적용된 컨테이너 기반 구성이 가능하도록 발전하고 있다. 또한, 클라우드 플랫폼은 점점 클라우드 벤더 의존적인 단일 클라우드 플랫폼을 활용하는 환경에서 벗어나 하이브리드 클라우드, 멀티 클라우드 등의 다양한 형태의 플랫폼 융합을 통해 클라우드 서비스를 제공하고 있다. 그러나, 클라우드 네이티브 지원에 대해서 퍼블릭 클라우드 플랫폼인 AWS(Amazon Web Service), GCP(Google Cloud Platform) 등의 자체적인 컨테이너 관리 기술 혹은 쿠버네티스 연계 기술을 제시하고 발전시키고 있으나, 프라이빗 클라우드 플랫폼인 오픈스택, 클라우드 스택 등은 컨테이너 관리 기술 체계를 정립하지 못하고 가상머신 위주의 서비스 제공기술에 집중하고 있다. 따라서, 본 논문에서는 오픈스택 클라우드 플랫폼을 활용하여 쿠버네티스 클러스터를 관리하는 컨테이너 인프라 관리 방법 및 구현 시스템을 제안한다. 제안한 방법을 검증하기 위해 대시보드 활용 사례 연구를 제시하여 적용성을 검증하며, 제안 방법과 기존 방법의 컨테이너 클러스터 구축 비교를 통해 제안 방법 및 시스템의 우수성을 검증한다.

1. 서 론

클라우드 플랫폼은 가상머신, 컨테이너 등을 통해 IT 서비스를 배포하고 관리하는 소프트웨어 프로젝트의 기반 기술로 활용된다. 클라우드를 활용하는 소프트웨어 프로젝트는 다양해지는 사용자 요구사항을 반영하고, 견고한 소프트웨어 개발을 위해 클라우드 네이티브(Cloud-native) 패러다임[1]을 적용하여 클라우드 플랫폼에서 자체적인 개발-배포-관리 사이클이 동작하도록 발전하고 있다.

클라우드 플랫폼에서 클라우드 네이티브를 적용하는 대표적인 방법은 컨테이너 구조를 적용하고 이들을 관리하는 쿠버네티스(Kubernetes), 도커(Docker) 등의 컨테이너 오케스트레이션 프레임워크를 주로 활용한다. 이를 적용하기 위해 다양한 클라우드 플랫폼에서는 ECS(Amazon Elastic Container Service), AKS(Azure Kubernetes Service) 등의 컨테이너 관리 솔루션을 제공하고 있다.

그러나, 프라이빗 클라우드 컴퓨팅 플랫폼인 오픈스택, 클라우드 스택 등은 컨테이너에 대한 지원보다

는 가상머신을 기반으로 하는 컴퓨팅 컴포넌트에 집중하고 있다. 최근 들어 마이크로서비스 등의 클라우드 네이티브 애플리케이션 개발이 부각되고 있지만 프라이빗 클라우드 플랫폼에서는 이를 지원하는 체계가 정립되어 있지 않다. 이는 프라이빗 구축시 오픈소스를 활용한 클라우드 컴퓨팅 환경의 활용도를 높이지 못하고, 플랫폼 확산을 저해하는 요인이 될 수 있다.

따라서, 본 논문에서는 오픈소스 클라우드 컴퓨팅 플랫폼인 오픈스택을 적용하여 클라우드 네이티브 개발을 지원하기 위한 컨테이너 인프라 관리 구조 및 시스템을 제안한다. 제안한 구조 및 시스템은 오픈소스 기반 클라우드 컴퓨팅을 활용하는 경우에 다양한 클라우드 기반 개발 패러다임의 접목이 가능하게 될 것으로 기대된다.

2. 관련 기술 및 연구

오픈스택[2]은 가상머신에 기반한 클라우드 서비스 제공을 지원하는 대표적인 IaaS(Infrastructure as a Service) 오픈소스이다. 오픈스택의 주요 컴포넌트인

이 논문은 23년도 정부(과학기술정보통신부), 정부(교육부)의 재원 및 한국연구재단의 지원을 받아 수행된 연구, 기초연구사업임(No. 2021R1A2C1006177, No. RS-2023-00243156)

* 교신저자(Corresponding Author)

노바(Nova)는 QEMU, KVM 등의 하이퍼바이저와 연계하여 가상머신 기반 가상화를 수행하고, 이를 클라우드 사용자가 활용할 수 있도록 관리 기능을 제공하는 역할을 수행한다. 그러나, 노바와 같은 오픈스택의 기초 컴포넌트들은 컨테이너 기반 가상화 기술을 제공하지 못해 클라우드 서비스의 제공 범위에 한계가 있다.

J. Alonso et al.[1]은 멀티 클라우드 기반의 클라우드 네이티브 애플리케이션에 대해 분석하였다. 해당 연구에 따르면 클라우드 네이티브와 하이브리드 클라우드(퍼블릭 클라우드와 프라이빗 클라우드의 동시적 활용)에 관해 구조적 측면, 관리적 측면에서 다양한 연구가 진행되며, 필요함을 제시하고 있다. 따라서, 퍼블릭 클라우드의 경우 제공되는 클라우드 네이티브 지원에 비해 미비한 오픈스택 등의 프라이빗 클라우드 플랫폼에서도 이에 대한 관련 기술을 지원할 필요가 있다.

3. 컨테이너 인프라 관리 구조 및 흐름

본 논문에서 제안하는 컨테이너 인프라 관리 구조는 [그림 1]과 같다.

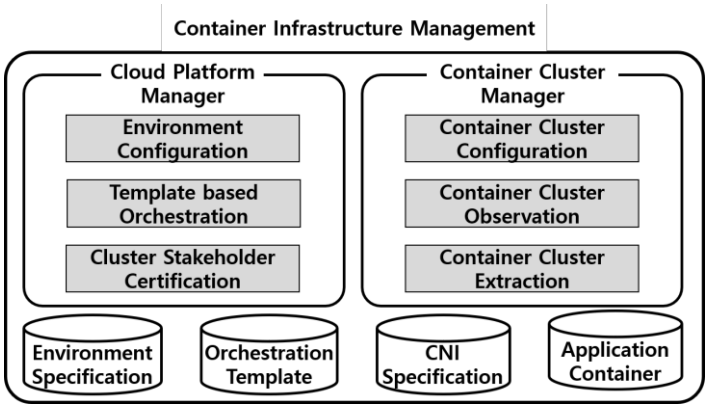


그림 1. 컨테이너 인프라 관리 구조

[그림 1]에서 컨테이너 인프라 관리 구조는 Cloud Platform Manager, Container Cluster Manager로 구분되며, 각 관리부에 필요한 리소스 및 데이터가 포함되어 있다. 각 관리부에 대한 설명은 다음과 같다.

- Cloud Platform Manager: 클라우드 플랫폼에서 컨테이너 클러스터 배포 및 관리를 수행하기 위해 플랫폼 레벨의 설정, 클러스터 배포, 이해관계자 인증을 수행함
- Container Cluster Manager: 배포된 컨테이너 클러스터를 요구사항에 맞게 설정하고, 인터페이스 도출, 클러스터 모니터링을 수행함

관리부에서 활용되는 각 리소스 혹은 데이터를 설명하면 다음과 같다.

- Environment Specification: 특정 플랫폼 배포를 위해(Kubernetes, Docker 등) 사전 클라우드 플랫폼에 필요한 의존성 정보나 환경 변수 설정을 위한 명세서
- Orchestration Template: 클라우드 플랫폼에 컨테이너 클러스터를 배포하기 위해 활용되는 오케스트레이션 템플릿
- CNI(Container Network Interface) Specification: 컨테이너 클러스터 내의 컨테이너 들이 네트워크를 형성하고 접근 가능한 인터페이스를 형성하기 위한 명세서
- Application Container: 컨테이너 클러스터를 통해 구현하고자 하는 애플리케이션이 탑재된 컨테이너 오브젝트

또한, 제안하는 관리 구조를 수행할 수 있는 흐름을 나타내면 다음 [그림 2]와 같다.

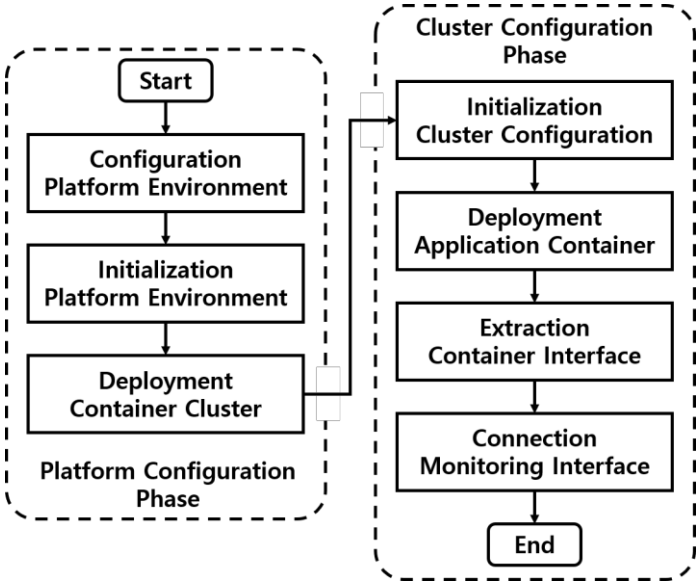


그림 2. 컨테이너 인프라 관리 흐름도

[그림 2]에서 나타난 바와 같이 컨테이너 인프라 관리 흐름은 플랫폼 설정 및 배포를 수행하는 Platform Configuration Phase와 컨테이너 클러스터의 설정 및 배포를 수행하는 Cluster Configuration Phase로 구분된다. 상세 설명은 다음과 같다.

Platform Configuration Phase는 플랫폼 환경 변수 및 플랫폼 컴포넌트를 준비하는 단계(Configuration Platform Environment), 컨테이너 환경 및 클러스터를 배포하기 위해 템플릿을 오케스트레이션 템플릿을 초기화하고 준비하는 단계(Initialization Platform Environment), 준비된 오케스트레이션 템플릿을 API와 CLI 인터페이스를 활용하여 클라우드 플랫폼에 전달하고 배포하는 단계(Deployment Container Cluster)의 순서로 동작한다.

Cluster Configuration Phase는 CNI 명세에 따라 컨테이너 네트워크를 준비하고 배포하여 컨테이너

클러스터가 동작할 수 있도록 초기화 하는 단계(Initialization Cluster Configuration), 애플리케이션 컨테이너를 클러스터 내에 배포하는 단계(Deployment Application Container), 컨테이너를 외부 호출 등의 이벤트로 동작가능 하도록 CNI를 활용하여 노출시키는 단계(Extraction Container Interface), 모니터링을 통해 컨테이너 클러스터가 관리되도록 연결하는 단계(Connection Monitoring Interface)의 순서로 동작한다.

4. 사례 연구 및 평가

4.1. 사례 연구

본 논문에서는 제안한 컨테이너 인프라 관리 방법의 적용성을 확인하기 위해 대시보드 기반의 쿠버네티스 인프라를 배포하는 시스템을 구현하였다. [그림 3]은 컨테이너 인프라 관리 구조 대시보드의 메인 화면이다.

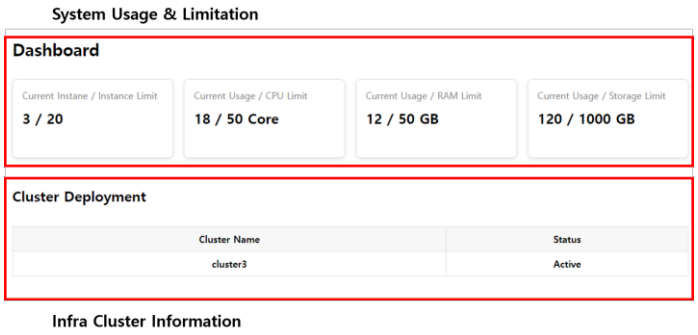


그림 3. 대시보드 메인 화면

[그림 3]에서 메인 화면은 인프라 배포 대상이 되는 오픈스택 환경의 사용량 및 최대치, 현재 배포된 상태의 인프라 클러스터의 상태를 표현한다.

[그림 4]는 신규 클러스터 생성을 위해 입력이 필요한 변수들을 나타낸다.

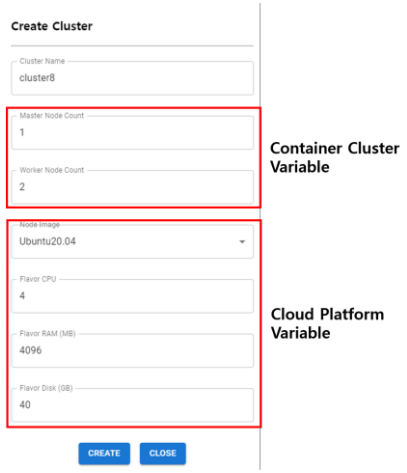


그림 4. 클러스터 생성 명세

[그림 4]에서 클러스터 생성을 수행할 때, 컨테이너 클러스터의 경우 Master-Slave 구조를 생성하기 위한 Master Node와 Worker Node의 개수를 지정한다. 클라우드 플랫폼의 경우는 클러스터의 각 노드가 동작할 수 있는 기반 OS를 명세하고, 클러스터 전체에 할당할 하드웨어 자원을 명세한다. 신규 생성하는 클러스터는 [그림 5], [그림 6]과 같이 나타난다.

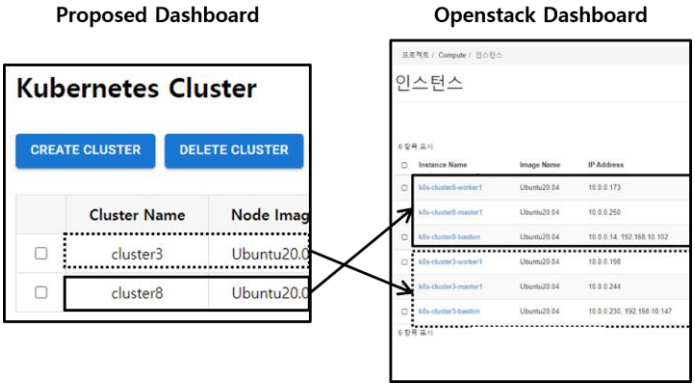


그림 5. 신규 클러스터 생성 반영

생성된 컨테이너 클러스터는 [그림 5]와 같이 배포 함께 오픈스택에 클러스터 구성에 필요한 노드들이 배포된다.

다음 [그림 6]은 배포된 클러스터의 모니터링 대시보드 중 일부이다.

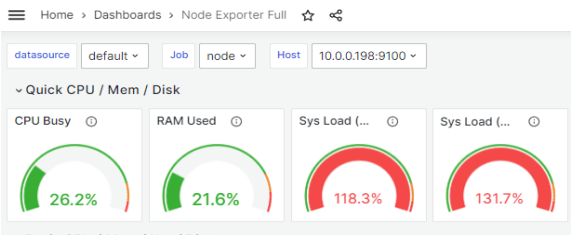


그림 6. 모니터링 대시보드

[그림 6]과 같이 컨테이너 클러스터 배포 시 연동하여 동작하는 모니터링 도구가 함께 배포되어 컨테이너 클러스터의 관리를 수행할 수 있는 기반으로 활용할 수 있다.

4.2. 평가

본 논문에서 제안한 방법의 평가는 대시보드 기반 관리 시스템의 클러스터 생성 과정과 해당 과정을 오픈스택을 통해 수동으로 구현하는 과정을 비교할 것이다. 우선 제안하는 방법의 경우 다음 [표 1]과 같은 과정을 갖는다.

표 1. 제안 방법 컨테이너 클러스터 배포 과정

Step	Step Function	Description
1	User Requirement Specification	사용자가 컨테이너 클러스터 구성요소를 명세함
2	Script Pipelining based on Requirement	사용자의 컨테이너 클러스터 구성요소를 각 요소 템플릿에 매핑하여 기 정의된 순서에 따라 실행함
3	Cluster Creation	클러스터가 생성됨

제안 방법은 구조와 흐름도를 바탕으로 구성요소를 오케스트레이션 템플릿과 환경 명세서의 순서에 따른 배치를 통해 스크립트 및 오케스트레이션 템플릿의 파이프라이닝이 가능하므로 위와 같이 수행할 수 있다.

다음 [표 2]는 기존 방법으로 수동적인 배포를 수행할 경우 컨테이너 클러스터를 구현하는 과정이다.

표 2. 기존 방법 컨테이너 클러스터 배포 과정

Step	Step Function	Description
1	User Requirement Specification	사용자가 컨테이너 클러스터 구성요소를 명세함
2	Platform Environment Configuration	사용자의 요구사항에 따라 필요한 환경변수나 오케스트레이션 템플릿을 준비함
3	Containerization Tool Deployment	컨테이너 배포 도구, 컨테이너 네트워킹 도구, 컨테이너 연결성 도구 등 컨테이너 클러스터 구성에 필요한 각종 도구를 배포함
4	Container Application Deployment	사용자 요구사항에 따라 컨테이너 애플리케이션을 배포함
5	Container Cluster Initialization	애플리케이션이 클러스터를 형성할 수 있도록 기 배포한 도구를 활용하여 초기화함
6	Container Interface Extraction	컨테이너 접근 및 이벤트 활용을 위한 각각 컨테이너의 인터페이스를 도출함
7	Cluster Creation	클러스터가 생성됨

기존 방법인 수동 생성은 [표 2]에서 나타난 대로 컨테이너 클러스터 구축을 위한 플랫폼 설정부터 컨테이너 클러스터화 도구를 배포하고, 애플리케이션 배포, 초기화, 인터페이스 도출의 전과정을 이해관계자의 요구사항을 바탕으로 작성하고 수행해야한다. 따라서 기존 방법은 클라우드 네이티브 환경을 다양하게 작성하고 활용해야 하는 소프트웨어 프로젝트 적용의 경우 인프라 배포 및 조율에 많은 시간을 소모하게 되는 단점이 나타날 수 있다. 따라서 본 제안한 방법은 이에 대한 시간 단축 효과를 얻을 것으로 기대된다.

5. 결론

본 논문에서는 오픈소스 클라우드 플랫폼에서 클라우드 네이티브 환경 배포 및 관리 지원이 부족한 점에 착안하여 오픈스택 플랫폼에 컨테이너 클러스터를 배포하고 지원하는 구조 및 흐름을 제안하였다.

제안한 방법은 컨테이너 클러스터를 작성하여 배포하는 대시보드를 구축하여 신규 컨테이너 클러스터가 작성되는 과정을 정립하고, 구현 가능성을 검증하였다. 또한, 컨테이너 클러스터를 제안한 방법과 기존 수동 배포 방법을 비교하여 본 방법이 단계적 축소를 지원하는 것을 나타냄으로써 제안 방법의 우수성을 제시하였다.

제안한 컨테이너 인프라 관리 방법은 오픈스택을 기반으로 클라우드 네이티브 환경을 활용하고자 하는 개발자 및 사용자의 플랫폼 활용 용이성을 제공할 수 있을 것이다. 또한 해당 구조를 확장하여 타 오픈소스 클라우드 플랫폼에 적용하는 것으로 오픈소스 기반의 클라우드 네이티브 환경을 제공하는 구조적 기반으로 활용될 수 있을 것이다.

참 고 문 헌

[1] J. Alonso, L. O. Echevarria, V. Casola, A. I. Torre, M. Huarte, E. Osaba and J. L. Lobo, “Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review,” Journal of Cloud Computing, Vol. 12, Issue. 6, pp. 1-34, 2023. 01.

[2] 오픈스택, https://www.openstack.org/software/

허가형 블록체인의 트랜잭션 로그 정제 방법

[1]강등원 ○, [2]정수민, [3]박준석, [1]염근혁*

부산대학교 [1]정보컴퓨터공학부, [2]정보융합공학과, [3]지능물류빅데이터연구소

okcdbu@naver.com, {sumin2708, pjs50, yeom*}@pusan.ac.kr

Transaction Log Refinement Method of Permissioned Blockchain

[1]Deungwon Kang○, [2]Sumin Jeong, [3]Joonseok Park, [1]Keunhyuk Yeom*

[1]School of Computer Science and Engineering, Pusan National University

[2]Department of Information Convergence Engineering, Pusan National University

[3]Research Institute of intelligent Logistics Big data, Pusan National University

요 약

하이퍼레저 패브릭은 접근 권한이 인가된 사용자들만 거래를 수행할 수 있도록 만들어진 허가형 블록체인 플랫폼이다. 허가형 블록체인에서 발생하는 거래는 매 수행 트랜잭션 수준으로 실행 이력이 누적된다. 그러나 하이퍼레저 패브릭 등의 허가형 블록체인 플랫폼의 폐쇄성으로 인해 실행 이력 즉, 로그 데이터의 활용이 제한적이다. 따라서 본 논문에서는 허가형 블록체인의 로그 데이터 활용성을 높이기 위해 로그 데이터에 접근하고 정제하는 방법을 제안한다. 또한 정제 방법을 적용하여 사례시스템에서 트랜잭션 함수의 수행 빈도를 정제하는 것을 실험하였다. 이를 통해, 본 방법이 트랜잭션의 중요도를 측정할 수 있는 기반 모델로 적용될 수 있음을 확인하였다.

1. 서 론

허가형 블록체인은 컨소시엄을 통해 허가된 사용자만 접근하여 활용할 수 있도록 구축한 블록체인 기술이다. 대표적인 허가형 블록체인인 하이퍼레저 패브릭(Hyperledger Fabric)에서는 체인코드와 스마트 컨트랙트 라는 프로그램 코드 기반의 계약서를 통해 트랜잭션을 수행하고 거래를 중재한다.

블록체인 원장에 누적되는 트랜잭션 실행 이력은 이해관계자 간의 거래 내역을 추적하는 데 활용될 수 있다. 또한 트랜잭션 실행 이력은 트랜잭션 정보의 효율적인 관리에 적용될 수 있는 중요한 요소 중의 하나이다. 따라서 트랜잭션 실행 이력을 관리하는 방법 중의 하나로 트랜잭션별로 발생하는 로그 데이터를 활용할 수 있다.

그러나 허가형 블록체인 구조는 폐쇄성을 가지고 있기 때문에 로그 데이터를 직접적으로 접근 및 활용하기 위한 개방형 인터페이스를 제공하지 않고 있다. 또한 내부적으로 쌓이는 로그 데이터의 형태도 사용자가 직접 분석을 통해 활용해야 하는 불편함이 존재한다.

따라서 본 논문에서는 허가형 블록체인의 로그 데이터를 활용하기 위해서 로그 데이터를 추출하고, 정제하는 방법 및 시스템을 제안한다. 제안한 방법은 블록체인의 트랜잭션 관리 및 트랜잭션 데이터를 활용한 분석 등 다양한 블록체인 트랜잭션 데이터의 활용도를 높이는 접근법으로 활용될 수 있다.

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 융합보안핵심인재양성사업,대학ICT연구센터육성지원사업의 연구결과로 수행되었음(IITP-2023-2022-0-01201, IITP-2024-2020-0-01797)

* 교신저자(Corresponding Author)

2. 관련 연구 및 기술

하이퍼레저 패브릭(Hyperledger Fabric)[1]은 허가형 블록체인 플랫폼으로 오픈소스 중 폐쇄적인 특징을 가지고 있기는 하나 피어와 오더러에 대한 접근을 수행하는 프로그래밍 가능한 API를 제공하고 있다. 이 API는 분석을 통해서 확인할 수 있으며, REST API의 형태로 접근하여 활용하기는 어렵다.

차동현 등[2]의 연구에서는 하이퍼레저 패브릭을 활용한 SaaS 환경의 블록체인 로그 관리 시스템에 대한 설계를 수행하였다. 해당 논문에서는 블록체인의 원장에서 추출할 수 있는 정보를 활용하여 로그 관리를 수행하는 시스템에 대한 설계를 제시하였다. 반면 본 논문에서는 API를 활용한 하이퍼레저 패브릭 트랜잭션 로그를 추출 및 정제하는 방법을 제시하였다.

3. 하이퍼레저 패브릭 로그 정제 방법

3.1. 로그 관리 분석

하이퍼레저 패브릭의 로그는 수준별로 FATAL, PANIC, ERROR, WARNING, INFO, DEBUG로 구분된다. 트랜잭션 로그는 구분된 수준 중 DEBUG 단계에서 나타난다. 하이퍼레저 패브릭의 로그 표출 구조는 하위 수준으로 이행할 수록 상위 수준의 내역을 포함해서 제공되는 구조이다. 이는 로그 분석을 위해 특정 수준 또는 특정 모듈에 대한 로그 즉, 로그 표현 시 필요 한 로그 부분만을 추출하지 못한다.

아래는 하이퍼레저 패브릭에서 발생한 DEBUG 로그 중 일부를 나타낸 것이다.

2023-12-12 08:42:27.101 UTC 0110 DEBUG [chaincode] HandleTransaction -> [8edeeef4] handling GET_STATE from chaincode

이 로그 구조를 분석하면, [Timestamp] [ID] [Log Level] [Module Name] [Function ID] -> [Message]의 구조이다.

3.2 로그 정제 방법

그림 1은 본 논문에서 제안하는 로그를 추출하고 정제하는 과정을 도식화한 것이다.

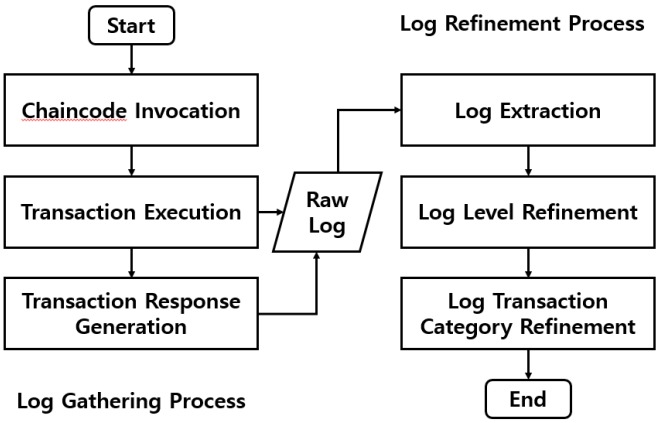


그림 1. Log Handling Process

그림 1에서 로그 처리 과정은 로그를 수집하는 단계(Log Gathering Process)와 로그를 정제하는 단계(Log Refinement Process)로 구분한다.

로그 수집 단계는 체인코드 호출(Chaincode Invocation), 트랜잭션 실행(Transaction Execution), 트랜잭션 응답 생성(Transaction Response Generation)의 순서로 수행되고, 최종적으로 Raw Level Log가 트랜잭션 실행과 응답 생성 시에 발생하고 누적된다.

로그 정제 단계는 로그 추출(Log Extraction), 로그 수준 정제(Log Level Refinement), 로그 트랜잭션 분류 정제(Log Transaction Category Refinement)의 순서로 수행된다. 최종적으로 도출되는 정제된 로그 구조는 트랜잭션 ID 별 정보의 획득 혹은 변경 수행 횟수의 형태로 구성된다.

그림 1에서 나타낸 로그 트랜잭션 분류 정제는 그림 2의 알고리즘에 따라 수행된다.

Algorithm 1 Hyperledger Log System

```
data <= get log data with HandleTransaction
namelist <= get unique txID list from data
cntdict <= {namelist: {"GetState": 0, "PutState": 0} }
for i in data do
    if i.message contains "Completed GET_STATE" then
        cntdict[i.name]['GetState']++
    if i.message contains "Completed PUT_STATE" then
        cntdict[i.name]['PutState']++
```

그림 2. 로그 트랜잭션 분류 정제 알고리즘 의사코드

예를 들어, “HandleTransaction” 이라는 키워드를 가지고 있는 로그 데이터를 추출한 이후, data 변수에 저장한다. 그리고 data의 txID(transaction ID) 리스트를 가져와 namelist에 저장한다. cntdict는 dictionary형태의 변수이다. 이 변수의 key값으로 namelist의 요소를, value에는 각각 호출한 함수의 이름, 횟수를 저장한다.

4. 로그 정제 시스템 구현

제안된 로그 정제 방법을 구현한 실험 환경은 표 1과 같다.

표 1. 실험 환경

실험 환경	구성 요소
하이퍼레저 패브릭	v2.5.4
사용 피어 노드	Peer1
스마트 컨트랙트	Basic/test-network
Chaincode 언어	Golang
피어 수	2

표 1에 기술된 실험 환경에 따라 로그 수집 단계를 적용하면 그림 2의 정제 전 로그 데이터가 수집된다.

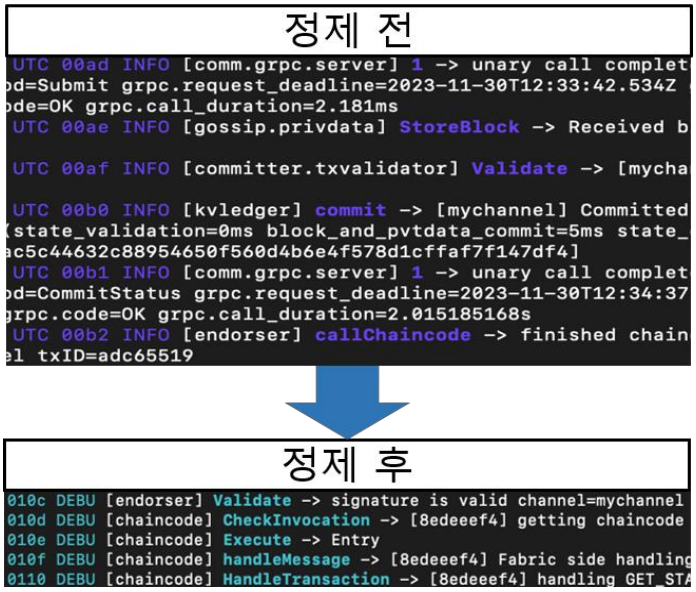


그림 3. 로그 데이터 로그 수준 정제 결과

그림 3의 정제 후 로그 데이터는 로그 수준 정제 단계가 완료된 후의 결과물로 기존 INFO 등의 기본 데이터만 도출하는 결과에서 DEBUG 수준의 데이터로 정제한 것이다.

도출 결과에서 로그 트랜잭션 분류 정제를 수행한 결과가 그림 4와 같다.

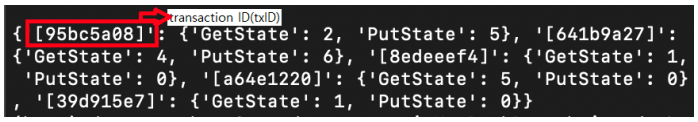


그림 4. 로그 데이터 정제 결과

그림 4는 각 txID(transaction ID)에 의해 호출된 getState, putState의 실행 횟수를 확인할 수 있다. 이 함수들은 Stub 구조에 접근하는 함수들로 호출에 의해 오버헤드가 발생한다. 이를 통해, 각 트랜잭션의 함수 호출 빈도에 따라 중요도를 식별할 수 있는 이점이 있다.

5. 결론 및 향후 연구

본 논문에서는 트랜잭션 실행 이력을 활용하기 위한 방안으로 트랜잭션 로그를 정제하는 방법 및 시스템을 제시하였다. 제시한 방법을 하이퍼레저 패브릭 환경에 적용하여 로그 정제가 실제 수행됨을 검증하였다.

로그 정제 결과는 트랜잭션의 실행 횟수에 주안점을 두어 트랜잭션의 중요도를 측정할 수 있는 메트릭으로 활용될 수 있다. 또한 각 함수의 실행 빈도를 통해 함수가 수행하고자 하는 역할을 도출할 수 있는 정적 분석의 요소로써 적용할 수 있을 것으로 기대된다.

향후 연구로는 정제 흐름 및 결과를 가시화하는 기법을 연구할 계획이다.

참 고 문 헌

[1] 하이퍼레저 패브릭, https://www.hyperledger.org/projects/fabric

[2] 차동현, 이현병, 유재수, “SaaS 환경에서의 블록체인 기반 로그 관리 시스템 설계,” 한국콘텐츠학회 2023 종합학술대회 논문집, pp. 275-276, 2023.05.

360도 어라운드 뷰 시스템의 SIFT 특징점 기반 동적 캘리브레이션 알고리즘

강대웅 [O], 조상훈, 이학승, 한정우, 염지환, 국중진
상명대학교 정보보안공학과 [1], 상명대학교 휴먼지능정보공학부 [2]
kangred@naver.com, {2022D3002, 201821269, 201821250, 202121313}@sangmyung.kr, kook@smu.ac.kr

SIFT Feature-Based Dynamic Calibration Algorithm for 360-Degree Around View System

Daewoong Kang[O], Sanghoon Cho, Hakseung Lee, Jungwoo Han, Jihwan Yeum, Joongjin Kook
Dept. of Information Security Engineering, Sangmyung University

요 약

본 논문에서는 360도 어라운드 뷰 시스템의 SIFT 특징점 기반 동적 캘리브레이션 알고리즘을 제안한다. 기존 시스템은 영상 정합을 위한 과정에서 기준이 되는 마커 패턴을 차량 주변에 배치하고 수동적인 방식으로 보정이 이루어지기 때문에 많은 시간과 비용이 요구된다. 따라서 본 연구에서는 어라운드 뷰 시스템에 장착된 4개의 어안렌즈 카메라로 촬영된 영상들을 왜곡보정 후, SIFT 기반의 특징점 추출을 통해 인접 영상과의 동일 특징점을 매칭함으로써 고정된 마커 패턴 없이도 영상 정합을 가능하게 한다.

1. 서 론

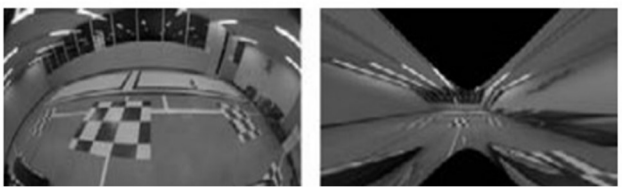
어라운드 뷰 시스템은 여러 대의 광학 카메라로 촬영한 영상을 병합하여 하나의 화면에 나타내야 하므로 필연적으로 공간의 왜곡이 발생한다. 광각 영상의 외곽 끝부분은 선명도의 저하 및 이미지의 왜곡이 발생하며, 이로 인해 캘리브레이션 결과 전체 환경구현에서의 필연적 성능 저하로 이어진다. 또한 현재 시판중인 어라운드 뷰 시스템은 전체 영상에서 각 이미지들이 정합되는 가장자리 부분에 왜곡 현상이 발생하는 것을 확인할 수 있어, 근거리에 초점을 두어 저속주행 및 주차에만 제한적으로 이용하도록 설정하여 제품화 되고있다.

본 논문에서는 차량에 4대의 광각 카메라를 설치하고 캘리브레이션을 통해 차량의 주변 환경을 실제와 같이 구현하여 운전자가 상황을 직관적으로 인식 및 위험 상황에 실시간으로 대처할 수 있게 하는 것을 목표로, 기존 Around View Monitoring System (AVM)의 캘리브레이션 과정을 개선하기 위한 SIFT 특징점 기반의 동적 캘리브레이션 알고리즘을 제안한다.

2. 본 론

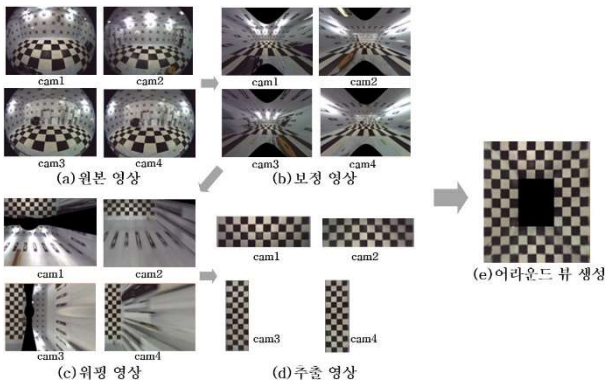
2.1 마커 기반의 캘리브레이션 방법과 4채널 AVM 생성

캘리브레이션은 먼저 촬영된 영상에서 특징점을 찾는 동작을 수행한다. VinhNinhDao와 MasanoriSugimoto[1]는 특징점 검색 시 체커보드 패턴을 활용하여 템플릿 매칭하는 방식을 적용했다. AVM은 차량의 전방, 후방, 좌측, 우측에 어안 렌즈 카메라를 설치한다. 영상을 획득하고 왜곡된 영상을 보정한 이후 워핑 작업을 수행한다. 최종적으로 마커 기반의 캘리브레이션을 통해 영상을 정합하여 어라운드 뷰 영상을 생성하고, 출력하는 장치이다. 어안 렌즈는 일반적으로 볼록한 형태의 베럴 디스토션(barrel distortion)을 발생시킨다. 따라서 AVM장치에 그대로 적용할 경우 영상의 왜곡이 심하게 발생하기에 왜곡된 부분의 보정을 통해 이러한 문제를 해결한다. 왜곡된 영상의 보정은 forward mapping과 inverse mapping 방식이 있다. forward mapping을 적용할 경우 보정의 결과 이미지에 hole 발생 및 연산의 속도가 매우 느려 실시간 보정 영상을 얻기가 어렵기 때문에, <그림 1>과 같이 inverse mapping 방식을 적용하는 것이 일반적이다.



<그림 1> 배럴 디스토션 영상과 Inverse mapping 결과

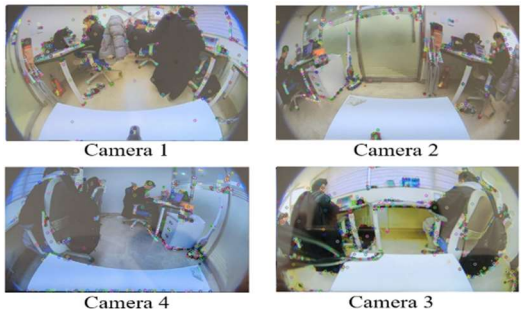
정합된 영상에 대해 색상 값을 조정하여 자연스러운 영상을 만들기 위해 alpha blending을 적용할 수 있다. 이러한 거쳐 최종적으로 <그림 2>와 같은 실시간 어라운드 뷰 영상의 생성이 가능하다.



<그림 2> 어라운드 뷰 생성 과정

2.2. SIFT 기반의 캘리브레이션 방법과 4채널 AVM 생성

본 논문에서는 특징점을 이용한 물체의 인식을 통해 마커 없이 캘리브레이션을 자동화할 수 있는 AVM 시스템을 제안한다. 특히, 특징점 기반의 객체인식 실험 중에 이전의 실험들에 비해 조도에 강하고 이미지 회전 및 리사이즈에도 변하지 않는 특성을 지닌 Scale Invariant Feature Transform(SIFT) 알고리즘을 이용하였다.



<그림 3> SIFT 알고리즘 검출 결과

SIFT를 이용한 왜곡 보정 알고리즘은 특징점 모델링을 기반으로 여러 대의 카메라에서 촬영된 영상들을 보정 및 워핑 후, 영상에 포함된 객체들의 특징점을 찾아 인접 영상과의 비교 매칭 과정을 거쳐 정합 작업을 하는 알고리즘이다.

본 논문에서 제안하는 어라운드 뷰 시스템의 SIFT 특징점 기반 동적 캘리브레이션 알고리즘은 4개의 카메라를 통해 연속적으로 입력되는 영상에 대해 인접한 두 영상의 공통된 특징점을 추출하여 두 영상의 매핑을 위한 위치 정보를 추출하여 영상을 정합하고, OpenGL을 기반으로 만들어진 3차원 반구체 모델에 텍스처 매핑을 수행한다.

2.3 실험 결과

<그림 4>은 SIFT 특징점 기반 동적 캘리브레이션 알고리즘을 통해 정합된 어라운드 뷰 영상을 나타낸 것이며, 실험을 통해 기존의 마커 정보를 사전에 입력하여 동작하는 캘리브레이션 방식의 결과물과 같은 캘리브레이션의 결과물을 얻을 수 있으며, AVM 시스템에서 올바르게 작동하는 것을 확인하였다.

<그림 4> SIFT 기반 동적 캘리브레이션의 영상 정합 결과

또한, SIFT 알고리즘을 활용한 자동 캘리브레이션을 적용했을 때, 영상의 획득부터 최종적인 영상 정합에 소요되는 시간은 평균 13.4(s)가 소요된다. 기존의 수동 캘리브레이션 방식이 소요되는 시간은 작업자의 숙련도, 프로그램의 편의성에 따라 약 6시간 내외의 시간이 소요된다. 특징점 기반의 객체인식 실험은 이전의 실험들에 비해 외부 조도에 강인하고 이미지 회전 및 리사이즈에도 변하지 않는 특성을 지니고 있어 기존의 방식보다 향상된 캘리브레이션 결과를 얻을 수 있다.

3. 결 론

본 논문에서는 어라운드 뷰의 캘리브레이션 과정에서 기존의 고정 마커를 이용하는 수동적 방식이 아닌, SIFT 특징점 기반 동적 캘리브레이션 알고리즘을 제안하였다. 이 알고리즘은 체커보드의 특정한 패턴을 사용하지 않아도 기존의 방식과 비슷한 캘리브레이션 영상을 생성할 수 있다. 제안하는 SIFT 특징점 기반 동적 캘리브레이션 알고리즘은 다음과 같은 장점들이 있다. 첫 번째는 마커를 설치하기 위한 시설에 대한 소요 경비가 발생하지 않는다는 것이다. 두 번째는 정합 시 필요한 패턴의 위치를 차량 주변에 고정하는 작업을 하지 않아도 되기 때문에 공간상의 제약을 받지 않는다. 세 번째는 마커를 필요치 않기 때문에 차량의 크기나 종류에 제약이 없으며, 차량이 아닌 드론, 로봇 등 다양한 물체에 적용이 가능하다는 것이다. 마지막으로 기존 AVM 시스템의 설치, 수리 시 캘리브레이션은 공장이나 서비스 센터에 입고가 되어야 가능했지만, 제안하는 방식은 환경의 제약이 없기 때문에 언제든 간단히 수행될 수 있다.

감사의 글

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(RS-2023-00263298, 딥러닝 기반 실시간 객체 인식 및 AR 디스플레이를 지원하는 360도 3D 어라운드뷰 시스템 개발)

4. 참고문헌

[1] Vinh Ninh Dao and M. Sugimoto, "A Robust Recognition Technique for Dense Checkerboard Patterns." 2010 20th International Conference on Pattern Recognition, pp. 3081-3084, 2010.

Multiple listener facial generation in face-to-face communication via Variational Autoencoder

Minh-Duc Nguyen, Dang-Khanh Nguyen, Prabesh Paudel, Hyung-Jeong Yang*
Department of Artificial Intelligence Convergence, Chonnam National University.
* Corresponding author: hjyang@jnu.ac.kr

Variational Autoencoder 를 통한 대면 커뮤니케이션에서 다중 청취자 얼굴 생성

응웬민득, 응웬 당 칸, 파우델 쁘러베스, 양형정*
전남대학교 인공지능융합학과

Abstract

The generation of listener facial responses focuses on modeling the interactive communication feedback from a listener in a face-to-face communication scenario. Our objective is to create realistic multiple listener-head videos that respond genuinely to one speaker, expressing a range of attitudes and viewpoints while accurately preserving listener identity information. To accomplish this, we propose the use of a non-deterministic network that leverages a Variational Autoencoder to learn a continuous latent representation of realistic listener facial motion. This enables the generation of multiple variations in response to the speaker. Through quantitative evaluation, our approach surpasses baseline methods.

Ⅰ. Introduction

Real-time observation of facial expressions in face-to-face interactions is vital for improving emotional understanding. Exploring this through computer vision, especially in dynamic talking human videos, is intriguing. Generating responsive listener reactions is crucial for authentic digital human interactions in diverse scenarios, from human-computer engagement to animation production.

Most previous works focus on speaker modeling, i.e. talking face generation while the generation of listener reaction is largely unexplored. Mohan Zhou, et al. [1] introduce a dataset and benchmark including LSTM [6] model for listening head generation task. However, as a deterministic model, it lacks diversity which is the key to real-world face-to-face scenarios.

In this work, our primary methods include: (1) We employ a variational autoencoder (VAE) [3] to train a probabilistic generative model capable of producing diverse head pose and expression features. (2) We apply several techniques to improve the performance of VAE training on limited data and prevent mode collapse leading to loss vanishing issues.

Ⅱ. Proposed method

Our objective is to generate comprehensive responsive listening head videos from a given speaker video clip comprising visual and audio information and a listener head image. The encoder-decoder architecture model first predicts the listener's head motion and facial expression features. These predicted features are adjusted to reconstruct the 3D Morphable Model (3DMM) coefficients, incorporating the past motion features of the reference listener. Subsequently, the adapted coefficients are fed into PIRender [4], a neural renderer, to generate a listening reaction video.

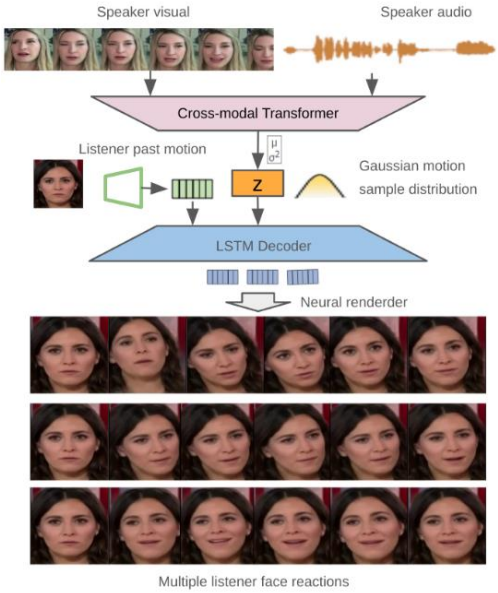


Figure 1. Overview of our pipeline, model can synthesize a variety of listener facial responses in different emotional attitudes.

Our model consists of (1) A Transformer encoder [7], which first integrates the 3DMM coefficients of the speaker with audio features to generate a fused feature embedding. Following this, the model makes predictions for a pair of tokens, μtoken and σtoken, signifying the Gaussian distribution of diverse facial reactions linked to input speaker behavior. These predictions rely on both the amalgamated speaker visual-audio fused embedding and a pair of learnable tokens; and (2) an LSTM decoder that creates instances of representations from the predicted distribution tokens, describing a response listener motion, which encompasses 3DMM coefficients. The final step involves using PIRender [4] to convert the generated

3DMM coefficients into a sequence of facial reaction images.

VAE model optimization includes reconstruction loss in terms of identity and Kullback-Leibler divergence loss for diversity. With reconstruction loss, we use L_2 distance separately within head pose and expression which are encompassed in 3DMM coefficients set. The loss is computed as:

$$\mathcal{L}_{\text{rec}} = \sum_{t=1}^{T} |\beta_t - \hat{\beta}_t|_2 + |c_t - \hat{c}_t|_2 + |p_t - \hat{p}_t|_2 + |\mu(c_t) - \mu(\hat{c}_t)|_2$$

where β_t, c_t, p_t and $\hat{\beta}_t, \hat{c}_t, \hat{p}_t$ denote the ground truth and predictions of angle, translation and expression coefficients respectively. The head motion inter-frame changes denote as $\mu(.)$.

We apply the Cyclical Annealing Schedule [2] with hyperparameter λ dynamically changed with 4 cycles for KL divergence loss during the training. This strategy effectively mitigates mode collapse. KL loss can be formulated as:

$$\mathcal{L}_{KL} = \lambda \text{KL}(\phi, \omega)$$

where ϕ is the learned Gaussian distribution with the standard normal distribution $\boldsymbol{\omega}$.

Additionally, we employ an energy-based diversity loss [5] to compute each pair of generated listener reactions to engage the diversity among predictions:

$$\mathcal{L}_{\text{div}} = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j \neq i}^{M} \exp\left(-\frac{|\hat{y} - y|_2^2}{\sigma_d}\right)$$

where $M = 2$ was set in the paper and an RBF kernel with scale σ_d is used. y and $\hat{y}$ are the goundtruth and predicted 3DMM listener parameters.

Ⅲ. Experiments

We evaluate our method on The ViCo dataset [1], which consists of 483 video clips showing real-world face-to-face communication between 67 speakers and 76 listeners in a variety of natural settings. The data contains rich samples of three main attitude categories: Positive, Natural and Negative. We follow the guidelines of PIRender [4] to extract 3DMM coefficients, encompassing identity, expression, texture, pose, and lighting, from videos at 30 fps with 256 x 256 size for each face video frame. Additionally, we extract acoustic features for audio, which include MFCC, MFCC-Delta, Zero Crossing Rate (ZCR), loudness, and energy.

Our evaluation aligns with ViCo [1] methodology. To assess the precision of the generated pose and expression features, we rely on $L1$ distance the as our chosen evaluation metric. For a comprehensive evaluation of video-level performance, we employ a set of metrics, including Structural Similarity (SSIM), Cumulative Probability of Blur Detection (CPBD), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID). To evaluate identity preservation, we measure the cosine similarity (CSIM) between the identity of generated and ground truth images. Additionally, we compute the total Mean Squared Error (MSE) between each pair of generated facial reactions for each input speaker, as a motions and expressions diversity measurement.

We compare the following baselines: Sequential models including LSTM [6] (adopted by ViCo [1]), GRU, and RNN. We further add Random motion and implement extra generative models: LSTM-GAN (LSTM-based generator), LSTM-VAE (LSTM encoder-decoder layers). Evaluated results are shown in Table 1.

Table 1. Quantitative comparison with other methods

	L1 ↓	SSIM ↑	CPBD ↑	PSNR↑	FID ↓	CSIM ↑	Diversity ↑
ViCO (LSTM)	0.114	0.581	0.164	17.64	25.55	0.573	0.000
RNN	0.118	0.590	0.166	17.85	28.12	0.554	0.000
GRU	0.106	0.580	0.168	17.54	27.02	0.545	0.000
GAN_LSTM	0.166	0.521	**0.173**	15.55	38.49	0.486	0.000
Random	0.145	0.556	0.159	16.68	41.28	0.516	**0.167**
VAE_LSTM	0.140	0.562	0.165	16.82	31.26	0.545	0.009
Ours	**0.095**	**0.624**	0.165	**18.84**	**25.45**	**0.641**	0.018

IV. Conclusion

In this study, we presented a multi-faceted listener motion generation network, introducing a novel approach that employed a continuous space learning model, which is designed to generate responsive listener reaction features. The model aimed to generate diverse, realistic, and natural outcomes. The approach underwent comprehensive evaluations, and both quantitative and experimental results confirmed its superior ability to generate precise and diverse listener motion responses.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00219107)" IITP "This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00256629) grant funded by the Korea government (MSIT).

REFERENCES

[1] Mohan Zhou, et al., 2022. "Responsive listening head generation: a benchmark dataset and baseline". In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII. Springer, pp 124–142.

[2] Hao Fu, et al., "Cyclical annealing schedule: A simple approach to mitigating kl vanishing". In North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.

[3] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In 2nd International Conference on Learning Representations, ICLR, 2014.

[4] Ren, Y., et al., "Pirenderer: Controllable portrait image generation via semantic neural rendering". In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Pp. 13759-13768, 2021.

[5] Ye Yuan and Kris Kitani. "Dlow: Diversifying latent flows for diverse human motion prediction". In Computer Vision– ECCV 2020: 16th European Conference, Glasgow, UK, Proceedings, Part IX 16, pp 346–364. Springer, 2020.

[6] Hochreiter, S., Schmidhuber, J.: "Long short-term memory". Neural computation 9(8), 1735–1780, 1997.

[7] Vaswani, A., et al., "Attention is all you need". Advances in neural information processing systems 30, 2017.

Enhanced Transformer Variational Auto Encoder for Multiple Appropriate Facial Reaction Generation

Dang-Khanh Nguyen, Minh-Duc Nguyen, Prabesh Paudel, Hyung-Jeong Yang*

Department of Artificial Intelligence Convergence, Chonnam National University.

* Corresponding author: hjyang@jnu.ac.kr

다양한 적절한 얼굴 반응 생성을 위한 향상된 트랜스포머 변이 오토 인코더

응웬당칸, 응웬민득, 파우델 쁘러베스, 양형정*
전남대학교 인공지능융합학과

Abstract

Facial reaction generation has been an emerging topic in recent years. However, various studies have been conducted on speaker-centric synthesis while the listening role generation is still an open topic. Predicting the listener's facial reaction in a dyadic conversation is a complex task because various reactions can be triggered by one specific speaker's behavior. In this paper, we propose the Multimodal Transformer-based Variational Autoencoder to learn the distribution of listener facial reaction given speaker audiovisual information. The proposed method utilizes the Multimodal Bottleneck Token to learn the interaction between acoustic and visual speaker features and employs the Variational Autoencoder to synthesize multiple listener reaction latent features. Our proposed method outperforms the baseline model and previous methods on REACT24 benchmarks.

Ⅰ. Introduction

Our daily lives involve frequent conversations, a dynamic interaction where individuals alternate between speaking and listening roles to transfer and receive information in face-to-face communication. The speaker conveys information verbally while the listener often responds through non-verbal cues, offering real-time feedback. According to Song et al. [1], the same information delivered by the speaker can prompt varied reactions from the listener depending on different contexts.

Although considerable research has focused on synthesizing speech from the speaker's perspective, insufficient attention has been paid to generating the listener's reaction. Song et al. [3] introduce a benchmark for multiple appropriate facial reactions including a multimodal dataset and a baseline non-deterministic model using the Transformer-based Variational Autoencoder (TransVAE). Luo et al. [2] propose an encoder-decoder architecture named ReactFace to resolve the ill-posed problem and synchronize the generated listener's reaction with speaker visual and acoustic features in the temporal dimension.

In this research, we aim to improve the interaction between the audio and visual modality of the speaker by using the Multimodal Bottleneck Transformer (MBT) [4] for the encoder part. With the informative interaction features extracted from the MBT, the effectiveness of the generative function performed by the TransVAE and the cross-modal transformer in the decoder can be boosted. Our proposed model attains a significant enhancement in appropriateness and diversity evaluation scores compared to existing methods.

Ⅱ. Proposed Method

Our proposed model adapts the encoder-decoder architecture receiving the video of a speaker as input and generating 3D facial features and facial reactions of the listener. The 3D facial feature is a sequence of 3D Morphable Model (3DMM) coefficients used to render the sequence of frames of the listener based on a static reference image. On the other hand, the facial reaction includes three widely-used facial descriptors: the probabilities of eight emotions, 15 well-defined facial movements also known as the action units, and the facial affect consisting of valence and arousal levels. A detailed illustration of our model is shown in Figure 1.

Initially, the speaker encoder extracts the visual and acoustic features from the video by exploiting the dedicated neural networks. Particularly, we use pre-trained VGGFace and VGGish for image and audio modalities, respectively. Sequentially, we propose the Multimodal Bottleneck Transformer for cross-modal learning. It is a low-cost transformer approach for fusing two time-series inputs with long sequence lengths. It adapts the idea of bottleneck tokens [4] to force the model to extract the most meaningful information from each modality. The speaker's enriched visual features are fed into a linear layer to predict the 3DMM attributes for the auxiliary task.

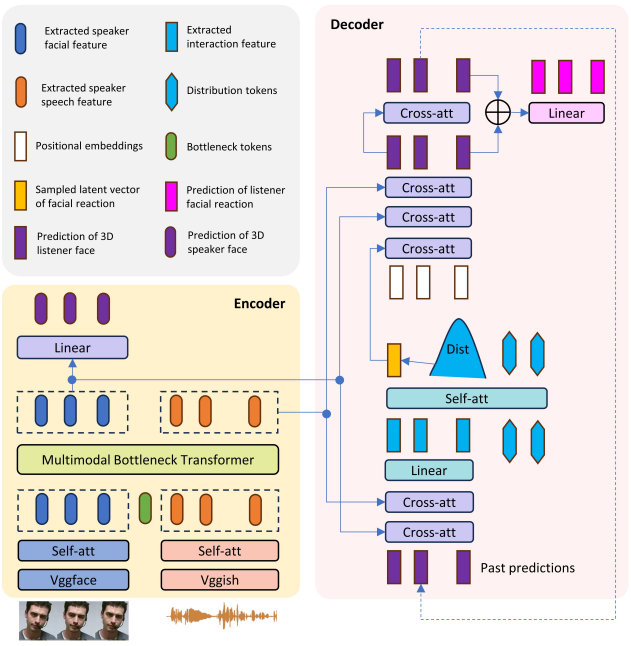


Figure 1. Block diagram of our proposed model

Discussing the decoder, we adapt the listener reaction decoder of the ReactFace [2] model. Firstly, the decoder leverages the output of the encoder including audio and visual features to extract the enhanced interaction features from the past predictions of the 3D listener face by transformer-based cross-attention. Afterward, a conventional transformer autoencoder is employed to learn a suitable facial reaction distribution for the listener based on the enhanced interaction features. A latent vector is sampled from the distribution and represents the facial reaction of the listener. Based on this latent feature, a sequence of 3D listener facial coefficients is synthesized by cross-modal transformers using the audio and visual features from the encoder. Finally, the prediction of the listener's facial reaction is obtained from a linear mapping layer with the predicted 3D features of the listener.

III. Experiments

We conducted our experiments on the REACT24 dataset which is used to evaluate ReactFace [2]. It is a compilation of two multimodal datasets NoXI and RECOLA. There are 2962 pairs of speaker's and listener's clips. To construct a multiple appropriate facial reaction dataset, Song et al. [1] apply the automatic appropriate facial reaction labeling strategy to define the correct facial reactions corresponding to each speaker in the dataset.

Regarding the assessment, we follow Song et al. [1] using a well-defined set of evaluation metrics to measure the appropriateness and diversity of our proposed model's output. Concisely, we use Facial reaction distance (FRDis) and Facial reaction correlation (FRC) for appropriateness measurement. To evaluate diversity, we adapt Facial reaction variance (FRVar), Diverseness among generated facial reactions (FRDiv), and Diversity among facial reactions generated from different speaker behavior (FRDvs). The higher value of the mentioned evaluation metrics is better, except FRDis.

Table 1 The quantitative results of our proposed method on validation split of REACT24 dataset.

Model	Appropriate		Diverse		
	FRC	FRDis	FRDiv	FRVar	FRDvs
Baseline	0.13	95.78	0.024	0.004	0.026
ReactFace	3.74	50.68	0.129	0.058	0.129
Ours	**4.15**	**50.33**	**0.150**	**0.072**	**0.152**

According to Table 1, our proposed model outperforms the baseline [3] and achieves better results than ReactFace [2] in all scores. Especially, FRC metric exhibits a significant improvement compared to previous studies. As a result, our model successfully synthesizes multiple appropriate facial reactions in a dyadic conversation setting.

IV. Conclusion

In this study, we propose a framework using the Multimodal Bottleneck Transformer and Transformer Variational Autoencoder to improve the extracted interaction feature between speaker and listener in conversation. This framework accomplishes a noticeable enhancement compared to prior methods in multiple appropriate facial reaction generation task. Our future research will focus on balancing the objective functions including the reconstruction loss, energy-based diversity, and the Kullback-Leibler loss in training process.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00219107). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00256629) grant funded by the Korea government (MSIT).

REFERENCES

[1] Song, Siyang, et al. "Multiple Appropriate Facial Reaction Generation in Dyadic Interaction Settings: What, Why and How?." arXiv e-prints (2023): arXiv-2302.

[2] Luo, Cheng, et al. "ReactFace: Multiple Appropriate Facial Reaction Generation in Dyadic Interactions." arXiv preprint arXiv:2305.15748 (2023).

[3] Song, Siyang, et al. "REACT2023: The First Multiple Appropriate Facial Reaction Generation Challenge." Proceedings of the 31st ACM International Conference on Multimedia. 2023.

[4] Nagrani, Arsha, et al. "Attention bottlenecks for multimodal fusion." Advances in Neural Information Processing Systems 34 (2021): 14200-14213.

그래프 합성곱 신경망을 활용한 웹 데이터 추출

김창영 O1 조영우 1 김명석 2 *양형정 1

1 전남대학교 인공지능융합학과 2㈜나로수

kcy13930@gmail.com, whduddn7720@naver.com, mskim@narosu.co.kr,

hjyang@jnu.ac.kr

Web Data Extraction using Graph Convolutional Network

Chang-yeong Kim O1 Young-woo Jo 1 Myung-suk Kim 2 *Hyung-Jeong Yang 1

1 Department of Artificial Intelligence Convergence, Chonnam National University

2 Narosu Co.,Ltd

요 약

4차 산업의 발달에 따라 모든 분야에서 대규모 데이터에 대한 중요성과 수요가 늘어나고 있다. 웹은 다양하면서 방대한 데이터를 포함하며 공개되어 있는 경우가 많아 누구나 접근이 가능하여 웹을 활용하는 연구가 지속적으로 수행되고 있다. 그러나 웹의 동적으로 변화하는 구조와 일관성이 없는 구조로 인해 데이터를 수집하기 어렵다는 문제점이 존재한다. 본 논문에서는 웹 페이지가 트리 구조라는 점을 이용하여 그래프 합성곱 신경망을 기반으로 데이터를 추출하는 방법을 제안하고 아마존 쇼핑몰에서 수집한 웹 페이지에서 상품의 정보를 추출하는 실험을 통해 제안 모델의 성능을 검증한다.

1. 서 론

최근 웹에서 수집된 대규모 데이터가 검색 엔진, 추천 시스템, 소셜 미디어 분석 등 비지니스나 의사 결정에 적극적으로 활용하고 있다. 웹의 풍부한 정보를 활용해 새로운 지식을 추출하기 위한 수집 및 분석 기술이 지속적으로 연구되어 왔으며, 최근 딥러닝 기술의 발전은 이를 더욱 가속화시켰다.

그러나 웹페이지는 동적이고 구조가 일관적이지 않아 웹에서 데이터를 수집하는 데 어려움이 발생한다. 웹페이지는 수시로 업데이트 될 수 있으며, 웹사이트에 따라 다양한 패턴의 레이아웃을 가지고 있다. 또한 웹페이지 자체가 무관하거나 좋지 않은 품질의 데이터를 포함하는 등 노이즈가 있을 수 있다. 이외에도 웹페이지의 정보가 영상, 음성, 텍스트 등 다양한 형태로 제공되기 때문에 멀티미디어 데이터에 대한 고려도 필요한다. 또한 대부분의 정보가 웹 인터페이스 뒤에 숨겨져 있다는 문제점이 존재한다. 이러한 문제점으로 인해 효율적으로 필요한 웹 데이터의 수집 및 분석 기술이 요구된다[1].

웹데이터를 수집하는 일반적인 방법으로 사람이 일일이 웹 문서를 분석해 데이터를 추출하는 규칙을 만드는 방식이 있다[2]. 이러한 방식은 웹 페이지가 업데이트 될 때마다 규칙을 추가하거나 변경해야 한다는 문제점이 존재한다. 최근 딥러닝이 활발히 적용되면서 매번 규칙을 추가하거나 수정할 필요 없이 모델이 패턴을 인식해 데이터를 추출하는 방식이 제안되고 있다. 그러나 웹 페이지는 일반적인 딥러닝 모델이 처리하기에는 입력의 크기가 매우 크다는 문제점이 존재한다[3].

본 논문에서는 이러한 문제점을 해결하기 위해 웹 페이지를 문서 객체 모델(Document Object Model, DOM)을 활용해 각 요소들을 기반으로 그래프로 구성하고 그래프 합성곱 신경망을 활용해 임베딩하여 데이터가 있는 영역을 추출하는 방법을 제안하고자 한다. 또한 아마존 쇼핑몰에서 수집한 상품들에 대한 정보가 담긴 웹 페이지 데이터셋으로 실험함으로써 그 성능을 검증한다.

2. 관련 연구

웹은 정형 데이터와 비정형 데이터가 모두 존재하는 반정형 데이터이다. 따라서 정보를 추출하기 위해서는 데이터를 표현하는 방식이 고려되어야 한다. 기존의 표현 방식에는 크게 Vision 기반[4, 5], DOM 기반[3, 6, 7, 8]의 방식이 존재한다.

Vision 기반 방식은 렌더링된 웹페이지를 OCR과 같은 영상 처리 기술을 통해 시각 정보를 분석하고 데이터를 추출하는 방식이다. 웹 페이지의 구조 변화에 강하면서 인간이 이해할 수 있는 시각적인 신호를 분석한다는 장점이 있다. 그러나 보통 웹 페이지는 일반적인 영상처리 모델의 최대 입력 길이를 초과하며, 렌더링 되는 기기에 따라 이미지 입력이 달라질 수 있다. 이를 해결하기 위해 이미지를 고정 크기로 변환하면 정보가 손실되며, 정보 손실을 막기 위해 이미지를 분할하면 정보가

*: 교신저자

분리될 수 있다는 단점이 존재한다.

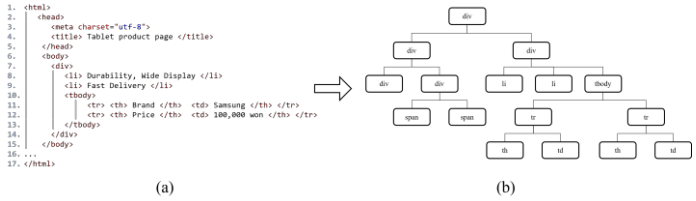


그림 1. (a)는 HTML, (b)는 DOM의 그래프 트리 구조를 나타낸다.

DOM 기반 방식은 그림 1과 같이 웹 페이지를 트리 구조로 변환하여 정보를 추출하는 방식이다. HTML의 요소들과 해당 요소들의 Tag, Attribute, Text 등이 노드 형태로 표현된다. 그래프 구조를 분석하여 추출하는 방식[6, 7]과 HTML은 Markup 언어 기반이라는 점에서 문서로 간주되어 자연어 처리 기술을 적용해 데이터를 추출하는 방식[3, 8]이 있다. 이는 웹의 소스를 통해 직접적으로 구조를 분석한다는 장점이 있다. 그러나 시각 정보가 손실되며, 문서 처리 모델의 경우, 일반적인 언어 모델에 비해 입력 사이즈가 매우 크다는 문제점이 존재한다.

3. 제안 방법

그래프 신경망(Graph Neural Network, GNN)은 노드(Node)와 노드 사이의 연결인 엣지(Edge)로 이루어진 그래프 구조를 처리하는 데 특화된 신경망이다. 각 층을 통과할 때마다 노드의 정보를 엣지를 통해 주변 노드에서 중심 노드로 메시지 전파(Message Passing)시키며 그래프를 효과적으로 임베딩하며 대규모 데이터에 효율적이다[9].

본 논문에서는 DOM 기반 방식으로 구조를 고려하기 위해 그래프 합성곱 신경망(Graph Convolutional Network, GCN)[10]을 사용한다. 웹 페이지는 무방향 그래프 $G = (V, E)$, 각 요소는 노드 $v_i \in V$, 노드 간 연결인 엣지는 $e_i \in E$로 정의된다. 엣지는 인접 행렬 $A \in \mathbb{R}^{N \times N}$로 표현되며 self-loop를 적용한 $\tilde{A} = A + I$를 모델의 입력으로 사용한다. D는 인접행렬 A의 차수행렬로 A의 정규화에 사용된다. GCN의 각 층의 출력 $H^{(l+1)}$은 다음과 같이 표현된다.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)} + b^{(l)}) \quad (1)$$

$H^{(0)}$은 노드의 특징을 나타내며 노드의 특징으로 각 요소의 tag, attribute가 사용된다. W는 모델의 가중치를 나타내며 합성곱 신경망과 같이 각 노드의 여러 특징맵을 추출한다. σ는 활성화 함수로 ReLU를 사용한다. Message Passing으로 $H^{(l+1)}$의 평균을 사용한다. 마지막으로 출력된 그래프 임베딩의 각 노드에 대해 선형 레이어와 그 출력에 Softmax를 적용해 추출하고자 하는 속성의 데이터 영역을 분류한다.

4. 실험

4.1 데이터

본 논문에서 제안하는 모델의 실험을 위해 아마존 쇼핑몰에서 코트, 신발, 셔츠, 속옷, 치마, 바지, 청바지 등 12개 카테고리에 해당하는 제품들을 수집하였다. 총 16,815개로 상품 페이지의 제목, 이미지 소스, 가격, 옵션, 스펙, 상세 설명, 요약 설명 7가지 속성의 데이터 영역과 그에 대한 Xpath를 구하였다. 데이터 통계는 그림 2와 표 1과 같다.

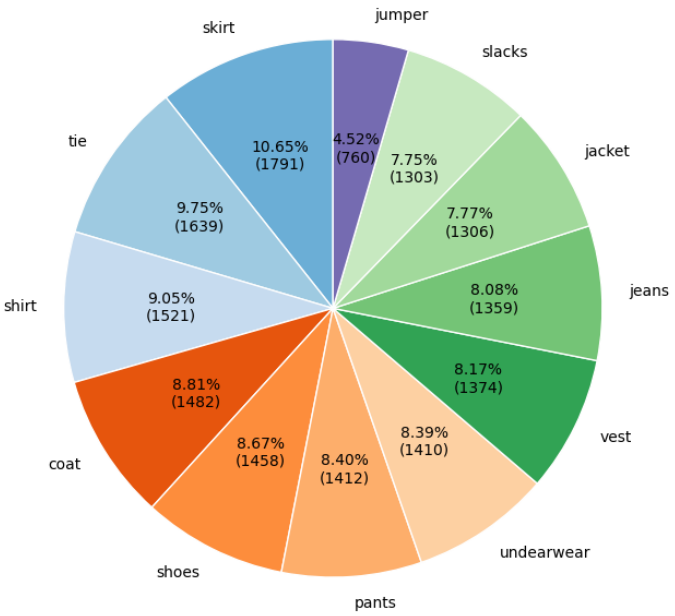


그림 2. 데이터 비율

표 1. 웹 페이지별 그래프 통계

Depth		Nodes		Edges	
mean	39.9979	mean	4043.2583	mean	4042.2583
std	2.6863	std	696.5542	std	696.5542
min	25	min	1769	min	1768
25%	41	25%	3580	25%	3579
50%	41	50%	4047	50%	4046
75%	41	75%	4502	75%	4501
max	43	max	7016	max	7015

4.2 실험 결과

본 논문에서는 3개의 GCN 층으로 구성된 모델을 사용하였다. 전체적인 과정은 그림 3과 같다. 학습 데이터 13,452개, 검증 데이터 1,681개, 평가 데이터 1,682개로 분할하여 학습 및 평가에 사용하였다. Optimizer로 Adam을 사용하였으며 데이터 불균형 해소를 위해 Focal Loss를 사용하였다. 실험 결과, macro f1 score가 98.71%의 성능을 달성하였으며 상품의 속성별 결과는 표 2와 같다.

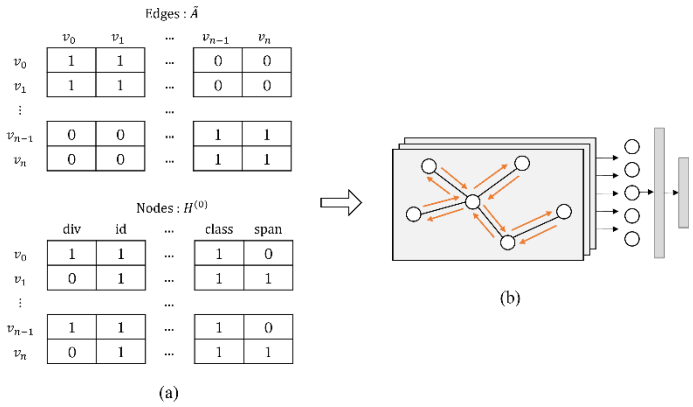


그림 3. (a)는 모델의 입력, (b)는 그래프 합성곱 신경망을 나타낸다.

표 2. 실험 결과

Class	Precision	Recall	F1-score	Support (Node)
Unrelated	99.99%	99.99%	99.99%	10206165
Title	100%	100%	100%	1456
Src	96.55%	98.07%	97.31%	1456
Price	97.14%	99.34%	98.23%	1370
Option	100%	98.18%	99.1%	1428
Specs	99.93%	94.64%	97.21%	1456
Detailed description	100%	98.23%	99.11%	509
Summarized description	99.58%	97.87%	98.72%	1456

5. 결론

본 논문에서는 그래프 합성곱 신경망을 통해 웹 페이지의 구조를 직접적으로 분석함으로써 데이터 영역을 추출하는 방법론을 제안하였다. 제안된 방법은 웹 페이지의 구조가 변하더라도 어느 정도 데이터를 추출할 수 있음을 보여준다. 그러나 아마존이라는 도메인에서만 수집된 한정된 데이터로는 일반화된 성능을 검증하기에 부족하다는 문제점이 있다.

향후 연구에서 벤치마킹 데이터[11, 12]와 기존 모델들과 비교[3, 6, 7, 8]를 통해 제안 모델의 일반성을 검증하고자 한다. 또한 웹 페이지의 구조 뿐만 아니라 텍스트, 이미지와 같은 비정형 데이터를 임베딩하여 노드의 특징으로 사용해 멀티미디어를 처리할 수 있는 모델을 개발하고자 한다.

Acknowledgment

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2023-RS-2023-00256629)

본 성과물은 중소벤처기업부에서 지원하는 2024년도 산학연Collabo R&D사업(RS-2023-00225642)의 연구수행으로 인한 결과물임을 밝힙니다.

참 고 문 헌

[1] Kumar, M., Bhatia, R. and Rattan, D. “A survey of Web crawlers for information retrieval,” WIREs Data Mining and Knowledge Discovery Volume 7, Issue 6, 2017.

[2] Khder, Moaiad Ahmad, “Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application,” International Journal of Advances in Soft Computing and its Applications, 2021.

[3] Li, Junlong, Yiheng Xu, Lei Cui and Furu Wei, “MarkupLM: Pre-training of Text and Markup Language for Visually Rich Document Understanding,” Annual Meeting of the Association for Computational Linguistics, 2021.

[4] S. K. Patnaik, C. N. Babu and M. Bhave, "Intelligent and adaptive web data extraction system using convolutional and long short-term memory deep learning networks," in Big Data Mining and Analytics, vol. 4, no. 4, pp. 279-297, Dec. 2021.

[5] Dr P. Tamije Selvy, Ms M. Anitha, L. R. Vishnu Varthan, P. Sethupathi and S. P. Adharsh, "Intelligent Web Data Extraction System for E-commerce," JOURNAL OF ALGEBRAIC STATISTICS, Vol. 13 No. 3, 2022.

[6] M. Mohammadi, M. J. Shayegan and N. Latifi, "Web Content Extraction by Weighing the Fundamental Contextual Rules," 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS), pp. 01-08, 2021.

[7] N. Utiu and V. -S. Ionescu, "Learning Web Content Extraction with DOM Features," 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 5-11, 2018.

[8] Wang, Qifan, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan and Dongfang Liu, “WebFormer: The Web-page Transformer for Structure Information Extraction,” Proceedings of the ACM Web

Conference, 2022.
[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in Proc. ICML, pp. 1263-1272, 2017.
[10] Kipf, Thomas and Max Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” ArXiv abs/1609.02907, 2016.
[11] Hao, Qiang, Rui Cai, Yanwei Pang and Lei Zhang, “From one tree to a forest: a unified solution for structured web data extraction,” Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, 2011.
[12] “Common crawl.” [Online]. (https://commoncrawl.org/).

감정인식 기반 Text-to-Speech : 감정표현 및 인식 훈련을 위한 플랫폼 개발

양현지 [1O], 조영우 [1], 유연수 [1], 양형정 [1*]

전남대학교 [1]

yhj22@jnu.ac.kr, 217433@jnu.ac.kr, powinz00@naver.com, *hjyang@jnu.ac.kr

Emotion Recognition Based Text-to-Speech : Development of a Platform for Emotion Expression and Recognition Training

Hyeon Ji Yang [1O], Young Woo Jo[1], Yeon Soo You[1], Hyung Jeong Yang[1*]

Chonnam National University[1]

요 약

본 논문은 감정표현에 어려움을 겪는 사람들을 위한 웹 기반 훈련 플랫폼을 제안한다. 플랫폼은 감정인식 모델(HuBERT)과 음성합성모델(VITS)을 통합해 사용자의 감정 상태를 인식하고, 해당 감정을 포함한 음성을 생성한다. HuBERT는 자기지도학습으로 음성의 세부 특징을 분석하며, VITS는 VAE와 GAN을 결합하여 다양한 감정과 스타일의 고품질 음성을 생성한다. 사용자는 훈련하고자 하는 감정을 선택하고 텍스트를 입력하여 음성을 생성한 후, 녹음한 자신의 음성과 생성된 음성을 비교함으로써 감정을 훈련할 수 있다. 감정 표현이 일치하면 다른 감정을 훈련하고, 불일치하면 반복 훈련을 진행한다. 실험 결과, 음성감정인식모델은 약 89%의 정확도를, 음성합성모델은 평균 MOS 점수 4.81을 달성하여 높은 음성 품질을 보여주었다.

1. 서 론

감정은 사람의 내면적 경험과 사회적 상호작용에서 핵심적인 역할을 한다. 일상생활에서 행복, 슬픔, 분노 등 다양한 감정을 경험하며, 이러한 감정들은 개인의 사고와 행동에 깊은 영향을 끼친다. 또한 타인의 감정을 이해하고 적절히 반응하는 것은 공감 능력을 발달시키고 인간관계에서 중요하다[1]. 일반적인 사람들은 성장 과정 중에 자연스럽게 감정을 표현하고 인식하는 방법을 배우지만 감정표현이 어려운 일부 사람들은 이러한 과정에서 어려움을 겪는다.

감정표현 및 인식의 어려움은 사회적 상호작용과 학습 능력에 영향을 미친다. 이들을 위해 다양한 연구들이 진행되고 있으며, 초기 연구에서는 일반적인 사람과 감정표현이 어려운 사람의 감정 이해 및 표현의 차이를 연구하였다. 최근에는 이러한 사람들의 사회적 기술 개발을 위해 타인의 감정을 더 잘 이해하고 자신의 감정을 효과적으로 표현할 수 있도록 하는 연구에 초점을 맞추고 있다[2].

M. Wysocka[3]에서는 발화 및 운율 개선을 위한 치료 프로그램을 제안하였으며, 이는 말하기 운율을 향상시키는 것을 목표로 한다. 발화 및 운율 개선 프로그램은 총 세 단계로 구성되어 있다. 1단계는 소리의 개별적인 특징(높이, 크기, 지속 시간, 음색)을 의식적으로 인식하고 청각 자가 통제 기능을 포함한다. 2단계는 말에서 개별 억양 현상(억양, 어휘 및 구문 강세, 리듬 및 템포 등)을 표현하기 위한 훈련으로 구성되며, 치료자의 구체적인 권장 사항을 따른다. 3단계는 이전 단계에서 훈련한 내용을 사용하여 음성 발화를 한다.

본 논문에서는 감정표현이 어려운 사람들의 감정표현 능력을 향상시키기 위해 감정인식과 음성합성기술을 활용한 감정표현 훈련 플랫폼을 제안한다. 이 플랫폼은 감정인식모델에서 발화자의 음성을 통해 감정을 인식하고, 인식된 감정으로 음성합성모델에서 감정이 포함된 음성을 생성한다. 이렇게 생성된 음성을 통해 감정인식 및 표현 학습이 진행되며, 자신의 감정을 더 잘 이해하고 표현하는 데 도움이 될 것으로 기대한다.

2. 감정표현 및 인식 훈련을 위한 플랫폼

본 연구에서 개발한 플랫폼은 모델의 결과를 빠르게 제공하기 위해 경량화된 FastAPI 기반의 백엔드와 HTML, CSS, JavaScript를 사용한 프론트엔드로 구성하였다. 백엔드는 ASGI (Asynchronous Server Gateway Interface) 웹 애플리케이션인 Uvicorn과 Starlette를 활용하여 감정인식 및 음성합성모델을 효율적으로 관리한다. 프론트엔드는 사용자 인터페이스를 제공하며, HTTP 요청을 통해 백엔드와 상호작용 한다.

플랫폼은 음성감정인식모델인 HuBERT[4]와 음성합성모델인 VITS[5]를 활용하여 감정표현 훈련프로그램을 설계하였다.

* : 교신저자

HuBERT는 자기지도 학습 기반이며, 음성의 숨겨진 특징을 잘 포착하고, 배경 소음이나 음성의 미묘한 변화에도 강하다는 점에서 음성감정인식모델로 선택하였다. VITS는 음성의 품질을 향상시키기 위해 VAE와 GAN을 사용하였으며, 빠른 학습 속도와 높은 데이터 효율성, 음성의 다양성과 뛰어난 표현력을 보여준다는 점에서 음성합성모델로 선택하였다.

플랫폼은 Figure 1과 같으며, 총 여섯 단계의 절차를 따른다. 사용자는 생성할 음성의 성별을 선택하고, 훈련할 감정을 기쁨, 슬픔, 분노 중에서 선택한다. 그 후 제공된 스크립트를 참고하여 예시 문장을 선택하거나 문장을 입력하고, 이를 바탕으로 처음 자신이 선택했던 감정으로 목소리를 녹음한다. 녹음된 음성은 감정인식모델을 통해 분석되어 자신의 발화 감정을 알 수 있다. 최종적으로 음성합성모델을 통해 자신의 감정을 포함하여 생성된 음성과 녹음된 자신의 음성을 비교하여 들어볼 수 있다.

플랫폼을 활용한 학습 방법은 다음과 같다. 처음 선택했던 감정과 자신의 발화 감정이 일치할 경우 감정표현이 가능하다고 판단하여 감정 선택 단계로 이동 후 다른 감정에 대한 훈련을 진행한다. 그러나 불일치할 경우 다시 녹음하는 단계로 돌아가 목소리를 다시 녹음하여 반복적인 훈련을 진행하게 된다.

Figure 1. 플랫폼 순서도

4. 실험 결과

HuBERT 모델은 AIHub의 감성 및 발화 스타일별 음성합성데이터[1]를 사용하여 학습되었으며, 25,930개의 음성 파일을 사용하였다. Train, Validation, Test는 8:1:1로 나누어 진행하였으며, Table 1과 같이 약 89%의 정확도를 달성하였다.

VITS 모델은 AIHub의 감성 및 발화 스타일 동시 고려 음성합성데이터[1]를 사용하여 학습되었으며, 290,559개의 음성 파일을 사용하였다. Train, Validation은 8:2로 나누어 진행하였으며, 평균 MOS는 4.81의 품질 결과를 보여주었다[2].

[1]이 연구는 과학기술정보통신부의 재원으로 한국지능정보사회진흥원의 지원을 받아 구축된 "감성 및 발화 스타일별 음성 합성 데이터, 감성 및 발화 스타일 동시 고려 음성 합성 데이터 "를 활용하여 수행된 연구입니다. 본 연구에 활용된 데이터는 AI 허브(aihub.or.kr)에서 다운로드 받으실 수 있습니다.

[2] 본 실험은 'AI 기반 김남주 시인 P-TTS 서비스 플랫폼 구축 프로젝트'의 일환으로, 모델의 음성 품질 평가를 목적으로 진행되었으며, (주)솔트룩스 이노베이션의 의뢰를 받아 어니컴에서 수행하였다.

Table 1. 음성감정인식모델(HuBERT) 실험 결과

	Precision	Recall	F1-score
Neutral	0.87	0.90	0.89
Happy	0.88	0.87	0.87
Sad	0.93	0.89	0.91
Angry	0.87	0.89	0.88
Accuracy	-	-	0.89
Macro avg	0.89	0.89	0.89
Weighted avg	0.89	0.89	0.89

5. 결론

본 논문에서 개발한 감정인식 기반 Text-to-Speech 플랫폼은 사용자가 감정표현 능력을 향상시킬 수 있도록 돕는다. 사용자는 원하는 감정과 텍스트를 선택하여 자신의 음성을 녹음하고, 음성합성모델이 생성한 음성과 비교해볼 수 있다. 이러한 반복적인 훈련을 통해 감정 표현 능력이 향상될 것으로 기대된다. 향후 연구에서는 개인 맞춤형 음성합성 플랫폼을 개발하여 사용자의 목소리로 감정표현이 가능한 모델을 연구하고자 한다.

ACKNOWLEDGMENT

이 논문은 2023년 광주광역시청의 재원으로 인공지능산업융합사업단의 지원을 받아 수행된 연구임(AI기반(김남주 시인)P-TTS 서비스 플랫폼 구축). 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00219107). 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음(IITP-2023-RS-2023-00256629).

참 고 문 헌

[1] M. Claudia, M. Penny, L. Pablo, P.Brian, “Emotions in Social Interactions : Unfolding Emotional Experience,” *Emotion-oriented systems: The humaine handbook,* pp.31-46, 2011.

[2] 송인혜, 김은경, "시각적 지원을 통한 이야기 읽기 교수가 자폐 스펙트럼 장애 아동의 기본 감정 이해 및 표현에 미치는 효과," *정서·행동장애연구,* Vol. 27, No. 3, pp. 87-122, 2011.

[3] M. Wysocka, M. Kwaterkiewicz, "Therapy Program for Improving the Expression Speech Prosody." *Logopedia*, Vol. 47, No. 2, pp. 251-270, 2018.

[4] W. Hsu, B. Bolte, Y.H. Tsai, K. Lakhotia et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 29, pp. 3451-3460, 2021.

[5] J. Kim, J. Kong, J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *International Conference on Machine Learning*, PMLR, pp. 5530-5540, 2021.

자연어 기반 요구 사항을 활용한 UML 상태 다이어그램을 통해 만화 이미지 생성

진예진 [O1], 공지훈 [2], 김기두 [3], 장우성 [4], 김영철 [5]
홍익대학교 소프트웨어공학연구실 [1,4,5], 툰스퀘어 [2], 한국정보통신기술협회 [3]
yejin_jin@g.hongik.ac.kr [O1], john.tooning@toonsquare.co[2], kdkim@tta.or.kr[3],
uriel200@hongik.ac.kr[4], bob@hongik.ac.kr[5]

Generating a Cartoon Image via UML State Diagram with Natural Language-based Requirement Specs.

Ye Jin Jin [O1], Ji Hoon Kong[2], Ki Du Kim[3], Woo Sung Jang[4], R. Young Chul Kim[5]
SELab., Hongik University[1,4,5], Toonsquare[2], Telecommunication Technology Association[3]

요 약

현재 소프트웨어 산업에서는 자연어 기반의 요구사항 이슈가 대두되고 있다. 그러나 이러한 요구사항 분석의 정확도가 문제이다. 현재에도 요구사항 단계에서 오류가 가장 많이 발생한다. 자연어 기반의 요구사항 정의 및 분석이 절대적으로 필요한 시점이다. 이러한 문제를 해결하기 위해 자연어 기반 요구사항 분석에 언어학자 Chomsky와 Fillmore 이론의 언어학을 적용한다. 즉 형태소와 명사의 의미를 식별한다. 이러한 메커니즘을 바탕으로 자연어 기반의 만화 이미지를 생성을 제안한다. 이를 통해 만화 이미지 생성에 새로운 접근 방식을 제안한다.

1. 서 론

본 연구는 인공지능 기반 사용자 대화형 멀티모달 인터랙티브 스토리텔링 3D 장면 제작 기술 개발에 관한 프로젝트의 성과를 담고 있다. 소프트웨어 개발 프로세스는 크게 요구사항 분석, 설계, 구현, 테스트, 유지보수로 나눠진다. 이 중 소프트웨어 오류의 66%가 시스템 명세 단계에서 나타난다[1].

본 연구에서는 요구사항을 명세하고 설계하는 과정에서 오류를 줄이기 위해 UML 모델을 사용한다. UML은 자연어에 비해 가시성이 높다는 장점이 있다. 자연어로부터 UML을 자동 추출하기 위해 촘스키의 구문 구조 분석을 통해 전체 문장의 구조를 파악하고, 형태소 단위로 분석한다. 분석된 문장 요소 간의 관계를 분석하기 위해 필모어의 의미역을 사용한다. 본 연구에서는 UML 중 상태 다이어그램을 추출하기 위해 언어학 기반의 추출 정보와 상태 다이어그램 생성에 필요한 항목을 매핑한다. 이를 통해 만화 이미지 생성이 가능한 Fabric.js를 활용하여 JavaScript 코드 템플릿을 구성한다.

본 논문의 구성은 다음과 같다. 2장은 연구와 관련된 어프로치에 대해 언급한다. 3장은 제안하는 메커니즘에 대해 언급하고 4장은 메커니즘을 적용한 사례 연구를 설명한다. 마지막으로 결론에 대해 언급한다.

2. 요구사항 모델링을 통한 상태 추출 어프로치

해당 연구는 시스템의 요구사항을 이해하기 위해 상태와 모드를 기반으로 자연어 요구사항을 모델링 한다[2]. 해당 연구에서 제안한 모델은 정해진 템플릿에 맞춰 자연어 요구사항이 표현된다. 이에 따라 모든 유형의 자연어 요구사항을 적용하기 어렵다는 단점이 있다.

본 연구에서는 언어학을 사용하여 복잡하게 구성된 요구사항을 체계적으로 분석한다. 규칙적인 템플릿의 제한 없이 사용할 수 있다.

3. 만화 이미지 생성 프로세스

본 연구에서 제안하는 자연어로부터 만화 이미지를 생성 프로세스는 그림 1과 같다.

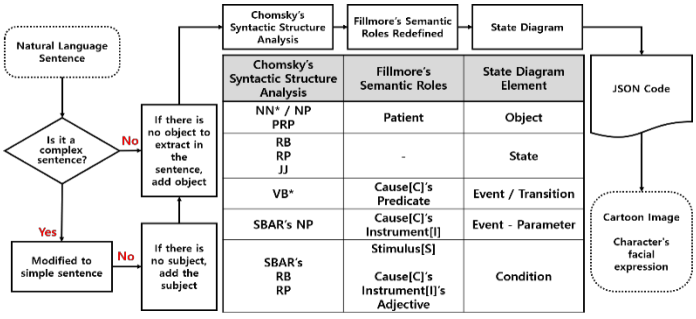


그림 1 자연어로부터 만화 이미지 생성 프로세스

자연어 문장이 입력으로 들어오면 복문을 단문으로 전처리한다. 버클리 파서를 통해 전처리 문장의 구문 구조를 분석한다. 분석된 구성 요소마다 재정의된 의미역을 부여한다. 언어학적 요소를 바탕으로 상태 다이어그램을 생성한다. 이 때, 상태 다이어그램의 상태는 구문 구조 분석 단계의 형태소와 매핑한다.

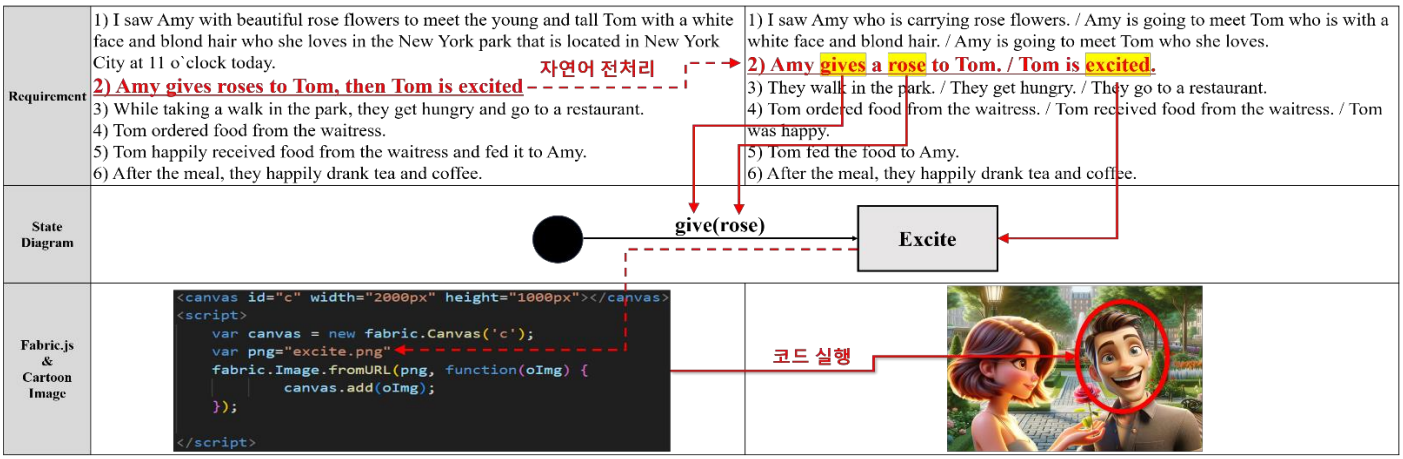


그림 2 제안하는 프로세스 적용 사례 예시

매핑된 요소들을 바탕으로 상태 다이어그램을 추출한다. 추출된 상태는 만화 객체의 상태와 연결되고, 객체의 상태는 감정에 따른 표정의 변화로 이미지를 생성한다.

4. 적용 사례

본 연구에서 제안한 메커니즘을 만화 스토리에 적용하면 그림 2와 같다. 그림 2는 요구사항 문장을 분석하여 상태 다이어그램 일부와 이미지 생성을 위한 JavaScript 코드를 생성한다.

1) 자연어 분석

우선, 자연어 요구사항 문장을 단문 화 과정에 따라 재구성해야 한다. 기존의 두 번째 문장은 두개의 문장이 연결되어 있다. 따라서 이를 단문 화한다.

2) 버클리 파서를 통한 문장의 형태소 식별

단문 화된 문장은 촘스키의 이론 바탕의 버클리 파서를 통해 형태소를 식별한다. 파서는 문장, 구, 절, 명사로 분석되고 POS 태그로 품사를 확인할 수 있다.

3) 형태소에 매핑되는 필모어의 역할 분석

품사 분석 결과를 통해 역할을 분석한다. 품사와 의미론적 역할을 매핑한 표를 바탕으로 역할을 유추하고, 주동사와의 관계와 의미에 따라 역할을 부여한다. 첫번째 문장의 동사 gives는 어떠한 감정의 변화를 일으킨 원인이기 때문에 Cause이고, 이 문장 안에 명사 rose는 이때 사용된 도구이기 때문에 Instrument이다. 두번째 문장의 주어 Tom은 상태의 변화를 겪기 때문에 Patient이고 이때, 형용사 excited가 변화된 상태가 된다.

4) 상태 다이어그램의 핵심 속성과 역할 매핑

기존 연구에서 만화 생성에 사용된 객체 다이어그램을 참고하여, 만화의 주요 객체를 식별한다[3]. 언어학 분석을 거친 문장의 각 요소는 상태 다이어그램의 요소와 매핑된다. 이를 바탕으로 주요 객체의 상태 다이어그램을 생성한다.

5) 상태 다이어그램 속성을 만화 속성과 매핑하여 만화 이미지 생성

주 객체의 감정이 나타나는 표정 에셋을 이미지 파일로 호출하고 이를 캔버스에 나타나도록 한다.

5. 결 론

본 연구는 자연어 요구사항 분석을 통한 상태 다이어그램 추출과 만화 이미지 생성 메커니즘을 제안한다. 제안된 메커니즘을 활용하면 체계적인 분석을 통해 만화 객체의 상태 변화를 자연스럽게 반영한 만화 이미지를 생성할 수 있다. 또한, 상태 다이어그램을 통해 만화 객체의 상태 변화를 명확하게 파악할 수 있다. 그러나, 주요 객체 하나의 상태 변화만을 나타내기 때문에 세부적인 객체의 상태는 확인할 수 없다.

향후 연구에서는 모든 객체의 상태를 다루고, 상세한 상태 다이어그램을 표현할 수 있도록 한다.

ACKNOWLEDGMENT

본 연구는 2023/2024년도 문화체육관광부의 재원으로 한국콘텐츠진흥원(과제명: 인공지능 기반 사용자 대화형 멀티모달인터랙티브스토리텔링 3D장면 저작 기술 개발, 과제번호: RS-2023-00227917,기여율:50%) 지원과 2023/2024년도 정부(교육부)의 재원으로 한국연구재단기초연구사업(과제명: NLP BERT Model 기반 자동 리팩토링을통한 무결점 코드화 연구, 과제번호: No.2021R1I1A3050407,기여율:50%)의 지원을 받아 수행된 연구임

참고 문헌

[1] Rupinder Kaur and Jyotsna Sengupta, “Software process models and analysis on failure of software development projects,” IJSER, vol.2(2), 2012.

[2] Yinling Liu, and Jean-Michel Bruel, “Modeling of Natural Language Requirements based on States and Modes,” 2022 IEEE 30th International Requirements Engineering Conference Workshops, pp.190-194, 2022.

[3] Jang Hwan Kim, and R. Young Chul Kim, “Cartoon Extraction Mechanism via UML Model based on Natural Language Requirement Specs.,” 10th Annual Conf. on Computational Science & Computational Intelligence, 2023

비정형 자연어 요구사항으로부터 UML 시퀀스 다이어그램 및 툰 이미지 생성 메커니즘

김현태 [1], 공지훈 [2], 박영식 [3], 박찬솔 [4], 이상호 [5], 김영철 [6]

홍익대학교 소프트웨어공학연구실 [1,3,4,6], 툰스퀘어 [2], 라스테크 [5]

[1]hyuntaekim@g.hongik.ac.kr, [2]john.tooning@toonsquare.co,
[3]park12160422@selab.hongik.ac.kr, [4]c2193102@g.hongik.ac.kr,
[5]shlee7200@gmail.com, [6]bob@hongik.ac.kr

A Toon Image and UML Sequence Diagram Generation Mechanism from Informal Natural Language Requirement Specifications

Hyun Tae Kim [1], Ji Hoon Kong[2], Young Sik Park[3], Chansol Park[4],

Sangho Lee[5], R. Young Chul Kim[6]

SELab., Hongik University[1,3,4,6], Toonsquare[2], RASTECH[5]

요 약

AI 이미지 생성 기술에서 자연어 기반 프롬프트 입력으로 이미지 생성이 되고 있다. 현재 대부분의 생성형 AI 도구들은 같은 질의에 동일한 그림을 생성 못한다. 그 이유는 같은 질의에도 같은 데이터 에셋을 사용하지 못하기 때문이다. 이러한 문제점을 해결하기 위해 비정형 자연어 요구사항 분석으로 핵심 개체 및 속성들을 분석 및 적용해 시퀀스 다이어그램 및 이미지 생성 메커니즘을 제안한다. 언어학적 방법을 통해, 자연어 분석으로 형태소 및 의미 역할을 식별하여, 시퀀스 다이어그램의 속성들과 연관성을 추출한다. 이를 이미지 내의 개체와 접목한다. 제안한 메커니즘을 통해 동일한 이미지 요소 에셋을 사용해 일관성 있는 그림 생성을 기대한다.

1. 서 론

본 연구는 인공지능 기반 사용자 대화형 멀티모달 인터랙티브 스토리텔링 3D 장면 제작 기술 개발에 관한 프로젝트의 성과를 담고 있다. 최근 AI 기술을 활용해 텍스트 기반의 이미지를 생성하는 툴이 빠른 성장을 하고 있다. 이미지 생성 툴을 사용하면 그림 수정, 합성과 생성을 할 수 있다. 대부분의 생성형 AI는 특정 그림의 요소를 계속 사용할 수 없다는 문제점이 있다. 또한 사용자가 상세하게 서술한 요구사항의 의도와 부합하지 않는 이미지를 생성한다는 문제점이 있다[1]. 이러한 문제들로 인해 추후 사용자가 수정을 하거나 계속 이미지 생성을 시도해야 하는 불편함이 존재한다.

특정 이미지를 계속해서 사용할 수 있도록 비정형 자연어 요구사항으로부터 UML 시퀀스 다이어그램 및 그림 생성 메커니즘을 제안하고자 한다. 2장에서는 자연어를 분석하기 위한 관련연구에 대해 언급한다. 3장은 비정형 자연어를 분석하여 시퀀스 다이어그램을 추출과 이미지를 생성하는 메커니즘에 대해 서술한다. 4장은 제시하는 메커니즘을 활용한 기대효과와 결론에 대해 언급한다.

2. 버클리 뉴럴 파서

비정형 자연어를 분석하기 위해서는 먼저 문장의 품사 판별하는 구문 구조 분석을 진행해야 한다. 문장 구조 분석을 통해 시퀀스 다이어그램에서 사용할 객체의 후보와 메시지를 정한다.

다이어그램의 요소를 찾기 위해 'NN*', 'PRP'와 'VB' 품사를 사용한다. 본 연구에서는 문장의 품사 분석을 하기 위해 버클리 뉴럴 파서를 사용한다. 파서는 문장을 중첩된 하위 구문으로 분해하여 구문 구조 및 품사 등을 분석하여 트리 구조로 표현한다[2]. 파서를 사용해 찾은 NN*과 PRP*를 객체의 후보로 정하고, VB*를 메시지로 사용한다.

3. 이미지 생성 메커니즘

3.1. 시퀀스 다이어그램의 핵심 속성과 격문법 매핑

비정형 자연어 요구사항을 기존 연구와 같이 파서와 격문법을 이용해 분석한다[3,4]. 분석한 결과 중 PRP*와 NN*의 요소에서 사람과 같이 행동을 할 수 있는 명사만을 다이어그램에서의 객체로 사용한다. VB*는 메시지로 사용한다.

표 1. 격문법과 시퀀스 다이어그램 요소 매핑

격문법	시퀀스 다이어그램 요소
Actor	Actor
Theme	Object
Source	
Target	
VB*	Message

표1은 필모어의 격문법의 요소와 다이어그램의 요소를 매핑한 결과다.

3.2. 시퀀스 다이어그램과 이미지 속성 매핑

문장을 분석하여 생성한 다이어그램을 기반으로 json파일의 요소를 구성한다. 다이어그램의 각 객체는 json파일에서 objects내에 객체의 속성이 정의된다.

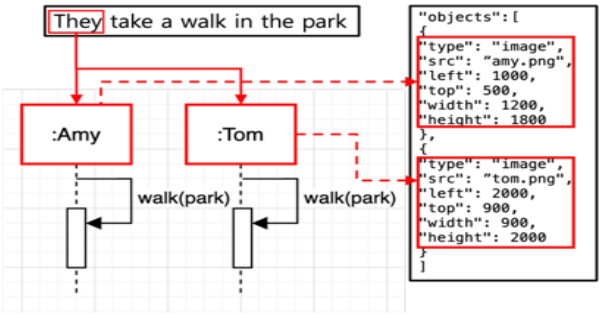


그림 2. 시퀀스 다이어그램 요소와 json 파일 매핑

그림2은 'They take a walk in the park.' 문장의 객체 속성을 정의한 json파일의 일부이다. 그림2의 json파일은 이미지 생성을 위해 사용한다. json파일에 객체의 속성을 정의할 때는 다이어그램에서 객체와 메시지 한 쌍과 매핑한다.

3.3. 이미지 생성

객체의 속성을 정의한 json파일을 FabricJS를 이용해 분석한다[5]. FabricJS를 이용해 json파일에 정의된 객체의 속성을 캔버스 안에 나타낸다. FabricJS를 이용해 생성한 그림은 다이어그램의 흐름에 따라 그림을 배치한다.

그림 3. json 파일을 기반으로 생성한 이미지 결과

'They take a walk in the park.' 문장과 'They go to a restaurant.' 문장을 기반으로 이미지 생성하면 그림 3과 같이 생성된다. 이미지는 json파일 내부에 정의된 객체의 속성을 기반으로 에셋의 위치와 크기가 정해진다.

4. 결론

제안하는 그림 생성 메커니즘을 이용하면 비정형 자연어 요구사항을 파서와 재정의된 필모어의 격문법을 사용하여 분석할 수 있다. 또한 분석한 자연어 요구사항을 이용하여 다이어그램을 생성할 수 있다. 현재 비공식적으로 명세 되고 있는 자연어 요구사항을 분석하여 다이어그램을 생성하면 소프트웨어 설계를 할 시간과 비용을 절약할 수 있다.

추가적으로 다이어그램을 이용하여 생성한 그림을 이용한다면 사용자의 요구사항에 맞는 그림을 그릴 수 있다. 따라서 사용자가 그림 그리는 시간을 줄여 비용을 절약할 수 있을 것이라고 기대된다. 또한 사용자가 원하는 그림 요소 에셋을 지속적으로 사용하여 일관성 있는 그림을 생성해 기존 그림 생성 AI의 단점을 보완할 수 있다.

향후 자연어 분석을 진행할 때 문장 내 단어의 의존성을 분석하여 더욱 정확한 요구사항 분석을 진행할 예정이다. 또한 기본적인 시퀀스 다이어그램의 요소 외의 부가적인 요소를 코드의 요소와 매핑하여 더욱 사용자의 요구사항에 맞는 그림을 생성할 것이다.

ACKNOWLEDGMENT

본 연구는 2023/2024년도 문화체육관광부의 재원으로 한국콘텐츠진흥원(과제명: 인공지능 기반 사용자 대화형 멀티모달인터랙티브스토리텔링 3D장면 저작 기술 개발, 과제번호: RS-2023-00227917,기여율:50%) 지원과 2023/2024년도 정부(교육부)의 재원으로 한국연구재단기초연구사업(과제명: NLP BERT Model 기반 자동 리팩토링을통한 무결점 코드화 연구, 과제번호: No.2021R1I1A3050407,기여율:50%)의 지원을 받아 수행된 연구임.

참고 문헌

[1] Marcus, Gary, Ernest Davis, and Scott Aaronson. "A very preliminary analysis of DALL-E 2." arXiv preprint arXiv:2204.13807, 2022

[2] Berkeley Neural Parser, [Internet], https://parser.kitaev.io/.

[3] H. T. Kim, and R. Y. C. Kim, "Extraction Practices on UML Sequence Diagram through Natural Language based Requirement Specifications," International Symposium on Advanced and Applied Convergence, AACL22, 2023

[4] J. H. Kim, and R. Y. C. Kim, "Cartoon Extraction Mechanism via UML Model based on Natural Language Requirement Specs.," 10th Annual Conf. on Computational Science & Computational Intelligence, Proceeding, 2023

[5] Fabric JS, [Internet], http://fabricjs.com/.

언어학적 의미 분석 기반 요구 공학을 통한 만화(Toon) 이미지 생성

김장환*[1], 공지훈 [2], 장우성 [3], 이근상 [4], 김기두 [5], 김영철 [6]
홍익대학교 소프트웨어공학연구실 [1,3,6], 툰스퀘어 [2],
전북테크노파크 [4], 한국정보통신기술협회 [5],
Janghwan.kim@g.hongik.ac.kr[1], john.tooning@toonsquare.co[2], uriel200@hongik.ac.kr[3]
ksoul406@naver.com[4], kdkim@tta.or.kr[5], bob@hongik.ac.kr[6]

Cartoon Image Generation via Requirement Engineering Based on Semantic Analysis

Janghwan Kim [1], Ji Hoon Kong[2], Woo Sung Jang[3],
Keunsang Yi[4], Kidu Kim[5], R. Young Chul Kim[6]
Hongik University Software Engineering Laboratory[1,3,6], Toonsquare[2],
Jeonbuk Technopark[4], Telecommunications Technology Association[5],

요 약

현재 생성형 인공지능이 전세계적으로 큰 화제다. 생성형 인공지능은 다양한 Image, Art, Video Clip, Advertisement등을 생성한다. 문제는 인공지능 내부의 작업에 대한 검증이 매우 어렵다. 현재 이슈에 요구공학자로써 언어학적 메커니즘을 접목하여 툰 이미지 생성을 시도한다. 이는 언어학자 촘스키와 필모어의 의미역할 분석기법을 통해, UML 객체 모델과의 접목한다. 그리고 도출한 속성들과 툰 생성 템플릿에 연계시킨다. 이는 툰 공학에 창의성 보다는 재사용성 기반 생산성을 보장에 있다. 앞으로는 소프트웨어 개발 프로세스과 재사용성을 접목하여 툰 이미지 생산성을 높이고자 한다.

1. 서 론

최근 딥 러닝을 기반으로 한 생성형 인공지능에 대한 관심이 증가하고 있다. 특히, 텍스트 기반의 이미지 생성 도구들(예: Dall-E3, Midjourney, ChatGPT)의 성장이 두드러진다. 이러한 도구들은 자연어 질의의 의미를 파악해 다양한 이미지를 생성하거나 합성한다[1]. 하지만, 이런 도구들은 동일한 텍스트 입력에도 결과가 일관성이 없다[2].

본 논문에서는 이런 문제를 개선하기 위해 기존 자연어 문장을 입력해 인공지능을 이용한 이미지 생성하는 방법들과 비교하여 자연어 질의를 통해 언어학적 분석과 소프트웨어공학적 설계 기법을 적용하여 결과를 도출하는 방법을 제안한다. 적용한 사례로 UML 다이어그램 중에서 클래스 다이어그램과 객체 다이어그램을 적용하여 자연어 문장으로부터 이미지 생성에 필요한 단계적 공정을 나타낸다.

2장 관련 연구에서는 자연어로 생성형 인공지능 도구에 적용되는 프롬프트와 문장분석 방법을 언급한다. 3장에서는 자연어 문장 분석으로 클래스 다이어그램을 생성하고 이를 기반으로 이미지를 생성하는 메커니즘을 언급한다. 4장에서는 적용사례에 대해 언급하고 5장에서는 결론과 향후연구에 대해 언급한다.

2. 관련 연구

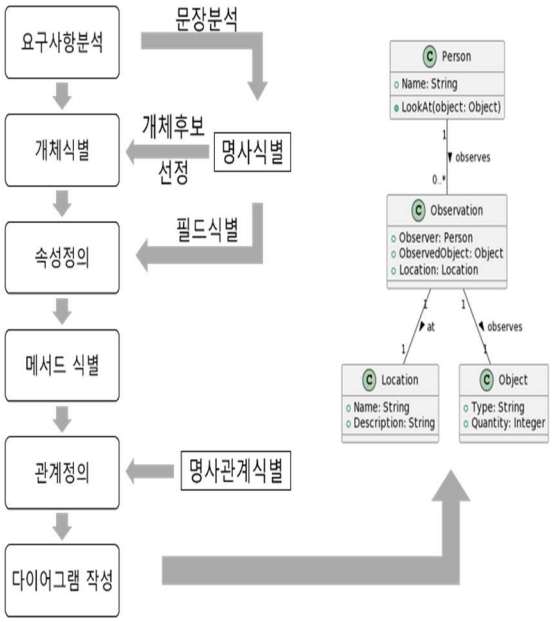


그림 1. 요구사항-클래스 다이어그램 메커니즘

클래스 다이어그램은 시스템의 정적 구조를 나타내는 UML의 핵심 요소이다. 이 다이어그램은 다양한 클래스, 클래스들의 속성, 작업, 그리고 클래스 간의 관계를 나타낸다. 객체 지향 분석과 설계에 중요한 역할을 하며, 시스템의 클래스, 그들의 상호 관계, 그리고 협력 방법을

상세하게 설명한다. 클래스 다이어그램은 소프트웨어 공학에서 클래스 구조를 시각화하고 문서화하는 데 널리 사용된다[3]. 그림 1은 요구사항명세로부터 요구사항 분석을 통한 클래스 다이어그램을 추출하는 방법을 나타낸다. 요구사항 분석 단계에서는 요구사항 명세의 문장들을 분석해 명사를 식별하고, 이를 클래스나 필드 정의에 활용한다.

3. 언어학 메커니즘을 접목한 요구사항 분석 프로세스

본 논문에서는 요구사항에서 클래스 다이어그램을 추출하기 위해 언어학적 접근 방법을 요구공학에 접목한다. 먼저, 구조 분석방법과 의미론적 분석 메커니즘을 통해 분석한다[4-5]. 구조적 분석방법을 통해 도출된 정보를 바탕으로 문장에서 명사들을 추출하고 명사들의 역할과 의미를 도출하기 위해 의미론적 접근방법을 적용한다. 각각의 명사로부터 추상화 정보를 바탕으로 클래스 다이어그램을 작성한다.

자연어 요구사항 입력: There is a house in the forest. Harry is looking at some mushrooms in front of the house.

그림 2. 자연어 요구사항 적용사례 입력 예시

그림2는 제안하는 방법을 적용하기 위한 자연어 요구사항 입력의 예이다. 문장 의미 분석을 통해 속성들을 찾고 클래스 다이어그램을 작성한다. 클래스 다이어그램이 작성되면 클래스 다이어그램의 속성을 갖는 객체 다이어그램을 생성하여 툰 이미지를 생성한다.

4. 생성형 인공지능 비교 적용 사례

제안하는 방법에 따라 생성한 이미지와 이미지를 생성하는 여러 생성형 인공지능 도구들의 차이를 알기 위해 두 방법에 같은 입력 값을 적용시켜 비교 분석한다.

그림3. 생성형 인공지능 도구 비교 테이블

그림3에서 그림2에 있는 동일한 입력을 생성형 인공지능 도구인 미드저니(Midjourney), DALLE-3, 그리고 제안하는 방법을 적용하여 비교한 결과물을 나타낸다. 입력 값에 특정한 묘사정도가 상세히 되어 있지 않기 때문에 이 도구는 다양한 형태로 일관되지 않은 이미지를 생성한다. 1차에서는 Harry(남성 이름)에 대한 이미지도 존재하지 않고 2차에서는 사람 2명이 존재한다. 해당 이미지는 해상도나 이미지의 품질면에서 미드저니보다 높은 품질을 나타낸다. 하지만 동일한 입력 값을 계속 입력할 경우 여전히 일관된 이미지를 생성하지 못한다.

본 논문에서 제안하는 방법은 다량의 이미지 자산(asset)을 보유해야 하는 제약이 있지만, 동일한 이미지를 지속적으로 재생산할 수 있다. 만화와 같이 여러 컷의 이미지를 계속적으로 생산해야 하는 작업에는 이러한 재생산성은 필수적이다.

5. 결 론

본 논문에서는 요구공학에 언어학적 메커니즘을 접목하여 툰 이미지 생성하는 방법을 제안한다. 언어학과 소프트웨어공학적 설계, 분석기법을 통해 자연어문장으로부터 이미지를 생성하는 중간 단계를 공정으로 만듦으로써 재생산성을 높인다. 이러한 방법을 통해 툰 이미지 생성에 소프트웨어 개발 프로세스과 재사용성을 접목하여 툰 이미지 생성에 대한 생산성을 높이고자 한다.

ACKNOWLEDGMENT

본 연구는 2023/2024년도 문화체육관광부의 재원으로 한국 콘텐츠 진흥원(과제명: 인공지능 기반 사용자 대화형 멀티 모달 인터랙티브 스토리텔링 3D장면 저작 기술 개발, 과제번호: RS-2023-00227917, 기여율:50%) 지원과 2023/2024년도 정부(교육부)의 재원으로 한국 연구재단 기초연구사업(과제명: NLP BERT Model 기반 자동 리팩토링을 통한 무결점 코드화 연구, 과제번호: No.2021R1I1A3050407, 기여율:50%)의 지원을 받아 수행된 연구임

참고 문헌

[1] H. Park, A Case Study On Application Of Text To Image Generator AI DALL. E. The Treatise on The Plastic Media, 26(1), 104.

[2] Creely, E. The possibilities, limitations, and dangers of generative AI in language learning and literacy practices.

[3] M. Ibrahim and R. Ahmad, "Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques," 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 2010, pp. 200-204, doi: 10.1109/ICCRD.2010.71.

[4] Chomsky, Noam. Syntactic structures. Mouton de Gruyter, 2002.

[5] C. J. Fillmore, (1967). The case for case.

선박무선통신 음성인식 기술 개발 연구

저자 1 김광일 [1] , 저자 2 유상록 [1] , 저자 3 문일주 [2]
저자 소속 [1]㈜미래해양정보기술, [2] 제주대학교 해양과학대학
저자1 kki@jejunu.ac.kr, 저자2 yoosangrok87@naver.com, 저자 3 ijmoon@jejunu.ac.kr

A Study on Development of Speech Recognition Technique for Ship Radio Communication

Author 1 Kim Kwang-il[1] , Author2 You Sang-rok[1] , 저자 3 Moon Il-ju[2]
저자 소속 [1]Future Ocean IT, [2] College of Ocean Science, Jeju National University

요 약

선박무선통신장비는 선박이 항해하는데 필요한 안전정보, 선박교통 모니터링 및 관제, 입·출항 정보를 교환하기 위한 필수 장비이므로 선박항해사는 무선통신 내용을 항상 주의 깊게 청취해야 함. 본 연구에서는 선박의 실제 음성 교신데이터 500시간 데이터를 수집 및 학습하고, Wav2vec 2.0 모델을 활용하여 음성인식 모델을 개발하고 실용화를 수행하였다. 음성인식 모델의 성능은 CER(Character Error Rate) 기준 94.5%로 향후 선박 운항 관련 다양한 분야에 적용이 가능할 것으로 사료된다.

1. 서 론

선박무선통신장비는 선박이 항해하는데 필요한 안전정보, 선박교통 모니터링 및 관제, 입·출항 정보를 교환하기 위한 필수 장비이므로 선박항해사는 무선통신 내용을 항상 주의 깊게 청취해야 한다[1]. 최근 빅데이터, 인공지능 기술발달로 해양분야에서도 e-Navigation, 자율운항선 등 신기술 도입을 적극 추진중임. 해상통신데이터는 선박 현재상태를 파악할 수 있는 객관적인 데이터로서 이를 문자화(전산화)하여 차세대 시스템과 연계 필요한 실정이다. 또한 향후 자율운항선박 기술이 실용화되면, 자율운항선과 주변선박간 항해의도 파악을 위해 선박무선통신 음성인식 기술이 필수적으로 요구되고 있다.

선박무선통신 교신 내용은 전문분야 대화로서 특수용어, 해사영어(Maritime English), 고유명사(선명 및 지명) 등이 포함되어 있고, 교신 규칙도 포함되어 있어 기존 음성인식 모델로 인식이 어렵다.

이에 본 연구에서는 선박 및 육상 해안무선국에서 송수신되는 선박무선통신 교신데이터를 수집, 가공 및 전처리하여 end-to-end 음성인식 딥러닝 모델을 활용하여 선박무선통신 전용 음성인식 모델을 개발하고자 한다.

2. 선박무선통신 음성인식 모델 개발

본 연구에서 음성인식 모델 개발은 ① 데이터 수집 및 전처리 ② 음향모델(Acoustic Model) ③ 언어모델(Language Model) ④ 후처리 변환 과정으로 수행한다.

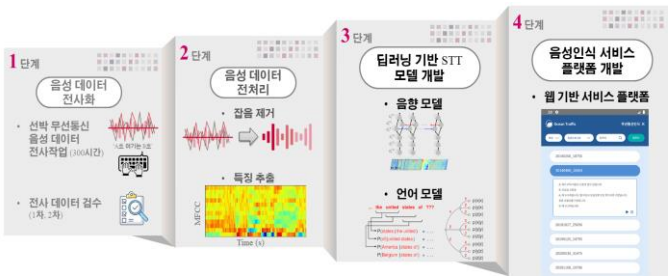


그림 1. 선박무선통신 음성인식 모델 개발 과정

Fig. 1. Process of Speech Recognition Model for Ship Radio Communication

2.1 데이터 수집 및 전처리

학습에 사용된 선박 무선통신 음성데이터는 제주대학교 및 실습선에서 약 3년치 데이터를 확보하여, 전문경력의 라벨러를 통해 약 200시간 학습을 수행하였다. 또한 음성이 장비마다 8 khz 또는 16 khz로 샘플링레이트가 차이가 있어서 8 khz대역으로 통일하였다.

2.2 음향모델 및 언어모델

음향모델은 음성을 만들어내는 음소(phone)와 해당 발음 사이의 관계를 나타내며, 전처리되어 사용되는 데이터와 해당 음성의 대응을 통계적인 정보로

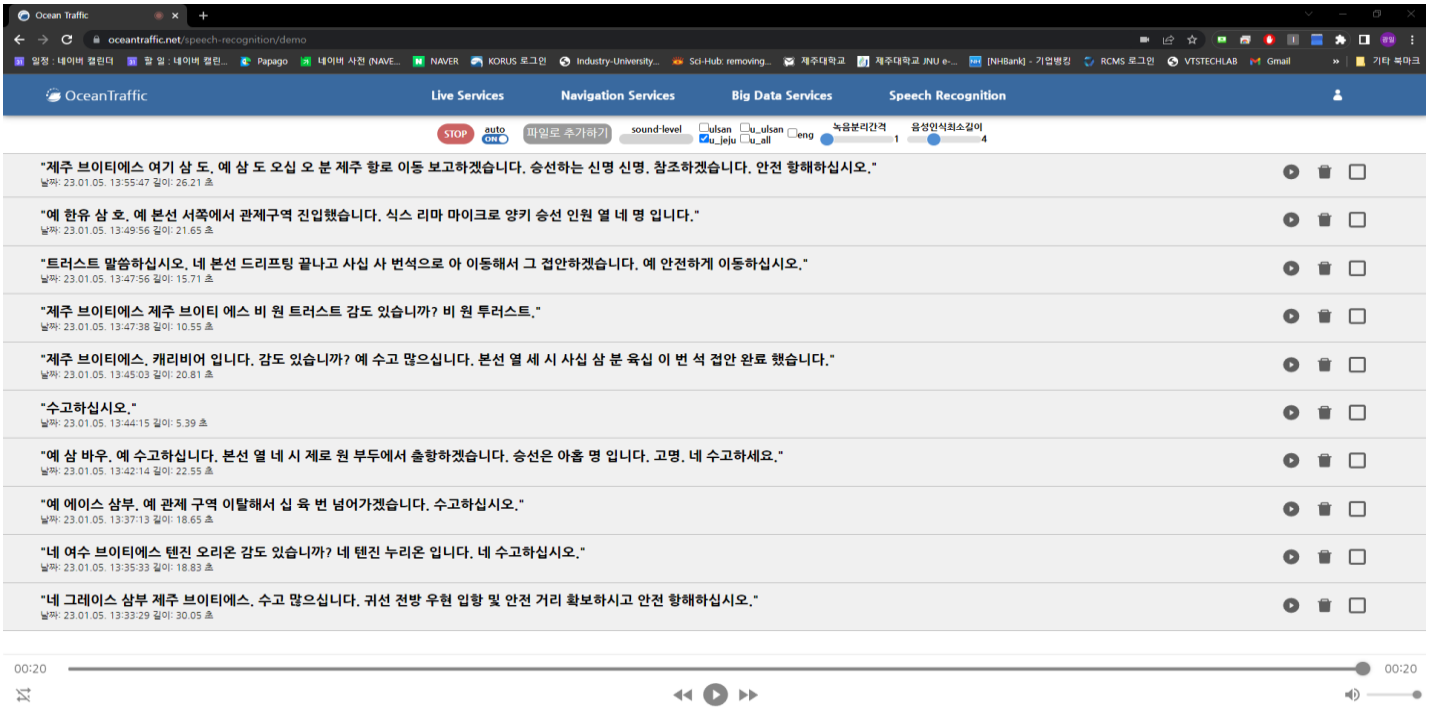


그림 2. 선박무선통신 음성인식 웹 플랫폼

Fig. 2. Web Platform of Speech Recognition Model for Ship Radio Communication

나타낸다. 본 연구에서는 페이스북에서 개발하여 사전 학습된 Wav2Vec 2.0 모델[2]을 사용한다. 이 모델은 적은양의 학습 데이터로도 음성인식모델 개발 장점이 있다.

언어모델은 음성인식에서 발화자의 음성에서 특징 추출하여 음절을 인식하고 글자, 단어 간 확률에 따른 텍스트를 생성을 한다. 언어 모델은 한국어 BERT(Bidirectional Encoder Representations from Transformers) 모델을 적용하여 선박무선통신 용어에 적합한 텍스트를 생성한다.

2.3 후처리 변환

언어모델의 결과 중 전문용어는 인식 과정에서 오류를 발생하기 쉬우므로 성능 고도화를 위해 선박 분야에서 사용되는 전문용어(해사영어, 지명, 선박이름, 부두번호 등) 표현의 특성을 반영한 N-gram 기반 후처리 모델을 개발하여 특정 단어 추론 향상하였다.

3. 모델 성능 및 활용

개발한 모델에 대한 성능을 검증하기 위해 추가로 수집한 30시간의 음성데이터를 라벨링하여 CER(Character Error Rate) 기준으로 모델의 성능을 평가하였다. 모델 사용 결과 한국어 교신 내용에 대한 인식율은 94.5%로 실무에서 활용하기 충분한 성능을 얻었다. 하지만 해사영어 교신에 대한 음성인식은 정확도가 낮아 향후 영어교신에 대한 추가적인 학습데이터를 수집하여 라벨링을 수행하겠다.

본 연구에서 개발한 한국어 음성인식 기술은 웹(oceantraffic.net) 기반으로 선박 관련 종사자들에게 서비스 활용할 계획이다.

Acknowledgement

본 연구논문은 2023년도『지자체-대학 협력기반 지역혁신사업』의 지원을 받아 수행한 "도민안전 디지털 해양안전정보 제공 기술 개발"의 연구 결과입니다.

참 고 문 헌

[1] 김진태, & 정훈. (2017). 음성인식 기술의 동향과 해군 정보통신분야 적용방안: 음성-문자 변환 (Speech To Text) 중심으로. 국방과 기술, (456), 120-127.

[2] Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations." Advances in neural information processing systems 33 (2020): 12449-12460.

[3] Li, F., Jin, Y., Liu, W., Rawat, B. P. S., Cai, P., & Yu, H. (2019). Fine-tuning bidirectional encoder representations from transformers (BERT)-based models on large-scale electronic health record notes: an empirical study. JMIR medical informatics, 7(3), e14830.

SAINT 기반의 소프트웨어 결함 예측

스리만 모하파트라[0], 김지영, 류덕산*
전북대학교 소프트웨어공학과
{srimanmohapatra0, jiyoung_kim, duksan.ryu}@jbnu.ac.kr

Software Defect Prediction Based on SAINT

Sriman Mohapatra[0], Jiyoung Kim, Duksan Ryu*
Department of Software Engineering, Jeonbuk National University

Abstract

Software Defect Prediction (SDP) efficiently allocates resources by identifying defect-prone modules, and recent research increasingly leverages deep learning techniques in this domain. This study investigates the applicability of SAINT, a state-of-the-art deep learning model for tabular data, in SDP. Through comprehensive comparisons with XGBoost and Random Forest, considering key metrics like PD, PF, Balance, and FIR, SAINT consistently outperforms its counterparts, showcasing its effectiveness in enhancing defect prediction accuracy. The findings contribute valuable insights into the role of deep learning techniques in SDP, emphasizing SAINT's potential to advance defect prediction methodologies in practical software development scenarios.

1. Introduction

Software Defect Prediction (SDP) efficiently allocates resources in software development by identifying error-prone modules using machine learning models on tabular datasets. SAINT, with self-attention mechanisms and advanced embedding, outperforms traditional methods in defect prediction. This research introduces SAINT-Transformer to address tabular data challenges, enhancing row classification and enabling exploration of various factors, including multi-task models and fusion with modalities like image and text. Anticipating superior performance, SAINT is positioned as a significant advancement in enhancing SDP.

2 Related Work

In recent years, Deep Learning (DL) has seen significant success, extending its application to Software Defect Prediction (SDP). Pan et al. [2] enhanced defect prediction using advanced Convolutional Neural Networks (CNNs), specifically addressing promise source code (PSC) issues. In a parallel study, Lee et al. [3] introduced a novel SDP model based on TabNet, outperforming traditional models like XGBoost (XGB) and Random Forest (RF). Inspired by these advancements, we explore applying the SAINT technique to further enhance SDP models.

3. Methodology

This study applies SAINT to Software Defect Prediction (SDP) with a meticulous process, addressing overfitting through cross-validation, training on designated data, and evaluating on separate test data. To enhance performance, MIN-MAX normalization ensures uniform feature ranges (0 to 1), mitigating the disparate scale impacts. This approach, combining rigorous cross-validation and feature scaling, enhances SAINT's effectiveness in SDP.

Algorithm 1 outlines SAINT's feature processing code, involving uniform tokenization (lines 1-3), transformation using the Transformer module (lines 4-6), and redefinition of [CLS] and top-of-the-line tokens for subsequent predictions (lines 7-9). The SAINT formulation succinctly describes key processes: tokenization, data transformation, and predictive steps.

Algorithm 1. SAINT

Input: Trained data X.
Output: Data Y predicted for fault.
1: /*Preprocessing X */
2: X is oversampled
3: Min-Max ← X
4: X is Normalization
5: Feature Tokenizer← X
6: X is Tokenized
7: T=X
8: T1= T[CLS]
9: /*[CLS] is top-level token containing the contents of all data */
10: Transformer ← T1
11: T2= Newly defined T1 in Transformer Model
12: /* Predict defects
13: Y= Conduct Defect Prediction using [CLS] */

4. Experimental Setup

4.1 Research Question

RQ: How does SAINT's defect prediction performance compare to other techniques?

This study quantifies SAINT's effectiveness through

*Corresponding Author

comparative analysis, aiming to provide valuable insights for enhancing software reliability.

4.2 Dataset

Table 1: Datasets used for experiment

Dataset	Project	Instances	Buggy (%)	No. of metrics	Granularity
AEEEM	EQ	324	129(39.81%)	61	Class
	JDT	997	206(20.66%)	61	Class
	LC	691	64(9.26%)	61	Class
Relink	Apache	194	98(50.52%)	26	Class
	Zxing	399	118(29.57%)	26	Class
Promise	Ant	769	187(24.32%)	20	Class
	Xerces	454	43(9.47%)	20	Class
AUDI	ProjectA	1909	191(4.45%)	13	Class
	ProjectK	2516	374(15%)	13	Class

Detailed information about the datasets is shown in Table 1.

4.3 Data Pre-processing

In this study, SAINT, XGBoost (XGB), and Random Forest (RF) underwent MIN-MAX scaling, 1:1 ratio learning, and SMOTE for imbalanced data to enhance model robustness. A total of 30 evaluations were performed through 10-fold cross-validation, conducted three times for each model.

4.4 Performance Metrics

In this study, evaluation metrics are derived from the confusion matrix. PD (Probability of Detection) measures the ratio of correctly identified defective instances to the total number of actual defective instances, while PF (Probability of False Detection) quantifies the ratio of non-defective instances misclassified as defective to the total number of non-defective instances. To address class imbalance, the study employs the Balance Metric. FIR (File Inspection Reduction) assesses the effectiveness of reducing code inspection efforts, defined as the ratio of files to be examined to the entire file.

5. Experimental result

Table 2. Comparison of performances

	MODEL		
Metric	SAINT	XGB	RF
PD	O.8126	0.7999	0.7000
PF	0.1650	0.4297	0.3702
BALANCE	0.8078	0.6634	0.6618
FIR	0.4741	0.4599	0.2676

In our study, SAINT consistently outperformed XGBoost and Random Forest (Table 2), demonstrating higher average PD (0.8127), lower PF (0.1650), and superior balance (average: 0.8078). In terms of FIR, SAINT outperformed both XGBoost (0.4599) and Random Forest (0.2676), highlighting its consistent effectiveness in software defect prediction.

6. Threats to Validity

This study's limitations include a potential compositional threat due to the consideration of only four performance metrics (PD, PF, BALANCE, and FIR). Simultaneously, the study acknowledges threats to validity stemming from limited dataset diversity and a narrow comparison scope with XGBoost and Random Forest, which future research should mitigate by exploring diverse datasets and incorporating a broader set of baseline methods.

7. Conclusion

In conclusion, our study introduces and evaluates SAINT for Software Defect Prediction (SDP). SAINT consistently outperforms traditional methods, demonstrating its efficacy in enhancing defect prediction accuracy. The research provides valuable insights into deep learning techniques, emphasizing SAINT's potential to advance SDP methodologies.

Acknowledgements

This research was supported by “Basic Science Research Program” (NRF- 2022R1I1A3069233) through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (MOE), and the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-2020-0-01795) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety(KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission(NSSC) of the Republic of Korea. (No. 2105030)

References

[1] Arik et al. Tabnet: Attentive interpretable tabular learning. arXiv preprint arXiv:1908.07442, 2019.

[2] Pan et al. "An improved CNN model for within-project software defect prediction." Applied Sciences 9.10 (2019): 2138.

[3] Lee et al. 'Software Defect Prediction Based on TabNet.' Proceedings of the Korean Information Science Society Annual Conference (2021): 1255-1257.

[4] Somepalli et al. “Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training.” arXiv:2106.01342v1 [cs. LG] 2 Jun 2021.

[5] Kim et al. “Software Defect Prediction based on Ft-Transformer” In Proceedings of Korea Computer Congress, KIISE, 2022.

[6] Chen et al. Learning s mantic annotations for tabular data. arXiv preprint arXiv:1906.00781, 2019.

[7] Dorogush et al. Catboost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363, 2018.

원자력 안전 소프트웨어 대상 신뢰도 평가 도구*

Lingjun Liu [1,O], 최우영[1], 지은경[1], 류덕산[2]
[1]한국과학기술원 전산학부 [2]전북대학교 소프트웨어공학과
riensha@se.kaist.ac.kr, uy0417@kaist.ac.kr, ekjee@se.kaist.ac.kr, duksan.ryu@jbnu.ac.kr

A Reliability Evaluation Tool for Nuclear Power Plant Safety Software

Lingjun Liu [1,O], Wooyoung Choi[1], Eunkyoung Jee[1], Duksan Ryu[2]
[1]School of Computing, Korea Advanced Institute of Science and Technology (KAIST)
[2]Department of Software Engineering, Jeonbuk National University

Abstract

Since nuclear power plants (NPPs) increasingly rely on digital I&C systems, reliability evaluation for NPP software has become crucial for NPP probabilistic risk assessment. Several reliability estimation methods have been proposed, but there is no available tool support for those methods. To support NPP software manufacturers, we propose a reliability measurement tool for NPP software. We designed our tool to provide reliability estimation depending on available qualitative and quantitative information that users can offer. We applied the proposed tool to an industrial reactor protection system to evaluate the functionality of this tool. This tool can considerably facilitate the reliability assessment of NPP software.

1. Introduction

With the advances in digital technology, the nuclear power industry has begun replacing traditional analog instrumentation and control (I&C) systems with digital ones and employing digital I&C systems in new nuclear power plants (NPPs) [1]. As digital systems are increasingly used in NPPs, it is essential to estimate the reliability of NPP software for NPP probabilistic risk assessment (PRA). Reliability estimation for NPP software focuses on failures on demand (i.e., an accidental scenario where the software should take safety actions). In this study, we utilize the reliability metric "probability of failure on demand (PFD)".

U.S. Nuclear Regulatory Commission (NRC) investigated various quantitative software reliability methods and selected potential candidates, the Bayesian belief network (BBN) method and the statistical black-box testing method, to support reliability modeling of digital systems in NPP PRA [2]. Chu et al. (2018) [3] developed a BBN model that estimates the number of faults considering software development life cycle (SDLC) processes and derives the PFD of NPP digital systems. Chu et al. (2017) [4] proposed a statistical black-box testing approach that validates the reliability goal by testing the software over test cases generated from operational profiles of demand scenarios. Cai et al. [5] presented a hybrid approach, combining BBN and statistical testing methods [3, 4], to fully consider the factors that affect software reliability.

Existing methods are not flexible enough to be applicable in situations where the availability of information is uncertain. In addition, there is a lack of reliability measurement tools for NPP software manufacturers to verify the NPP software reliability. Without a systematic tool, it is challenging for NPP software manufacturers to utilize existing methods since they need to build from scratch.

We propose a systematic reliability evaluation tool for NPP software. Inspired by Cai et al.'s approach, we developed the tool to support the flexibility for reliability estimation, which can adjust to available information, including qualitative and quantitative evaluation results. This tool can considerably ease the reliability assessment of NPP software for NPP software manufacturers and regulatory agencies.

*This research was supported by This research was partly supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (No. 2105030) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2022R1I1A1A01072004).

2. Reliability measurement tool for NPP software

The overall process of the proposed reliability measurement tool is shown in Figure 1. With our tool, target system experts can evaluate qualitative attributes based on ratings and input the number of function points (FPs). Our tool then starts PFD inference based on the BBN model using the WinBUGS tool [6]. The parameters of the BBN model are specific to NPP software and recorded in the U.S.NRC technical report [3]. Users can obtain the number of remaining faults in NPP software and the initial PFD before considering operational conditions.

Given a reliability target, our tool can calculate the number of demands (i.e., test cases) required for statistical testing. After testing, our tool can update the prior PFD with the number of failures and estimate the reliability (i.e., updated PFD) to consider development and V&V qualities during SDLC phases and software operation under demand conditions. Furthermore, our tool can validate if the reliability goal has been accomplished based on testing results.

2.1. Attribute evaluation

For each SDLC phase, development and V&V qualities are required for estimating the number of remaining defects. Development and V&V qualities of each SDLC phase depend on the qualities of performing corresponding development and V&V activities. The activities performed in the development/V&V process are considered as the attributes affecting the development/V&V quality. For instance,

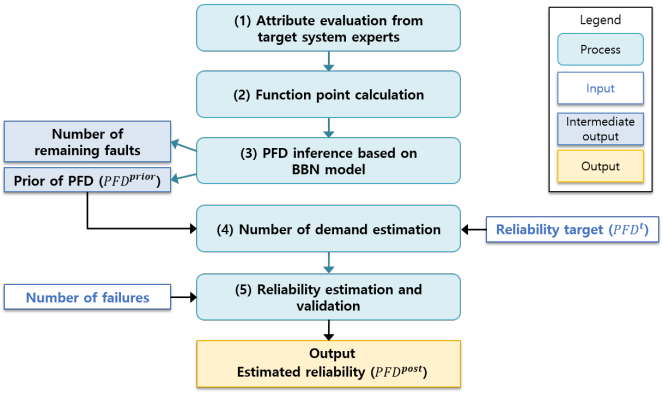


Figure 1. Overall process of reliability measurement tool

V&V quality in the requirement phase includes attributes: software V&V planning, concept documentation evaluation, etc. Three levels, High, Medium, and Low, are used to evaluate the attributes based on the status of performing required tasks related to each attribute.

For each attribute, we set the default value to a Medium level so that our tool can still calculate the reliability even if the information on attribute quality is unavailable. For example, after the implementation phase, users can evaluate all the attributes before the test phase. However, our tool can still deliver an expected reliability (PFD) for users with the default values for the rest of the phases. During software development, our tool can support users to anticipate the reliability results and further monitor reliability progress.

2.2. Function point calculation

The number of FPs was used to calculate the number of introduced faults and as an indicator of software complexity. The number of FPs is assumed to be provided by users.

2.3. PFD inference based on BBN model

With available attribute qualities and the number of FPs, our tool uses the WinBUGS tool to quantify the number of remaining faults and the prior PFD and shows the mean values and simulation traces of the results.

2.4. Number of demand estimation

Given a reliability target, our tool can calculate the required number of demands with a prior distribution of PFD based on Bayes' theorem. The reliability target is considered as achieved if the software is tested over the required test cases and no failures occur.

2.5. Reliability estimation and validation

We assumed that test generation and execution have already been performed by users. With the number of test cases and failures, our tool can estimate the reliability (PFD) based on the prior PFD. The prior PFD can be either the informative distribution inferred from the qualitative information of SDLC or the uninformative uniform distribution. Based on the testing results, software reliability can be validated if no failure occurs among the required tests. Our tool can adapt to the uncertainty in the availability of qualitative information and various reliability test results.

Table 1. The number of remaining faults of SDLC phase

Phase	Mean	Standard deviation	5%	Median	95%
Requirement	7.937	8.935	0.4593	4.9	25.9
Design	26.33	18.57	5.462	21.79	62.57
Implementation	40.8	24.3	11.28	35.82	87.4
Test	28.3	16.07	8.882	25.06	59.16
Installation	11.27	9.107	1.714	8.829	28.97

Table 2. The PFD for the IDiPS-RPS

Mean	Standard deviation	5%	Median	95%
1.142E-3	1.054E-2	9.927E-7	5.789E-5	3.491E-3

3. Case Study and Evaluation

3.1. Case Study

To assess the tool functionality, we chose the same target system in previous BBN-based reliability estimation work [4], i.e., the Integrated Digital Protection System–Reactor Protection System (IDiPS-RPS) developed in the Korea Nuclear Instrumentation and Control System project [7]. With the attribute qualities collected from Chu et al. (2018)'s work, we applied our tool to obtain the reliability results for the IDiPS-RPS. The attributes were evaluated before the installation phase, so all the attribute qualities for the installation phase were set to a default Medium level. The number of FPs was set to 56, the mean of the number of FPs estimated in Chu et al. (2018)'s work. Tables 1 shows the estimated number of remaining faults at the end of each SDLC phase. With the number of remaining faults in the installation phase and generic fault size distribution, the distribution of PFD for the IDiPS-RPS is inferred and presented in Table 2.

3.2. Evaluation

To evaluate the functionality of the BBN model in our tool, we performed hypothesis testing to check if the results obtained from our tool were significantly different from those presented in Chu et al. (2018)'s work. We collected a total of 30 pairs of data points, including mean, standard deviation, 5%, median, and 95% values from PFD and numbers of remaining faults for SDLC phases. We conducted the Wilcoxon matched-pairs signed-rank test, a non-parametric test because the distributions of data points and differences between two data sets did not follow a normal distribution. The p-value is 0.4645, which is higher than 0.05. Thus, we failed to reject the null hypothesis that there is no significant difference between the results from our tool and Chu et al. (2018)'s work. Based on the hypothesis testing results, the developed BBN model in our tool conforms to the functionality of Chu et al. (2018)'s approach.

4. Conclusion

This paper proposed a systematic reliability estimation tool for NPP software. Our tool provides the flexibility of available qualitative information during SDLC and quantitative testing results. We evaluated our tool functionality by applying it to the IDiPS-RPS. The reliability results generated from our tool have no significant difference from Chu et al. (2018)'s results. The parameters of the BBN model can be further extended for safety-critical software in other domains once expert knowledge is available. In future work, we plan to develop a tool, which can generate reliability tests for NPP software.

References

[1] International Atomic Energy Agency (IAEA), "Instrumentation and Control (I&C) Systems in Nuclear Power Plants: A Time of Transition", Nuclear Technology Review, pp. 83-94, 2008.

[2] T. Chu, M. Yue, G. Martinez-Guridi et al., "Development of quantitative software reliability models for digital protection systems of nuclear power plant," (NUREG/CR-7044), 2013.

[3] T. Chu, A. Varuttamaseni, M. Yue et al., "Developing a Bayesian belief network model for quantifying the probability of software failure of a protection system," (NUREG CR-7233), 2018.

[4] T. Chu, A. Varuttamaseni, J. Baek et al., "Development of a statistical testing approach for quantifying safety-related digital system on demand failure probability," (NUREG/CR-7234), 2017.

[5] Y. Cai, Y. Wu, J. Zhou et al., "Quantitative software reliability assessment methodology based on Bayesian belief networks and statistical testing for safety-critical software," Annals of Nuclear Energy, Vol. 145, 107593, 2020.

[6] D. Spiegelhalter, A. Thomas, N. Best, & D. Lunn, "WinBUGS user manual," 2003.

[7] J. H. Park, D. Y. Lee, and C. H. Kim, "Development of KNICS RPS Prototype," Proceedings of ISOFIC (International Symposium on Future I&C) 2005, pp. 160-161, 2005.

모바일 환경에서 실시간 족부 불균형 판별을 위한 딥러닝 앙상블 시스템 설계 및 구현

정혜선 [1),O], 김태구 [1)], 조용훈 [1)], 신기훈 [1)], 신채림 [1)], 이수경 [2)], 백윤주*

[1),*]부산대학교, [2)]동의대학교

[1)][ahttjd55, tbg8577, kchoyh95, skh2929209, cofla0429]@pusan.ac.kr, [2)]ptlsk@deu.ac.kr,
*yunju@pusan.ac.kr

Design and implementation of deep learning ensemble system for real-time foot imbalance determination in mobile environment

Hyesun Jeong[1),O], Taegu Kim[1)], Yonghun cho[1)], Kihun Shin[1)], Chaerim Shin[1)],

Sugyeong Lee[2)], Yunju Baek*

[1),*]Pusan National Univ, [2)]Dong-eui Univ

요 약

족부 불균형이 몸 전체의 균형과 관련된 다양한 질환으로 영향을 미칠 수 있음을 인식하고, 이를 효과적으로 진단하며 모니터링할 수 있는 모바일 앱 기반 딥러닝 앙상블 시스템을 설계하고 구현하는 것을 목표로 한다. 족부 불균형의 유형은 과내전, 외전, 중립의 세 유형으로 분류되며, 이를 정확하게 진단하기 위해 전문가의 지식과 경험이 필요하다. 본 연구에서는 불균형한 데이터에 강건한 멀티입력 앙상블 모델과 경량화를 통한 모바일 환경의 실시간 유형 판별하는 시스템을 제안한다. 먼저 모드별 데이터와 전체 데이터에 머신러닝 기법과 딥러닝 기법을 적용하여 데이터 특성 분석한다. 통합된 Integrated mode 데이터셋에서 MLP 95.7%와 Integrated image 데이터셋에서 CNN 92.9%의 성능을 확인하였으며, 멀티입력을 위한 Integrated mode 데이터셋의 MLP와 CNN모델과 Integrated image 데이터셋의 CNN와 ViT모델을 선정하여 앙상블한 결과 MLP와 ViT가 결합된 모델이 94.5%의 정확도를 가진다. TensorFlow Lite를 사용한 모델의 경량화 및 최적화를 통해 모바일 환경에서의 실시간 추론이 가능하다.

1. 서 론

최근 건강한 생활 방식과 신체 건강에 대한 관심이 증가함에 따라 신체 각 부위의 건강 상태를 정확하게 이해하고 관리하는 것이 중요해지고 있다. 특히 족부의 건강은 삶의 질에 직접적인 영향을 미치며, 장기적인 건강과 웰빙에 중요한 역할을 한다. 족부는 몸을 지탱하는 정적 기능과 이동을 위한 동적 기능을 담당하는 중요한 부위로 족부의 균형이 올바르지 못하면 몸 전체의 중심이 흔들리게 되어 아킬레스건염, 무릎 통증 등 다양한 관절 질환[1]으로 이어질 수 있다.

족부 불균형은 과내전(Overpronation), 중립(Neutral) 외전(Underpronation)[2] 세 가지 유형으로 나눌 수 있으며, 각각 발의 구조와 기능에 영향을 미친다. 중립 상태에서의 좌우 밸런스 불균형은 과내전, 외전과는 달리 진단을 위해 전문적인 지식이 필요하다.

현재 족부 불균형의 진단 기술은 대부분 물리적 검사 및 영상 진단 도구를 필요로 한다. 하지만 이러한 기술은 종합적인 검사 기기의 부재로 복합적인 검사를 거쳐야 한다. 이는 환자에게 육체적, 경제적 부담을 주는 한계가 존재하며, 의료기관에 한정된 진단 방식으로 개인 모니터링이 불가능하다.

본 논문에서는 가정에서 족부 불균형의 유형을 진단 및 모니터링하는 모바일 앱 기반의 딥러닝 앙상블 모델을 제안한다. 부족한 수집된 데이터를 보완하기 위해 로드셀 센서로 부터 얻어진 Raw data와 센서의 수치 데이터 평균을 2D 이미지로 변환한 데이터를 멀티 입력으로 모델을 학습한 후 모바일 앱에 탑재하기 위해 양자화 및 FlatBuffers를 활용하여 모델을 경량화한다. 경량된 모델은 모바일 앱에 탑재하여 실시간 추론과 올바른 유형 판정을 하는지 검증한다.

2. 본 론

2.1 데이터 수집

본 논문에서는 총 14명의 참가자를 대상으로 데이터를 수집했다. 참가자들은 20~30대 남성10명과 20대 여성 4명으로 구성되었다. 각각 다른 자세를 취하는 5가지 모드로(정적 자세, 측면 굽힘, 앞뒤 굽힘, 앉았다 서기, 한발 서기) 구성되어 있고 모드1을 제외한 4개 모드

는 모두 동적 자세로 수행되었다. 각 참가자는 각 모드는 좌우 5초씩 6회 반복 측정한다. 측정 기기에 올라서서 의식적으로 양쪽 족부를 서로 맞추기 위해 움직이는 행동을 제한하고 무의식적 데이터 수집을 위해 측정기기에 올라갈 때는 정면을 바라본 상태에서 진행한다. 의식적 데이터는 본인의 족부 형태를 이상적인 형태로 고치려는 행동들로 신뢰성이 낮은 데이터로 변질될 가능성이 높기 때문에 본 논문에서는 의식적 데이터는 제외한다. 데이터 수집 시나리오 및 모드에 따른 자세는 표 1과 그림 1에서 나타난다.

표 1. 족부 유형 판단 시나리오

Action No.	Action	
	Description	Repeat
Mode 01	정적 자세	바로 선 자세를 유지 6회
Mode 02	측면 굽힘	소리에 맞춰 바로 선 자세에서 배꼽 위치를 중심으로 유지하고 몸통을 좌우 기울이기 6회 (오른쪽3, 왼쪽 3)
Mode 03	앞뒤 굽힘	소리에 맞춰 바로 선 자세에서 배꼽 위치를 중심으로 유지하고 몸통을 앞뒤 젖힘 6회 (앞쪽3, 뒤쪽3)
Mode 04	앉았다 서기	소리에 맞춰 바로 선 자세에서 쪼그려 앉았다 서기 6회
Mode 05	한발 서기	소리에 맞춰 바로 선 자세에서 좌우 한발 서기를 유지 6회 (오른쪽3, 왼쪽3)

그림 1. 모드별 자세

수집된 Raw data는 측정기기와 모바일 앱 간 Bluetooth 통신을 통해 전송되며, 연결된 모바일 기기에 저장된다. 이후 http 통신을 통해 서버에 Raw data를 저장한다. 저장된 데이터는 족부 압력 분포와 관련된 데이터 포인트들로 구성되어 있다. 본 연구에서 사용된 측정기기는 그림 2와 같다. 측정기기는 최대 180kg의 하중을 견딜 수 있는 로드셀(Load cell) 센서 8개를 사용한다.

그림 2. 측정기기 및 데이터 수집

2.2 데이터 전처리

데이터 전처리를 위해 정규 표현식을 사용한 데이터 정제 및 LableEncoder를 활용한다. 초기 수집된 Raw data는 통신 확인을 위한 특수 문자와 특정 패턴을 포함하고 있다. 이러한 문자와 패턴들은 데이터의 분석 및 처리를 복잡하게 만들 수 있으므로 정규 표현식을 활용하여 제거하였고, 범주형 데이터는 LableEncoder를 통해 수치형으로 변환한다. 이는 딥러닝 모델의 학습을 위한 형태로 데이터를 변환하는 과정으로, 특히 데이터 표현 간소화로 인해 데이터 크기와 계산의 효율성을 높여주기 때문에 범주형 데이터를 효과적으로 처리하는게 중요하다.

수집된 Raw data의 5초 동안 얻은 좌우 압력 포인트 값을 1초씩 샘플링하여 센서 수치 데이터를 2D 이미지로 변환한다. 변환된 2D 이미지는 1개의 이미지 데이터로 통합한다. 이 과정은 전문의가 데이터 라벨링하는 과정에서 보다 특징을 직관적이고 빠른 판단을 가능하게 한다. 그림 3과 같이 2D 이미지 시각화는 전문의들이 복잡한 수치 데이터 해석 대신 임상 경험을 활용하여 미세한 특징 패턴을 식별하고, 전문적인 진단하기 유용하다.

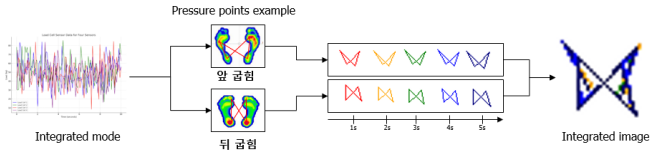


그림 3. 이미지 데이터로 변환

2.3 모델 설계

2.3.1 데이터 통합 접근법

로드셀 센서로부터 얻은 Raw data와 센서 데이터 기반 변환된 2D 이미지 데이터의 멀티입력으로 모델을 학습한다. 멀티입력은 모델에 다차원적인 정보를 제공하며 데이터 불균형 문제에 대해 모델의 강건성을 향상에 이점을 가진다. 모델 설계에 있어 모드별 데이터 특성 분석은 중요한 단계로 각 유형이 특정 정보와 패턴을 지닐 가능성을 탐색합니다. 이러한 과정은 앙상블 모델 설계의 기반을 형성하기 위해 진행된다. 본 논문에서는 모드별 독립적인 실험을 수행한 후, 그 결과를 통합함으로써 데이터의 다양성을 확보하고 불균형한 데이터 분포와 부족에 모델의 강건성을 개선하는 방법을 검증한다. 이를 위해 전통적인 머신러닝 기법인 Decision Tree, Random Forest, Support Vector Machine(SVM) 세 가지 기법과 딥러닝 기법 Multi-Layer Perceptron(MLP), CNN, Long Short-Term Memory(LSTM) 세 가지 기법으로 평가한다. 각 모드가 제공하는 정보의 한계성과 데이터의 다양성을 고려하여 모드별 Raw data와 2D 이미지 데이터를 각각 통합하여 Integrated mode 데이터셋과 Integrated image 데이터셋을 구성하여 Integrated mode 데이터셋은 MLP, CNN, LSTM, Transformer기법을 Integrated image 데이터셋은 CNN, Vision Transformer(ViT)기법을 적용한다. 이는 각 기법이 데이터 특성을 포착하는 방식이 다르며, 모델의 성능을 극대화하는 조합의 앙상블 모델을 설계하기 위함이다.

2.3.2 멀티입력 데이터를 활용한 앙상블 모델 설계

본 논문에서 구현한 모바일 환경 기반 딥러닝 앙상블 시스템의 전체 구성은 그림 4와 같다. 모델 학습의 입력은 수치 데이터와 이미지 데이터 스트림의 두 가지 분기로 나뉘며, 각 스트림은 주요 특징을 추출하는 데 효율적인 로컬 모델에 의해 처리된다. 각 로컬 모델은 로컬 및 글로벌 특징을 추출하여 서로 결합한다. 특징 벡터들은 직접 결합하여 더 큰 차원의 벡터를 형성하고 Dense layer에 전달한다. dense layer는 각 로컬 출력 차원 간의 상호 관계를 학습하여 더 깊은 추론을 수행한다.

학습된 모델은 TensorFlow Lite로 변환하는 과정을 통해 모델의 파라미터를 감소시켜 경량화 한다. FlatBuffers 파일 포맷의 사용을 통해 메모리에 직접 액세스가 가능하여 파싱이나 언패킹 과정이 불필요하게 되어 모바일 환경에서 빠른 모델 로딩과 효율적인 메모리 사용을 가능하다. 그리고 학습에 사용되었지만 추론에는 불필요한 메타데이터 제거를 통해 모델의 크기를 줄이고, 실행 속도를 향상시킨다. 모바일 환경에서 실시간 추론을 위해 양자화(Quantization)을 적용하여 모바일 앱에 탑재한다.

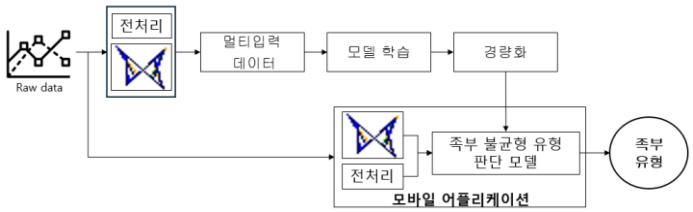


그림 4. 모바일 환경의 족부 유형 판정 flow 다이어그램

3. 실험 및 성능평가

3.1 모드별 성능평가

본 논문에서는 족부 불균형 유형 판별 성능 비교를 위해 다양한 머신러닝 및 딥러닝 기법과 앙상블 시험 및 비교하였다. 데이터는 8:2로 나누어 실험한다. 먼저 모드별 분류 성능을 평가하기 위해 머신러닝 기법인 Decision Tree, Random Forest, Support Vector Machine(SVM) 세 가지 기법을 적용하였다. 그 결과는 표 2와 같으며, 데이터 부족으로 인한 과적합과 미세한 특징을 분류하지 못하는 것을 확인할 수 있다.

표 2. 모드별 머신러닝 기법을 활용한 성능 비교

모드	정확도 (%)		
	Decision Tree	Random Forest	SVM
Mode 1	90.0	90.0	90.0
Mode 2	84.5	100	100
Mode 3	84.5	100	100
Mode 4	100	100	100
Mode 5	100	100	100

그리고 MLP, CNN, LSTM 세 가지 종류의 딥러닝 기법을 사용하여 비교 분석을 수행하였다. MLP와 CNN은 평균 정확도 19.32%이고 LSTM은 평균 정확도 84.1%이다. 해당 기법들의 결과는 표 3과 같으며, 입력 Raw data의 시계열 데이터 처리에 특화된 LSTM 구조 상 타 모델 대비 좋은 성능을 달성한 것으로 보인다. 또한 모드 1의 정적 자세는 동적 자세에 비해 상대적으로 특징이 적어 더 쉽게 학습되고 인식되어 다른 모드에 비해 2배 이상의 높은 정확도를 보인다.

표 3. 모드별 딥러닝 기법을 활용한 성능 비교

모드	정확도 (%)		
	MLP	CNN	LSTM
Mode 1	35.0	35.0	66.7
Mode 2	15.4	15.4	100
Mode 3	15.4	15.4	84.6
Mode 4	15.4	15.4	84.6
Mode 5	15.4	15.4	84.6
평 균	19.32	19.32	84.1

3.2 통합된 데이터 성능평가

모드별 실험 결과 각 모드의 특성이 두드러지지 않았음을 보여준다. 이에 따라, 각 모드별 데이터를 통합하여 데이터 양을 증가시켜 성능 비교를 하였으며, 데이터의 불균형으로 편향되기 쉬운 머신러닝 기법은 사용하지 않는다. Integrated mode 데이터셋의 딥러닝 기법의 성능 평가를 위해 네 가지 딥러닝 기법 MLP, CNN, LSTM, Transformer[3]의 성능 비교와 Integrated image 데이터셋은 CNN과 Vision Transformer(ViT)[4] 기법으로 성능을 비교 분석하였다. 실험 결과는 표 4와 같다. 통합 데이터셋에서 MLP가 95.7%로 가장 좋은 성능을 보이며 타 모델 대비 다양한 유형의 특징을 포착한 것으로 보인다. CNN의 경우 92.9%로 MLP보다 성능은 낮지만 수치데이터와 이미지 데이터 입력에 일관된 성능을 유지할 수 있음을 시사한다. LSTM은 모드별 성능 평가에서는 가장 좋은 성능을 보였으나, 통합 데이터셋에서는 타 모델 대비 80%로 비교적 낮게 나타남을 볼 수 있다. LSTM이 주로 시간적 연속성이 중요한 데이터에 더 적합하며 현재 데이터에는 이러한 특성이 잘 활용되지 않았음을 확인할 수 있다. Transformer의 경우 77.9%로 가장 낮은 성능을 보여주었지만, Transformer 특성상 많은 데이터양이 필요하며 데이터양에 따라 성능이 크게 달라진다. 본 논문의 데이터로는 Transformer에서 높은 성능을 확인하기 어렵다는 것을 확인할 수 있다.

3.3 딥러닝 앙상블 성능평가

Integrated mode 데이터셋에서 좋은 성능을 보여준 MLP와 CNN과 Integrated image 데이터셋에서 CNN과 ViT의 앙상블을 통해 성능 비교를 수행하였다. 그 결과

는 표 5와 같으며, MLP와ViT이 결합된 앙상블 모델이 94.5%로 가장 좋은 성능을 보인다. MLP는 수치 데이터를 처리하는 동안 ViT는 이미지 데이터에 대해 글로벌하고 복잡한 패턴을 학습하여 각 데이터 특징에 맞게 처리 능력이 잘 결합되었음을 확인할 수 있다. CNN과 CNN 앙상블결합은 MLP와 ViT의 결합 보다 0.8% 성능이 낮지만, CNN이 다양한 데이터 유형에 대해 유연하게 적용될 수 있음을 확인할 수 있다. CNN과 ViT 앙상블 결합의 경우 84.8%로 가장 낮은 정확도를 보인다. 이는 두 모델이 모두 이미지 처리에 초점을 맞추고 있기 때문에 수치 데이터 처리에 있어서 최적의 조합이 아님을 확인할 수 있다.

표 4. 통합 데이터셋 딥러닝 기법을 활용한 성능 비교

통합 데이터셋	정확도 (%)			
	MLP	CNN	LSTM	Transformer/ ViT
Integrated mode	95.7	92.9	80.0	77.9
Integrated image	-	92.9	-	81.1

표 5. 딥러닝 앙상블 성능 비교

수치데이터+ 이미지데이터	정확도 (%)
MLP+CNN	91.6
CNN+CNN	93.7
MLP+ViT	94.5
CNN+ViT	84.8

3.4 모바일 환경 추론 실험 결과

성능 실험을 통해 선정한 MLP와 ViT 앙상블 모델을 선정하여 TensorFlow Lite로 변환하여 모바일 앱에 탑재한다. 변환 과정을 통해 모델의 파라미터 감소 및 연산 속도의 향상을 이루었으며, 모바일 앱은 측정된 족부 데이터 유형을 정확하게 분류하여 추론할 수 있음을 보였다. 앱의 사용자 인터페이스(UI)는 다양한 측정 모드를 지원하며, 사용자는 5개 모드 중 어떤 모드를 선택하더라도 분류된 족부 유형 결과를 실시간으로 확인할 수 있다. 이는 그림 4와 그림 5를 통해 확인할 수 있다.

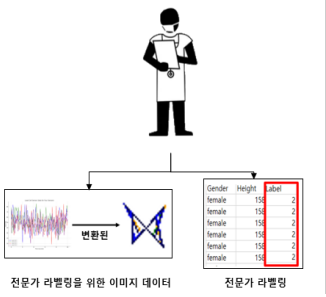


그림 5. 전문가 데이터 라벨링

그림 6. 유형 판정 확인

4. 결 론

본 논문은 모바일 앱 기반의 MLP 와 ViT 결합된 앙상블 모델을 통해 족부 불균형 진단을 위한 새로운 방식을 제안한다. TensorFlow Lite 로 모델을 변환하여 모바일 장치에 적합하게 최적화하였으며, 이를 통해 빠르고 정확한 진단 정보를 환자에게 제공한다. 모델은 다양한 데이터 형태에 대해 강건한 성능을 보였으며, 환자가 쉽게 사용할 수 있는 앱을 통해 자신의 족부 유형을 쉽게 파악하고 관리할 수 있다. 향후 연구로는 추가적인 데이터 수집을 통해 모델을 더욱 세밀하게 조정하고 개인별 맞춤형 진단을 위한 다양한 딥러닝 기법과 데이터 증강 등 확대하여 적용할 계획이다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터사업의 연구결과로 수행되었음" (IITP-2023-RS-2023-00260098)

이 연구는 2024 년도 산업통상자원부 및 한국산업기술기획평가원(KEIT) 연구비 지원에 의한 연구임(20015052)

이 논문은 2024 년도 부산광역시의 재원으로 (재)부산테크노파크의 지원을 받아 수행된 연구 결과입니다 (사업명: 시장수요맞춤개방형연구실사업)

참 고 문 헌

[1] Neal, Bradley S, "Foot posture as a risk factor for lower limb overuse injury: a systematic review and meta-analysis." Journal of foot and ankle research. pp. 1-13, July. 2014.

[2] Cote, Karen P, "Effects of pronated and supinated foot postures on static and dynamic postural stability." Journal of athletic training. pp. 41-46, Mar. 2005.

[3] Vaswani, Ashish, "Attention is all you need." Advances in neural information processing systemsm, 2017.

[4] Dosovitskiy, Alexey, "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929, Jun. 2020.

AIS데이터를 이용한 대형선망어선의 조업 해역 분석

송은아 ○, 정은주, 김광일*
제주대학교 해양과학대학
C0209@jejunu.ac.kr, karkdi01@naver.com, kki@jejunu.ac.kr

Analysis of Fishing Area of Large Purse Seine Vessels Using AIS Data

Euna Song○, Eunju Jung, Kwangil Kim*
Jeju National Univeersity, College of Ocean Science

요 약

기존 어선의 조업해역을 파악하기 위해 어선 선장의 위치 보고에만 의존하던 한계를 극복하기 위하여 본 논문에서는 선망어선 선박의 실제 위치 데이터인 AIS 데이터를 수집하고, Python의 Folium 라이브러리와 Matplot 라이브러리를 이용하여 대형 선망 어선의 조업 어장을 분석한다. 향후 연구에서는 이 논문의 분석 결과와 대형 수협 위판 자료를 결합하여 대형 선망 어선의 실제 조업어장에서의 어획량을 추정하고자 한다.

1. 서 론

해양오염, 해수 온도의 상승 등 해양 환경의 변화, 무분별한 남획 등의 사유로 인하여 수산 자원은 계속하여 감소하고 있다. 지속적인 수산자원 관리를 위하여 자원량을 파악하고 고갈되지 않게 노력하는 것은 중요하다. 따라서, 우리나라에서는 총 허용 어획량 할당 제도를 통해 어종별로 연간 잡을 수 있는 어획량을 한정하여 각 어선에 할당하고 있다. 기존에는 어선 선장이 수협 어선안전조업국에 무선으로 하는 위치 보고를 통해서만 각 어선들의 조업해역을 알 수 있었다. 하지만, 위치 보고는 출항 시각을 기준으로 하여 24시간 중 1회만 진행되고 있으며, 선장이 위치 보고를 잘못할 수 있다는 한계를 갖고 있다. 풍랑특보가 발효될 경우에도 12시간 중 1회만 위치 보고를 진행하고 있어 어선의 실시간 위치 및 상황을 파악하기 힘든 실정이다. 따라서, 본 연구에서는 선박 위치 발신장치(AIS) 데이터를 이용하여 대형선망 선단의 항적 분석을 통해 대형선망 어선이 조업중인지 아닌지를 판별하고 어선의 실제 조업 구역을 판별하 파악하고자 한다.

2. 데이터 라벨링

2.1 AIS 데이터 수집

선박위치발신장치(Automatic Identification System, AIS)는 선박의 항해정보를 실시간으로 송·수신하는 장치이다. AIS 데이터에는 선명, mmsi 번호, 선박의 위치(위·경도), 속력, 침로, 선박의 재원 정보 등이 포함되어있다. AIS 데이터는 선박의 속력이 0~14knot일 때 12초간격, 14~23knot일 경우 6초간격, 23knot 이상일 경우 3초 간격의 발신주기를 갖는다. AIS 데이터는 제주대학교와 한라산 근처에 설치되어 있는 AIS 수신기를 통해 대형선망 어선의 항적 데이터를 수집하였다. 2019년도 7월 1일부터 2022년도 6월 30일까지의 데이터를 수집하였으며, 데이터 수집 대상 해역은 북위 28°00′~ 38°00′, 동경 120°00′~ 135°00′ 까지이다.

2.2 대형 선망 선단 정보 수집

대형선망의 한 선단은 본선 한 척, 등선 두 척, 운반선 세 척으로 구성된다. 조업 시 본선 한 척과 등선 두 척이 짝을 이루어 진행하며, 운반선 세 척이 번갈아 가면서 항구와 본선을 오가며 위판을 진행한다. 부산 대형선망수협에 등록되어 있는 어선 명부를 참고하여 같은 선단에 소속된 선박의 정보를 수집하였다. 어선 명부에는 선단명, 동일 선단에 소속된 선명, 선박 번호, 어업허가번호 등이 포함되어있다.

2.3 데이터 라벨링

대형 선망 어선의 조업시간은 어군을 발견하여 투망을 진행한 후 양망을 완료하기까지 1시간 30분에서 2시간 정도이다. 투망의 경우 8~10knot의 속력으로 원형을 그리는 패턴을 보이며, 5분에서 10분 정도의 짧은 시간동안 진행된다. 양망 진행시에는 어군이 도망가지 못하도록 투망 완료한 그물의 아래를 조이고 그물을 천천히 거두어들여야 하므로 1~3knot의 낮은 속력을 나타낸다. 이러한 항적 패턴과 속력을 이용하여 대형선망선단의 본선에 대한 라벨링을 [그림1]과 같이 진행하였다.

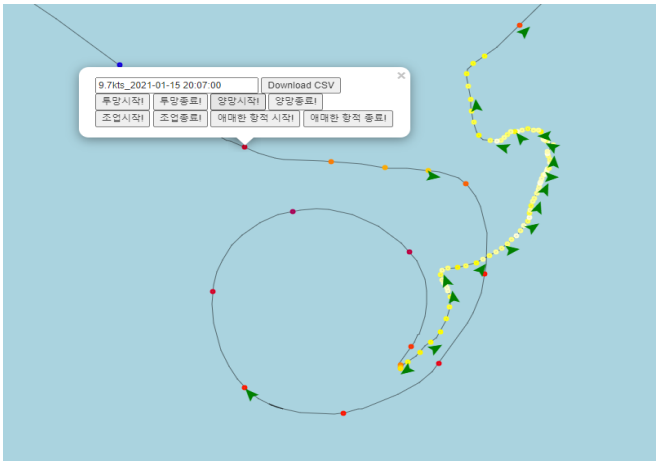


그림1 대형 선망 어선의 라벨링 과정

라벨링을 통해 선망 어선의 투망 시작, 투망 종료, 양망 시작, 양망 종료 시점을 분류하였다.

3. 결과

Python의 Folium 라이브러리를 사용하여 AIS를 통해 수신된 대형 선망 선단의 선박 위치를 선으로 연결하여 [그림 2]와 같이 나타내었다. 빨간색은 본선, 주황색과 노란색은 등선, 파란색은 운반선의 항적을 나타낸다.

그림 2 대형 선망 선단 항적 추출 결과

등선(주황색)이 어군을 모으면 본선(빨간색)과 등선(노란색)이 근처로 와서 투망을 진행한다. 이후 양망을 진행하면 운반선(파란색)이 본선에 접근하여 어획물을 끌어올리는 것을 확인할 수 있다.

[그림 3]은 선망 선단의 본선과 등선만을 나타냈다. 라벨링 결과를 토대로 선박이 조업을 진행한 것으로 판단되는 항적 구간위에 색이 있는 점을 나타냈다. 동일한 색상의 점은 각 선박의 동일한 시간대의 움직임을 나타낸다.

등선(주황색)이 어군을 모으면 본선(빨간색)과 또 다른 등선(노란색)이 어군을 탐지하다가 등선(주황색) 근처로 접근하여 투망을 진행하는 것을 확인할 수 있다.

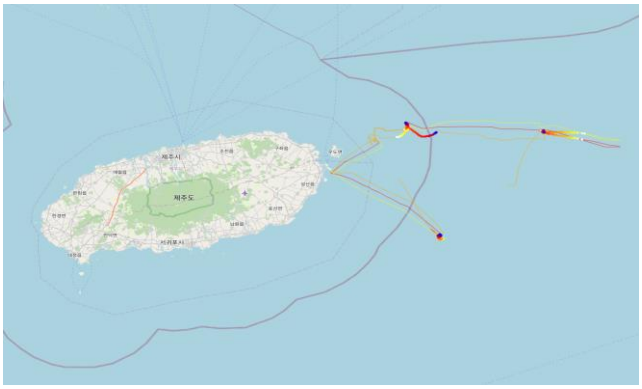


그림 3 대형 선망 선단 항적 추출 결과

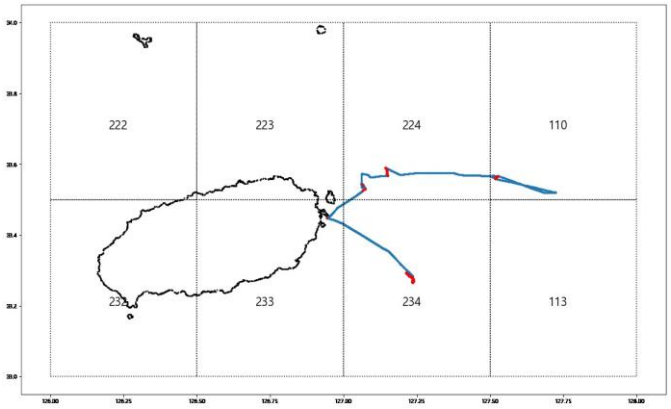


그림 4 대형 선망 선단 조업 해역

[그림 3]에서 조업을 진행한 해역을 파악하기 위하여 본선의 항적을 Python의 Matplot 라이브러리를 이용하여 [그림 4]와 같이 나타냈다. 해구의 위·경도 정보는 국립수산과학원 위성해양정보시스템에서 제공하는 해구코드 좌표를 이용하였다. 이때 조업 중인 본선의 속력이 3노트 미만일 경우 선박이 양망을 진행 중인 것으로 판단하여 빨간색으로 표시하였다. 이를 통해 위 선박은 224, 234, 110해구에서 조업을 진행한 것을 확인할 수 있다.

4. 결 론

본 논문에서는 대형선망 선단의 실제 조업 구역을 분석하기 위하여 자동 선박 위치발신장치(AIS) 데이터를 수집하였다. 수집한 항적 데이터를 Python 프로그램의 Folium 라이브러리를 이용하여 본선 한 척, 등선 두 척, 운반선 세 척으로 이루어진 한 선단의 항적을 각 선박별로 색상으로 구분하여 항적을 나타내었으며, 선망어선의 조업 속력 및 항적 패턴을 파악하였다.

기존 선장의 위치 보고는 24시간 중에 1회만 진행되어 위치 보고 당시의 선박의 위치와 조업 여부만 파악할 수 있었으나, AIS 데이터는 선박의 속력에 따라 최소 3초에서 최대 12초 간격으로 선박의 속력, 위치 등의 정보가 전송되므로 매 시각 어선의 조업 위치뿐만 아니라 어선이 이동한 항적까지 보다 명확하게 파악할 수 있었다.

선망어선의 조업 속력 및 항적 패턴을 이용하여 본선 항적의 투망 시작, 투망 종료, 양망 시작, 양망 종료를 분류하는 데이터 라벨링을 진행하였으며, Python 프로그램의 Matplot 라이브러리를 이용하여 선박이 조업을 진행한 해구를 분석하였다.

향후 연구에서는 대형선망어선이 수협에 판매를 진행한 위판량 자료와 본 연구의 결과로 나타난 대형선망어선의 실제 조업 해역을 결합하여 각 해역별 어획량을 분석하고 이를 토대로 각 해역별 자원량을 추정하고자 한다.

사 사

본 논문은 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다. (2023RIS-009)

참고문헌

[1] 김형석, and 이햇님. "대형선망어업에 있어서 고등어 어장의 어황변동." 수산해양기술연구 (구 한국어업기술학회지) 47.2 108-117. 2011.

[2] Owiredu, Solomon Amoah, and Kwang-Il Kim. "Spatio-Temporal Fish Catch Assessments Using Fishing Vessel Trajectories and Coastal Fish Landing Data from around Jeju Island." Sustainability 13.24 13841. 2021.

[3] Zhang CI. Evaluation and Management of Fisheries Resources. 56-57. 2020.

[4] 김광일, et al. "어선 항적데이터를 활용한 어업손실보상을 위한 조업일수 산출 방법." 수산해양기술연구 57.4 (2021): 334-341.

[5] Kim Byung-Ok, “Radiocommunication Networks for Vessel Monitoring System”. 2006.

ChatGPT 활용에 대한 고찰: 다양한 도메인에 적용되는 프롬프트 엔지니어링 전략 분석

Ivan Stanislavov Ivanov ○ 송지영
한남대학교
20234218@gm.hannam.ac.kr, jysong@hnu.kr

Unlocking ChatGPT's Potential: A Literature Review of Prompt Engineering Strategies Across Diverse Domains

Ivan Stanislavov Ivanov ○ Jiyoung Song
Hannam University

요 약

Large Language Models (LLMs), exemplified by OpenAI's ChatGPT, represent a significant advancement in artificial intelligence, demonstrating human-like text generation and contextual understanding. Contrary to misconceptions, ChatGPT enhances creativity and efficiency instead of reducing human involvement. The model's efficacy relies on well-crafted prompts, emphasizing the importance of clear and context-rich communication. The literature review of seven papers underscores the crucial role of tailored prompts in optimizing ChatGPT's performance across diverse domains. This analysis reveals both commonalities and distinctions in prompt engineering strategies, with a consistent theme of specificity and structure. The adaptability of ChatGPT across various fields is evident, showcasing its potential in scientific research, entrepreneurship, and education.

1. Introduction

The advent of Large Language Models (LLMs) marks a significant leap in the field of artificial intelligence and natural language processing [1]. LLMs, such as OpenAI's GPT series, are sophisticated algorithms trained on vast amounts of text data and they have the remarkable ability to generate human-like text, comprehend complex queries, and provide relevant responses [2]. ChatGPT, in particular, can offer more accurate, context-aware, and interactive responses, making it an invaluable tool in diverse applications ranging from customer service to content creation and educational support [3]. Contrary to belief, while ChatGPT can allow people to do less, and people that want to do more, be more creative, save time, will be able to do so [4].

The efficacy of ChatGPT, while impressive, is significantly influenced by the input it receives – this is where the concept of 'prompting' becomes crucial [5]. Prompting in the context of LLMs refers to the way users communicate their requests or queries to the model. The effectiveness of ChatGPT, therefore, hinges on how well these prompts are engineered. Effective prompting involves creating queries that are clear, concise, and contextually rich, guiding the AI to understand the intended task and respond accurately.

In the following literature review, we will review and compare 7 papers related to ChatGPT prompting and how to improve the LLM's responses.

2. Analysis of Prompt Engineering Strategies

This literature review delves into the realm of optimized prompting strategies to enhance the utilization of ChatGPT. By examining various studies, this review explores how different approaches to prompt engineering can significantly impact the performance of ChatGPT across various domains. From scientific data extraction to educational applications, each paper in this review contributes insights into the art and science of prompting, showcasing the potential of well created prompts in unlocking the full capabilities of this advanced LLM. Through this exploration, we aim to provide a comprehensive understanding of how strategic prompting can transform the way we interact with and benefit from conversational AI models like ChatGPT.

Table 1 Prompting papers' summary

Author	*Field*	*Prompting strategy*	*Additional remarks*
Polak and Morgan [1]	Scientific Data Extraction	Zero-shot	AI prompted to recognize and interpret complex scientific data
Short and Short [5]	Business communication	Elevator, Crowdfunding, Twitter pitch prompts	ChatGPT crafting diverse business narratives with specific prompts
Poola [6]	General AI	Prompt – Evaluate result – Give feedback – Prompt again	Focus on improving ChatGPT's accuracy and conveying uncertainty
Lo [7]	Education	Concise, Logical, Explicit, Adaptive, Reflective prompts	Promoting information literacy through nuanced prompt engineering
Mesko [8]	Healthcare	No specific strategy, but recommends zero-shot, few-shot, roleplaying prompting	Refining healthcare interactions, decision support, and patient engagement
Shi et al. [10]	Regulation	3 step approach: 1) Summarize text 2) Ask to add specific facts 3) Summarize text again	Refining summarization abilities for FDA guidelines
Oh [11]	Education	Prompts including Role-playing, Rules, Example, Problem, Process	Integrating specific strategy to solve mathematical problems

Table 1 summarizes the papers and related prompt strategies discussed in this section. In the realm of advancing conversational AI, Polak and Morgan [1] present a compelling case for the extraction of accurate materials data from scientific literature using ChatGPT. The objective was to leverage the model for precise data mining tasks, which often requires an understanding of complex scientific terminologies and concepts. The researchers employed a zero-shot learning approach where the AI was prompted to recognize and interpret data without prior specific training on the task, and the results were promising – showing high precision (around 91%) and recall (around 85%) in identifying material properties, a testament to the model's potential for scientific data extraction. This study's contribution is particularly noteworthy as it underscores the capacity of well-engineered prompts to enable AI to navigate and extract information from highly specialized and technical texts.

Transitioning from scientific research to the business world, Short and Short [5] explore ChatGPT's utility in creating entrepreneurial rhetoric. The investigation centered on the model's ability to emulate the communicative styles of various CEO archetypes, thereby generating diverse and compelling business narratives. While the findings revealed ChatGPT's adeptness at crafting pitches that resonated with the qualities of different leadership personas when provided with strategically designed prompts, due to the risk of fake announcements, the authors warn that in the future we need to be careful when it comes to business communication. Overall, this research broadens the horizon of conversational AI's applicability in the business sector, demonstrating that with well-structured prompts it has the potential for creating nuanced and persuasive content that aligns with the user's intent and the audience's expectations

On the back of distinguishing between real and fake, Poola [6] focuses on addressing the inaccuracies often encountered in AI outputs, particularly in ChatGPT. The study delves into the model's problem-solving capabilities and its expression of uncertainty—a critical aspect of trust and reliability in AI systems. The researchers believe that ChatGPT cannot indicate of any uncertainties in its answers and that its answers are not consistent. This can lead basic question and answer

users to end up with inaccurate responses. Thus, by experimenting with various prompting strategies, the researchers aimed to enhance the model's accuracy and its ability to convey uncertainty appropriately. The results suggest users to prompt ChatGPT, evaluate the result and provide feedback and prompt again. This can lead to more accurate and trustworthy interactions. This contribution is pivotal as it provides a pathway to improve conversational AI's performance in tasks that require not only precision but also the nuanced expression of confidence levels in the provided information.

An example of such a more nuanced approach is the CLEAR framework, designed to promote information literacy through prompt engineering [7]. The aim was to devise a method that would enable ChatGPT to support educational endeavors, specifically aiding librarians and students in their research. The CLEAR framework—standing for Concise, Logical, Explicit, Adaptive, and Reflective prompts—was shown to be effective in guiding students through the research process, enhancing their critical thinking and analytical skills. Not only do they provide a prompting framework, but the researcher suggests users should be aware of technical terms like tokens, model temperature and top-p. Knowing how to adjust these parameters might require a little more technical knowledge from the user and can introduce even more variance in ChatGPT's responses if an unaware of their nuance's person adjusts them.

Moving into the healthcare domain, Mesko [8] discusses the emerging role of prompt engineering for medical professionals. The scope of this research extends to various healthcare-related interactions, including decision support and patient engagement. By providing concrete steps for healthcare professionals to construct effective prompts, the study aims to refine the interaction with AI, thus improving the quality of healthcare services. The outcomes underscore the potential of prompt engineering to significantly enhance the delivery of healthcare by augmenting the decision-making process and facilitating patient communication. The study does not provide any specific prompting solution that can be useful in the medical scene, but rather suggests various approaches that were also mentioned in Schmidt et al. [9] - use one-shot/few-shot, ask ChatGPT to play roles, ask open ended questions and request examples.

Shi et al. [10] present a distinct application of ChatGPT in the regulatory sphere, specifically in summarizing food effect studies pertinent to FDA guidelines. The researchers proposed an iterative prompting strategy (1st stage: summarize a text, 2nd stage: ask to add specific facts to it, 3rd stage: again, summarize the text into 2-3 sentences) to refine the model's summarization abilities, thereby aiding in the drafting of clear and concise food regulatory documents. After being tested on 100 drugs, the assessors concluded that the text after stage 3's prompt is the most concise, but noted that even for 42 of the cases, even just after stage 1, the text was satisfactory. This emphasizes the idea of writing a simple prompt, even without knowing any technical aspects of ChatGPT, which counters the ideas of some of the above studies which suggest looking into technical details as well, only to write a good prompt.

Finally, Oh [11] examines the use of ChatGPT in educational settings, focusing on the resolution of mathematical problems. The study developed a structured prompting approach that significantly enhanced the model's problem-solving accuracy. This approach, integrating role-play, rules, example-solving, and process articulation within the prompts, was shown to markedly improve ChatGPT's performance in mathematical tasks.

The implications of this research are far-reaching for, not only, the field of education, suggesting that conversational AI, when combined with carefully designed prompts, can serve as an effective supplementary tool for learning and instruction. This study's result may indicate that specific areas may need their own versions of efficient prompts, because what might work for mathematics, might not work for medicine or food regulation.

3. Application in Software Engineering

Applying insights from existing studies can enhance ChatGPT's usage in software engineering tasks. Drawing inspiration from Polak and Morgan's precision approach, a zero-shot learning strategy could help ChatGPT understand and interpret programming-related data without specific task training. Short and Short's entrepreneurial rhetoric exploration may be adapted for crafting code snippets tailored to different programming needs. The CLEAR framework, designed for education, could guide developers in creating prompts that can result in better code snippets. Insights from prompt engineering for medical professionals may provide invaluable insights into medical LLM and how to

improve their training and testing. Shi et al.'s iterative prompting strategy could refine code documentation or generate concise technical summaries. Finally, Oh's structured prompting approach for mathematical problem-solving may be extended to enhance ChatGPT's assistance in addressing coding challenges (as in LeetCode). In summary, these strategies offer practical ways to optimize ChatGPT for software engineering tasks, focusing on precision, clarity, collaboration, and problem-solving.

4. Conclusion

In conclusion, a comprehensive analysis of the identified studies reveals both commonalities and distinctions in the application of prompt engineering to enhance the performance of ChatGPT. The emphasis on tailored prompts emerges as a shared theme across all papers, underscoring the critical role of specificity and structure in input prompts for optimizing ChatGPT's effectiveness. Furthermore, the studies showcase the versatility of ChatGPT in diverse domains, ranging from scientific research and entrepreneurship to education. Despite this commonality, the application contexts and objectives vary, emphasizing the adaptability of prompt engineering across different fields. Finally, it becomes evident that prompt engineering consistently enhances ChatGPT's performance across all papers. However, the degree of effectiveness varies, with different studies reporting varying levels of impact in specific domains. The complexity of prompt design also differs, with some studies presenting structured frameworks (i.e. CLEAR) while others adopt a more flexible and creative approach (as in [11]).

5. Future research

Future research in optimizing ChatGPT and LLM prompting for general and software engineering tasks should focus on tailored strategies and addressing challenges in handling ambiguous prompts. Dynamic prompt adjustment mechanisms based on user interactions and ethical considerations for preventing misuse are crucial aspects to explore. Additionally, investigating multimodal interactions, adaptability across software domains, and developing seamless user interfaces will contribute to enhancing ChatGPT's effectiveness and responsible use in the evolving landscape of AI and artificial general intelligence.

Reference

[1] Polak, M. P., & Morgan, D. Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering--Example of ChatGPT. arXiv preprint arXiv:2303.05352. (2023)
[2] OpenAI. Introducing ChatGPT. (2022)
[3] Mollick, E. A turning point in AI, ChatGPT is coming, Harvard Business Review. 2023.
[4] Altman, S. (2023). Sam Altman & OpenAI speech| 2023 Hawking Fellow, 2023
[5] Short, C. E., & Short, J. C. The artificially intelligent entrepreneur: ChatGPT, prompt engineering, and entrepreneurial rhetoric creation. Journal of Business Venturing Insights, (2023).
[6] Poola, I. Overcoming ChatGPTs inaccuracies with Pre-Trained AI Prompt Engineering Sequencing Process. International Journal of Technology and Emerging Sciences, 3(3), pp. 16-19. (2023).
[7] Lo, L. S.. The CLEAR path: A framework for enhancing information literacy through prompt engineering. The Journal of Academic Librarianship, 49(4), 102720. (2023)
[8] Meskó, B. Prompt engineering as an important emerging skill for medical professionals: tutorial. Journal of Medical Internet Research, (2023).
[9] Schmidt DC, Spencer-Smith J, Fu Q, White J. Cataloging Prompt Patterns to Enhance the Discipline of Prompt Engineering. (2023)
[10] Shi, Y., Ren, P., Wang, J., Han, B., ValizadehAslani, T., Agbavor, F., Zhang, Y., Hu, M., Zhao, L., & Liang, H. Leveraging GPT-4 for Food Effect Summarization to Enhance Product-Specific Guidance Development via Iterative Prompting. Journal of Biomedical Informatics, 148, 104533. (2023)
[11] 오세준. 수학 문제 해결에서 효과적인 ChatGPT의 프롬프트 고찰: 이차방정식과 이차함수를 중심으로. 수학교육 논문집, 37(3), 545-567. (2023).

분산 식별자를 사용한 머신 러닝 데이터 수집 과정의 개선

한윤경 O1, 고한경 1, 이주희 2, 서중원 2, 조성우 2, 박수용 2

1 서강대학교 메타버스전문대학원, 2 서강대학교 컴퓨터공학과

hanyk1221@gmail.com, hko920920@gmail.com, wngml01101@gmail.com,

jungwonrs@gmail.com, csw2479@gmail.com, sypark@sogang.ac.kr

Refining the Process of Machine Learning Data Collection Using Decentralized Identifier

Yoonkyung Han O1, Hankyeong Ko 1, Juhui Lee 2,

Jungwon Seo 2, Sungwoo Cho 2, Sooyong Park 2

1 Sogang University Graduate School of Metaverse,

2 Sogang University Computer Science Department

요 약

전통적인 머신 러닝 데이터 수집 방식은 중앙 서버에서 데이터를 수집하고 저장한 후 일괄 처리하는 방식이었지만, 현재의 대규모 데이터 환경에서는 이 방식이 여러 문제점을 내포하고 있다. 특히, 데이터의 중앙 집중화는 통신 오버헤드, 데이터 프라이버시 및 보안 문제, 단일 실패 지점의 위험, 그리고 법적 규제의 도전과 같은 문제들을 야기한다. 또한, 데이터 수집 과정에서 다양한 소유자 및 참여자가 관련되어 있는 상황에서 권한 관리의 어려움이 있다. 이러한 문제들을 해결하기 위해, 논문은 머신 러닝에 분산 신원 인증(DID) 기술을 도입하는 방안을 제시한다. 이를 통해 데이터 소유자와 데이터에 대한 접근 권한을 명확하게 정의하고, 데이터 참여자의 접근 제어 및 개인정보보호 문제를 해결하는 블록체인 기반의 머신러닝 데이터 수집과정 개선 시스템을 제안한다.

1. 서 론

기술의 급속한 발전과 데이터 중심의 의사 결정은 데이터 수집 및 처리 방식에 대한 변화를 요구하고 있다. 전통적으로, 머신 러닝(Machine Learning: ML) 데이터 수집 과정은 중앙 서버에서 데이터를 수집하고 저장한 후 일괄 처리하는 방식으로 이루어졌다. 이 방법은 간결성과 직관성에서 장점을 가지지만, 현대의 데이터 환경에서는 큰 문제점을 지니고 있다.

현대의 대규모 데이터는 다수의 이해당사자에 의해 생성되며 지리적으로 분산된 방식으로 저장된다. 이러한 방대한 양의 데이터를 여러 데이터 보유자로부터 중앙 서버로 전송하는 것은 상당한 비용 및 지연 시간을 발생시킨다. 또한, 중대한 데이터 프라이버시 및 보안 문제를 야기한다. 데이터의 중앙화는 단일 실패 지점(Single Point of Failure)을 만들어 데이터 손실 및 침해의 위험을 증가시키며, 특히 개인정보보호 및 데이터 관리와 관련된 법적 규제가 강화되는 현 상황에서 큰 도전이 된다. 또한, 데이터 수집과정에서 다수의 소유자 및 참여자가 관련되어 있기에 권한을 효과적으로 관리하는 것이 어려울 수 있다.

본 연구에서는 이러한 문제점들을 해결하기 위해 머신 러닝 기술에 분산 신원 인증(Decentralized Identifier: DID) 기술을 도입하여 데이터 소유자 및 데이터에 대한 접근 권한에 대한 명확한 정의를 하여 데이터에 대한 참여자의 접근제어 및 개인 정보보호 문제를 해결하고자 한다.

2. 배경지식

2.1 Machine Learning (ML)

머신러닝(ML)은 작업을 수행하기 위해 '경험 및 데이터'를 학습한다는 목표를 가진 학습 프로세스를 포함한다. 특정 작업에서의 머신러닝(ML) 모델의 성능은 시간이 지남에 따라 경험을 통해 향상되는 성능 지표로 측정이 되며, 머신러닝(ML)모델과 알고리즘의 성능을 계산하기 위해서 다양한 통계기법 및 수학적 모델이 사용이 된다. 학습 과정이 끝나면 훈련된 모델을 사용해 훈련과정에서 얻은 경험을 바탕으로 새로운 예시를 분류, 예측 또는 클러스터링할 수 있다. 즉, 머신러닝은 데이터에서 패턴을 학습하고 이를 기반으로 새로운 입력에 대한 의사 결정을 내리는 프로세스를 의미한다.

그림 1은 머신러닝(ML)의 일반적인 접근 방식을 시각적으로 나타낸 것이다[1].

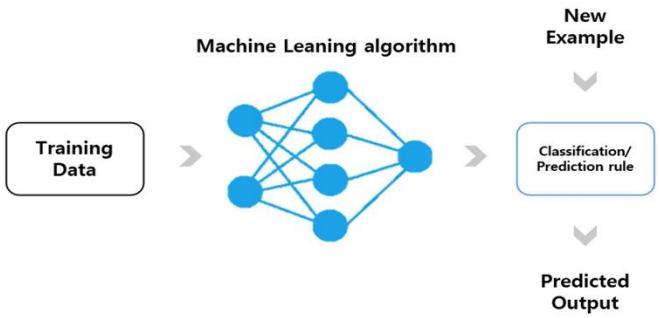


그림 1. 일반적인 ML 접근방식

2.2 Data Collection

머신러닝(ML)에서 데이터 수집은 모델을 훈련시키기 위해 필요한 정보를 수집하는 과정이다. 데이터의 품질과 양은 모델의 성능에 직접적인 영향을 미친다. 데이터 수집 및 훈련 과정은 그림 2와 같이 총 4단계로 이루어진다. 첫째, 여러 명의 데이터 수집가들이 데이터를 모은다. 둘째, 어노테이터들이 데이터에 대한 정답지를 만든다. 셋째, 이렇게 만들어진 데이터와 정답지를 중앙화된 서버에 보낸다. 마지막으로 중앙화된 서버에서 모아진 데이터와 정답지를 가지고 모델을 훈련한다. 이렇게 여러 참여자들이 데이터에 접근하게 되는데 이는 데이터 보안 및 개인 정보 문제를 발생시킨다. 따라서 본 논문에서는 데이터 접근 권한을 확실히 구분하여 부여하는 방안을 제안한다.

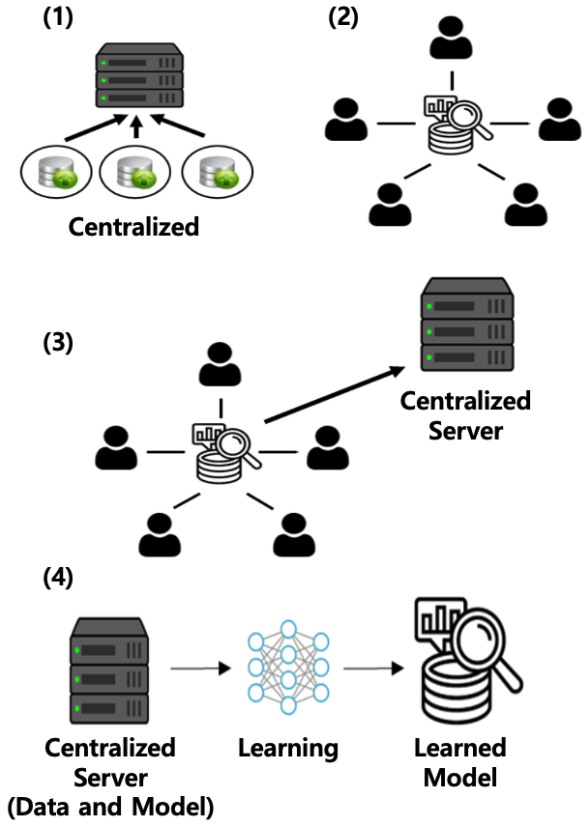


그림 2. 데이터 수집 및 훈련 과정

2.3 Blockchain

블록체인은 2008년 사토시 나카모토라는 익명의 사용자가 비트코인을 위해서 제안하였으며, 이러한 기술의 특징은 P2P(Peer-to-Peer) 네트워크 상에서 데이터의 무결성 및 불변성 등을 보장하고, 블록체인 내에서 발생한 트랜잭션 등 모든 상태 업데이트를 블록내에 기록하게 되는 분산 원장 기술(DLT)이다[2]. 그림 3은 트랜잭션들을 모아 하나의 블록을 생성하고, 연결된 형태를 방식을 나타낸다. 또한, 블록체인은 퍼블릭 블록체인(Public Blockchain), 프라이빗 블록체인(Private Blockchain)으로 나뉜다.

퍼블릭 블록체인은 누구나 자유롭게 참여할 수 있는 개방형 블록체인 네트워크를 말하며, 공공 및 개방형 블록체인이라고 한다. 프라이빗 블록체인은 미리 정해진 단체 및 조직 개인들만 참여할 수 있는 폐쇄형 블록체인 네트워크를 말하며, 허가 받은 사용자들만 네트워크에 참여할 수 있다는 특징을 가지고 있다.

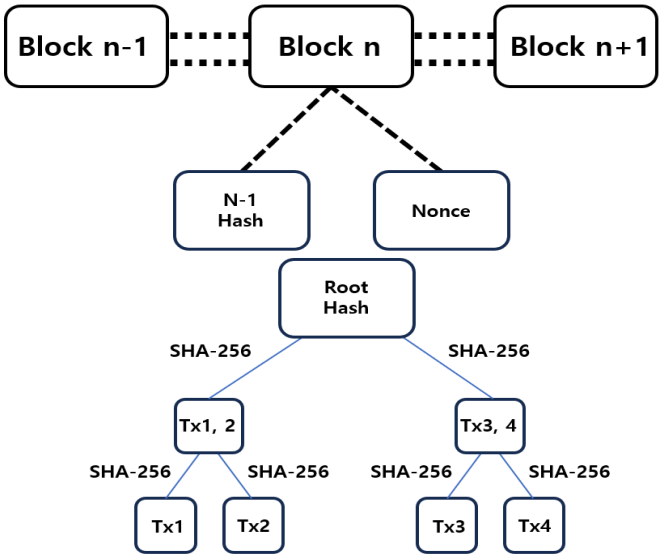


그림 3. 블록체인의 구조

2.4 DID(Decentralized Identifier)

DID는 디지털 ID(Identity)의 자체 관리 및 주권 관리 접근방식을 강조하는 디지털 ID분야의 움직임으로, 중앙 집중 서버 없이 식별자 및 기타 ID 관련 데이터 관리 및 제어를 나타낸다[3]. 이는 사용자들에게 자신의 신원 정보를 효과적으로 소유하고 관리할 수 있는 독립적인 방식을 제공한다. DID는 블록체인과 분산 원장 기술을 활용하여 신뢰성 있는 식별 시스템을 구축할 수 있는 독특한 방식으로 주목받고 있다. 그림 4를 통하여, DID의 전체 프로세스를 확인할 수 있다. 이 프로세스는 사용자가 자신의 디지털 ID를 생성하고 유지하는 과정부터 시작하여, 해당 ID를 사용하여 다양한 온라인 서비스 및 트랜잭션에서 신원을 확인하는 단계까지 포괄적으로 이루어진다. DID는 중앙 집중 서버를 거치지 않고도 안전하고 투명한 방식으로 신원을 관리함으로써 사용자에게 높은 수준의 개인정보 보호와 자율성을 제공한다.

또한, DID의 도입은 기존의 중앙집중식 ID관리 시스템에서 발생하는 보안 문제와 개인 정보 노출

위험을 줄이고, 사용자가 자신의 신원을 보다 적극적으로 통제할 수 있는 새로운 패러다임을 제시하고 있다. 이러한 특징은 현대 디지털 생태계에서 개인정보 보호와 신원 관리에 대한 중요한 이슈에 대한 대안으로 주목받고 있다.

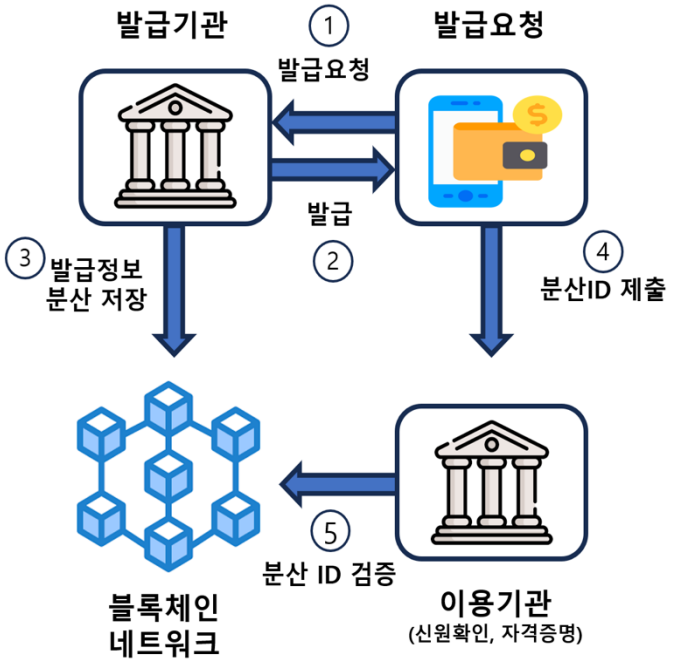


그림 4. DID 발급 프로세스

3. 관련연구

머신러닝 데이터 수집과정에서 데이터 프라이버시 및 보안을 위한 연구로는 X. Chen이 개발한 분산 머신러닝 알고리즘인 Leaning Chain이 있다[4]. 이는 블록체인 기술을 이용하여 선형 및 비선형 학습 모델에 적용 가능한 탈중앙화된 프라이버시 보호 및 보안 메커니즘을 구현하며, 이를 통해 차등적인 프라이버시와 비잔틴 공격 방어를 위한 알고리즘이 포함이 된다. 하지만 위 연구에선 이익의 분배에만 초점을 맞추고 있고 데이터에 대한 권한 관리 및 접근 제어는 고려하지 않는 한계가 있다.

Al-Rubaie은 머신 러닝(ML)에서의 프라이버시를 보호하는 데 관련된 여러 위협과 솔루션에 대한 논문을 작성했다[5]. 머신러닝 모델의 훈련 데이터에는 민감한 정보가 포함되어 있어, 모델이 학습한 정보를 통해 개인의 프라이버시가 침해될 수 있음을 강조한다. 특정 머신 러닝 애플리케이션이 여러 입력 당사자의 데이터를 필요로 할 때, 암호 프로토콜을 사용하여 암호화된 데이터에서 머신 러닝 훈련/테스트를 수행할 것을 제안했다. 이러한 기술 중 많은 방법에서는 효율성을 높이기 위해 데이터 소유자가 계산 서버에 자신의 암호화된 데이터를 기여하도록 하여 문제를 안전한 이중/삼중 계산 설정으로 간소화 시킨다. 그러나 이러한 접근 방식은 서버 중앙화라는 한계점을 가지고 있다.

Reed, Drummond은 개인이나 조직이 중앙 기관의 중재 없이 자신의 디지털 식별자를 만들고 관리할 수 있도록 하는 분산 식별 시스템 DID를 제안했다[6]. 이는 개인의 식별 정보에 대한 더 많은 통제와 프라이버시를 제공하며, 블록체인과 같은 분산 기술을 기반으로 하여 안전하고 탈중앙화된 식별 체계를 구축하는 방향으로 나아가는 중요한 아이디어를 제시하고 있다. 하지만 해당 시스템의 구체적인 적용방안 등에 대한 제안은 부족하다.

따라서 본 논문에서는 머신 러닝 데이터 수집에 블록체인과 DID를 활용하여 데이터의 접근 권한을 명확히 정의하며 개인정보 보호를 달성할 수 있는 탈중앙화 시스템을 제안한다.

4. 접근방안

ML에서 데이터를 수집하기 위해서는 아래 4가지 순서를 따른다. (1)Gathering : 여러 명의 데이터 수집가들이 데이터를 모음, (2)Labeling : 어노테이터들이 데이터에 대한 정답지를 만듦, (3)Sending : 중앙화된 서버에 해당 데이터와 정답지를 보냄, (4)Tranining : 중앙화된 서버에서 모아진 데이터와 정답지를 가지고 모델을 훈련한다.

이러한 과정에서 대규모 데이터는 일반적으로 여러 명에 의해 수집이 되고, 편집되며 지리적으로 분산된 방식으로 저장된다. 이러한 방식은 중요한 데이터 개인정보 보호 및 보안 문제를 야기하며, 데이터 소유자와 데이터에 대한 접근 권한에 대한 명확한 정의를 필요로 한다. 또한, 데이터 수집 과정에서 다수의 소유자 및 참여자가 관련되어 있기 때문에, 각각의 역할 및 권한을 효과적으로 관리하는 것이 어려울 수 있다.

이러한 문제를 해결하기 위해서 그림 5의 접근 방안을 제안하며, 표1은 프로세스에 대한 주체를 표기한다. 주어진 접근 방안은 Claim Holder(CH) 즉, 데이터 수집 과정에서의 참여자들인 소유자, 수집가, 어노테이터, 모델 훈련 담당자는 Claim Issuer(CI)에게 데이터 접근 가능 권한을 요청한다. CI는 CH의 신원을 확인하고, Claim(C)를 발급한다. 발급된 C는 블록체인을 통해 분산 저장된다. CH가 데이터 접근을 필요로 한다면, Claim Verifier Computing Server (CVCS)에 발급받은 C를 제출한다. CVCS는 블록체인에 분산 저장된 C를 통해 신원을 확인하고, CH에게 데이터 접근 권한을 부여한다.

Symbol	Symbol Definition
CH	Claim Holder
CI	Claim Issuer
CVCS	Claim Verifier Computing Server
C	Claim

표 1. 접근방안 표기

이러한 접근방안은 분산된 신원 관리 시스템을 제공하여, 각 참여자가 자체적으로 신원을 관리하며, 이는 머신 러닝 데이터 수집 과정에서 참여자들의 개인 신원을 보다 안전하게 관리 및 보호할 수 있고, 데이터 소유자가 자신의 데이터에 대한 소유 및 접근 권한을

효과적으로 제어할 수 있으며, 각 데이터 소유자는 데이터에 대한 권한을 획득하고 관리할 수 있다. 또한 신원 검증이 가능하므로 데이터 교환 시에 참여자 간의 신뢰성 있는 소통이 가능하고, 이는 데이터의 출처를 확인하고 변조를 방지하는데 도움이 될 수 있다.

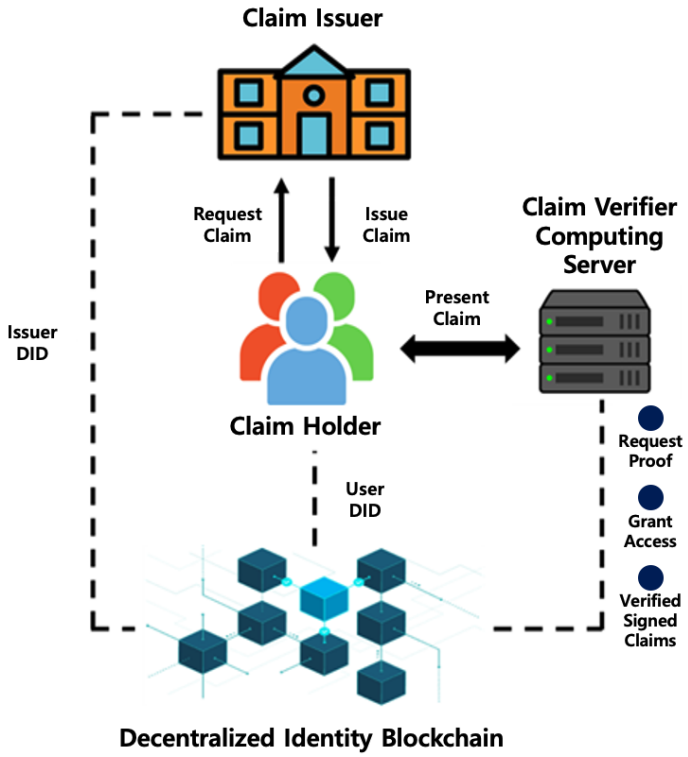


그림 5. MLDID Chain 프로세스

5. 결론 및 향후 연구

본 연구에서 제안된 MLDID Chain 접근 방식은 데이터 수집 과정에서의 개인 정보 보호 및 보안 문제를 주요 고려 대상으로 삼는다. 블록체인 기술을 활용하여 분산된 신원 인증을 통해 각 참여자의 신원을 확인하고 데이터 접근 권한을 부여함으로써, 데이터의 출처를 확인하고 데이터 변조를 방지할 수 있다. 이는 머신 러닝 데이터 수집 과정의 투명성과 신뢰성을 크게 향상시킬 수 있어 이를 활용한다면 더욱 많은 데이터 참여자를 모을 수 있을 것이고, 더욱 향상된 학습 결과를 도출할 수 있을 것으로 보인다.

향후 연구에서는 이러한 접근 방식의 효과를 실증적으로 검증하기 위한 실제 사례 연구 및 실험을 수행할 예정이다. 또한, 데이터 접근 권한 관리와 관련된 법적 및 윤리적 측면도 중요한 연구 주제로 부각된다. 분산 신원 인증 방식의 확장 가능성과 다양한 머신 러닝 환경에서의 적용 가능성에 대해서도 추가적인 탐구가 필요하다. 이러한 폭넓은 범위에서의 적용을 통해 제안된 시스템의 유효성과 실제 현장에서의 적용 가능성을 평가할 수 있을 것이다.

뿐만 아니라, 시스템의 확장성과 효율성을 개선하기 위한 기술적인 측면에서의 연구가 필요하다. 대용량 및 다양한 종류의 데이터에 대한 효율적인 처리 및 분산 시스템에서의 성능 최적화는 중요한 과제이다. 나아가, 보다 사용자 친화적이고 실용적인 환경에서의 적용을 위해 사용성과 안전성에 대한 개선도 고려되어야 한다. 이러한 다양한 측면을 고려하여 향후 연구 방향을 설계하고, 제안된 시스템의 현장 적용 가능성을 더욱 강화하는 데 기여할 것으로 기대된다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 2024년도 메타버스 융합대학원의 연구 (RS-2022-00156318)와 문화체육관광부 및 한국콘텐츠 진흥원의 2024년도 문화기술 연구개발 사업(RS-2023-00219237)으로 수행되었음

참고문헌

[1] Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. Sensors 2018, 18,2674.https://doi.org/10.3390/s18082674

[2] Z. Li et al., "Byzantine Resistant Secure Blockchained Federated Learning at the Edge," in IEEE Network, vol. 35, no. 4, pp. 295-301, July/August 2021, doi: 10.1109/MNET.011.2000604.

[3] V. Kersic, U. Vidovic, A. Vrecko, M. Domajnko and M. Turkanovic, "Orchestrating Digital Wallets for On- and Off-Chain Decentralized Identity Management," in IEEE Access, vol. 11, pp. 78135-78151, 2023, doi: 10.1109/ACCESS.2023.3299047.

[4] X. Chen, J. Ji, C. Luo, W. Liao and P. Li, "When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 1178-1187, doi: 10.1109/BigData.2018.8622598.

[5] Al-Rubaie, M., & Chang, J. M. (2019). Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy, 17(2), 49-58.

[6] Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., & Holt, J. (2020). Decentralized identifiers (dids) v1. 0. Draft Community Group Report.

SAND: 거대 언어 모델을 이용한 소프트웨어 요구사항 명세서에서의 비원자 문장 탐지 및 수정

박상준 ○ 이선구 백종문

한국과학기술원

j010119@kaist.ac.kr, sungu0027@kaist.ac.kr, jbaik@kaist.ac.kr

SAND: Detecting and Correcting Non-Atomic Statements in Software Requirements Specification Using Large Language Model

Sangjun Park○ Sungu Lee Jongmoon Baik

KAIST

요 약

소프트웨어 요구사항 명세서(SRS)는 소프트웨어 개발의 기초이다. 높은 품질의 SRS는 소프트웨어 개발 과정에서의 오류 발생확률을 감소시키고, 오류 수정과정에서 수반되는 비용과 개발기간 지연을 최소화한다. SRS에서의 비원자 문장은 기능 서술의 독자성을 손상시키고, 모호성과 같은 SRS 평가지표를 악화시키며, 독자의 신속하고 정확한 이해를 어렵게 하여, 궁극적으로 오류 가능성을 증가시킨다. 기존에 인공신경망을 이용해 원자성을 향상하려는 시도가 있었으나, SRS을 분해/분석하는 과정에서 성능이 저해되고, 기울기 갱신(gradient update)이 필연적이므로 사용에 어려움이 있다. 최근에, 거대 언어 모델은 그 강력한 자연 언어 처리 능력을 바탕으로 소프트웨어 개발 영역에서 다양하게 이용되고 있다. 본 논문에서는 SRS 내 비원자 문장을 거대 언어 모델, GPT-4를 이용해 탐지 및 수정하는 새로운 도구, SAND를 제안한다. SAND는 기존의 어휘 기반 인공신경망 모델에 비해 비원자 문장 탐지에서 4.1%의 F1-score 향상과, 83.7%의 수정 정확도를 보였다. SAND는 그래픽 사용자 인터페이스(GUI)를 이용해 소프트웨어 개발자로 하여금 쉽고 정확하며, 한정된 컴퓨터 자원으로도 SRS 내 비원자 문장을 탐지 및 수정할 수 있게 촉진한다. 더 나아가, SAND는 프롬프트 변경 등의 간단한 방법을 통해 모호성, 평가가능성, 측정가능성 등의 다른 SRS 평가지표 또한 향상시킬 수 있는 확장성을 가져, 소프트웨어 개발자에게 상당한 편의를 제공할 수 있다.

1. 서 론

높은 품질의 소프트웨어 요구사항 명세서(Software Requirements Specification, SRS)는 소프트웨어 품질 향상에 중요한 역할을 하는 것으로 알려져 있다. 약 50%-80%의 소프트웨어 개발 프로젝트는 낮은 품질의 SRS 또는 SRS의 오독으로 인해 실패한다 [1]. SRS는 소프트웨어 개발의 초기 단계로, SRS의 품질은 전체 개발과정에 상당한 영향을 미친다 [2]. 결함 있는 SRS는 후속 과정에서 오류 확률을 높이고, 오류 수정을 위한 추가적 개발 비용과 지연을 야기한다 [3].

SRS을 평가하는 기준으로는 주로 모호성, 평가가능성, 측정가능성 그리고 원자성이 있다. 그 중 원자성은 SRS 내의 문장이 단일 시스템의 기능, 특징, 요구, 능력을 모든 정보와 세부사항, 한계, 특성을 포함하여 완전히 기술하는 지를 의미한다 [4]. 비원자 문장은 요구사항의 독자적 인식을 저해하고, 다른 SRS 평가지표를 향상하는 알고리즘 등에 의한 분석을 어렵게 한다 [5]. 또한, 비원자 문장은 일반적으로 독자의 구문분석을 느리고 부정확하게 하여 [6], 소프트웨어 개발 중 오류 발생확률을 증가시킨다.

기존의 연구들은 주로 다른 SRS 평가지표 향상, 특히 모호성 개선을 다루었기 때문에 [7], 원자성 향상 관련 연구는 현저히 부족한 상황이다 [4]. 따라서, 본 논문에서는 원자성을 향상하는 방법에 집중하였다.

최근에, 거대 언어 모델은 독해와 문맥 추론을 포함한 강력한 자연 언어 처리 능력을 바탕으로 [8] 헬스케어, 경제 등 다양한 분야에 활발히 이용되고 있다 [9]. 거대 언어 모델은 소프트웨어 공학에서도 요구사항 생성, 개발자 피드백, 소프트웨어 테스트, 문서화 등의 과정에서 사용되고 있다 [10]. 최근 연구에서는 LLaMA를 이용한 코드 리뷰 방법(Junyi Lu et al., 2023) [11], GPT-4를 이용한 버그 수정 방법 (Guoyang Weng et al., 2023) [12] 등이 제안되었다. 이러한 발전을 바탕으로, 본 연구는 소프트웨어 요구공학에 거대 언어 모델을 도입하는데 목적을 뒀다.

이러한 점들을 고려하여, 본 논문은 거대 언어 모델, 특히 GPT-4 [13]을 이용해 SRS 내의 비원자 문장을 탐지 및 수정을 자동화하는 새로운 도구, SAND(Software requirements Automatic Non-atomicity Detection and correction)을 제안한다.

2. 관련 연구

기존에 SRS의 평가지표를 위한 다양한 연구가 있어왔다.

2.1 타 SRS 평가지표 개선

기존의 연구는 주로 모호성과 평가가능성 개선에 집중 되어있다. 주 방법은 어휘, 의미론 기반 [14][15][16], 또는 인공신경망을 이용한다 [17]. 이러한 방법들은 좋은 성능을 보여주지만, 모델에 입력하기 위해 SRS 내 문장을 분해/분석하는 과정이 수반된다. 이 과정에서 오류가 발생하기 쉽고, 성능에 큰 영향을 미친다. 따라서 다양한 SRS 환경에 적용하기 어려움이 존재한다.

2.2 원자성 개선

원자성 개선을 위한 연구도 존재한다. Fahrizal Halim et al. (2019) [18]에서는 베이즈 네트워크, 랜덤 포레스트 트리, 다층 퍼셉트론을 사용하여 비원자 문장을 감지했다. 그러나, 이 역시 SRS 문장 분해/분석 과정이 필요하고, gradient update 과정 때문에 모델이 무거워 사용하기 어렵다. 추가로, 오로지 비원자 문장 감지기능만 존재하여 수정할 수 없기에 사용에 제약이 있다.

3. SAND: 비원자 문장 탐지 및 수정

SAND는 Python으로 작성된 GUI 멀티 플랫폼 어플리케이션으로, Microsoft Windows와 Unix 기반 시스템인 Apple macOS, Linux 등의 환경에서 호환된다. 이 장에서는 SAND의 전반적인 구조에 대해서 소개한다.

3.1 구조

SAND는 그림 1과 같이 4개의 주요 요소로 구성된다.

*Processor*는 다른 요소들에 대해 중추적인 역할을 한다. 자세한 내용은 2절에 설명된다.

*Data loader*는 SAND에 SRS를 로드하는 첫번째 요소이다. .txt, .doc, .docx SRS 파일 형식을 지원한다. *Data loader*는 SRS를 문장 단위로 처리하여 각 문장에 고유 번호를 부여한다.

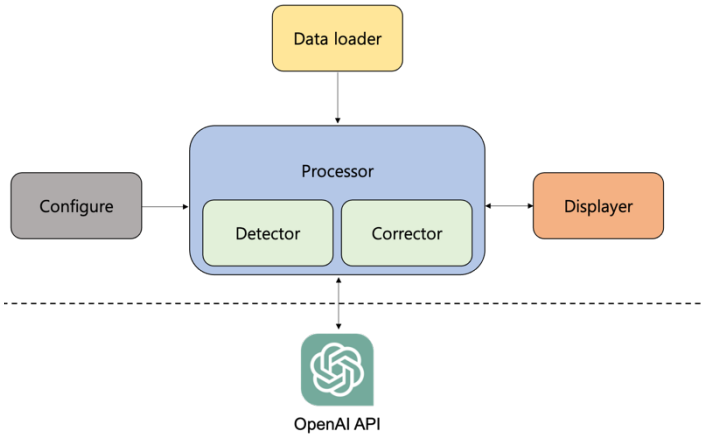


그림 1 SAND 주요 구성요소

이 과정에서 불필요한 글머리기호나 개행 문자를 제거하여 OpenAI API로의 전송을 원활하게 한다. 이후, *Processor*로 데이터를 이동시킨다.

*Displayer*는 SAND의 전반적 GUI를 담당한다. Processor로부터 로드, 탐지, 수정 출력를 받아 사용자에게 시각적 결과를 제공한다. 또한, 디스플레이된 정보에 대해 이후 작업을 위한 사용자와의 상호작용을 *Processor*에게 전달한다.

*Configure*는 SAND의 API 관련 설정을 조정한다. 사용자가 Detector와 Corrector의 프롬프트를 변경하고 OpenAI API와의 통신 지연시간을 속도와 안정성 사이에서 결정할 수 있도록 한다. 이러한 설정 값은 *Processor*에게 적용된다.

3.2 Processor

Processor는 Data loader로부터 SRS 데이터를 받아 SRS 내 비원자 문장을 탐지하고 수정한다. Cofigure로부터 받은 설정으로 OpenAI와 상호작용하고 결과를 반환한다. 이 결과를 Displayer로 전해 사용자가 후속작업을 진행할 수 있도록 한다. Processor는 Detector와 Corrector, 2개의 내부 요소를 포함한다.

3.2.1 Detector

비원자 문장을 탐지하기 위한 프롬프트는 표 1과 같다. 시스템 프롬프트는 GPT-4에게 20년 경력 이상의 소프트 요구공학자라는 Persona [19]를 부여하여 결과 품질을 높이고, 원자 문장과 비원자 문장의 예시를 제공한다.

3.2.2 Corrector

비원자 수정을 위한 프롬프트는 표 2와 같다. 시스템 프롬프트는 Detector와 같다.

표 1 Detector 프롬프트

System	You are a very professional software requirements analyst with more than 20 years of experience. "If the ATM is running out of money, than no card should be accepted and an error message is displayed.", "The system shall allow users to log in using their username", "The system shall allow users to search for products by entering keywords in a search bar.", "The system shall store user passwords securely using industry-standard encryption algorithms." are atomic statements. "The authorization starts after a customer has entered card and his PIN number.", "The system shall allow users to log in using their username and password.", "The system shall provide a user profile page where users can update their personal information, view order history, and change their email address.", "The system should perform well under heavy load, with response times not exceeding 2 seconds for any user action, and it should be able to handle a minimum of 1,000 concurrent users.", "The system shall implement secure user authentication, including password-based authentication, multi-factor authentication (MFA), and integration with external identity providers." are non-atomic statements.
User	Is the following sentence atomic?: "*input sentence*". If it is, say A. Or it is not, say N.

표 2 Corrector 프롬프트

User	Correct this non-atomic statement: "*input sentence*".

3.3 사용자 인터페이스

이 절에서는 세부적인 UI와 유스케이스에 대해 설명한다. 사용자는 그림 2와 같은 화면에서 작업을 시작한다.

3.3.1 로드

사용자가 원하는 SRS를 로드하면, SRS의 각 문장에 고유 번호가 표시된다. 사용자는 'Input Requirements' 영역에서 로드된 SRS를 문장 단위로 볼 수 있고, SRS의 전체 문장 수가 'Detection Panel'에 표시된다. 로드 과정의 결과 예시는 그림 3에 나타나 있고, 시퀀스 다이어그램은 그림 4와 같다.

3.3.2 탐지

탐지 과정은 로드 과정이 선행되었음을 전제로 진행된다. 사용자가 탐지 버튼을 누르면 로드된 SRS 중 해당 문장이 비원자 문장인지 여부를 알려주는 인디케이터가 생성된다. Detector의 탐지 결과가 인디케이터에 반영되고, 사용자가 SRS 내 문장의 비원자 문장 여부를 알 수 있게된다. 또한 SRS의 비원자 문장 비율이 'Detection Panel'에 표시된다. 탐지 과정 결과 예시는 그림 5에 나타나 있고, 시퀀스 다이어그램은 그림 6과 같다.

그림 2 SAND 초기 화면

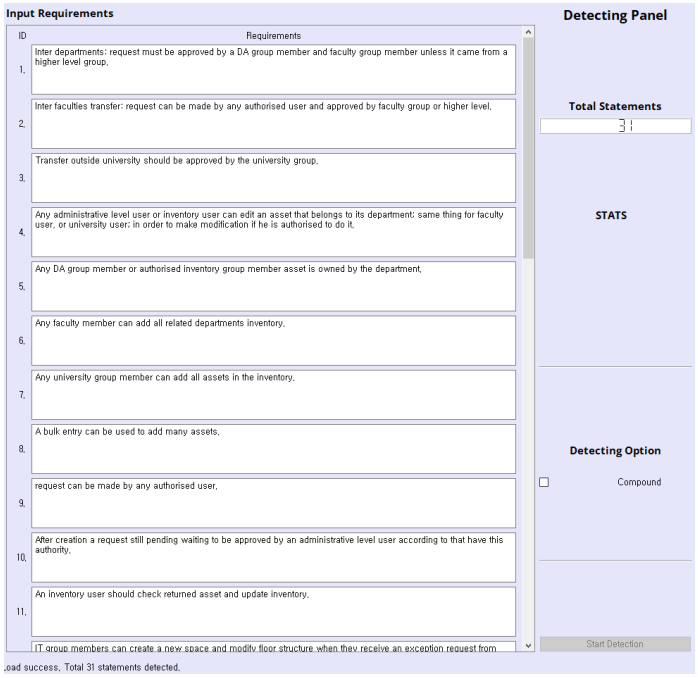


그림 3 로드 과정 결과 예시

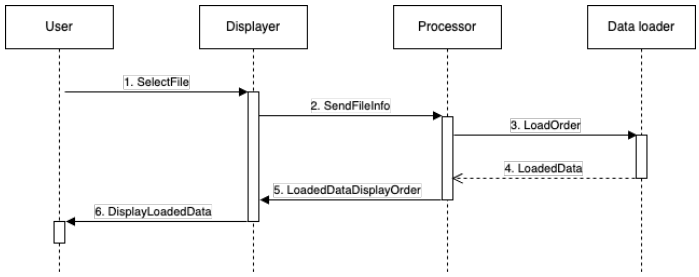


그림 4 로드 과정 시퀀스 다이어그램

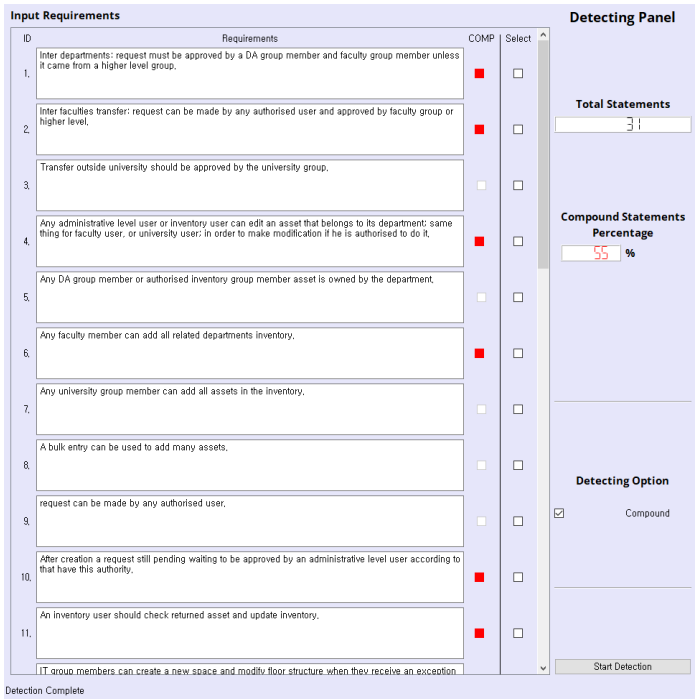


그림 5 탐지 과정 결과 예시

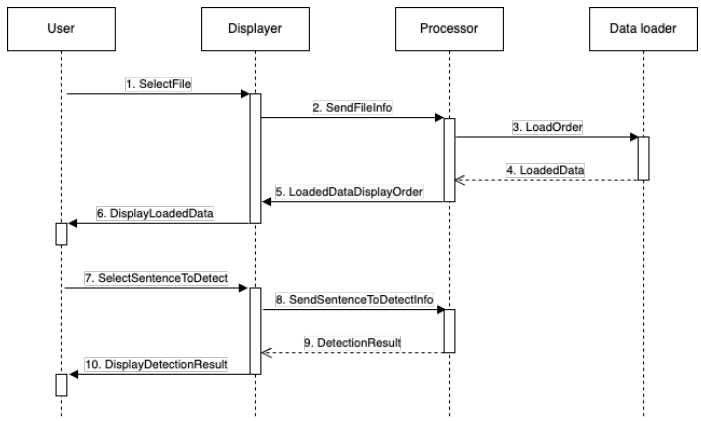


그림 6 탐지 과정 시퀀스 다이어그램

3.3.3 수정

수정 과정은 로드 과정과 탐지 과정이 선행되었음을 전제로 진행된다. 탐지 결과를 바탕으로, 사용자가 수정을 원하는 비원자 문장을 선택할 수 있다. 선택이 완료된 후 수정 버튼을 누르면 Corrector에 의해 수정된 문장이 'Corrected Requirements' 영역에 나타난다. 이후, 사용자는 수정된 결과가 반영된 SRS를 저장할 수 있다. 수정 과정 결과 예시는 그림 7에 나타나 있고, 시퀀스 다이어그램은 그림 8과 같다.

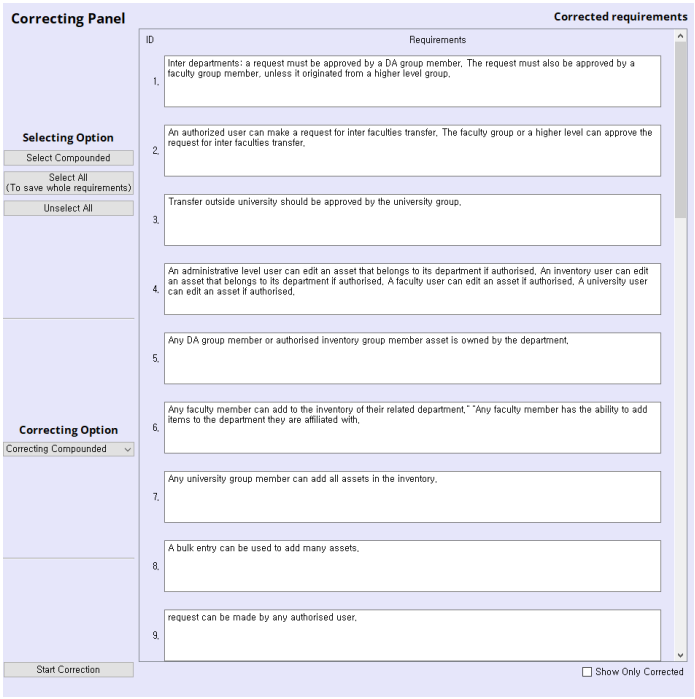


그림 7 수정 과정 결과 예시

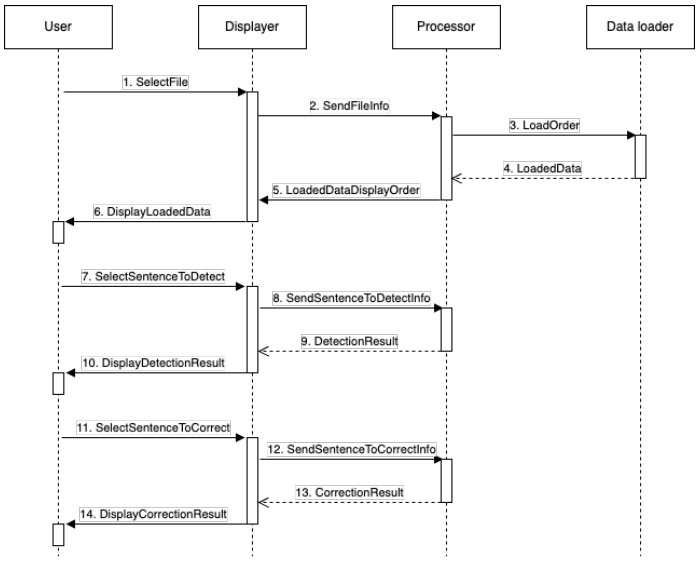


그림 8 수정 과정 시퀀스 다이어그램

4. 실험 설계

이 장에서는 SAND의 효과를 평가하기 위한 연구 질문과 실험의 설계를 설명한다.

연구 질문은 다음과 같다.

Q1. GPT-4가 기존 모델보다 더 정확하게 SRS 내 비원자 문장을 탐지할 수 있는가?

Q2. GPT-4가 SRS 내 비원자 문장을 신뢰성있게 수정할 수 있는가?

Q2.1. GPT-4의 비원자 문장 수정 과정이 정확성을 보이는가?

Q2.2 GPT-4가 비원자 문장 수정 과정이 일관성을 보이는가?

탐지 과정에서는 선행 연구가 존재하여, 기존 모델의 정확도와 SAND의 정확도를 직접적으로 비교해 평가할 수 있으나, 수정 과정은 선행 연구가 존재하지 않아 같은 방법을 사용할 수 없다. 따라서 Aman Madaan et al. [20]에서 LLM의 성능을 자기 피드백으로 향상하는 방법을 사용한것에서 착안하여, 정확성과 일관성을 이용해 평가하는 방법을 채택했다.

실험에 사용된 데이터셋은 PURE 데이터셋 [21]에서 ReqExp [22]을 이용해 추출한 것을 이용했다. 10개의 프로젝트에서 1687개의 문장을 무작위로 추출하고, 각 문장의 원자성 여부가 연구자에 의해 주석되었다.

4.1 탐지 정확도 평가

GPT-4에게 Detector 프롬프트를 이용해서 데이터셋 문장의 비원자성 여부 탐지를 요청한다. 이후 반환된 결과를 분석한다. 분석지표로는 F1-score를 이용했다. 자세한 과정은 그림 9와 같다.

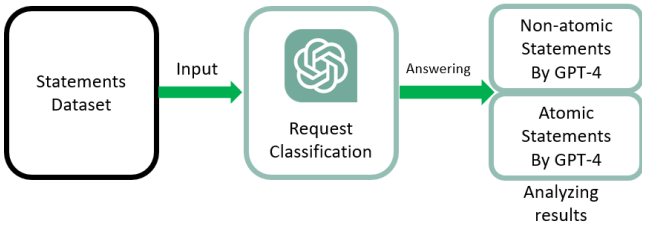


그림 9 탐지 정확도 평가 과정

4.2 수정 신뢰성 평가

4.2.1 수정 정확도 평가

탐지 결과를 바탕으로, 비원자성으로 판별된 문장을 추출하여 GPT-4에게 Corrector 프롬프트를

이용해서 해당 문장을 원자 문장으로 수정할 것을 요청한다. 이후 반환된 결과를 분석한다. 분석 지표로는 Accuracy를 이용했다. 자세한 과정은 그림 10과 같다.

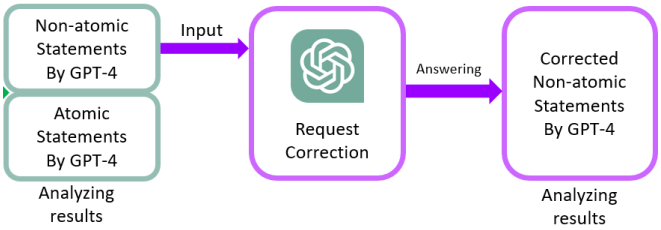


그림 10 수정 정확도 평가 과정

4.2.2 수정 일관성 평가

전체적인 과정은 탐지, 수정 정확도 평가와 유사하다. 그러나 수정 결과를 탐지 평가 과정에서 원자 문장이라 판별된 문장들과 병합한 뒤, 다시 위의 과정을 반복한다. 이후 각 회차마다 반환된 결과를 분석한다. 자세한 과정은 그림 11과 같다.

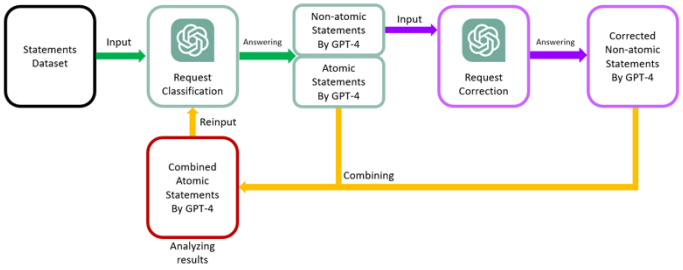


그림 11 수정 일관성 평가 과정

5. 결과 및 분석

5.1 탐지 정확도 평가

표 3을 바탕으로, F1-score는 0.814398로 측정되었다. 기존 모델의 F1-score인 0.782222 [9]에 비해 약 4.1% 향상된 탐지 정확도를 보였다. 이를 바탕으로 GPT-4가 기존 모델에 비해 상대적으로 정확한 탐지 정확도를 보인다 말할 수 있다.

표 3 탐지 정확도 평가 혼동 행렬

		Model		
		Non-atomic	Atomic	
Label	Non-atomic	724	226	950
	Atomic	104	633	737
		828	859	1,687

5.2 수정 신뢰성 평가

5.2.1 수정 정확도 평가

표 4를 바탕으로, Accuracy는 0.837582로 측정되었다. 수정 과정이 상당한 정확도를 보인다 말할 수 있다.

표 4 수정 정확도 평가 결과

Well-correction	Wrong-correction	
1413	274	1,687

5.2.2 수정 일관성 평가

결과는 총 3회차를 통해 산출되었다. 각 회차마다, 총 입력 문장 수는 증가했는데, 하나의 비원자 문장이 여러 개의 원자 문장으로 수정되면서 발생한 현상이다. 그림 12와 같이, 비원자 문장 판별 비율은 1회차에서 0.5417, 2회차에서 0.1411, 3회차에서 0.0552로, 회차가 반복될수록 감소하는 것을 볼 수 있다. 또한 2회차와 3회차는 이미 1회차에서 판별된 이후에 진행된 것인데, 1회차의 비율에 비해 상당히 낮은 것을 확인 할 수 있다. 또한 표 5, 6, 7에서 볼 수 있듯, 모든 배치에서 회차가 반복될 수록 비원자 문장 판별 비율이 하강하는 것을 확인할 수 있다.

이를 통해, GPT-4가 LLM의 필연적 속성인 비결정성을 가짐에도 불구하고, 수정 과정에서 특정 수준의 신뢰성을 가짐을 알 수 있다.

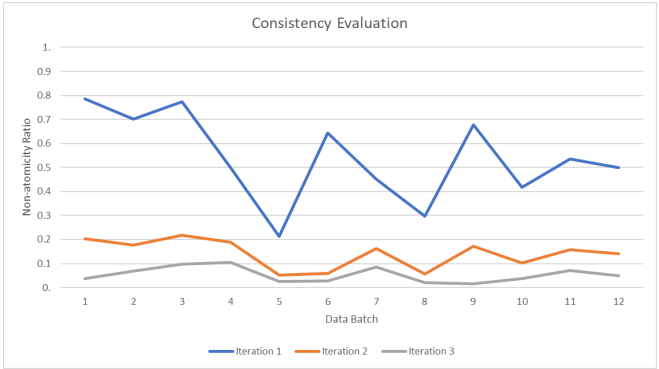


그림 12 수정 일관성 평가 결과 그래프

표 5 수정 일관성 평가 1회차 결과

Data batch	iteration 1			
	Input	Atomic	Non-atomic	Non-atomic ratio
1	84	18	66	0.7857
2	84	25	59	0.7024
3	84	19	65	0.7738
4	84	42	42	0.5
5	84	66	18	0.2143
6	84	30	54	0.6429
7	84	46	38	0.4524
8	84	59	25	0.2976
9	84	27	57	0.6786
10	84	49	35	0.4167
11	84	39	45	0.5357
12	84	42	42	0.5
Total	1008	462	546	0.5417

표 6 수정 일관성 평가 2회차 결과

Data batch	iteration 2			
	Input	Atomic	Non-atomic	Non-atomic ratio
1	252	201	51	0.2024
2	197	162	35	0.1777
3	233	182	51	0.2189
4	174	141	33	0.1897
5	116	110	6	0.0517
6	423	398	25	0.0591
7	167	140	27	0.1617
8	87	82	5	0.0575
9	105	87	18	0.1714
10	137	123	14	0.1022
11	107	90	17	0.1589
12	157	135	22	0.1401
Total	2155	1851	304	0.1411

표 7 수정 일관성 평가 3회차 결과

Data batch	iteration 3			
	Input	Atomic	Non-atomic	Non-atomic ratio
1	320	308	12	0.0375
2	236	220	16	0.0678
3	298	269	29	0.0973
4	208	186	22	0.1058
5	123	120	3	0.0244
6	448	436	12	0.0268
7	196	179	17	0.0867
8	92	90	2	0.0217
9	128	126	2	0.0156
10	160	154	6	0.0375
11	127	118	9	0.0709
12	182	173	9	0.0495
Total	2518	2379	139	0.0552

6. 결 론

본 논문에서는 SRS 내 비원자 문장을 자동으로 탐지 및 수정하는 새로운 도구 SAND를 제안했다. SAND는 기존 모델이 탐지만 가능했던 한계에서 벗어나 수정도 지원하여 편의를 제공한다. SAND는 비원자 문장 탐지에서 기존 모델보다 4.1% F1-score 향상을 보였으며, 수정에서 83.7%의 정확도와 일정 수준의 일관성을 통해 신뢰성을 가졌다.

SAND의 주요 과정은 OpenAI API를 이용하기 때문에 적은 로컬 컴퓨팅 파워로도 작동할 수 있고, GUI를 적용해서 간단하고 쉽게 이용할 수 있어 소프트웨어 개발자에게 강력한 편의를 제공한다.

향후 연구로는, 프롬프트 변경 등의 간단한 방법을 통해 SAND에 원자성 개선 뿐만이 아닌, 모호성, 평가가능성 등 다른 SRS 평가지표를 개선하는 방향으로 진행할 수 있다.

참고문헌

[1] R. Kaur, J. Sengupta, “Software Process Models and Analysis on Failure of Software Development Projects”, International Journal of Scientific & Engineering Research, Vol. 2, Issue 2, pp. 2-3, 2011.

[2] "IEEE Standard for Developing Software Life Cycle Processes", in IEEE Std 1074-1991, p. 45, 1992.

[3] I. Somerville, "Software Engineering", 9th, Addison-Wesley, p. 110, 2009.

[4] W. L. Honig, N. Noda, S. Takada, "Lack of Attention to Singular (or Atomic) Requirements Despite Benefits for Quality Metrics and Management", ACM SIGSOFT Softw. Eng. Notes, vol. 41, no. 4, pp. 1-5, 2016.

[5] W. M. Wilson, L. H. Rosenberg, L. E. Hyatt, "Automated Analysis of Requirement Specifications", International Conference on Software Engineering, 1997.

[6] R. Evans, C. ORĂSAN, "Identifying signs of syntactic complexity for rule-based sentence simplification", Natural Language Engineering, 25(1), 2019.

[7] V. Pekar, M. Felderer, R. Breu, "Improvement Methods for Software Requirement Specifications: A Mapping Study," 2014 9th International Conference on the Quality of Information and Communications Technology, Guimaraes, Portugal, pp. 242-245, 2014.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clack, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, "Language models are few-shot learners", Advances in neural information processing systems, vol. 33, pp. 1877-1901, 2020.

[9] P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope", Internet of Things and Cyber-Physical Systems, Volume 3, pp. 121-154, 2023.

[10] I. Ozkaya, "Application of Large Language Models to Software Engineering Tasks: Opportunities, Risks, and Implications" in IEEE Software, vol. 40, no. 03, pp. 4-8, 2023.

[11] J. Lu, L. Yu, X. Li, L. Yang, C. Zuo, "LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models through Parameter-Efficient Fine-Tuning," 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), Florence, Italy, pp. 647-658, 2023.

[12] G. Weng, A. Andrzejak, "Automatic Bug Fixing via Deliberate Problem Solving with Large Language Models," 2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW), Florence, Italy, pp. 34-36, 2023.

[13] OpenAI, "GPT-4 Technical Report", 2023.

[14] A. Rani, G. Aggarwal, "Advanced Practices to Detect Ambiguities and Inconsistencies from Software Requirements," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 17-21, 2018.

[15] S. Raikar, N. G. Cholli, "An Analysis of Ambiguity Detection Techniques for Software Requirement Specification", 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, pp. 1-4, 2021.

[16] J. H. Hayes, W. Li, T. Yu, X. Han, M. Hays, C. Woodson, "Measuring Requirement Quality to Predict Testability," 2015 IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), Ottawa, ON, Canada, pp. 1-8, 2015.

[17] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, "Automated Handling of Anaphoric Ambiguity in Requirements: A Multi-solution Study," 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), Pittsburgh, PA, USA, pp. 187-199, 2022.

[18] F. Halim, D. Siahaan, "Detecting Non-Atomic Requirements in Software Requirements Specifications Using Classification Methods," 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS), Denpasar, Indonesia, pp. 269-273, 2009.

[19] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilber, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT", 2023.

[20] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, P. Clark, “SELF-REFINE: Iterative Refinement with Self-Feedback”, 2023.

[21] A. Ferrari, G. O. Spagnolo, S. Gnesi, "PURE: A dataset of public requirements documents.", 2017 IEEE 25th International Requirements Engineering Conference (RE), IEEE, pp. 502-505, 2017.

[22] V. Ivanov, A. Sadovykh, A. Naumchev, A. Bagnato, K. Yakovelv, "Extracting Software Requirements from Unstructured Documents", 2022.

Explainable AI 기법을 활용한 다중 오믹스 데이터 기반 COVID-19 중증도 예측 모델

장호중[O] 주정진[O] 김준구 조겨리
충북대학교 컴퓨터공학과
[O] These authors contributed equally to this work
hojoong310@gmail.com, wnwjdwls321@gmail.com, jk901@chungbuk.ac.kr
kyurijo@chungbuk.ac.kr

COVID-19 Severity Prediction from Multi-Omics Data Using Explainable AI Method

Hojoong Jang[O] Jeongjin Ju[O] Junku Kim Kyuri Jo
Department of Computer Engineering. Chungbuk National University

요 약

생물 실험 데이터 중 단일 세포 RNA 시퀀싱(Single Cell RNA-sequencing)으로부터 획득한 유전자 발현량 데이터는 최근 환자의 상태 분류, 예후 예측 등을 위한 인공신경망 모델의 입력 특징으로 활용되고 있다. 특히, 이러한 인공신경망 모델 구조에 생물학적 패스웨이(biological pathway) 정보를 반영하고자 하는 시도가 활발히 이루어지고 있으나, 기존 연구는 특정 질병에 대한 모델이거나 정량적 평가가 어려운 한계를 가지고 있다. 본 연구에서는 유전자 발현량을 학습하여 질병의 중증도를 예측하는 심층 신경망 모델 구조를 유전자와 생물학적 패스웨이의 계층적인 관계를 반영할 수 있도록 구성하고, Layer-wise Relevance Propagation(LRP)[1] 기법을 적용하여 예측에 중요한 기전으로 작용한 생물학적 패스웨이를 순위화하는 모델을 제안한다. 제안된 모델의 성능 검증 및 비교를 위해 COVID-19 환자 데이터를 적용하였으며, 그 결과 환자의 중증도 예측 성능이 유사 기법들보다 높음을 확인하였다. 또한, 세포의 종류별로 환자 중증도에 영향을 준 후보 생물학적 패스웨이를 선별하였다.

1. 서 론

차세대 시퀀싱(next generation sequencing) 기술의 발전에 의해 대규모 오믹스(omics) 데이터 세트가 대량 생산되어 이를 분석하고 활용하는 연구가 활발하게 진행되고 있다. 이 가운데 RNA 시퀀싱 데이터는 각 유전자로부터 생성된 단백질의 양을 추정할 수 있어 세포나 조직의 기능 및 상태를 이해하는데 큰 도움이 된다. 이에 따라 단일 세포 RNA 시퀀싱(Single Cell RNA sequencing, 이하 scRNA-seq) 데이터로부터 얻은 각 유전자의 발현량을 인공신경망 모델의 입력 특징으로 하여 환자의 상태 분류 및 예후 예측 등을 수행하는 모델이 다수 개발되어 왔다[2].

생물학적 패스웨이(Biological Pathway)는 생물체 내의 단백질, 유전자, 세포 등과 같은 요소들이 상호작용하고, 변화하는 과정을 상세하게 설명할 수 있는 생물학적 개념으로, 세포의 변화를 이끌어내어 질병에 어떠한 영향을 미치는지에 대한 해석에 도움을 준다. 유전자 발현량을 입력으로 사용하는 인공신경망 모델 가운데, 최근 유전자와 패스웨이와의 계층적 구조를 모델 구조에 반영하고자 하는 시도가 이루어지고 있다.

해당 연구로는 DeepHisCoM(Deep learning pathway analysis using hierarchical structural component models) [3]과 PASNet(Pathway-Associated Sparse

[1] 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업(2019-0-01183), 질병관리본부 연구개발과제(project No. 2023-ER0803-00) 지원을 받아 작성되었음

Deep Neural Network)[4]이 있다. DeepHisCoM[3]은 유전 데이터를 통해 중요한 패스웨이를 확인하는 모델로, 질병과 바이오마커 간의 생물학적 매커니즘을 확인하며 일종의 앙상블[5] 기법을 활용하여 학습된다. 하지만 모델 구조에 따라서 입력 데이터가 증가할수록 학습량이 급증하여 정량적인 평가가 어렵다는 문제점이 있다. PASNet[4]은 대규모 유전체 데이터를 사용하여 환자의 예후를 정확히 예측하려는 모델로, 특히 Glioblastoma multiforme(GBM)환자 데이터를 활용하여 GBM 의 생물학적 분석과 패스웨이를 이해하려한다. 하지만 PASNet[4]의 모델 구조는 GBM 데이터에 특화되어 있어 다양한 형태와 스펙트럼을 가진 유전자 발현량을 가진 데이터를 활용하는데 있어서 범용성이 떨어진다는 문제점이 있다.

본 연구에서는 특정 질병에 국한되지 않고 범용적으로 사용될 수 있고, 연산량도 적으면서 검출된 패스웨이에 대해서 정량적인 값을 도출할 수 있는 모델을 제안한다. 제안된 모델을 통해 환자의 중증도와 결과에 크게 기여한 패스웨이를 확인하고, 이에 대한 성능을 COVID-19 환자들의 scRNA-seq 데이터를 통해 타 모델과 비교하였다.

2. Explainable AI와 유전자-패스웨이 계층 구조를 활용한 COVID-19 중증도 예측 모델

2.1 데이터셋 구성

제안한 모델의 검증을 위해 질병관리청 국립보건연구원에서 구축한 COVID-19 환자의 시계열 scRNA-seq 데이터를 사용하였다. 해당 데이터는 경북대 정인욱 교수 연구진에 의해 분류 모델의 학습 데이터로 활용될 수 있도록 라벨링 되었다. 그림 1은 데이터 라벨링에 사용된 표준 시퀀스를 나타낸다. 표준 시퀀스는 중증도가 크게 높아졌다가 떨어지는 COVID-19 환자의 시퀀스(Severe)와 일반인의 시퀀스(Normal)로 구성되어 있다. 각 scRNA-seq 샘플은 해당 샘플의 중증도가 표준 시퀀스의 어떤 위치에 해당하는지 빨간색으로 매핑되어 표현되며, 크게는 중증도가 상승하는 위치이면 Deterioration Phase(DP), 하강하는 위치이면 Recovery Phase(RP) 두 가지로 구분하였다.

유전자와 패스웨이의 계층관계를 확인하기 위해 Reactome[6]의 데이터베이스에서 패스웨이 관련 정보를 세포별로 활용하였다. 활용한 세포는 Monocyte(Mono), T heler cell(CD4T, CD8T, otherT), Natural Killer cell(NK), B cell(B), Dendritic cell(DC), other이다.

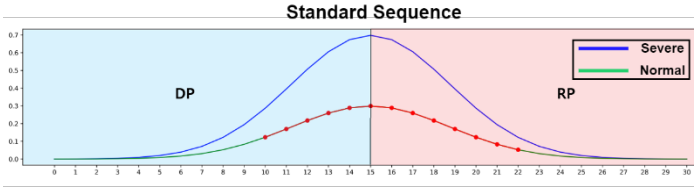


그림 1 중증도 데이터 라벨링을 위한 표준 시퀀스

2.2 데이터 전처리

scRNA-seq 데이터를 세포별로 분석하기 위해 세포별로 pseudobulk를 생성하여 유전자 발현량을 따로 구성하고자 하였다. 이에 따라 중증도 라벨링 역시 세포별로 분리된 데이터에 대해, 각 시간대에서 공통적으로 관찰되는 유전자들을 고려하여 수행되었다. 최종적으로 사용된 세포별 유전자 개수 및 샘플 개수는 표 1과 같다.

이후 인공신경망 모델에서는 Reactome[6]에서 수집한 패스웨이 데이터를 기반으로 하여 DC는 15개, 나머지 세포는 13개의 패스웨이를 사용하였다.

표 1 scRNA-seq 데이터의 세포별 유전자 개수와 전체 시간대 및 라벨별 샘플 개수

Cell-type	유전자 개수	전체 샘플	RP 샘플	DP 샘플
CD4T	288	213	126	87
CD8T	288	213	126	87
otherT	288	190	113	77
NK	288	212	125	87
Mono	288	213	126	87
Other	288	213	126	87
DC	305	198	117	81
B	288	213	126	87

2.2.1 Pseudobulk 분석을 위한 유전자 발현량 정규화

유전자 발현량 분석은 생명 과학에서 핵심적인 연구 도구로 자리 잡았다. 특히, RNA-seq 기술은 그 높은 해상도와 정확성으로 인해 유전자 발현량을 측정하는데 널리 사용되고 있다. 하지만 RNA-Seq 데이터는 실험 조건, 시퀀싱 깊이 등 여러 요인에 의해 영향을 받을 수 있어, 이를 보정하는 정규화 과정이 필수적이다. 시퀀싱 깊이가 더 깊은 샘플에서는 동일한 read count가 발현량이 더 낮음을 의미하기 때문이다.

본 실험에서는 각 세포에서 추출된 샘플 간의 시퀀싱 깊이 차이를 보정하기 위해 DESeq2(Differential gene

expression analysis based on the negative binomial distribution)[7]에서 유전자 발현량을 정규화하는 방법을 선택하였다. 유전자 발현량을 정규화한 공식은 다음과 같다.

$$\boldsymbol{Normalized\ Count} = \frac{\boldsymbol{Raw\ count}}{\boldsymbol{Size\ Factor}} * \mathbf{10000}$$

$$\boldsymbol{Size\ Factor} = \boldsymbol{median}\left(\frac{sample`s\ read\ count}{median\ of\ other\ samples`\ read\ counts}\right)$$

2.3 학습 모델

본 논문에서 제안한 모델(그림 2)은 유전자와 발현량을 입력 데이터로 받아 중증도(DP/RP)를 분류하는 모델이다. 인체 내에서는 여러 유전자들이 모여 하나의 기능을 수행하며, 각 기능별 유전자 집합은 패스웨이 데이터베이스에 공개되어 있다. 이러한 생물학적 매커니즘을 반영하기 위해, 제안한 모델은 입력층, 은닉층, 패스웨이층, 출력층의 4 개의 계층으로 구성되어 있다. 입력 데이터로 들어온 유전자 발현량은 은닉층을 거쳐 해당 유전자가 속한 패스웨이 노드로 전달된다. 이 때 각 패스웨이 노드는 패스웨이에 속한 유전자의 개수에 따라 상위 30%, 중위 30%, 하위 40%로 구분되어, 유전자의 개수가 많은 패스웨이는 더 많은 은닉층 노드와 연결되도록 구성되었다. 패스웨이 계층과 출력층은 Fully Connected 되어있으며, 출력층을 통해 중증도를 DP 또는 RP 로 이진분류한다. 순전파가 진행될 때 입력층과 은닉층, 은닉층과 패스웨이 층 사이의 가중치는 별도로 저장되어 LRP[1] 연산에 활용되고, 세 개 계층 사이의 masking 매트릭스에 파일로 저장된다.

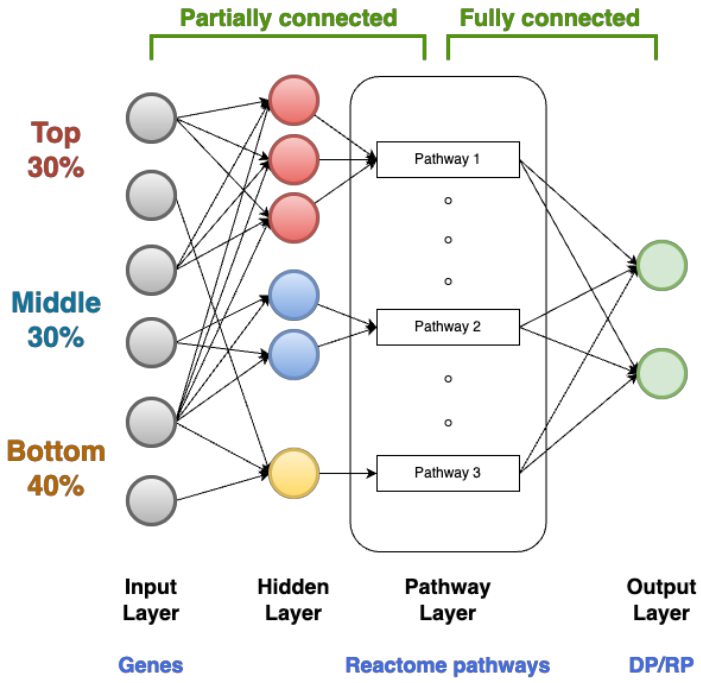


그림 2 제안한 모델의 구조

패스웨이의 개수는 방대하며, 모든 패스웨이를 확인하는 것은 비효율적이다. 특히, 적은 수의 유전자가 배정되는 작은 패스웨이는 종종 더 많은 유전자가 배정된 패스웨이와 중복되며, 모델의 학습을 방해하고 결과의 복잡성을 증대시킨다. 또한, 인간의 전반적인 동작에 관여하는 신진대사와 연결된 패스웨이는 많은 유전자와 연결되어 있어 결과에 대한 기여도가 과도하게 분배된다는 문제점이 있다. 이에 따라 Reactome 데이터베이스의 전체 패스웨이 가운데 유전자가 1500 개 이상 또는 50 개 이하인 패스웨이는 본 모델에서 배제되었다.

그림 3 을 통해 masking 행렬에 대한 pseudocode 를 살펴볼 수 있다. Input-hidden layer 의 masking 행렬은 유전자와 은닉층의 경로 간의 관계를 나타내며, 해당 관계가 존재하는 경우에만 해당 위치에 1 이 표시되며, hidden-pathway layer 의 masking 행렬은 은닉층과 패스웨이 목록 간의 연결 여부를 나타내며, 연결된 경우에만 해당 위치에 1 이 표시된다.

Algorithm 1: Making Masking Matrix of input-hidden layer

```
Data: gene_list, hidden_one_list
Result: input_to_h1_masking
input_to_h1_masking = zeros matrix with shape (len(gene_list),
  len(hidden_one_list));
for i ← 0 to len(gene_list) - 1 do
    gene_pathways = GetPathwaysForGene(gene_list[i]);
    for j ← 0 to len(hidden_one_list) - 1 do
        pathway_name = ExtractPathwayName(hidden_one_list[j]);
        if pathway_name in gene_pathways then
            input_to_h1_masking[i, j] = 1;
        end
    end
end
```

Algorithm 2: Making Masking Matrix of hidden-pathway layer

```
Data: hidden_one_list, sorted_pathways
Result: h1_to_pathway_masking
h1_to_pathway_masking = zeros matrix with shape
  (len(hidden_one_list), len(sorted_pathways));
for i ← 0 to len(hidden_one_list) - 1 do
    pathway_name = ExtractPathwayName(hidden_one_list[i]);
    for j ← 0 to len(sorted_pathways) - 1 do
        if pathway_name == sorted_pathways[j] then
            // Set corresponding element in the matrix to 1
            h1_to_pathway_masking[i, j] = 1;
        end
    end
end
```

그림 3 Masking 행렬에 대한 알고리즘

3. 실험 결과

3.1 모델의 성능 평가:

모델의 성능을 평가하는 대표적인 방법으로는 ROC-AUC[8]가 있다. ROC(Receiver Operating Characteristics) curve는 종속변수값을 무엇으로 예측할 것인지의 기준이 되는 확률(threshold probability) 값에 따른 TPR(True Positive Rate)과 FPR(False Positive Rate)값을 활용한다. TPR은 실제 양성 중 양성으로 예측한 비율을, FPR은 실제 음성 중 양성으로 잘못 예측한 비율을 나타내며, TPR을 Y축으로, FPR을 X축으로 놓은 곡선으로 표현된다. AUC(Area Under the Curve)는 ROC curve 아래의 면적을 의미하며, 값이 1에 가까울수록 좋은 모델을 나타낸다.

설명가능한 인공지능 기법인 LRP[1]를 활용하는 모델을 구축하고 결과에 기여한 패스웨이의 기여도를 표 2의 LRP score 열을 통해 확인하였다. 또한, 모델의 성능을 확인하기 위해 제안된 모델과 비교모델의 AUC값을 시각화하였다(그림 4). 본 논문에서 제안한 모델을 제외하면 Logistic Regression이 AUC가 가장 높기에 환자의 예후를 예측하는데 활용된 데이터셋이 비교적 선형적인 형태를 띄고 있다고 예측되며, 비슷한 AUC를 가진 DeepHisCoM[3]과 PASNet[4]은 본 논문에서 제안한 모델과 구조적으로 유사하기에 높은 AUC를 나타낸 것으로 보인다. DeepHisCoM[3]은 각 패스웨이 별로 독립적인 네트워크를 형성하지만 모델의 구조는 입력층, 은닉층, 패스웨이층, 출력층으로 유사하고, PASNet[4]은 은닉층이 패스웨이층과 출력층 사이에 존재하지만, 희소 코딩을 활용하여 관련이 있는 유전자와 패스웨이들에게 집중한다는 유사성이 존재한다. 반면, 기본적인 인공신경망에서 과적합 문제를 해결하여 학습의 효율을 향상시키는 dropoutNN은 0.34로 매우 낮은 AUC를 나타내었는데, 이는 유전자와 패스웨이 간의 관계에 집중하지 않고 훈련에서 일정 비율의 뉴런을 무작위로 제외하였고, 하이퍼파라미터 튜닝을 거치지 않고 기본적인 값을 활용하였으며, 데이터의 절대적인 양이 많지 않아 적절하게 학습되지 못한것으로 예측된다.

모델을 효율적으로 학습시키기 위해 Optuna[9]를 통해 세포별로 학습률, 은닉층에 배정되는 입력 데이터의 개수, 최적화 기법(Adaptive moment estimation(Adam)), Stochastic gradient descent(SGD), Root Mean Squared Propagation(RmsProp))에 대한 하이퍼 파라미터 튜닝을 진행하였다. 튜닝을 통해 각 세포별로 모델을 최적화시키고, 세포별로 얻은 결과를 앙상블[5] 기법을 통해 각 세포가 아닌, 데이터로 들어간 모든 세포에 대해 비교적 일관적인 성능이 출력되도록 일반화하였다. 모든 세포에 대해 앙상블[5] 기법을 활용하고, 추가적으로 특정 세포의 출력 특징을 제외하여 모델의 성능을 확인하였는데, 이를 통해 특정 세포에 대한 데이터가 모델 성능에 어떠한 영향을 끼치는지 간접적으로 확인하였다.

실험 결과 제안된 모델은 은닉층을 통해 질병에 유의한 패스웨이에 집중하여 모든 노드들이 학습되지 않지만 유전자와 패스웨이의 계층적인 관계를 효율적으로 구성하면서 COVID-19 데이터를 효율적으로 학습하였음을 그림 5를 통해 확인하였다. 그림 6은 제안한 모델에 앙상블[5] 기법을 적용하여 각 세포에 대해 예측한 모델의 성능을 나타내며, 세포를 하나씩 제외해가며 얻은 모델의 성능분포를 세포별로 시각화하였다. DC세포와 otherT 세포의 데이터를 예측하였을 때 모델의 성능이 뚜렷하게 감소한 것을 확인하였고, 모든 세포가 포함되었을 때와 otherT 세포의 데이터를 제외했을 때 모델의 성능이 모든 세포에서 가장 높았으며, NK세포 데이터를 제외하였을 때 otherT 외의 다른 세포에 대해 모델의 성능이 최소치가 되는 것을 알 수 있다.

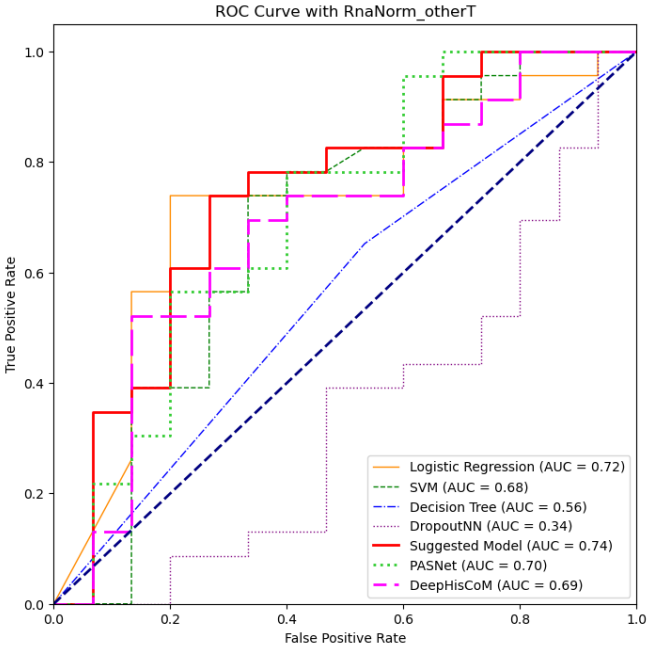


그림 4 otherT 세포에 대한 모델별 ROC Curve

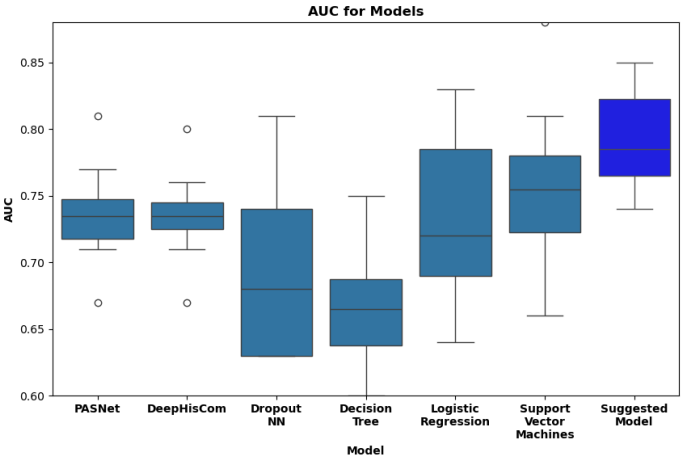


그림 5 실험한 세포에 대한 모델별 AUC

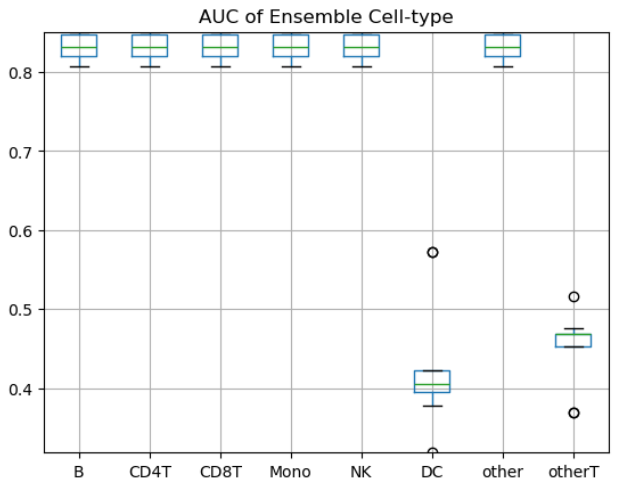


그림 6 앙상블 기법을 활용하여 예측한 세포별 AUC

표 2-1 세포별 상위 5위 패스웨이에 대한 LRP score

Cell	Pathway	Score
CD8T	Signal Transduction	73.5
	Disease	46.33
	Innate Immune System	38.49
	Cellular responses to stimuli	14.61
	Gene expression (Transcription)	6.16
CD4T	Metabolism of proteins	157.82
	RNA Polymerase II Transcription	110.74
	Signal Transduction	104.97
	Cellular responses to stress	95.17
	Innate Immune System	51.28
B	Gene expression (Transcription)	68.03
	Innate Immune System	7.04
	Cellular responses to stimuli	4.65
	Cytokine Signaling in Immune system	2.82
	Metabolism of proteins	1.91
DC	Metabolism of proteins	356.74
	Platelet activation, signaling and aggregation	28.51
	Generic Transcription Pathway	19.51
	Developmental Biology	10.88
	Innate Immune System	9.59
other	Cellular responses to stress	86.62
	Metabolism of proteins	77.47
	Innate Immune System	4.97
	Neutrophil degranulation	1.19
	Gene expression (Transcription)	0.62
NK	Cellular responses to stress	81.42
	Metabolism of proteins	62.65
	Cellular responses to stimuli	45.07
	Hemostasis	40.03
	Signal Transduction	34.41

표 2-2 세포별 상위 5위 패스웨이에 대한 LRP score

Mono	Metabolism of proteins	701.05
	Disease	285.64
	Innate Immune System	84.64
	Signal Transduction	78.74
	Hemostasis	20.06
otherT	Disease	65.44
	Innate Immune System	34.39
	Neutrophil degranulation	17.14
	Immune System	4.43
	Signal Transduction	4.33

4. 결론 및 향후 연구

본 연구에서는 COVID-19 환자에게서 획득한 유전자 발현량 데이터를 기반으로 환자의 중증도를 세포별로 예측하는 심층 신경망 모델을 제시하였다. 질병에 유의한 영향을 준 세포를 구분하기 위해 세포별로 학습을 진행하였고, 세포를 제외해가며 결과를 확인하는 앙상블[5] 기법을 활용하여 질병에 유의한 영향을 준 세포를 간접적으로 구분하였다.

데이터셋 분석 결과 비교 모델보다 COVID-19 데이터를 효율적으로 학습하였음을 확인하였고, 중증도 예측에 중요하게 작용한 패스웨이를 LRP score를 기반으로 검출하였다. 앙상블[5] 적용 결과 특정 세포를 제외하여도 성능의 뚜렷한 차이는 없으나, DC세포와 otherT세포는 모델 성능의 일반화에 악영향을 끼치며, NK세포가 비교적 큰 영향을 주는 것으로 예측된다. 하지만 활용한 데이터의 절대적인 양이 많지 않다는 한계점이 존재한다.

향후 연구에서는 보다 많은 양의 데이터를 활용하여 환자의 예후를 더욱 세밀하게 예측하기 위해 중증도를 이진분류가 아닌 다중분류를 진행해야 하며, 결과에 기여한 패스웨이를 지식 기반 그래프에 적용하여 검출한 패스웨이가 실질적인지 확인하며 환자에 맞는 compound를 찾아내는데에 활용하고자 한다.

5. 참고 문헌

[1]Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, 193-209.

[2] Alharbi, F., & Vakanski, A. (2023). Machine learning methods for cancer classification using gene expression data: A review. Bioengineering, 10(2), 173.

[3]Park, C., Kim, B., & Park, T. (2022). DeepHisCoM: deep learning pathway analysis using hierarchical structural component models. *Briefings in Bioinformatics*, *23*(5), bbac171.

[4]Hao, J., Kim, Y., Kim, TK. *et al.* PASNet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data. *BMC Bioinformatics* 19, 510 (2018). https://doi.org/10.1186/s12859-018-2500-z

[5]Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249.

[6]Griss J, Viteri G, Sidiropoulos K, Nguyen V, Fabregat A, Hermjakob H. ReactomeGSA - Efficient Multi-Omics Comparative Pathway Analysis. Mol Cell Proteomics. 2020 Sep 9. doi: 10.1074/mcp. PubMed PMID: 32907876.

[7]Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*, *15*(12), 1-21.

[8]Huang, Jin, and Charles X. Ling. "Using AUC and accuracy in evaluating learning algorithms." *IEEE Transactions on knowledge and Data Engineering* 17.3 (2005): 299-310.

[9]Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2623-2631)

오픈소스 시뮬레이터 기반 자율주행 구현 및 검증

윤수한, 기석철
충북대학교 전자공학부, 지능로봇공학과
soohan98@chungbuk.ac.kr, sckee@chungbuk.ac.kr

Implementation and Verification of Autonomous Driving Based on Open Source Simulator

Soo-Han Yoon, Seok-Cheol Kee
School of Electronics Engineering, Dept. of Intelligent Systems & Robotics, Chungbuk National University, Chungbuk, South Korea

요 약

본 연구에서는 오픈소스 시뮬레이터를 기반으로 자율주행 구현 및 검증을 목표로 한다. 사용하는 프록램은 Carla, ROS2, Autoware로 각각의 오픈 소스 시뮬레이터를 분석하여 각 프로그램이 본 연구에서 사용이 용이한지 타당성을 보인다. 그 후 자율 주행 프로그램을 구현 후 필요한 구성요소와 구조를 정리하여 정확한 자율 주행 원리를 밝힌다. 실험은 다음과 같이 이루어 진다. Lidar 기능을 이용하여 시뮬레이션 상에서 자율주행을 시킨다. 단 주행할 때마다 다른 종류의 장애물을 배치하여 Autoware의 Lidar 기능의 성능을 측정하고 보완점을 찾는다. 장애물은 차량, 보행자, 자전거로 하며 차량에 경우 5가지 시나리오(일반주행, 주행방해, 끼어들기, 차량 회피 및 차선 변경, 다수의 차량 회피)로 진행한다. 주행 성공 여부는 충돌 혹은 정지하지 않고 목표지점에 도달하였는 가로 한다. 필자는 위에 7가지 실험을 진행한 결과로 성공 시나리오에서는 Autoware의 Lidar 기능의 신뢰도를 확인할 수 있었으며 실패 시나리오를 통해서 Lidar의 인식문제, 프로그램상 충분히 회피 할 수 있음에도 경로를 수정하지 않고 멈추는 문제, 목표지점을 재 지정하면 일정 부분 정상가동이 되는 문제, 차가 이동 할 수 없는 경로로 이탈하면서 정지하는 문제, 비상 상태에서의 Autoware의 대응문제, H/W의 한계점과 같은 보완점을 도출할 수 있었다.

1. 서 론

현대 사회에서 자동차 산업은 현재 기술적으로 혁신과 변화가 일어나고 있다. 첫째, 전통적인 내연기관에서 전기차로의 변화가 일어나고 있으며 둘째, 자동차가 기계공학 중심의 단순 이동수단에서 소프트웨어와 전장부품이 중심이 되는 산업으로 변화되고 있다. 이에 따라 내연기관에서 유명했던 기업들이 현재 매출이 높음에도 저평가 받기도 하며 비교적 신생기업들이 고평가 받는 경우가 생기고 있다. 그리고 이것은 자동차 산업을 핵심 산업으로 가지고 있는 한국에게도 매우 중요한 일이다. 산업의 중요성과 반대로 자율주행이 대중의 관심을 받은 것은 얼마되지 않았으며 이에 따라 자율주행 관련해서 연구해야 할 과제들이 많다. 하지만 누군가가 자율 주행에 대하여 공부 혹은 사업을 시작한다면 과연 그 사람이 제대로 프로젝트를 진행할 수 있을까? 비싼 전자기기들과 여러 프로그램의 라이센스 가격으로 인해서 시작하는 것조차 힘들 것이며 무엇을 사용해야 하는지도 어떻게 사용해야 하는지도 모를 것이다. 하지만 오픈 소스를 활용한다면 이러한 문제점을 보완할 수 있을 것이다. 그 이유로 첫째, 오픈소스는 사용자 무료이기 때문에 연구에 필요한 비용을 절감할 수 있다. 둘째, 다수의 사용자를 통해 생성된 커뮤니티로 인해서 보다 많은 정보를 쉽게 공유할 수 있게 되고 이로 인하여 빠르고 유연한 개발이 가능하다. 셋째, 오픈소스는 대체로 표준기준을 따르기 때문에 호환성이 좋아 여러 프로그램을 연동하는데 이점이 있다. 때문에 본 연구에서는 오픈소스 프로그램을 사용하여 자율주행을 구현하고 장애물이 있는 시뮬레이션 속을 주행하여 자율 주행 성능을 평가하고 보완점을 제시하려고 한다.

2. 오픈 소스 시뮬레이터

2.1 Carla

CARLA는 자율주행 시스템을 위해서 만들어진 시뮬레이터로 프로그램을 검증하고 개발, 교육에 사용할 수 있다. 또한 오픈소스 기반으로 되어 있으며 건물, 차량, 지형들을 구성 및 제작이 가능하도록 설계되어 있다. 또한 자동차에 들어가는 여러가지 센서, 기계, 카메라 등을 이용할 수 있다. CARLA는 Unreal Engine을 기반으로 하는 시뮬레이터로 사실적인 표현이 가능하다.[1]

자율 주행 오픈 소스 프로그램은 CARLA를 제외하고도 Pre Scan, GAZEBO, LGSVL 등이 있다. 각 프로그램의 특징은

다음과 같다. “PreScan은 자율 주행 자동차와 ADAS를 설계하기 위한 시뮬레이터 프레임워크를 제공한다. 제조업체가 자율 내비게이션 아키텍처를 검증할 수 있도록 하는 자동 트래픽 생성기를 제공하여 다양하고 사실적인 환경과 교통 조건을 제공한다. 이 시뮬레이터는 또한 실제 어플리케이션에서 사용되는 전자 제어 장치(ECU)를 평가하는 데 매우 일반적인 HIL 시뮬레이션을 지원한다.[2]” PreScan은 MATLAB, Simulink를 기반으로 움직인다. GAZEBO는 ROS를 기반으로 하는 오픈소스 다중 로봇 3D 시뮬레이터이다. 3D 장면을 실내 및 실외 환경에서 재현을 할 수 있다. GAZEBO는 ODE(Open Dynamic Engine)를 물리 엔진으로 사용한다. “LGSVL(LG Electronics America R&D Center)는 다중 로봇 시뮬레이션에 중점을 둔 자율주행 기술 테스트를 위한 최신 시뮬레이터이다. Unity 게임 엔진을 기반으로 하며, 시뮬레이터 백본과 자율 주행 스택 간에 메시지를 전달하기 위한 다양한 브리지를 제공한다. LGSVL은 CARLA 시뮬레이터와 유사한 방식으로 기상 조건, 적의 위치 등과 같은 다양한 환경 엔터티를 제어하는 PythonAPI를 제공한다. 올바른 시뮬레이터를 선택하기 위해서는 인식(센서), 멀티뷰 지오메트리, 교통 인프라, 차량 제어, 교통 시나리오 시뮬레이션, 3D 가상 환경, 2D/3D 실측 자료, 서버 다중 클라이언트 아키텍처를 통한 확장성 등과 같이 우리의 목적에 가장 적합한 시뮬레이터를 식별하는 지표 역할을 할 수 있는 일련의 기준이 있다.[2]” PreScan은 사실적인 환경을 구축할 수 있도록 해준다. Gazebo는 로봇 시뮬레이 터로 상당히 인기가 있다. 하지만 두 프로그램은 복잡하고 장면을 만드는데 많은 시간과 노력이 필요하다. 때문에 자율 주행 기술 테스트에서 선호 되지 않는다. 그렇기 때문에 LGSVL과 CARLA 두 시뮬레이터가 유력하다. 하지만 LGSVL은 “멀티뷰 지오메트리 또는 SLAM(Simultaneous Localization and Mapping)을 수행하기 위한 카메라 보정을 제공하지 않는다[2]”는 점 때문에 CARLA를 선택하게 되었다.

2.2 Autoware

Autoware는 ROS를 기반으로 하는 오픈 소스 자율주행 소프트웨어로 주행, 탐지, 계획 제어, 지도 제작 등 자율주행에 필요한 다양한 것들을 제공한다.

“Autoware는 크게 Ego vehicle 위치 추정, 장애물 탐지, 경로 계획, 경로 추종, 이상 4가지 모듈들로 구성된다.

첫째, Ego vehicle 위치 추정은 GPS와 라이다 센서를 함께 사용함으로써 요구되는 정밀도를 달성한다. 즉, 위치 추정 모듈은 GPS로부터 차량의 위도값과 경도값을 수신하여 원점으로 삼고, 이 원점을 중심으로 하는 오차 반경의 원 안에서 차량의 실제 위치를 라이다 스캔 정보를 이용하여 확률적으로 추정한다. 둘째, 장애물 탐지는 객체 탐지와 객체 추적으로 구성된다. Ego vehicle 주변에 존재하는 장애물들의 위치와 크기를 파악해야만 뒤에서 설명할 주행 상태 기계가 주행을 위해 계획된 후보 궤적들 중에서 어느 궤적을 선택할지 결 정할 수 있다. 객체 탐지 모듈은 카메라 영상 처리 알고리즘에 의해 얻어진 이미지 객체들과 라이다 스캔들에 대한 점군 분할 알고리즘에 의해 얻어진 점군 객체들의 위치와 크기를 출력하며, 객체 추적 모듈은 칼만 필터 (Kalman filter) 알고리즘을 이용하여 이동 객체들의 위치와 속도를 함께 고려하여 다음 단계 위치를 추정한다. 셋째, 경로 계획은 전역적 계획과 지역적 계획으로 나뉜다. 전역적 계획은 벡터 지도, 시작 포즈, 목표 포즈를 입력 받아 주행 기준이 되는 기준 경로(reference path)를 계산한다.

마지막으로 경로 추종은 경로 계획 모듈이 제공한 최 종 궤적을 작은 시간 단위들로 나누어 차량이 안정적으 로 추종하기 위한 타겟 속도 v와 타겟 조향 각속도 w를 계산한다. 매 순간 타겟 속도는 사용자 입력 속도와 지 역적 경로 계획 알고리즘이 강제하는 가속, 등속, 감속 단계들에 의해서 결정된다. 한편, 타겟 조향 각속도는 타겟 속도를 고려하여 순수 추종(pure pursuit) 알고리에 의해서 계산된다. 최종적으로 (v, w)는 차량 제어 프로그램에 입력되어 차량 제어에 사용된다[3].”

2.3 ROS

ROS는 로봇 운영체제(Robot Operating System)의 약자로 이름과 같이 로봇과 관련된 소프트웨어 개발을 위한 오픈 소스 프로그램이다. ROS는 센서나 로봇과 같은 다양한 기기 간의 미들웨어로 사용되어 통신을 할 수 있게 해준다.

ROS는 ROS1과 ROS2로 나뉘어져 있는데 둘은 크게 마스터 노드의 유무로 차이가 난다. ROS1은 마스터 노드가 존재하여 모든 노드를 마스터 노드가 관리하지만 ROS2는 마스터 노드가 없으며 노드들간 DDS로 통신을 한다.

ROS는 여러가지 패키지 말고도 편의 기능을 제공하는데 ROS에는 사용하는 프로그램들을 시각화 하는 도구인 RViz와 ROS의 데이터를 효과적으로 이용할 수 있게 해주는 ROSBAG이 포함되어 있다.[5]

본 논문에서는 Carla와 Autoware를 Carla ROS Brdige를 이용하여 연결해 주는 정도로만 사용된다.

2.4 Carla ROS Bridge

CARLA 시뮬레이터와 Autoware은 각각의 Message Type이 서로 다르기 때문에 두 프로그램이 연결되기 위해서는 publish된 Topic을 연결해주는 프로그램이 추가로 필요하며 그 역할을 ROS Bridge가 해주게 된다. CARLA와 Autoware가 같은 ROS Domain에 연결이 되어 있다면 두 프로그램이 ROS를 통해서 상호작용할 수 있게 된다. 예를 들어 CARLA 시뮬레이터에서 건물과 지형, 차량 및 센서를 생성하고 ROS Bridge를 통해서 ROS Node에 보내지게 된다. 보내진 데이터는 ROS Bridge가 Autoware의 Message Type으로 변환을 해준다. 때문에 ROS Bridge는 Carla와 ROS계열 프로그램을 사용하기 위한 필수 조건이라고 볼 수 있다[4].

3. 자율 주행 구현 및 검증

3.1 구조

본 실험에서는 Carla라는 시뮬레이션 툴로 가상환경을 구현한 후에 Autoware를 이용하여 차량을 제어하는 것을

목표로 한다. Autoware를 Carla를 통해서 제어 할 수 있는 이유는 Autoware가 ROS를 기반으로 되어있으며 Carla는 ROS와 ROS bridge를 통해서 연결이 되기 때문이다. 구조를 그림으로 설명하자면 다음과 같다.

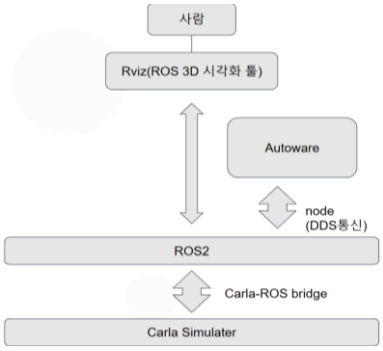


그림 1. 프로그램 구성도

3.2 OSM

Autoware는 OSM 파일을 이용하여 차선, 정지선, 주행 유도선 등의 정보를 가지고 있다. 때문에 Autoware는 vector map builder라는 툴을 이용해서 차선의 성질을 부여해 주어야 한다. 요번 실험에서는 lane change 기능을 사용해야 함으로 각 lane을 차선 변경이 가능하도록 성질을 부여해 주어야 한다.

“이 때 주행 유도선으로 사용될 경로점들은 다음과 같은 요구사항들을 만족해야 한다. 첫째, 모든 경로점들은 등간격(약 1 미터)으로 배치되어야 한다. 둘째, 차량이 교차로를 건너가거나, 차 로 변경을 수행하기 위해서는 하나의 경로점이 2개 이상의 다른 경로점들로 분기하도록 미리 연결 링크가 만들어져 있어야 한다. 셋째, 곡선로 상에 위치하는 경로점들은 차량의 조향휠이 급격하게 회전하는 일이 없도록 완만하게 변화하는 곡률을 가져야 한다. 이러한 요구사항들은 Autoware의 전역적 경로 계획 알고리즘이 목적지까지의 최단 경로를 탐색하기 위해 서, 그리고 주행 궤적을 생성하는 지역적 경로 계획 알고리즘이 곡률 변화가 작은 궤적들을 생성하기 위해서 필요하다[7].”

3.3 장애물 구현

장애물을 구현하는 것은 크게 4가지 방법이 있다. Unreal Engine, Roadrunner Scenario, Scenario Runner, Carla Python API이다. 이 프로그램들은 다른 장단점을 가지고 있다. Unreal Engine은 UI를 가지고 있으며 많은 기능을 지원하며 Carla 자체가 Unreal Engine으로 실행이 가능하기 때문에 추가적인 연동을 시킬 필요가 없다, 하지만 Unreal Engine은 상대적으로 매우 무거운 프로그램이며 기술 난이도도 높다는 단점이 있다. Roadrunner Scenario는 간편한 UI에 프로그램도 가볍고 다루기도 쉽지만 파일 양식을 바꿔야 한다는 단점이 있다. Scenario runner는 python을 이용하기 때문에 호환성이 좋고 가볍지만 UI가 없어서 직접 코딩을 해줘야 한다는 단점이 있다. 마지막으로 Carla PythonAPI는 Carla 내부 python example을 이용하는 것으로 가볍고 빠르게 사용 가능하지만 기능이 적고 직접 조종해야 한다는 단점이 있다. 나는 위와 같은 4가지 방식 중에 Unreal Engine과 Carla PythonAPI를 이용하여 장애물을 생성하였다,

3.4 프로그램 버전

본 실험에서 사용한 컴퓨터 및 프로그램은 다음과 같다. S/W – Ubuntu 20.04, Carla 0.9.15, ROS2 Galactic, Autoware universe, H/W – 인텔 13세대 i7, RAM DDR5 64GB, Nvidia RTX 4090

4. 실험 결과

4.1 자전거 인식

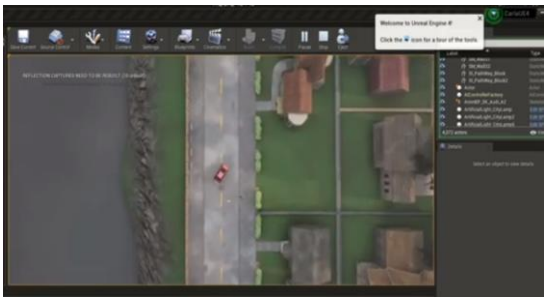


그림 2. 자전거 인식 실행 화면 캡쳐

2차선 도로에 자전거를 두고 Ego vehicle이 정지 할 수 있는지 관측하였다. 회피가 아닌 정지로 한 이유는 대부분에 경우에서 자전거는 횡단보도에서 마주치는 경우가 훨씬 많고 사람이 운전하기 때문에 동작 예측이 힘들어 회피보다는 정지가 더 맞다고 생각했기 때문이다. 그 결과는 차량은 회피하지 못하였고 부딪혀서 튕겨져 나간 후에도 다시 움직이다 가 부딪혀서 2차 충돌을 해버렸다. 진행이 불가능 했을 때 (충돌, 정지) 대응을 할 수 있는 능력이 없다는 것을 확인 할 수 있었다. 앞에 공간이 부족해서 정지되는 경우라면 후진하는 방법을 생각 할 수 있어야 하며 충돌한 경우라면 사고가 커지지 않도록 차량을 정지해야 하지만 목표 지점을 통해 전진하며 방향만 바꿀 수 있도록 설계되어 있었다.

4.2 보행자 인식

그림 3. 보행자 인식 실행 화면 캡쳐

보행자를 2차선 도로에 두고 Ego vehicle이 정지 할 수 있는지 관측하였다. 결과는 자전거와 마찬가지로 충돌하는 결과를 낳았다. 사람이나 자전거와 같은 작은 물체는 Lidar로 인식이 안되는 문제점이 있다. 이를 해결하기 위해서는 카메라 같은 기능을 사용해서 추가적으로 사물 인식을 해야만 사람과 자전거 인식이 가능 할 것이다.

4.3 일반주행

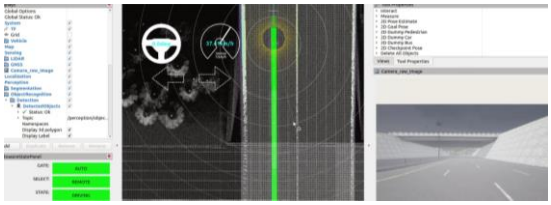


그림 4. 일반 주행 실행 화면 캡쳐

목표 지점을 선택하고 이탈, 정지 없이 주행을 하는지 관측한다. 이 실험은 Autoware가 잘 연동되었는지 확인하기 위해 관측하였다. 목표 지점을 정해주면 경로 계산을 하여 주행하는 방식이며 정상적으로 주행 성공했다.

4.4 주행방해

그림 5. 주행방해 실행 화면 캡쳐

차량이 앞을 가로지르며 급정거를 하게 한 후에도 다시 프로그램이 회복되어 골지점까지 가는지를 관측했다. 옆 차선에서 끼어든 들기를 반복하여도 목표지점까지 도달하는 것을 볼 수 있다.

4.5 끼어들기

그림 6. 차량접근 실행 화면 캡쳐

일반 주행과 같이 Ego vehicle을 전진시킨 후에 Python API를 통해서 차량을 생성한 후 빠른 속도로 앞을 가로질렀다. Ego vehicle는 이에 대해 곧바로 정확히 측정하였으며 급정거하여 충돌도 하지 않았다.

4.6 차량 회피 및 차선 변경

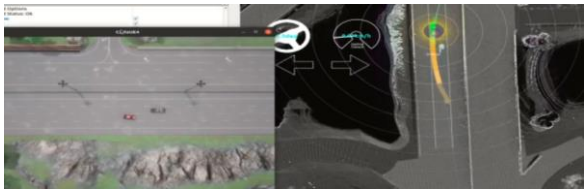


그림 7. 차량 회피 및 차선 변경 실행 화면 캡쳐

Ego vehicle 앞에 장애물을 세운 후 Autoware가 회피 할 수 있는지에 대해 관측했다. Autoware는 장애물을 인식한 후에 계속해서 자신의 궤도를 수정해가면서 목표지점에 도달하였다.

4.7 다수의 차량 회피

4.7.1 실험결과

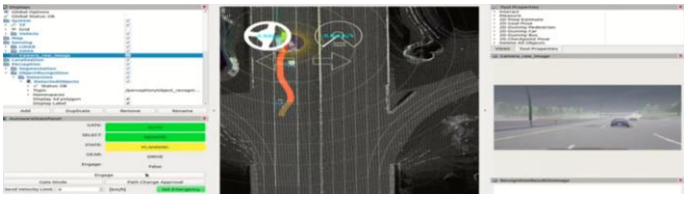


그림 8. 다수의 차량 회피 실행 화면 캡쳐

2개의 차선에 5대의 차량을 교차로 둔다. 단 차량 폭을 점점 갈수록 줄인다. 이 때 Ego vehicle이 5대에 차량을 지그재그로 통과 할 수 있는지 관측하였다. 차량의 속도는 5km/h 장애물 역할을 하는 차량들의 간격은 19m로 하였다. 결과는 첫 차량을 회피하는 것 외에는 전부 실패하였다. 처음에는 잘 회피하는 듯 했지만 동시에 두개의 장애물이 나타나자 경로 수정이 안되기 시작하였고 결국 멈춰 버렸다. 다시 목표 지점을 선택해주자 두번째 차량을 넘어가는 듯 싶었지만 다시 3번째 차량에서 멈추어 버렸다. 그 이후로는 차가 경로를 이탈하면서 물리적으로 움직일 수 없게 되어버렸고 그대로 시스템이 멈추어 버렸다.

4.7.2 다수의 차량 회피에 대한 추가적 실험

발생한 문제는 3가지이다. 첫번째로 프로그램상 충분히 회피 할 수 있음에도 경로를 수정하지 않고 멈추는 문제, 두번째로 목표 지점을 재 지정하면 일정 부분 정상가동이 되는 문제, 세번째로 차가 이동 할 수 없는 경로로 이탈하면서 정지하는 문제이다. 위에 문제가 생기는 정확한 원인을 분석하기 위해 다수의 차량 회피에 대한 실험을 추가적으로 진행하였다.

4.7.2.1 목표 지점 분리 실험

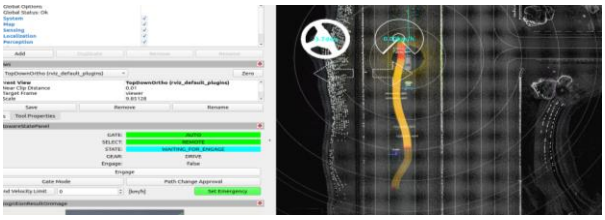


그림 9. 목표 지점 세분화 실험 캡쳐

첫번째 실험은 목표 지점 분리이다. 원래 목표지점(5번째 차량 뒤쪽)에 지정하는 것이 아니라 목표 지점을 여러 개로 분리해서 목표지점에 들어가게 하는 실험이다. 차량의 속도는 5km/h 장애물 역할을 하는 차량들의 간격은 19m로 하였다. 이 실험의 목표는 목표 지점과 자율주행 경로 계산의 연관성을 보기 위해서 진행하였다.

골 지점	1,2,3,4,5번 차 뒤	2,4,5번 차 뒤	3,5번 차 뒤	4번 차 뒤
성공 여부	성공	실패	실패	실패

표1. 목표 지점과의 관계 실험 결과 표

1,2,3,4,5번 차량 뒤에 각각 목표 지점을 선택해 주어야지만 성공적으로 주행이 가능했다.

4.7.2.2 속도, 장애물 차량의 간격의 변화 실험

두번째 실험은 속도와 차량의 간격에 변화에 따른 자율주행 성공 여부이다. 목표 지점은 5번째 차량 뒤로 두고 속도와 간격이 자율 주행에 어떠한 연관이 있는지 보기 위해서 진행하였다.

속도/간격	12m	19m	26m
2 km/h	성공	성공	성공
5 km/h	실패	성공	성공
10 km/h	실패	실패	성공

표2. 속도, 장애물 차량의 간격의 변화 실험 결과 표

주행 차량의 속도가 감소하거나 장애물 차량의 간격이 넓을수록 성공하였다.

4.7.2.3 장애물 차량의 위치 변경

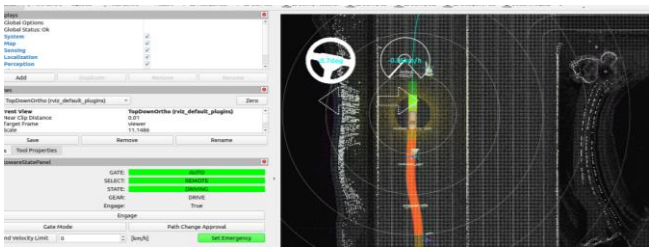


그림 10. 허용되지 않는 차선에 충돌한 차량 캡쳐

3번째 실험은 장애물 차량의 위치를 도로 중심 혹은 측면으로 변경 한 후에 본 실험(4.7)과 같은 실험을 진행하는 것이다. 본 실험(4.7)에서 경로를 이탈해서 허용되지 않는 차선에 충돌하는 경우가 관측되었다. 이 문제가 주행 차량과 장애물 차량의 좌, 우 위치에 영향을 받는다는 추측을 하였고 이를 증명하기 위해서 진행하였다.

시도 횟수/도로 위치	중심	측면
1차	실패	성공
2차	실패	성공
3차	실패	성공

표3. 차선 충돌 여부 실험 결과 표

주행 자체는 성공과 실패를 반복하였지만 차선 충돌은 장애물 차량의 위치를 측면으로 변경 한 후에 일어나지 않았다.

4.7.3 원인 분석

위에 3가지 실험과 Autoware university Documentary에 나오는 Avoidance design 부분을 통해서 원인을 분석해 보았다.

4.7.3.1 필수적인 여분 거리

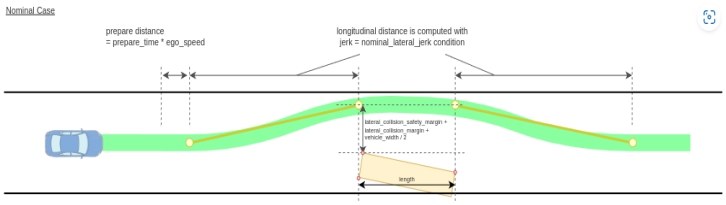


그림11. Autoware 장애물 회피 원리[8]

Autoware가 장애물을 회피하기 위해서는 prepare distance와 이동하기 위한 longitudinal distance가 필요하다. 하지만 여기서 prepare distance는 prepare time * ego_speed인데 이 시간을 확보하지 못하면 정지가 된다. 이 때문에 사람이 보았을 때 충분히 회피가 가능해 보이지만 앞에 차를 회피 할 때의 longitudinal distance와 prepare distance를 확보하지 못해서 정지해 버리는 것이다. 때문에 실험 4.7.2.1에서 목표지점을 분리하면 운행 도중 속도가 줄어서 preapare distance가 충분히 확보되기 때문에 성공하는 것이며 실험 4.7.2.2에서 속도가 느리고 장애물 차량의 간격 넓어야 prepare distance를 확보 할 수 있어 실험을 성공하는 것이다.

4.7.3.2 주행 차량의 center line

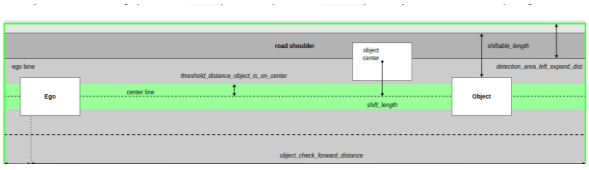
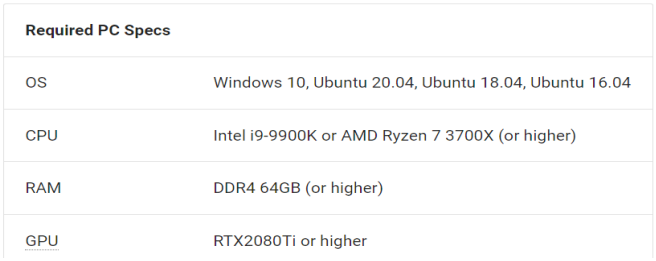


그림12. Autoware의 장애물 회피 판단[8]

Autoware는 주행차량 전방 중심에 center line을 설정하고 장애물의 위치와 center line을 비교하여 장애물이 피할 가치가 있는지와 어느 방향으로 회피해야 할지를 판단한다. 때문에 차량의 이동 혹은 장애물의 애매한 위치 등의 변수는 정확한 방향으로의 회피를 힘들게 만든다.

4.7.4 다수의 차량 회피 결론

다수의 차량 회피에서 Autoware의 많은 보완점들이 나왔다. 프로그램상 충분히 회피 할 수 있음에도 경로를 수정하지 않고 멈추는 문제, 목표 지점을 재 지정하면 일정 부분 정상가동이 되는 문제, 차가 이동 할 수 없는 경로로 이탈하면서 정지하는 문제가 있었다. 또한 처음부터 5개의 차량이 인식되는 것이 아니기 때문에 주행 도중에 계속해서 연산이 추가 되서 컴퓨터와 프로그램에 큰 부담을 줬다. 이는 실제 차량에서 상당히 심각한 문제가 될 수 있다. 실제 차량에서는 차량이 5대만 있을 리도 없고 정지한 상태로 존재하지도 않을 것이기 때문이다.



Required PC Specs	
OS	Windows 10, Ubuntu 20.04, Ubuntu 18.04, Ubuntu 16.04
CPU	Intel i9-9900K or AMD Ryzen 7 3700X (or higher)
RAM	DDR4 64GB (or higher)
GPU	RTX2080Ti or higher

그림 13. Autoware 성능 요구[8]

위 사진은 Autoware의 요구 성능 표이다. Autoware는 그냥 실행시키는 것만으로 큰 컴퓨터 자원이 필요하며 Autoware가 실제 차량에 들어간다면 다수의 차량 회피와 같이 많은 연산이 필요한 상황에서 차량에 삽입된 컴퓨터에 많은 부담을 줘서 신뢰성을 해칠 수 있으며 이에 관한 문제 해결이 필요할 것이다.

4.8 결과 정리

실험 번호	실험 목록	1차	2차
4.1	자전거 인식	실패(충돌)	실패(충돌)
4.2	보행자 인식	실패(충돌)	실패(충돌)
4.3	일반주행	주행 성공	주행 성공
4.4	주행방해	정지 성공	정지 성공
4.5	끼어들기	정지 성공	정지 성공
4.6	차량 회피 및 차선 변경	회피 성공	회피 성공
4.7	다수의 차량 회피	실패(정지)	실패(정지)

표 4. 실험 결과 정리

5. 결 론

본 연구에서는 오픈소스 시뮬레이터를 기반으로 자율주행 구현 및 검증을 하였다. Lidar를 통한 자율주행에서 자전거와 보행자에 경우 아예 인지를 하지 못하였으며 실패 후 대응에도 문제가 존재하였다. 장애물이 차량인 경우에는 단일 장애물 상황에서는 높은 신뢰도를 보여 주었지만 다수의 차량 회피는 실패했으며 실패 원인에 대해 Autoware Document와 추가 실험을 통해 기술적으로 분석하였다. 또한 H/W적으로도 무거운 프로그램으로 인해서 신뢰도가 의심되기도 하였다. 본 논문을 통해 확인한 문제점을 통해서 현재 프로그램의 한계를 알 수 있었고 추후 Autoware 장애물 회피 관련 연구에 참고될 것이라고 예상된다.

6. 사사 문구

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1A5A8026986).

7. 참고 문헌

[1] 이흥구 등 2명, “가상시뮬레이터 CARLA 이용한 AUTOWARE 연동 및 모델 예측 제어”, 한국 차세대 컴퓨팅 학회 춘계학술대회, 2021

[2]Carlos Gómez-Huélamo1·Javier Del Egido1· Luis M. Bergasa1·Rafael Barea1·Elena López-Guillén1·Felipe Arango1· Javier Araluce1 · Joaquín López, “Train here, drive there: ROS based end-to-end autonomous-driving pipeline validation in CARLA simulator using the NHTSA typology”, Multimedia Tools and Applications, 2022

[3] 이효은 등 2명, “자율주행 소프트웨어 Autoware의 실행 환경 분석”, 한국정보과학회 학술발표논문집, 2018

[4] 김두엽 등 3명, “CARLA와 Autoware를 연동하는 ROS2 Package 개발”, 제어로봇시스템학회 국내학술 대회 , 2023

[5] 이화성, 한창진, 박무성, 유찬곤, “ROS 기반 취약점 분석 기술 조사”. 한국정보기술학회 하계 종합학술대회 논문집, 2023

[6] S. Kato et al., "Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems", 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, 2018, pp. 287-296, doi: 10.1109/ICCPS.2018.00035.

[7] 최규진 등 5명, “오픈 소스 자율주행 플랫폼 Autoware 이해와 이식”, 정보과학회지, 2019

[8] Autoware Universe Documentation, https://autowarefoundation.github.io/autoware.universe/main/

[9]Carla Documentation, https://carla.org

[10]ROS2 galactic Documentation, https://docs.ros.org/en/galactic/index.html

[11]Hatem Github, https://github.com/hatem-darweesh

KNN 기반의 가짜 리뷰 계정 분류 모델 및 서비스 개발

한철현 ○, 권세빈, 박상근
경희대학교 소프트웨어융합학과
hch2454@khu.ac.kr, sebin4548@khu.ac.kr, sk.park@khu.ac.kr

Developing an Account-Level Fake Reviewer Detection Model and Service using KNN

Cheolhyeon Han○, Sebin Kwon, Sangkeun Park
Department of Software Convergence, Kyung Hee University

요 약

온라인 쇼핑에서 리뷰는 소비자가 수많은 제품을 선택하는 데 중요한 기준이다. 많은 업체들이 리뷰를 관리하기 시작했으며, 리뷰를 전문적으로 조작하는 업체도 등장했다. 가짜 리뷰를 구분하기 위한 여러 연구가 진행되었지만, 일반 리뷰의 특성을 파악하고 이를 따라한 가짜 리뷰 분류의 어려움 등의 한계점이 존재한다. 본 연구에서는 이런 한계점을 극복하기 위해 가짜 리뷰를 작성하는 계정을 식별하는 알고리즘을 개발한다. 이를 위해, 쿠팡에서 가짜 리뷰 계정과 일반 리뷰 계정을 수집하고 두 계정 간 유의미한 차이가 있는 다섯 가지 속성을 찾아냈다. 이 다섯 가지 속성을 기반으로 가짜 리뷰 계정과 일반 리뷰 계정을 분류할 수 있는 KNN 분류 모델을 개발했다. 그리고 사용자가 실제 쇼핑몰 웹사이트에 접속했을 때 이 분류 모델을 사용할 수 있도록 크롬 확장 플러그인을 제작하여 가짜 리뷰 계정 분류 모델의 활용 가능성을 확인했다.

1. 서 론

온라인 쇼핑몰에서 리뷰는 소비자가 상품을 선택하는 중요한 판단 기준이다[1]. 이는 사회적 디폴트 효과로 인해 특정 선호를 정해두지 않은 상황에서 다른 사람의 선택을 따라가는 경향이 높기 때문이라고 볼 수 있다[2]. 따라서 리뷰의 개수와 담긴 정보에 따라 상품의 판매량이 영향을 받을 수 있으므로 많은 판매자가 리뷰 관리에 각별한 신경을 쓰고 있다. 이렇게 리뷰의 중요성이 커지자, 돈을 받고 리뷰를 전문적으로 관리하는 업체가 등장하였고, 이 업체들로 인한 가짜 리뷰가 사회적 문제로 떠오르고 있다.

이런 조작된 가짜 리뷰는 소비자의 잘못된 판단을 유도하기 때문에 경쟁 질서를 해치는 악의적인 행위이다. He et al.[3]은 가짜 리뷰가 제품의 평점 및 순위에 영향을 줄 수 있음을 밝혔다. 가짜 리뷰가 양산되면 크게 두 가지 문제가 발생한다. 첫 번째로, 소비자들은 상품에 등록된 리뷰가 정직하게 작성된 리뷰인지, 돈을 받고 가짜로 작성된 리뷰인지 알 수 없어서 리뷰의 신뢰도가 낮아진다. 두 번째로, 돈을 받고 가짜 리뷰를 작성해 주는 업체를 이용하는 판매자가 많아질수록, 리뷰를 조작하지 않는 상품 판매자가 검색 순위에서 밀리는 등의 피해를 보게 된다.

가짜 리뷰를 식별하기 위한 다양한 연구가 진행되었다. 각각의 리뷰 텍스트를 분석하여 가짜 리뷰인지 식별하는 방법[4, 6]뿐 아니라 리뷰의 길이, 리뷰 작성 날짜 등 다양한 데이터를 활용한 가짜 리뷰 식별 연구[5, 8] 등이 있다. 이런 방법을 통해 높은 정확도로 가짜 리뷰를 식별하는 것은 가능하지만, 기존 연구에서 제시한 일반 리뷰와 가짜 리뷰를 구분하는 특징을 찾아낸 다음에 일반 리뷰의 특징을 흉내 낸 가짜 리뷰를 작성하면 구분하기가 어려워진다는 한계가 존재한다.

이 한계를 극복하기 위해, 본 논문에서는 가짜 리뷰를 식별하는 대신 가짜 리뷰어를 식별해 내는 새로운 접근 방식을 제시한다. 이를 위해 가짜 리뷰어를 식별하기 위한 새로운 특징들을 찾아내고, 이를 기반으로 머신러닝 기법을 활용해 가짜 리뷰어와 일반 리뷰어를 구분한다. 나아가 실제 온라인 쇼핑몰에 접속했을 때, 사용자가 보는 제품의 리뷰 중에서 특정 리뷰는 가짜 리뷰를 주로 작성하는 가짜 리뷰어에 의해 작성된 리뷰임을 알려줄 수 있는 기능을 구현하여 소비자의 올바른 상품 판단을 돕고 리뷰 조작의 영향력을 감소시키는 서비스를 구현해 그 활용성을 확인한다.

2. 관련 연구

리뷰 텍스트 분석을 활용해서 가짜 리뷰를 찾아내기 위한 다양한 연구가 수행되었다. Alsubari et al. [4] 은 TF-IDF 와 N-gram 기법을 활용해서 호텔 리뷰 텍스트에서 특징을 추출하고, 여러 가지 머신러닝을 활용해 높은 정확도로 가짜 리뷰를 식별할 수 있음을 보였다. 강지우 등[6]은 한국어 형태소 분석 라이브러리를 활용해 음식점 리뷰 텍스트를 어간 단위로 재구성하고, 이를 기반으로 머신러닝을 활용해 가짜 리뷰를 판별했다.

가짜 리뷰를 판별하기 위해, 리뷰 텍스트 외에 다양한 특징을 활용하는 연구도 있다. 이민철 & 윤형식[8]은 블로그에서 가짜 리뷰(광고 포스팅)를 식별하기 위해 텍스트뿐 아니라 해당 포스트의 길이, 작성 날짜, 사용된 이미지 수 등의 다양한 데이터를 활용했다. 이를 통해, 텍스트 외의 다양한 데이터도 가짜 리뷰를 식별하는 것에 중요한 요인이 될 수 있음을 확인하였다. Mohawesh et al. [5] 또한 리뷰 텍스트의 특징뿐 아니라 리뷰 작성자의 행동 패턴(예: 긍정 리뷰 비율, 평균 리뷰 길이 등)을 활용해 딥러닝으로 가짜 리뷰를 식별할 수 있음을 보였다.

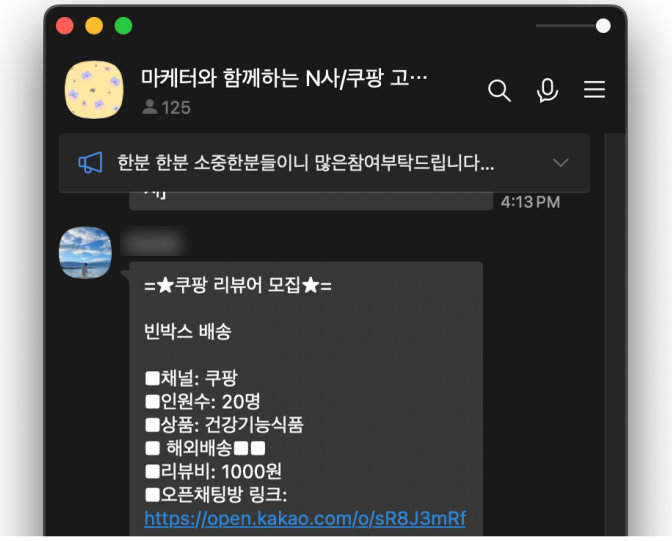
Patel & Patel [7]은 다양한 알고리즘을 비교 분석하면서 기존의 가짜 리뷰 식별 연구의 한계점을 지적했다. 가짜 리뷰와 일반 리뷰를 구분하고, 각 리뷰의 특징을 추출하여 학습시키는 방법으로는 일반 리뷰를 흉내 낸 가짜 리뷰를 식별하기가 매우 어렵다는 점이다. 예를 들어, 특정 가짜 리뷰가 일반 리뷰와 비슷한 길이로 작성되고, 비슷한 이미지 개수를 포함하며, 비슷한 단어를 사용하면 일반 리뷰인지 가짜 리뷰인지 식별이 어렵게 된다.

본 논문에서는 기존 연구에서 제시한 다양한 특징 추출 방법을 참고하되, 가짜 리뷰를 리뷰 단위로 분석하는 것이 아니라 가짜 리뷰를 작성하는 리뷰어 계정을 찾아내는 새로운 접근 방법을 제시한다. 리뷰 하나하나의 특징에 크게 의존하지 않기 때문에, 특정 리뷰가 일반 리뷰처럼 보인다고 하더라도 가짜 리뷰어의 리뷰라는 것을 알아내는 것 자체만으로 해당 리뷰를 의심할 수 있는 합리적 근거를 제시할 수 있다. 그뿐만 아니라, 가짜 리뷰어 식별 알고리즘을 활용하여 실제 온라인 쇼핑몰에서 작동하는 서비스를 구축했고, 이를 통해 이 알고리즘의 활용 가능성을 확인했다.

3. 학습 데이터 수집

3-1 가짜 리뷰 계정 수집

가짜 리뷰 계정과 일반 리뷰 계정을 분류하기 위한 데이터를 확보하기 위해, 가짜 리뷰어를 모집하는 카카오톡 오픈채팅방에 직접 접속해서 어떤 과정으로 가짜 리뷰가 양산되고 있는지 관찰했다 [그림 1]. 쿠팡에서 '빈 박스 배송'을 활용해 가짜 리뷰를 작성하는 경우가 많음을 확인했다. 빈 박스 배송이란, 가짜 리뷰 작성을 위해 고용된 가짜 리뷰어가 가짜 리뷰 작성 대상 제품을 구매하면, 판매자는 가짜 리뷰어 구매자에게 실제 제품을 배송하지 않고 빈 박스만 배송하는 것을 말한다. 판매자는 실제로 제품을 보내지 않고 빈 박스만 배송했지만, 쿠팡에는 마치 실제로 구매가 이뤄진 것으로 기록이 남는다. 그러면 빈



[그림 1]가짜 리뷰 작성자 모집 모습

박스를 배송받은 가짜 리뷰어는 마치 실제 제품을 구매한 것처럼 쿠팡에 가짜 리뷰를 작성할 수 있다.

가짜 리뷰어 모집 오픈채팅방을 통해 가짜 리뷰 작성 대상 제품 리스트 52 개와 제품별 빈 박스 배송 일정을 확인했다. 제품별 리뷰 중 빈 박스 배송 시작 이후 3 일 이내에 리뷰를 작성한 계정의 ID 를 모두 수집했다. 이렇게 수집한 리뷰어 ID 중 15 개 이상의 가짜 리뷰 작성 대상 제품에 리뷰를 작성한 ID 를 가짜 리뷰어 ID 로 가정하였다.

3-2 가짜 리뷰 계정 분류를 위한 가설 설정

특정 계정이 가짜 리뷰를 작성하는 계정인지 구분하기 위한 분류 모델을 학습하기 위해서는 가짜 리뷰 계정과 일반 계정의 차이를 구분할 수 있는 특징을 찾아야 한다. 가짜 리뷰 계정과 일반 계정을 분류하는데 적절한 데이터를 찾기 위해 다음 다섯 가지 가설을 선정했다.

a. **리뷰수**: 가짜 리뷰 계정은 일반 고객보다 리뷰를 더 많이 작성한다.

b. **성실도**: 가짜 리뷰 계정에서 구입한 제품은 성실한 리뷰의 비중이 높다. (성실한 리뷰: 쿠팡에서 별점 리뷰(1~5 점) 외에 별도로 "예상보다 맛있어요", "괜찮아요", "예상보다 맛없어요" 등을 선택하는 리뷰)

c. **쿠팡비율**: 가짜 리뷰 계정의 구매 제품은 로켓배송 등 판매자가 쿠팡(주)인 제품인 비율이 낮다. (가짜 리뷰는 빈 박스 배송을 통해 이뤄지는데, 로켓 배송 제품은 빈 박스 배송이 불가능하기 때문)

d. **글자수**: 가짜 리뷰 계정은 일반 고객보다 리뷰를 더 길게 작성한다.

e. **작성일수**: 가짜 리뷰 계정은 리뷰를 작성한 날짜가 많다. (리뷰 아르바이트라는 특성상 가짜 리뷰 작성자는 일반 고객보다 더 많은 리뷰를 작성한다고 가정)

3-3 가설 검증

2023 년 10 월 05 일부터 10 월 22 일까지 총 18 일 동안 가짜 리뷰 작성 중개인에게 안내받은 가짜 리뷰어 모집 제품 52 개의 쿠팡 URL 을 확보했다. 구글에서 만든 Node.js 라이브러리인 Puppeteer[1]를 활용해 해당 제품들의 쿠팡 제품 페이지에 접속한 후, 리뷰를 작성한 계정의 ID 를 모두 수집했다. 이 중 빈 박스 배송 시작 이후 3 일 이내에 리뷰를 작성한 379 개 계정을 확보하고 이를 가짜 리뷰 계정 ID 로 가정했다.

가짜 리뷰 계정이 아닌 일반 계정 확보를 위해, 쿠팡에서 애플(Apple)의 맥북 제품 판매 페이지에 접속하여 리뷰를 작성한 계정의 ID 380 개를 수집했다. 애플 같은 유명 제조사 들은 리뷰 조작을 하지 않고도 높은 품질과 평판이 보장되었기에 가짜 리뷰를 사용하지 않기 때문이다. 유명 제조사의 제품 리뷰 페이지 속에서 평점별로 동일한 수의 ID 를 랜덤 추출하여 일반 계정 380 개를 선정했다.

Puppeteer 를 다시 한번 활용해서, 수집된 모든 가짜 리뷰 계정과 일반 계정의 리뷰 이력 페이지에서 계정별로 '리뷰수', '성실도', '쿠팡비율', '글자수', '작성일수' 데이터[2]를 수집했다.

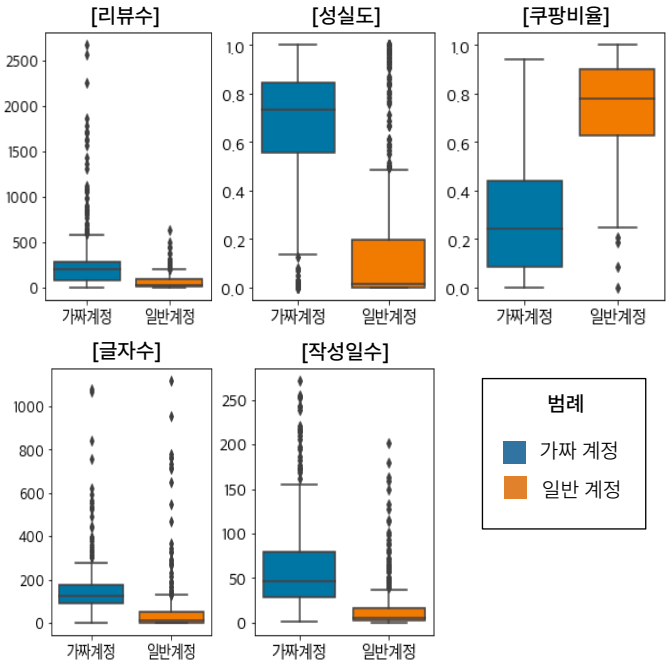
- 리뷰수 : 해당 계정이 작성한 리뷰 개수
- 성실도 : 전체 리뷰 중 '성실한 리뷰' 비율
- 쿠팡비율 :전체 리뷰 중 판매자가 쿠팡(주)로 표기된 제품에 작성한 리뷰의 비율
- 글자수 : 평균 리뷰 글자수
- 작성일수 : 리뷰를 작성된 날짜의 수

[표 1] 속성별 가짜/일반 계정의 평균 및 표준편차

	리뷰수	성실도	쿠팡비율	글자수	작성일수
가짜 계정	293 (SD=372)	0.66 (SD=0.25)	0.29 (SD=0.24)	153 (SD=127)	62 (SD=53)
일반 계정	69 (SD=94)	0.17 (SD=0.29)	0.75 (SD=0.20)	56 (SD=137)	16 (SD=28)

수집된 데이터를 기반으로, 가짜 리뷰 계정과 일반 계정의 차이를 확인하였다 [표 1]. 가짜 리뷰 계정과 일반 계정을 분류하는데 해당 데이터를 사용하는 것이 적절한지 확인하기 위해, 속성별로 t-test 를 수행했다. 모든 속성에서 p-value 가 유의수준(0.05) 보다 작음을 확인했으며, 이를 통해 각 속성의 분포는 가짜 리뷰 계정과 일반 계정 간 유의미한 차이가 있음을 알 수 있었다.

이렇게 통계적으로 검증된 일반 계정과 가짜 계정 간의 분포 차이를 시각화하면 가짜 리뷰 계정과 일반 계정 간에 차이가 있음을 한눈에 확인할 수 있다 [그림 2].



[그림 2] 5 가지 속성별 전체 데이터 Boxplot

4. 가짜 리뷰 계정 & 일반 계정 분류 모델

4-1 학습 데이터 전처리

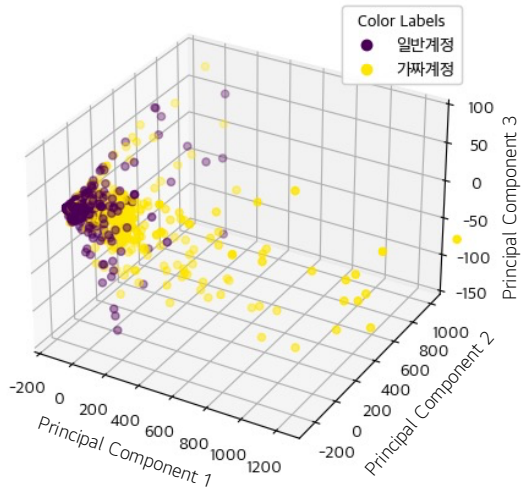
[표 2] 5 가지 속성 간 상관관계

	리뷰수	성실도	쿠팡비율	글자수	작성일수
리뷰수	1.00	0.38	-0.33	0.15	0.80
성실도	0.38	1.00	-0.57	0.56	0.53
쿠팡비율	-0.33	-0.57	1.00	-0.12	-0.36
글자수	0.15	0.56	-0.12	1.00	0.34
작성일수	0.80	0.53	-0.36	0.34	1.00

모델을 학습하기 전 다중공선성 문제를 확인하기 위해, 속성 간 상관관계를 계산했다 [표 2]. 리뷰수와 작성일수 사이에는 0.8 의 강한 상관관계가 있음을 확인했다. 강한 상관관계를 가진 속성 들은 다중공선성의 우려가 있기에, PCA 를 이용하여 차원을 축소했다. PCA 결과로 3 개의 주성분이 누적 분산 99.99% 로 설명할 수 있다는 결과를 확인할 수 있었다. [그림 3]은 PCA 를 수행한 후 일반 계정과 가짜 리뷰 계정 집단 간 주성분 분포를 확인한 결과이다. 두 집단이 비교적 명확하게 분리된 것을 확인할 수 있다.

[1] https://pptr.dev/

[2] https://github.com/festring/coupang_review_dataset



[그림 3] PCA 시각화

4-2 모델 구축 및 평가

앞에서 차원 축소를 통해 얻은 주성분을 기반으로, 가짜 리뷰 계정과 일반 계정을 분류하기 위해 노이즈에 강한 K-Nearest Neighbors (KNN) 알고리즘을 사용했다. K 값을 적절히 선택하여 노이즈에도 일관된 결괏값을 얻기 위해서이다. 데이터의 총개수가 적은 부분을 보완하기 위해 K-Fold 교차 검증을 사용했고, 데이터의 개수를 고려하여 Fold 값은 3 으로 설정했다. scikit-learn 1.3.2 [3] 의 KNeighborsClassifier 클래스와 cross_val_score 클래스를 활용했고, 파라미터는 모두 기본값으로 설정했다.

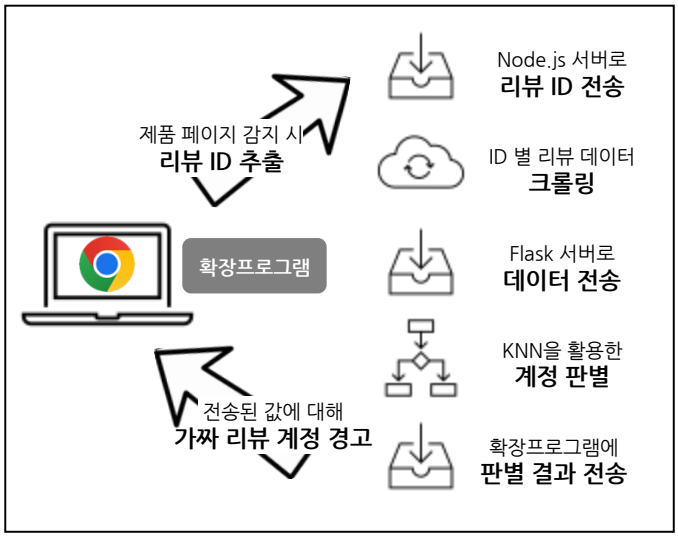
[표 3] K 값에 따른 모델의 3-fold 교차검증 결과

	k = 1	k = 3	k = 5	k = 7	k = 9
F1 Score	0.83 (SD=0.05)	0.86 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)
Recall	0.83 (SD=0.05)	0.86 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)
Precision	0.84 (SD=0.05)	0.86 (SD=0.02)	0.86 (SD=0.02)	0.85 (SD=0.02)	0.86 (SD=0.01)
Accuracy	0.83 (SD=0.05)	0.86 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)	0.85 (SD=0.02)

[표 3]은 홀수인 K 값에 따른 KNN 모델 평가 지표들이다. K 값이 1 일 때 값이 가장 낮고, 3 이상에서는 큰 변화 없이 일정한 값들이 나타난다. 이를 통해 현재 데이터셋에서 모델은 K 값이 3 이상 일 경우 안정적으로 작동함을 알 수 있다. 본 연구에서는 K 값을 7 로 설정하고 온라인 쇼핑몰 가짜 리뷰 계정 판별 서비스 구현에 활용하기로 결정했다.

5. 온라인 쇼핑몰 가짜 리뷰 계정 판별 서비스 개발

5-1 작동 구조



[그림 4] 가짜 리뷰 계정 판별 서비스 작동 구조

본 논문에서 구현한 가짜 리뷰 계정 판별 알고리즘을 사용자가 직접 사용해 볼 수 있는 서비스를 개발했다. 서비스의 작동 구조는 [그림 4]와 같다. JavaScript 를 기반으로 한 크롬 플러그인 형태로 개발하여 사용자가 간편하게 설치하고 사용할 수 있다. 사용자가 본 크롬 플러그인의 모드를 ON 으로 해놓고 쿠팡의 특정 제품 페이지에 접속하면, 해당 리뷰 페이지에 리뷰를 작성한 계정의 ID 가 모두 Node.js 로 구현한 서버로 전송된다. 전송된 ID 를 서버에서 각 리뷰 계정 ID 가 작성한 모든 리뷰를 수집한 다음 [표 1]에서 언급한 다섯 가지 속성을 Flask [4] 로 구현한 서버로 전송한다. 이 과정이 별도의 서버에서 이뤄지기 때문에, 사용자는 다른 방해를 받지 않고 계속 쇼핑을 진행할 수 있다.

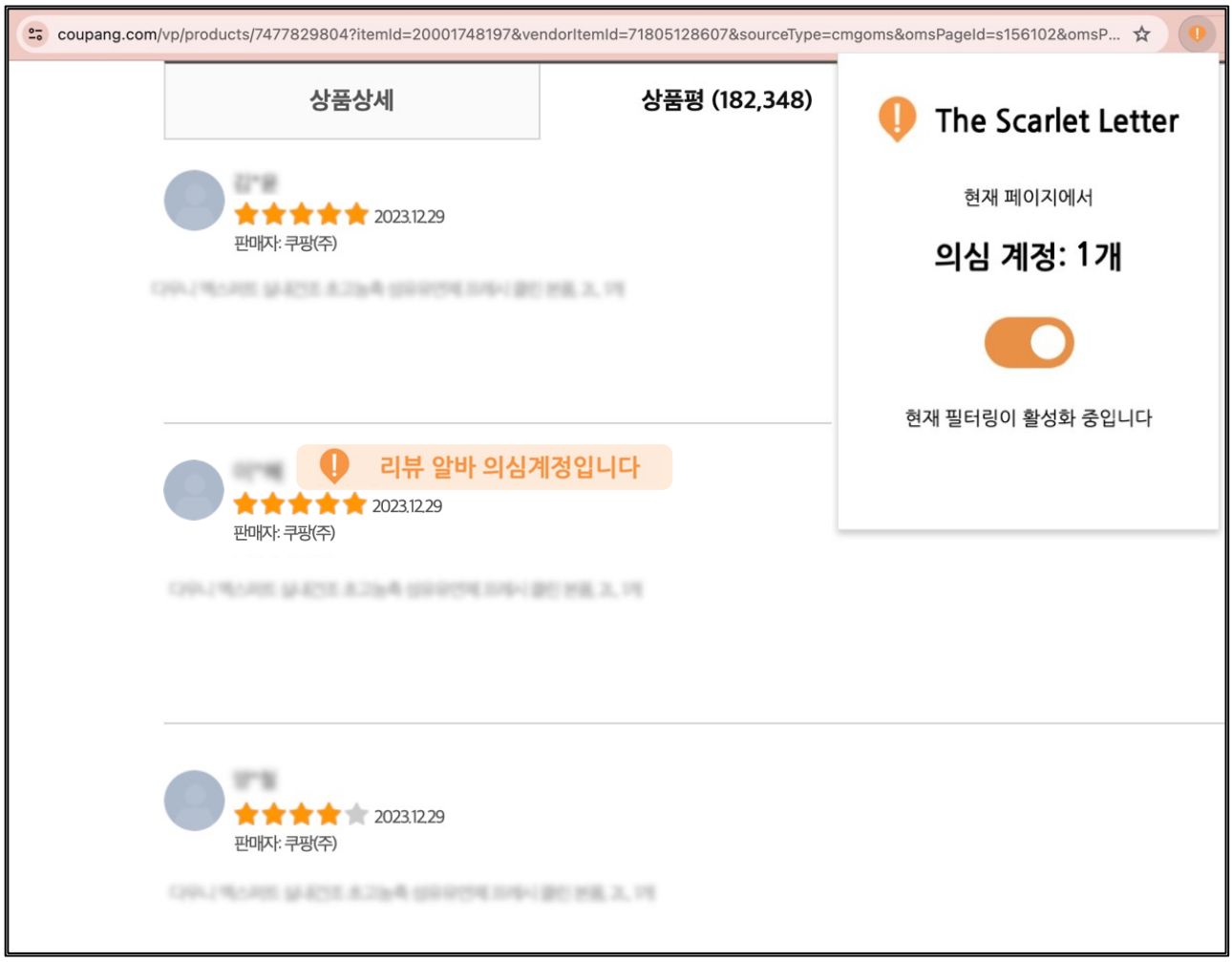
Flask 서버에서는 PCA 를 수행하여 속성을 3 개로 축소하고, 미리 학습된 KNN 모델에 입력하여 각 리뷰 계정 ID 가 가짜 리뷰 계정인지 일반 계정인지 분류한다. 분류된 결과는 다시 확장 플러그인에 전송되고, 확장 플러그인은 가짜 리뷰 계정으로 분류된 계정의 오른쪽에 경고 표시를 띄운다 [그림 5].

5-2 서비스 기능

최종 결과물의 핵심 기능으로 대시보드 기능과 의심 계정 표시 기능이 있다. 대시보드는 크롬 브라우저의 우측 상단에 존재하는 크롬 확장 플러그인 중 해당 플러그인 아이콘을 클릭하면 실행된다 [그림 5. 우측 상단]. 대시보드의 ON/OFF 기능으로 사용자가 원할 때만 가짜 리뷰 계정 경고 기능을 활성화/비활성화할 수 있다. 또한, 가짜 리뷰 계정으로 의심되는 계정이 몇 개인지 바로

[3] https://scikit-learn.org

[4] https://flask.palletsprojects.com/en/3.0.x/



[그림 5] 쿠팡 페이지에서 실제 작동 모습

표시함으로써, 현재 접속 중인 쿠팡 제품의 리뷰를 볼 때보다 주의를 기울일 수 있도록 구현하였다.

쿠팡 제품의 리뷰 페이지에서, 가짜 리뷰 계정으로 의심되는 계정에는 직접 HTML 요소를 삽입하여 “리뷰 알바 의심계정입니다”라는 문구를 표시한다 [그림 5. 가운데]. 이를 통해, 사용자는 가짜 리뷰 계정으로 추정되는 계정이 작성한 리뷰를 주의 깊게 살펴보면서 구매 여부를 결정하는 데 참고할 수 있다.

6. 결론 및 향후 연구

본 연구에서는 인터넷 쇼핑몰에 만연한 가짜 리뷰 문제를 해결하기 위해, 기존의 가짜 리뷰 판별이 아닌 가짜 리뷰 계정을 판별하는 머신러닝 모델을 개발하고, 이를 활용한 서비스를 직접 구현하여 그 활용 가능성을 확인했다. 본 연구의 아이디어를 확장하면, 가짜 리뷰 계정뿐 아니라 가짜 정보를 양산하는 계정을 찾아내는 등, 기존과 다른 방법으로 사용자의 정상적인 판단을 저해하는 거짓 정보를 차단하는 데 도움이 될 수 있다.

본 연구를 수행하기 위해서, 실제로 가짜 리뷰어를 모집하는 오픈채팅방에 접속하여 가짜 리뷰가 작성되는 제품 리스트를 확보하고, 이를 기반으로 가짜 리뷰의 특징 등을 선별했다. 하지만 여전히 가짜 리뷰라고 가정한 계정이 100% 가짜 리뷰 계정이라고 결정지을 수 없다는 한계가 존재한다. 이에 따라 생존 편향(Survivorship bias) 문제가 있을 수 있다. 향후 연구에서는 가짜 리뷰 계정과 일반 계정을 보다 정확하게 분류할 수 있는 기준을 마련하고, 적절한 K 값 설정을 통한 기존 KNN 알고리즘의 성능 향상 및 앙상블 등 다양한 기법을 활용해 모델의 정확성과 서비스의 활용성을 높이고자 한다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 2023년도 SW중심대학사업의 결과로 수행되었음" (2023-0-00042)

참고 문헌

[1] 김세라. 온라인 쇼핑 이용후기 '실제 구매에 큰 영향' 미쳐. 소비자 경제, https://www.dailycnc.com/news/articleView.html?idxno=209683, 2023.

[2] Young Eun Huh, Joachim Vosgerau, Carey K. Morewedge. Social Defaults: Observed Choices Become Choice Defaults. Journal of Consumer Research, 41, 3, 746-760, 2014.

[3] Sherry He, Brett Hollenbeck, Davide Proserpio. The Market for Fake Reviews. Marketing Science, 41, 5, 896-921, 2022.
[4] Saleh Nagi Alsubari, Sachin N. Deshmukh, Ahmed Abdullah Alqarni, Nizar Alsharif, Theyazn H. H. Aldhyani, Fawaz Waselallah Alsaade, Osamah I. Khalaf. Data Analytics for the Identification of Fake Reviews Using Supervised Learning, Computers, Materials & Continua, 70, 2, 3189-3204, 2022.
[5] Rami Mohawesh, Shuxiang Xu, Son N. Tran, Robert Ollington, Matthew Springer, Yaser Jararweh, Sumbal Maqsood. Fake Reviews Detection: A Survey, IEEE Access, 9, 65771-65802, 2021.
[6] 강지우, 김동욱, 송이현, 이석범, 이범진, 정윤경. 음식점 가짜 리뷰 판별을 위한 기계학습 방법 비교. 한국정보과학회 학술발표논문집, 1980-1982, 2017.
[7] N. A. Patel & R. Patel. A Survey on Fake Review Detection using Machine Learning Techniques, 2018 4th International Conference on Computing Communication and Automation (ICCCA), 1-6, 2018.
[8] 이민철 & 윤현식. 머신러닝을 활용한 가짜리뷰 탐지 연구: 사용자 행동 분석을 중심으로. 지식경영연구, 21, 3, 177-195, 2020.

가속도 센서를 이용한 CNN 기반의 스마트폰 터치 에러 감소 기법 개발

김은호 ○ 박상근
경희대학교 소프트웨어융합대학
taemin4u@khu.ac.kr, sk.park@khu.ac.kr

Development of a CNN-Based Smartphone Touch Error Reduction Technique Utilizing Acceleration Sensor

Eunnho Kim○ Sangkeun Park
Software Convergence, Kyung Hee University

요 약

스마트폰의 보급률이 높아지면서 사람들은 일상생활에서 여러 모바일 서비스를 편리하게 이용할 수 있게 되었다. 스마트폰 사용자는 정지된 상황뿐 아니라 이동 중에도 한 손으로 스마트폰을 조작하며 모바일 서비스를 이용하기도 한다. 이동 중에 스마트폰을 사용하면 사용자의 신체 흔들림 등으로 사용자의 한 손 터치 정확도가 떨어지는 불편함이 있다. 본 논문에서는 가속도 센서를 통해 측정되는 스마트폰의 기울어진 정도를 기반으로 사용자가 터치하려는 버튼을 예측하는 CNN 모델을 개발했다. 또한 예측된 버튼의 터치 허용 영역을 유연하게 확장하는 터치 에러 감소 기법을 제시하고 실험을 통해 그 유용성에 대하여 검증했다.

1. 서론

2022 년 기준 스마트폰 사용률이 97%를 기록할 정도로 많은 사람이 스마트폰을 사용하고 있다[1]. 스마트폰은 한 손으로도 간편하게 소지하고 다닐 수 있으므로, 출퇴근 등 이동 중에도 스마트폰을 사용하는 사람을 쉽게 찾아볼 수 있다.

스마트폰을 사용할 때는 버튼 클릭 등의 다양한 인터랙션이 필요하다. 이동 중에 스마트폰을 사용할 때는 신체의 흔들림이 발생하므로, 제자리에 멈춰서 스마트폰을 사용하는 것에 비해 원하는 부분을 정확하게 터치하지 못하는 상황이 생기기도 한다. 특히, 이동 중 스마트폰을 조작할 때 한 손으로 기기를 조작하는 경우가 많다. Karlson [2]은 모바일 기기 사용자 50 명의 사용 패턴을 조사한 결과, 74%가 한 손으로 기기를 주로 조작하고 그중 54%는 이동 중에 조작한다는 사실을 밝혀냈다.

이러한 한 손 조작은 스마트폰 터치 정확도에 부정적인 영향을 미친다[3]. 한 손으로 스마트폰을 다룰 때는 스마트폰을 안정적으로 잡기 위해 엄지손가락을 제외한 네 손가락으로 기기 뒷면을 받친 상태에서 엄지로 조작하는 경우가 많다. 그러나 한 손으로 스마트폰을 쥐고 있는 상태에서는 자유로운 손가락 움직임에 제약이 있어 터치 정확도가 떨어지게 된다. 오른손잡이를 기준으로, 스마트폰을 터치할 때 오른쪽 하단 모서리와 거리가 먼 곳일수록 터치하는데 소요되는 시간은 증가하고 터치 정확도는 떨어진다[4]. 심지어, 한 손 스마트폰 인터랙션 과정에서는 스마트폰을 떨어트릴 수 있는 위험성도 존재한다[5].

낮은 터치 정확도에도 불구하고 많은 사용자들이 한 손으로 모바일 기기를 사용하므로, 한 손으로 기기를 조작할 때의 터치 정확도를 향상하기 위한 다양한 연구가 수행되었다. 예를 들어, 모바일 기기 사용 시 버튼의 위치 및 크기와 터치 정확도의 관계 연구[6], 가속도 센서를 활용해 스마트폰 키보드 타이핑 속도와 정확도를 올리기 위한 연구[7] 등이 수행되었다. 그러나 정지 상태에서의 스마트폰 터치가 아닌 이동 중 한 손 스마트폰 사용에서의 터치 정확도에 대한 연구는 크게 주목받지 못했다.

이동 중에 한 손으로 스마트폰을 사용하게 되면, 터치하려는 대상(예: 버튼)을 정확하게 터치하지 못하고 인접한 영역을 터치하게 되는 경우가 자주 발생할 수 있다. 이때, 사용자가 터치하려는 대상을 예측할 수 있다면, 사용자가 정확하게 해당 대상을 터치하지 못하더라도 터치가 예상되는 대상의 터치 허용 영역을 유연하게 확장하여 터치 정확도를 높일 수 있다.

본 논문에서는 스마트폰에 내장된 가속도 센서를 활용해 사용자의 이동 중 터치 경향을 학습하고, 이를 기반으로 터치 대상을 예측하는 개인화 모델을 개발한다. 그리고 이 모델을 활용하여 터치 에러를 개선할 수 있는 터치 영역 확장 기법을 개발하고 실험을 통해 그 효과를 검증한다.

2. 관련 연구

대표적인 스마트폰 제조사인 삼성과 애플은 한 손으로 기기를 편하게 조작할 수 있는 한 손 모드 기능을 탑재하고 있다. 아이폰은 화면 상단 부분을 터치하기 쉽도록 화면을

아래로 옮길 수 있는 기능을 제공하며, 갤럭시 시리즈는 전체적인 화면을 축소하여 사용자가 한 손으로 쉽게 상호작용할 수 있도록 하는 기능을 제공한다. 또한, 두 제조사의 스마트폰은 자판을 오른쪽 또는 왼쪽으로 쏠리게 하여 한 손으로 원활하게 타이핑할 수 있도록 하는 방식도 제공하고 있다. 이렇게 한 손으로 기기 조작을 쉽게 하도록 하는 기능들은 엄지로 쉽게 도달할 수 없는 영역을 터치할 수 있게 해주지만 전체적인 사용자 인터페이스를 변경시킨다는 한계가 존재한다. 예를 들어, 전체 화면 크기가 축소되면 사용자의 스크롤 활동이 많아지게 되므로, 한 손으로 스마트폰을 조작하다가 스마트폰을 떨어뜨릴 우려가 있다[5].

그림 1. 데이터 수집을 위한 웹 애플리케이션
(클릭해야 하는 버튼을 1 초 동안 빨간색으로 표시)

한 손으로 스마트폰을 사용할 때, 터치 정확도를 올리기 위한 다양한 연구도 수행되었다. Park and Han[6]은 한 손 엄지손가락으로 터치할 때 버튼 크기와 위치에 따른 정확도의 차이를 밝혔다. 그 결과로, HP iPAQ rz1717 기기에서 최적의 버튼 크기는 10mm 이고 중앙에 버튼을 배치할 때 정확도가 가장 높음을 밝혔다. Weir et al.[8]는 사용자의 터치 행동을 반영하여 정확도를 향상하려는 연구를 수행했다. 사용자별 터치 정확도를 향상하기 위해 사용자의 터치 위치를 학습시키는 기계 학습 접근 방식을 제시하였으며, 실험을 통해 최대 23.47%만큼의 터치 정확도가 상승했음을 보였다. 그러나 [8]는 이동 중인 상황은 고려하지 않았다는 점에서 한계가 존재한다.

이동 중 스마트폰을 사용할 때 터치 정확도를 올리기 위한 선행 연구도 존재한다. Goel et al.[7]은 이동하면서 스마트폰을 사용하는 사람들의 키보드 타이핑 정확도를 향상하기 위해 가속도 센서의 값으로 정확도를 보정하는 모델을 제시하였다. 그러나 양손을 사용할 때를 기준으로 연구가 수행되었다는 한계가 존재한다.

이에 본 논문에서는 사용자가 이동 중 한 손으로 조작할 때 발생하는 시계열 가속도 센서 데이터 패턴을 학습하고, 이를 기반으로 사용자가 이동 중에 터치하려는 대상을 예측하는 개인화 모델을 구축한다. 그리고 이 모델을 활용해서 이동 중 터치 정확도 향상을 위한 터치 영역 확장 기법을 개발한다.

3. 터치 영역 예측을 위한 학습 데이터 수집

본 절에서는 사용자가 터치하려는 버튼을 예측하기 위해 학습 데이터 수집을 목적으로 한 실험에 대해 설명하고, 그 실험 결과를 시각화하여 이동 중 터치 정확도 향상 모델에 대한 필요성을 확인한다.

3.1 데이터 수집을 위한 모바일 웹 애플리케이션 개발

사용자의 터치 영역 예측에 사용하기 위한 학습 데이터를 수집하기 위해, 사용자의 스마트폰 터치 좌표와 가속도 센서값을 수집할 수 있는 모바일 웹 애플리케이션을 개발했다 [그림 1]. 애플리케이션을 처음 실행하면 9 개의 회색 버튼이 3x3 형태로 배치되어 있다. 9 개의 버튼이 정상적으로 로드되고 나면 9 개의 버튼 중 무작위로 선정된 버튼 하나의 색깔이 1 초 동안 빨간색으로 바뀐다. 이때 사용자가 빨간색 버튼을 터치하면 해당 버튼의 번호, 터치된 좌표, 터치 직전 0.2 초 동안의 가속도 센서 시계열 데이터가 데이터베이스에 저장된다. 이제 다시 남은 버튼 중 하나가 무작위로 빨간색으로 변하게 되고, 데이터가 저장되는 과정이 반복된다. 9 개의 모든 버튼이 모두 한 번씩 선정되면 데이터 수집 세션이 종료된다.

3.2 실험 방법

스마트폰을 5 년 이상 사용한 20 대 오른손잡이 남성 3 명을 모집했다. 참가자는 오른손으로 스마트폰을 쥐고, 엄지손가락만 사용하여 앞서 설명한 모바일 웹 애플리케이션을 통해 무작위 버튼 터치 미션을 수행하도록 했다. 단, 버튼 터치 정확도에 영향을 끼칠 수 있는 스마트폰의 크기, 무게 등의 외적 요인을 제거하기 위해 모든 참여자는 미리 준비된 스마트폰(아이폰 12, 71.5X146.7X7.4mm)으로 해당 모바일 웹 애플리케이션을 사용했다.

실험은 1) 의자에 앉은 상황(정지 모드)과 2) 복도에서 걷고 있는 상황(걷기 모드) 두 가지 모드로 진행되었다. 참가자는 각 모드에서 모바일 웹 애플리케이션에 대해 50 회의 세션을 수행하였다. 따라서 각 참가자는 총 900 회의 터치 미션을 수행하였다(1 세션(9 번 터치) X 50 회 X 2 개 모드). 단, 빨간색 버튼을 제시간에 터치하지 못한 경우도 있으며, 반대로 한 번에 2 회 이상 연속으로 터치한 경우도 있기 때문에, 참가자별 실제 기록된 터치 횟수는 900 회보다 조금 더 많았다. 실험에는 참가자마다 약 30 분의 시간이 소요되었다.

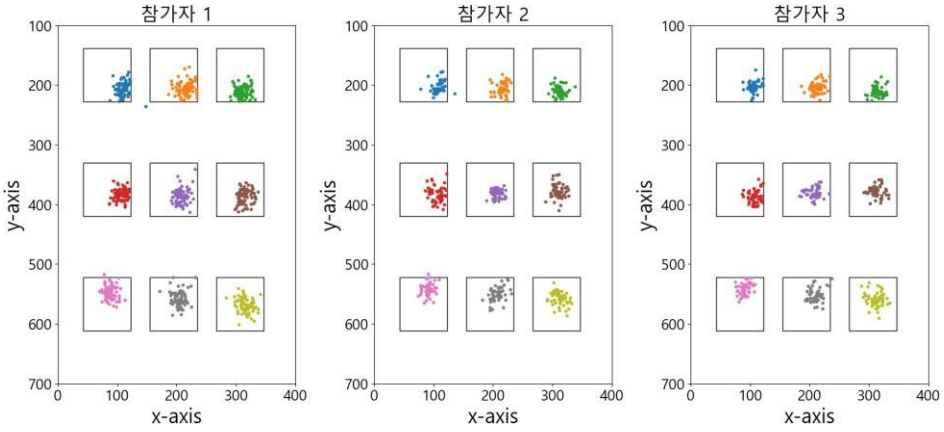


그림 2. 정지 모드 터치 좌표 분포

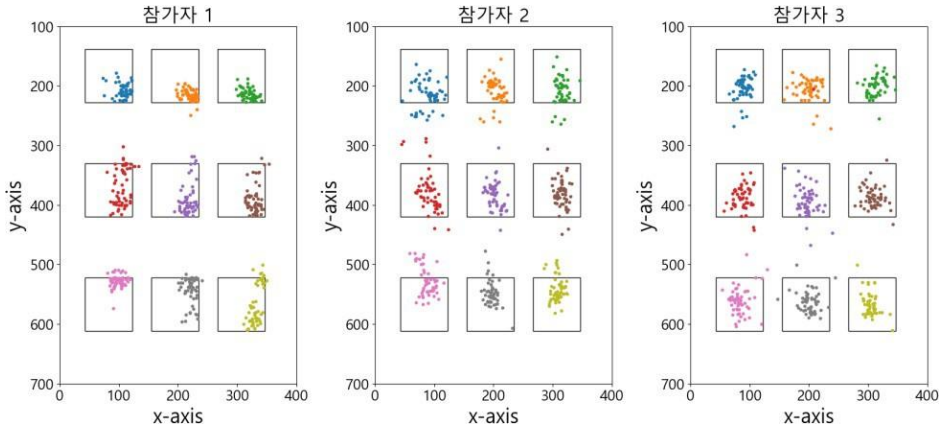


그림 3. 걷기 모드 터치 좌표 분포

표 1. 정지 모드 터치 에러 비율

	버튼1	버튼2	버튼3	버튼4	버튼5	버튼6	버튼7	버튼8	버튼9	Error
참가자 1	1/75	0/76	0/76	0/76	0/77	0/75	1/77	0/77	0/76	0.29%
참가자 2	1/48	0/47	0/51	0/51	0/50	0/51	1/49	0/50	0/51	0.45%
참가자 3	0/50	0/50	0/49	0/50	0/50	0/50	0/51	1/51	0/51	0.22%

표 2. 걷기 모드 터치 에러 비율

	버튼1	버튼2	버튼3	버튼4	버튼5	버튼6	버튼7	버튼8	버튼9	Error
참가자 1	0/53	2/57	0/55	5/56	3/55	2/55	8/57	2/56	6/56	5.60%
참가자 2	11/52	5/50	4/48	7/52	2/52	3/51	15/51	5/52	9/54	13.20%
참가자 3	4/56	3/53	1/51	2/55	3/55	2/53	2/56	3/54	1/53	4.32%

3.3 실험 결과

[그림 2]와 [그림 3]은 각각 참가자들이 앉아있는 상태(정지 모드)에서 실험을 진행했을 때의 좌표와 복도를 걷는 상태(걷기 모드)에서 실험을 진행했을 때의 터치 좌표 분포를 나타낸다. [표 1]은 정지 모드일 때, [표 2]는 걷기 모드일 때의 참가자별 버튼 터치 에러의 개수와 전체 에러 비율을 나타낸다. 정지 모드일 때는 터치 좌표의 분포가 균일하고 터치 에러의 비율도 매우 작지만, 걷기 모드일 때의 터치 좌표의 분포는 상대적으로 산발적이며 터치 에러의 비율도 더 높은 것을 확인할 수 있다.

4. 터치 에러 감소 기법 개발 및 유용성 검증

본 절에서는 실험에서 수집한 데이터를 기반으로 터치 버튼 예측 모델을 개발하고, 이를 활용해 터치 에러를 감소시킬 수 있는 방법에 대해 알아본다.

4.1 CNN 기반 터치 에러 감소 모델 및 기법 개발

본 논문의 목적은 이동 중 한 손 터치 정확도를 향상하는 모델을 개발하는 것이므로, 걷기 모드의 실험에서 수집된 버튼 번호, 터치 좌표, 버튼이 터치되기 직전부터 버튼이

터치될 때까지 0.2 초 동안의 시계열 가속도 센서 데이터를 사용해 개인화된 터치 버튼 예측 모델을 개발했다. 수집한 시계열 데이터 시간 간격이 매우 짧은 점을 고려하여 PyTorch 로 1D-CNN 모델을 구성하였다. 합성곱 계층의 입력으로는 배치(batch) 크기가 8 인 가속도 센서 세 축의 시계열 데이터를 사용하였다(채널 수 3, time step 15). 이때 합성곱 계층과 풀링 계층의 커널의 크기는 2 로 두었으며, 합성곱 계층의 연산 결과로 출력되는 데이터의 크기는 8X128X3 으로 설정했다(채널 수 128, time step 3). 각 출력층의 마지막 활성화 함수로 SoftMax 를 사용하여 버튼별 터치 확률을 구하였다. 참가자마다 개인화된 모델의 정확도를 확인하기 위해 예측 확률이 가장 높은 버튼의 번호와 터치된 실제 버튼 번호의 일치 여부를 계산한 결과, 참가자 1, 2, 3 에 대한 모델의 평균 정확도는 66.8%로 확인되었다.

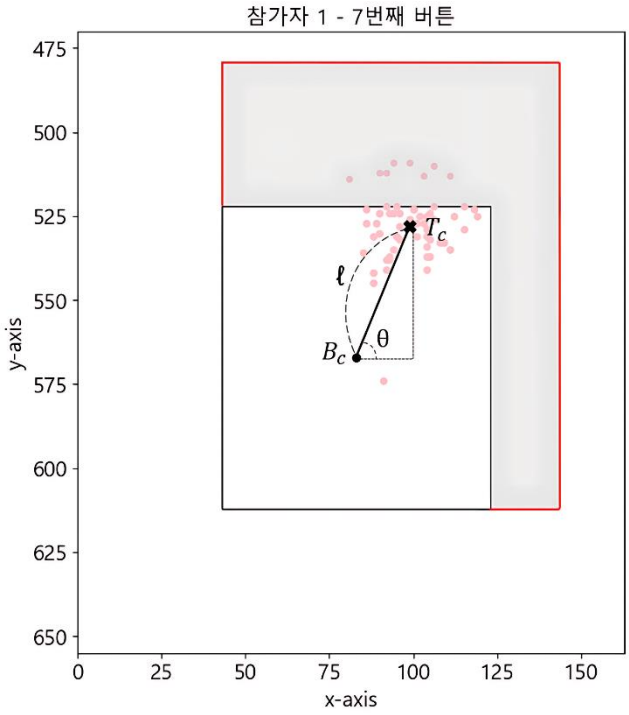


그림 4. 터치 에러 감소 기법

사용자마다 각 버튼에 대한 터치 경향이 모두 다르고, 한 손 엄지손가락으로 버튼을 터치할 때 필연적으로 기기를 기울이게 되므로 사용자가 기기를 기울이는 정도에 따라 각 버튼이 터치될 확률도 모두 다르다. 따라서 터치 에러를 줄이기 위해 터치 영역을 확장할 때 사용자의 터치 경향을 고려해야 한다.

본 연구에서 제안하는 터치 에러 감소 기법은 다음과 같다. 먼저 각 버튼의 중심점 B_c와 실험 3 에서 수집한 걷기 모드의 버튼별 터치 좌표의 중심점(centroid) T_c를 찾고, 이 둘 사이의 거리 ℓ 과 B_c 를 원점으로 두고 x 축, y 축을 설정하였을 때 B_c와 T_c가 이루는 각도 θ 를 계산한다. 그 후 B_c와 T_c의 가로 길이의 크기인 $\ell|\cos\theta|$와 세로 길이의 크기인 $\ell|\sin\theta|$을 구한다.

터치 허용 영역의 확장 방향을 결정하기 위해 B_c 를 기준으로 하여 버튼을 사사분면으로 나눈다(오른쪽 위, 왼쪽 위, 왼쪽 아래, 오른쪽 아래). T_c는 사용자가 그 버튼을 터치하기 위해 시도한 좌표 분포의 중심이므로 T_c가 위치한 사분면 방향으로 터치 허용 영역을 확장한다.

터치 허용 영역의 확장 크기를 결정하기 위해, 버튼별 터치 확률을 활용한다. 각 버튼의 터치 확률을 $\ell|\cos\theta|$와 $\ell|\sin\theta|$에 곱한 뒤, 그 값만큼 버튼의 기존 터치 허용 영역을 확장한다. 단, 터치 확률이 10% 이하인 버튼은 터치 허용 영역을 확장하지 않았다.

[그림 4]는 걷기 모드에서 참가자 1 의 7 번째 버튼에 대한 터치 좌표 분포와 그 중심점을 나타낸다. 터치 좌표의 중심점 T_c가 1 사분면에 위치하므로, 버튼을 터치할 확률이 p 일 때 터치 영역의 가로 방향에 대해서는 오른쪽으로 $p\ell|\cos\theta|$만큼 기존 터치 영역에서 더 확장하고, 세로 방향에 대해서는 위쪽으로 $p\ell|\sin\theta|$만큼 더 확장하여 그 방향이 1 사분면이 되도록 한다.

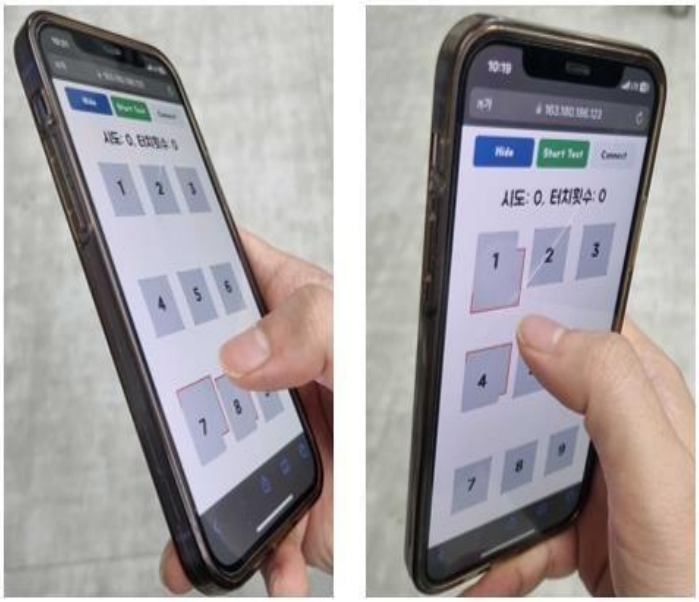


그림 5. 추가 실험을 위한 웹 애플리케이션 화면

4.2. 터치 에러 감소 기법 평가

CNN 기반의 모델 및 터치 에러 감소 기법의 성능을 평가하기 위해 사용자가 터치하려는 버튼을 예측하고 해당 버튼의 터치 허용 영역(빨간색 영역)을 순간적으로 확장하는 모바일 웹 애플리케이션을 개발했다[그림 5].

[그림 5]의 왼쪽 사진은 학습된 해당 사용자의 터치 패턴에 기반하여 현재 스마트폰의 기울임 정도가 7 번 버튼을 터치할 확률이 가장 높고, 8 번 버튼을 터치할 확률이 두 번째로 높은 순간임을 보여준다. 그리고 기존의 학습된 7 번 버튼 터치 좌표 분포에 따라, 해당 방향으로 7 번 버튼의 터치 허용 영역이 가장 많이 확장되었다. 8 번 버튼의 터치 허용 영역도 해당 버튼의 터치 확률에 비례하여 조금 확장되었다. [그림 5]의 오른쪽 사진은 해당 사용자가 1 번 버튼을 터치할 확률이 가장 높은 상황임을 모델이 예측하고, 1 번 버튼의 터치 영역을 순간적으로 확장한 상황이다.

해당 터치 에러 감소 기법의 효과를 확인하기 위해, 이전 실험에 참여했던 동일한 실험 참가자 1, 2, 3 을 대상으로 이전의 실험과 동일한 방법으로 실험을 한 번 더 진행했다.

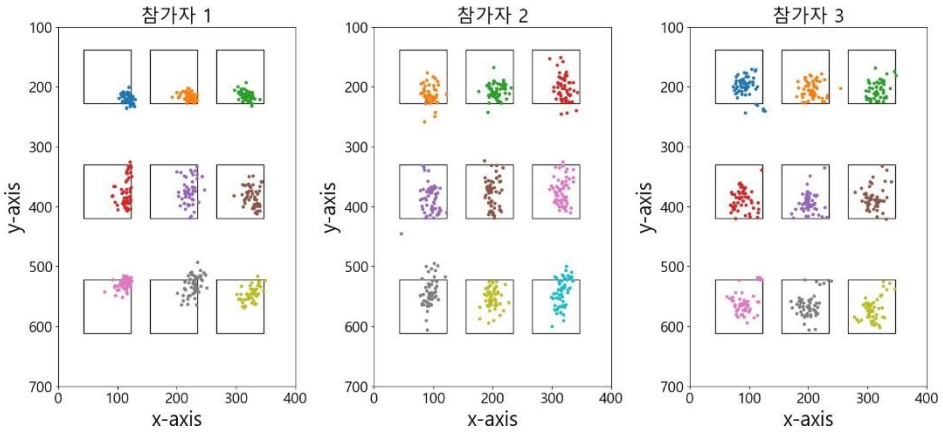


그림 6. 추가 실험 결과 터치 좌표 분포

표 3. 모델 적용을 하지 않았을 때의 터치 에러

	버튼1	버튼2	버튼3	버튼4	버튼5	버튼6	버튼7	버튼8	버튼9	Error
참가자 1	15/52	0/52	0/52	2/52	3/52	0/51	8/52	21/52	2/51	10.94%
참가자 2	3/52	1/53	3/53	1/55	1/56	1/58	9/54	0/53	13/56	6.04%
참가자 3	7/55	1/51	1/53	0/55	0/55	0/53	2/53	1/53	0/53	2.52%

표 4. 실제 터치 에러 비율

	버튼1	버튼2	버튼3	버튼4	버튼5	버튼6	버튼7	버튼8	버튼9	Error
참가자 1	4/52	0/52	0/52	1/52	1/52	0/51	3/52	2/52	2/51	2.79%
참가자 2	2/52	1/53	1/53	1/55	1/56	1/58	5/54	0/53	1/56	2.64%
참가자 3	1/55	1/51	0/53	0/55	0/55	0/53	0/53	1/53	0/53	0.63%

단, 정지 모드에서는 이미 터치 정확도가 매우 높음을 확인했으므로 이번에는 걷기 모드에서만 실험을 진행하였다. 그리고 사용자가 터치 영역이 확장된 것을 눈치채지 못하도록 [그림 5]에 보이는 확장된 터치 영역(빨간색 영역)은 보여주지 않았으며, 참가자에게도 터치 확장 영역에 대해 언급하지 않았다.

추가로 수행한 실험에서 참가자들의 버튼 터치 좌표 분포를 나타내면 [그림 6]과 같다. [그림 3]과 비슷한 패턴으로 터치 에러가 발생한 것을 확인할 수 있다. [표 3]은 터치 에러 감소 기법을 적용하지 않았을 때 고정된 버튼의 영역에 대한 터치 에러 개수 및 터치 에러의 비율을 나타내며 [표 4]는 터치 에러 감소 기법을 적용하여 터치 영역이 확장된 것을 감안한 터치 에러 개수 및 터치 에러의 비율을 나타낸다. 참가자 1, 2, 3 에 대해서 각각 8.15%, 3.4%, 1.89%만큼 터치 에러가 감소하였으며(평균 4.48%), 맥니마 검정 결과 모든 참가자에게서 유의미하게 터치 에러가 줄어들었음을 확인했다($p<.05$). 이를 통해 본 연구에서 제안한 터치 에러 감소 기법이 이동 중 발생하는 터치 에러를 감소시키는 데 효과적임을 확인할 수 있다.

5. 결론

본 논문에서는 직접 개발한 버튼 터치 기록 모바일 웹 애플리케이션을 활용해서 정지 상태와 이동 중일 때의 터치 좌표를 수집/분석하여 이동 중에 터치 에러가 더 많이 발생한다는 것을 확인했다. 이 문제를 해결하기 위해, 이동 중에 한 손으로 스마트폰을 다룰 때 발생할 수 있는 터치 에러를 감소시키기 위한 기법을 개발했다. 스마트폰에 내장된 가속도 센서를 활용해 버튼별 터치 확률을 예측하는 1D-CNN 모델을 만들고, 이를 활용해 버튼별 터치 확률을 가중치로 둔 터치 영역 확장 기법을 개발했다. 해당 기법의 유용성을 검증하기 위한 실험을 진행한 결과, 이동 중 터치 에러가 평균 4.48% 감소했음을 보였고 통계적으로 유의미함을 검증하였다.

6. 향후 연구

본 논문은 다음과 같은 한계를 가지고 있다. 첫째, 실험에 참여한 모든 사람에 대해 통계적으로 유의하게 터치 에러가 감소하였다는 것을 검증하였으나 그 피실험자의 수가 매우 작고, 에러 감소율이 낮아 그 결과가 객관적이라고 보기 힘들다. 둘째, 피실험자가 모두 20 대이고 남자인 점과 하나의 기기로만 실험을 진행한 점을 고려하였을 때 본 연구의 결과를 일반화하기에 어려움이 있다.

향후 연구로 다양한 연령, 성별 및 기타 사용자 특징을 고려하여 참가자를 모집하고 다양한 모바일 기기로 실험을 진행하여 결과의 일반화 가능성을 높일 계획이다. 또한 스마트폰에 내장된 여러 센서를 결합하고 시계열 데이터를 처리할 수 있는 다른 모델에 대해 탐구하면 터치 에러 감소 모델을 개선할 수 있으며, 실제 이용 중에 발생하는 여러 상황(터치 손가락의 변경, 우천 중 상호작용 등)에 대응할 수 있는 기법에 대해 연구할 계획이다.

사사의 글

“본 연구는 과학기술정보통신부 및 정보통신기획평가원의 2023 년도 SW 중심대학사업의 결과로 수행되었음”(2023-0-00042)

참고 문헌

[1] 김동석. “2022 년 한국 성인 97% ‘스마트폰 사용한다’” 뉴스인, 2022

[2] Karlson, A. K., “Interface and Interaction Design for One-Handed Mobile Computing.” Ph.D. Dissertation, University of Maryland, College Park, 2007.

[3] Ng, A., Brewster, S. A., & Williamson, J. H. “Investigating the Effects of Encumbrance on One-and Two-Handed Interactions with Mobile Devices.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1981-1990, 2014

[4] Trudeau, M. B., Asakawa, D. S., Jindrich, D. L., & Dennerlein, J. T. “Two-Handed Grip on a Mobile Phone Affords Greater Thumb Motor Performance, Decreased Variability, and a More Extended Thumb Posture Than a One-Handed Grip.” Applied Ergonomics, 52, pp. 24-28, 2016

[5] Le, H. V., Bader, P., Kosch, T., & Henze, N. “Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage.” In Proceedings of the 9th Nordic Conference on Human-Computer Interaction, pp. 1-10, 2016

[6] Park, Y. S., & Han, S. H. “Touch Key Design for One-Handed Thumb Interaction with a Mobile Phone: Effects of Touch Key Size and Touch Key Location.” International Journal of Industrial Ergonomics, 40(1), pp. 68-76, 2010

[7] Goel, M., Findlater, L., & Wobbrock, J. “Walktype: Using Accelerometer Data to Accomodate Situational Impairments in Mobile Touch Screen Text Entry.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2687-2696, 2012

[8] Weir, D., Rogers, S., Murray-Smith, R., & Löchtefeld, M. “A User-Specific Machine Learning Approach for Improving Touch Accuracy on Mobile Devices.” In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, pp. 465-476, 2012

스마트 시티 실현을 위한 성북구 도로의 혼잡시간과 혼잡시간이 아닐 때의 교통사고 비교 분석

정가은O, 이다영, 강종구
성신여자대학교 AI융합학부
{20221421, 20221392, jonggu.kang}@sungshin.ac.kr

Comparative analysis of traffic accidents when congestion and non-congestion hours on roads in Seongbuk-gu for the realization of a smart city

Gaeun JungO, Dayoung Lee, Jonggu Kang
School of AI Convergence, Sungshin Women's University

요 약

미래 도시 산업으로 주목받고 있는 스마트시티로의 전환에 있어 교통 인프라 구축을 위한 지역 맞춤형 교통 예측 모델은 중요하게 인식되고 있다. 본 연구의 목표는 성북구 맞춤형 교통사고 예측 모델을 만들기 위한 성북구 도로의 혼잡도와 교통사고 발생률의 유의미한 연관성을 파악하는 것이다. 이를 위해 성북구 도로를 혼잡시간대와 혼잡하지 않은 시간대로 분류하여 86개의 도로로 세분화하고 각각의 도로에 성북구 교통사고 데이터를 매핑하여 성북구 도로의 혼잡한 시간대와 혼잡하지 않은 시간대의 교통사고 데이터를 얻어 비교 분석한다. 분석 결과 혼잡한 시간대 보다 혼잡하지 않은 시간대에 발생한 교통사고가 더 많았으며, 두 경우 모두 교통사고 발생 시 가장 많은 비율을 차지하는 노면상태와 기상상태가 동일함으로 노면상태 및 기상상태는 혼잡도와 교통사고 발생률의 연관성 도출 과정에서 많은 영향을 끼치지 않는 지표임을 검증했다. 본 연구에서 도출한 성북구 도로의 혼잡도와 교통사고 발생률의 연관성에 관한 분석 결과들은 GRU, LSTM 모델을 이용한 성북구 맞춤형 교통사고 예측 모델 개발 시 효과적인 지표로 사용될 것으로 기대된다.

1. 서 론

현대 도시화의 진전과 함께 자동차 수요의 급증은 도로 교통체계에 새로운 도전을 제시하고 있다. 최근 도시 개발은 스마트시티로의 전환을 강조하고 있으며, 도로 교통사고 예방은 도시 안전과 도시 교통체계 향상을 위한 전 세계의 핵심적인 과제 중 하나로 인식되어 왔다.[1] 특히 한국은 2022년 기준 사상자 발생 수(Accidents involving casualties number)가 196,863명으로 OECD 국가 중 4위를 차지하여[2] 이에 대한 적절한 대응이 필요한 시점이다.

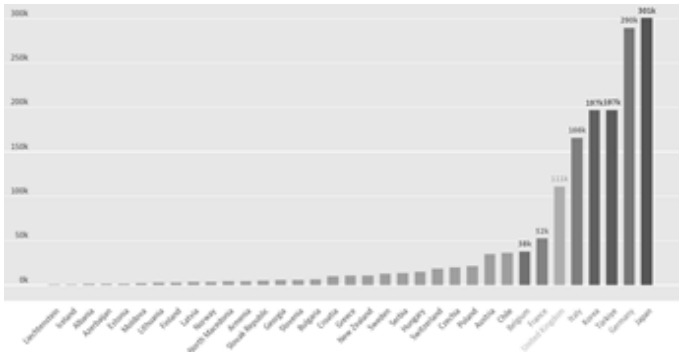


그림 1. Accidents involving casualties (OECD) [2]

2022년 2월 서울시는 스마트시티 추진계획[3]을 발표하면서 스마트시티로의 전환의 중요성을 강조했다. 해당 계획에서 언급된 성북구는 서울시 주요자치구 중 하나이며 교육기관, 문화시설이 밀집된 지역으로 여러 교통수단이 교차하기 때문에 다양한 교통상황이 발생할 수 있다. 본 연구에서는 성북구 도로의 혼잡시간과 혼잡하지 않은 시간의 교통사고를 사고유형, 도로 환경에 따라 비교 분석하여, 도로의 혼잡 상태와 교통사고 발생률의 연관성을 찾고, 이를 통해 교통사고 예방에 기여할 수 있는 통찰을 도출하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 분석하고, 3장에서는 성북구 도로 및 교통사고 분류 방법 및 결과에 대해 기술하고, 4장에서는 3장에서 분류한 교통사고 데이터를 비교 분석한 결과를 설명한다. 5장은 위협요소에 대해 기술하고, 6장에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

본 연구와 관련된 연구로는 Spatial-Temporal Geographical Module 및 Spatial-Temporal Semantic Module과 GRU, 합성곱 모델을 사용한 GSNet 모델과 상관분석, 데이터마이닝기법[4]인 CHAID를 바탕으로 토지이용 및 교통의 특성을 반영하여 서울시의 교통사고 예측 모형을 개발한 연구가 있다.

Wang, B. et al.[5]은 시카고와 뉴욕 지역 데이터를 기반으로 하여 시공간적 특성 뿐만 아니라 지리 및 의미적 공간-시간 특성을 융합하여 교통사고 위험 예측의 정확성을 향상시킨 모델이다. [6][7]

박준태 et al.[8]는 서울시를 토지 특성에 맞게 4개의 유형으로 분류하여 연구를 진행하였다. 종사자수, 교차로 밀도, 고령자 인구 구성비를 변수로 설정하여 교통사고 발생을 예측할 수 있는 유형별 모형을 도출하였다.

기존의 연구에서는 모델의 배경이 한국이 아니거나 사고 유형을 고려하지 않고 사고 빈도 데이터만을 활용하여 교통사고를 예측한다는 아쉬움이 있다. 따라서 한국에 최적화되며 교통 사고 유형을 반영한 연구가 필요하다.

3. 성북구 도로 및 교통사고 분류

3.1 통계 분석도구

본 연구에서 사용된 통계 분석 도구는 교통 분석 서비스를 제공하는 View-T와 교통사고 데이터를 제공하는 TAAS이다. View-T에서는 도로의 평균 속도와 교통량을 지표 삼아 혼잡 도로를 선정한다. 이때 평균 속도는 도로를 주행하는 모든 차량들의 평균 속도이며 상세도로망 Level6 네트워크 기반으로 산출되어 주요도로망 Level 5.5 네트워크로 구축된 데이터가 제공된다. View-T 통행지표 설명자료에 따르면 행정구역 C가 Level 6의 A,B 링크로 구성되어 있다고 가정했을 때의 C의 평균속도는 다음과 같다. [9]

$$\frac{(A\text{링크길이}+B\text{링크길이})}{(\frac{A\text{링크길이}}{A\text{링크속도}})+(\frac{B\text{링크길이}}{B\text{링크속도}})}$$

View-T의 혼잡도 분석에서 평균속도를 지표로 삼을 경우 평균속도는 빨간색, 주황색, 노란색, 초록색, 연두색, 5가지 색으로 구분된다. 빨간색은 0-30 km/h, 주황색은 30-40 km/h, 노란색은 40-50 km/h, 초록색은 50-60 km/h, 연두색은 60-150 km/h로 성북구 도로의 혼잡 시간대는 빨간색인 평균속도 0~30Km/h의 시간대를 혼잡 시간대로 선정하였다. [10]

표 1. View-T 혼잡기준 속도표

구분		기준속도
고속도로	4 차로 이상	90 Km/h
	2 차로	75 Km/h
도시고속도로	-	75 Km/h
일반국도	4 차로 이상	80 Km/h
	2 차로	70 Km/h
특별광역시도	-	27 Km/h
국가지원지방도	-	60 Km/h
지방도	-	60 Km/h
시군도	-	27 Km/h

3.2 도로 명칭 부여

그림 2. 성북구 주요 도로 [10]

성북구 도로는 성북n이라는 변수명을 부여하여 성북 1부터 성북 86까지 총 86개의 도로로 세분화하였다. 성북구 도로의 세분화 방법은 View-T에서 얻은 혼잡도 정보와 TAAS에서 얻은 교통사고 현황 정보를 도로에 1:1 대응하여 성북구 주요 도로 위주로 세분화하되, 주요도로가 아닌 도로 중 다량의 교통사고 발생한 도로에 한해 추가하여 명칭을 부여하였다.

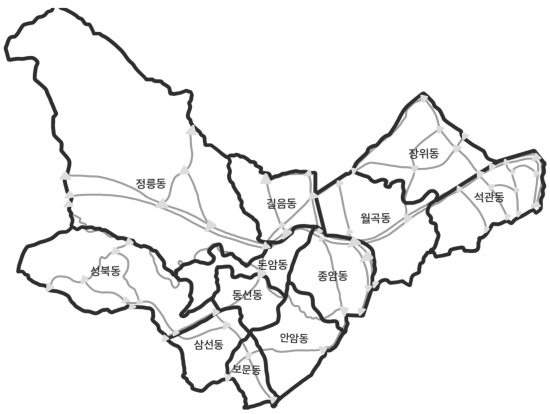


그림 3. 성북구 주요 도로

86개의 도로는 IC, 4차로 이상의 일반 국도의 경우 양방향 2가지 도로의 집합으로, 2차로 이상의 일반 국도, 시군도의 경우 1가지 도로의 집합으로 분류하여 명칭이 부여된 각 도로 집합의 원소 도로로 분류했다. 즉 성북구의 A에서 B까지 가는 거리의 시군도 2개가 있다면 시군도 2개는 성북n이라는 집합의 원소 도로가 된다.

3.3 혼잡시간대 분석

위에서 세분화한 성북 1-86의 혼잡 시간대 분석은 86개의 도로 집합을 이루는 원소 도로들의 모든 시간대(00시-24시)의 혼잡 시간대 분석을 한 후 원소 도로들의 혼잡 시간대를 집합 도로들의 혼잡시간대로 확장하여 진행하였다. 원소 도로들의 혼잡 시간대를 구한 뒤, 혼잡 시간대 교집합에 위치한 시간을 집합 도로들의 혼잡 시간대로 선정하였다. 즉 성북 83을 이루는 원소 도로가 A, B가 있다고 가정할 때, A의 혼잡 시간대가 00-24시이고, B의 혼잡 시간대가 06-24시이면 성북 83의 혼잡 시간대는 A도로와 B도로의 교집합인 06-24시가 된다.

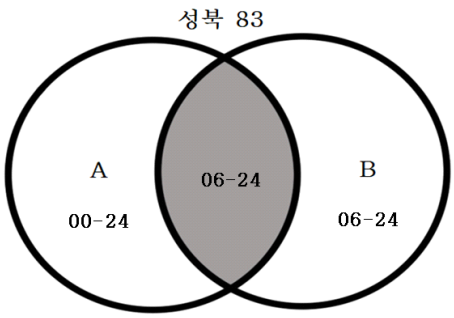


그림 4. 성북 83 집합

3.4 교통사고 분류

TAAS의 GIS 분석 시스템을 이용하여 2021년 성북구 교통사고 데이터를 추출하고 그림 4와 같이 세분화된 도로에 추출한 교통사고 데이터를 매핑하여 아래 표와 같이 각 도로의 시간대 별로 분류한다.

표 2. 성북 36,37,38의 혼잡시간대 분류표

	08-09시	09-10시
성북 36	중1, cc-측면충돌, d, s	-
	경1, cp-횡단중, d, s	-
	경1, cc-측면충돌, w, r	-
성북 37	경1, cp-횡단중, d, s	경1, cc-측면충돌, d, s
	경1, cc-측면충돌, d, s	경1, cc-측면충돌, d, s
	-	경1, cc-기타, d, s
성북 38	-	경1, cc-추돌, d, s
	-	경2, cc-측면충돌, d, s
	-	경1, cc-기타, d, s

위 과정을 통해 성북 1-86 도로에 해당하는 교통 데이터(사고유형, 피해유형, 노면상태, 기상상태)를 얻을 수 있으며 교통데이터는 변수로 표시하였다.

표 3. 교통데이터 표기

사고 유형	차대 사람 (cp)	길가장자리구역 통행중			
		보도 통행중			
		차도 통행중			
		횡단중			
		기타			
	차대차 (cc)	정면충돌			
		추돌			
		측면충돌			
		후진중충돌			
		기타			
	차량 단독 (c)	공작물 충돌			
		도로 외 이탈-추락			
		전도전복-전도			
		전도전복-전복			
		기타			
피해 유형	사망 (사)	교통사고 발생일로부터 30일 이내에 사망한 경우			
	중상 (중)	교통사고로 인하여 3주 이상의 치료를 요하는 부상을 입은 경우			
	경상 (경)	교통사고로 인하여 5일 이상 3주 미만의 치료를 요하는 부상을 입은 경우			
	부상 (부)	교통사고로 인하여 5일 미만의 치료를 요하는 부상을 입은 경우			
노면 상태	건조(d)	젖음/ 습기(w)	서리/ 결빙(f)	적설 (sc)	기타(e)
기상 상태	맑음(s)	눈(i)	흐림(b)	비(r)	기타 (etc)

피해유형은 각 단어의 앞글자로 표현하였고, 사고유형, 노면상태, 기상상태는 영어 철자를 따와 변수명을 부여하였다. 노면 상태에 대해서는 건조, 젖음/습기, 서리/결빙, 적설, 기타 로 분류할 수 있으며, 기상 상태에 대해서는 맑음, 눈, 흐림, 비, 기타로 분류할 수 있다.

표 4. 성북 46 교통데이터 정리표

	혼잡함	혼잡하지 않음	사고유형	혼잡함	혼잡하지 않음
성북 46	6	33	차대사람-기타	-	-
			차대사람-길가장자리구역통행중	-	-
			차대사람-보도통행중	-	1
			차대사람-차도통행중	-	-
			차대사람-횡단중	1	4
			차대차-기타	1	3
			차대차-정면충돌	-	-
			차대차-추돌	1	6
			차대차-측면충돌	3	19
			차대차-후진중충돌	-	-
			차량단독-공작물충돌	-	-
			차량단독-기타	-	-
			차량단독-도로외이탈-추락	-	-
			차량단독-전도전복-전도	-	-
			차량단독-전도전복-전복	-	-

피해정도	혼잡함	혼잡하지 않음	노면상태	혼잡함	혼잡하지 않음	기상상태	혼잡함	혼잡하지 않음
부상	-	3	건조(d)	6	30	맑음(s)	6	26
경상	7	29	젖음/습기(w)	-	3	눈(i)	-	-
중상	-	-	서리/결빙(f)	-	-	흐림(b)	-	5
사망	-	-	적설(sc)	-	-	비(r)	-	1
			기타(e)	-	-	기타(etc)	-	1

표 4은 그림 4의 성북 1-86 도로에 교통사고 데이터를 매핑하여 표 2에서 설명한 변수명에 따라 교통사고 데이터를 분류한 것 중 성북 46 도로의 데이터 분류표로, 성북 46 도로에서는 혼잡한 시간대의 교통사고가 6건, 혼잡하지 않은 시간대의 교통사고가 33건으로 총 39건의 교통사고가 발생했으며, 교통사고 발생 시 노면상태는 건조하고 기상상태는 맑은 날이 많았다는 것을 확인할 수 있다.

4. 성북구 교통사고 비교 분석

4.1 연구 질문

본 논문에서 수행한 분석은 다음과 같은 연구 질문(RQ)를 설정한다.

RQ1: 성북구 도로의 혼잡도는 교통사고 발생률과 연관성이 있는가?

성북구 도로의 혼잡한 시간대에 발생한 교통사고 발생건수와 혼잡하지 않은 시간에 발생한 교통사고 발생건수의 차이를 비교한다.

RQ2: 성북구 도로의 혼잡도에 따른 교통사고 발생률에 악화된 노면상태, 기상상태는 영향을 주는가?

성북구 도로의 혼잡도에 따른 교통사고 발생 건수와 교통사고 발생 시의 노면상태, 기상상태를 분석한다.

RQ3: 성북구 도로의 혼잡도에 따라 발생한 교통사고의 사고유형도는 달라지는가?

성북구 교통사고 데이터 중 사고유형을 혼잡도에 따라 분류하고 비교 분석하여 유의미한 결과를 도출한다.

4.2 성북구 평일 도로 데이터

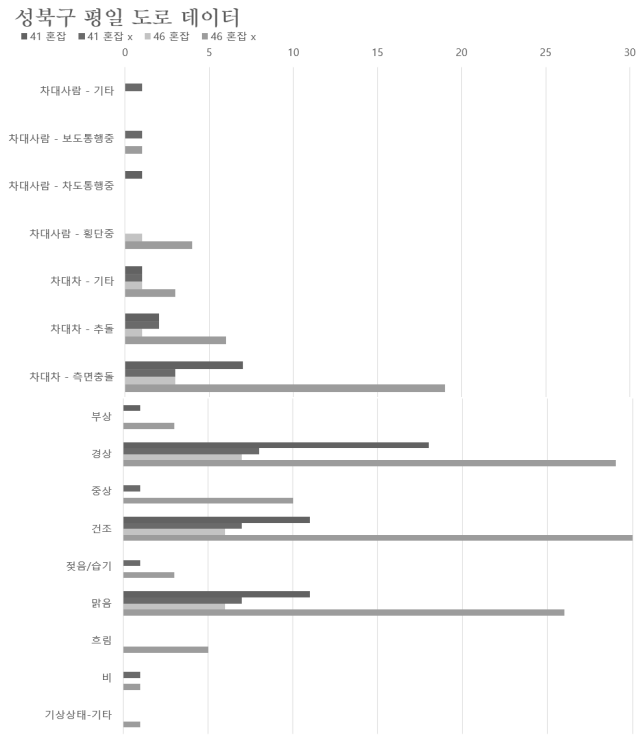


그림 5. 성북구 평일 도로 데이터 그래프

그림 5와 같이 성북 1-86 도로를 통계내면 표 5와 같다. 성북구의 평일 전체 사고 건수는 혼잡한 경우가 246건, 혼잡하지 않은 경우가 406건으로 사고 발생률을 다음과 같이 정의할 때

$$사고발생률 = \frac{시간대별사고발생(수)}{전체사고발생(수)}$$

혼잡하지 않은 경우가 혼잡한 경우보다 사고 발생률이 약 1.65배 높다는 것을 알 수 있다. 혼잡한 경우와 혼잡하지 않은 경우 모두 사고유형은 차대차(cc)의 경우가 각각 78%, 75%를 차지하며 그중 측면충돌의 경우가 95건, 150건으로 가장 높게 나타났다. 두 경우 모두 노면상태는 건조하고, 기상상태는 맑은 날이 가장 많은 비율을 차지하였고 혼잡한 경우와 혼잡하지 않은 경우의 사고 발생 건수를 고려하면 거의 동일한 비율을 차지한다는 것을 알 수 있다.

표 5. 성북구 도로 평일 통계

교통사고 데이터 \ 혼잡여부	혼잡	혼잡 하지 않음
사고 건수	246	406
차대사람(cp) -길가장자리구역 통행중	0	0
차대사람(cp) -보도통행중	4	10
차대사람(cp) -차도통행중	8	10
차대사람(cp) -횡단중	22	39
차대사람(cp) -기타	0	0
차대차(cc) -정면충돌	7	5
차대차(cc) -추돌	29	85
차대차(cc) -측면충돌	95	150
차대차(cc) -후진중충돌	2	1
차대차(cc) -기타	59	65
차량단독(c) -공작물충돌	0	5
차량단독(c) -도로외이탈 -추락	0	0
차량단독(c) -전도전복 -전도	1	1
차량단독(c) -전도전복 -전복	0	0
차량단독(c) -기타	4	6

혼잡도 \ 피해유형	혼잡함	혼잡 하지 않음	혼잡도 \ 피해유형	혼잡함	혼잡 하지 않음
부상	12	20	중상	61	89
경상	210	398	사망	2	4

혼잡도 \ 노면상태	혼잡함	혼잡 하지 않음	혼잡도 \ 기상상태	혼잡함	혼잡 하지 않음
건조(d)	225	377	맑음(s)	220	355
젖음/습기(w)	17	25	눈(i)	1	2
서리/결빙(f)	1	1	흐림(b)	9	20
적설(sc)	1	1	비(r)	12	16
기타(e)	2	2	기타(etc)	4	13

4.3 성북구 주말 도로 데이터

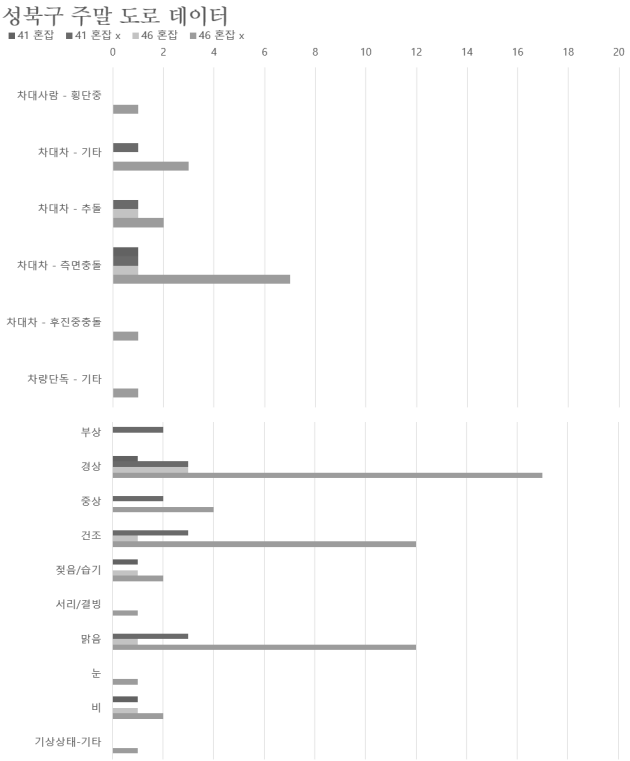


그림 6. 성북구 주말 도로 데이터 그래프

그림 6과 같이 성북 1-86 도로를 통계내면 표 6과 같다. 성북구의 주말 전체 사고 건수는 혼잡한 경우가 98건, 혼잡하지 않은 경우가 130건으로 혼잡하지 않은 경우가 혼잡한 경우보다 사고 발생률이 약 1.32배 높다는 것을 알 수 있다. 혼잡한 경우와 혼잡하지 않은 경우 모두 사고유형은 차대차(cc)의 경우가 각각 80%, 86%를 차지하며 그중 측면충돌의 경우가 36건, 50건으로 가장 높게 나타났다. 두 경우 모두 노면상태는 건조하고, 기상상태는 맑은 날이 가장 많은 비율을 차지하였고 혼잡한 경우와 혼잡하지 않은 경우의 사고 발생 건수를 고려하면 거의 동일한 비율을 차지한다는 것을 알 수 있다.

표 6. 성북구 도로 주말 통계

교통사고 데이터 \ 혼잡여부	혼잡	혼잡 하지 않음
사고 건수	98	130
차대사람(cp) -길가장자리구역 통행중	2	0
차대사람(cp) -보도통행중	1	2
차대사람(cp) -차도통행중	1	2
차대사람(cp) -횡단중	6	5
차대사람(cp) -기타	9	4
차대차(cc) -정면충돌	4	0
차대차(cc) -추돌	9	35
차대차(cc) -측면충돌	36	50
차대차(cc) -후진중충돌	4	1
차대차(cc) -기타	26	26
차량단독(c) -공작물충돌	0	1
차량단독(c) -도로외이탈 -추락	0	0
차량단독(c) -전도전복 -전도	0	1
차량단독(c) -전도전복 -전복	0	0
차량단독(c) -기타	0	3

혼잡도 \ 피해유형	혼잡함	혼잡 하지 않음	혼잡도 \ 피해유형	혼잡함	혼잡 하지 않음
부상	8	5	중상	16	27
경상	94	146	사망	0	2

혼잡도 \ 노면상태	혼잡함	혼잡 하지 않음	혼잡도 \ 기상상태	혼잡함	혼잡 하지 않음
건조(d)	80	110	맑음(s)	75	109
젖음/습기 (w)	14	11	눈(i)	1	2
서리/결빙 (f)	0	2	흐림(b)	7	4
적설 (sc)	0	0	비(r)	10	9
기타(e)	4	6	기타 (etc)	5	6

4.4 분석 결과

4.2~4.3에서 수행한 분석 결과를 정리하면 다음과 같다.

RQ1: 성북구 도로의 혼잡도는 교통사고 발생률과 연관성이 있는가?

2021년 성북구의 교통사고 분석 결과 혼잡한 시간대보다 혼잡하지 않은 시간대 발생한 교통사고가 평일의 경우 1.32배, 주말의 경우 1.65배 많다. 주어진 결과에 따르면 교통사고 발생률은 혼잡도가 낮을 때 더 많이 발생함을 알 수 있다

RQ2: 성북구 도로의 혼잡도에 따른 교통사고 발생률에 악화된 노면상태, 기상상태는 영향을 주는가?

혼잡시간대와 혼잡하지 않은 시간대 두 경우 모두 교통사고 발생 시 가장 많은 비율을 차지하는 노면상태는 건조, 기상상태 맑음으로 혼잡도에 따른 교통사고 발생률에 악화된 노면상태, 기상상태는 영향을 끼치지 않는 것으로 확인된다.

RQ3: 성북구 도로의 혼잡도에 따른 교통사고 사고유형은 달라지는가?

혼잡 시간대와 혼잡하지 않은 시간대 모두 교통사고 발생 시 가장 많은 비율을 차지하는 사고 유형은 차대차 유형이며, 그 중 비중이 높은 유형은 측면추돌, 추돌 2가지 유형으로, 혼잡도와 관계없이 발생률이 높은 사고유형이 동일하므로 성북구 도로의 혼잡도에 따른 사고유형은 달라지지 않음을 확인할 수 있다.

5. 위협요소

본 연구에서 분석 시 사용한 데이터는 View-T에서 제공하는 성북구 혼잡도로 중 주요도로에 성북구 교통사고 데이터를 1:1 대응시켜 도출한 교통 데이터로, 성북구 맞춤 교통사고 예측 모델을 만들기 위해 지역 기반 데이터를 사용하여 다른 지역 기반의 데이터를 사용하거나, 다른 비교 분석 방법을 적용 가능한지 여부는 추가적인 검증이 필요하다.

6. 결론과 향후 연구 방향

최근 도시 개발이 인공지능 기반의 스마트시티 개발로 발전함에 따라 교통체증, 교통사고 등 다양한 교통 문제를 예측할 수 있는 교통 예측 모델 개발은 스마트시티 구축에 있어서 중요하다. 본 연구에서는 성북구 교통사고 예측 모델 개발 시 필요한 수치들을

실험적으로 분석했다. 분석 결과에 따르면 성북구 교통사고 발생률은 도로의 혼잡도가 낮을 때 더 많이 발생했으며, 혼잡도에 따른 교통사고 발생률에 악화된 노면상태 및 기상상태는 영향을 끼치지 않는다. 또한, 혼잡도와 관계없이 교통사고 발생 시 가장 많이 발생하는 사고유형은 차대차 유형이다.

향후 본 연구에서 검증된 수치를 바탕으로 성공적인 시계열 데이터 예측을 위해 GRU 및 LSTM[11]과 같은 순환 신경망을 사용하여 성북구 도로의 혼잡도와 교통사고 발생률의 연관성을 적용시킨 성북구 교통사고 예측 모델을 개발할 것이다.

7. Acknowledgement

이 논문은 성신여자대학교 2023년도 동계 학부생 연구참여 프로그램(UROP)의 지원을 받아 수행된 연구임

8. 참고문헌

[1] Feng, S., Zhong, S., Hu, L., & Sun, L. (2015). “Traffic Flow Prediction With Big Data: A Deep Learning Approach. IEEE Transactions on Intelligent Transportation Systems”, 16(2), 865-873.

[2] OECD. (2022). “Road accidents data: Accidents involving casualties”, Number, 2022. Retrieved from https://data.oecd.org/transport/road-accidents.htm

[3] 디지털정책관. (2023). “2023 스마트도시 및 정보화 시행계획”. 서울특별시

[4] 류귀열, 문영수. (2006). 연관분석을 이용한 데이터마이닝 기법에 관한 사례연구. Journal of The Korean Data Analysis Society, 8(3), 1021-1033.

[5] Wang, B., Lin, Y., Guo, S., & Wan, H. (2021). “GSNet: Learning Spatial-Temporal Correlations from Geographical and Semantic Aspects for Traffic Accident Risk Forecasting”, In AAAI-21 Technical Tracks, Vol. 35, No. 5, 4402-4409. AAAI Press.

[6] Ren, H., Song, Y., Wang, J., Hu, Y., & Lei, J. (2018). “A deep learning approach to the citywide traffic accident risk prediction.” In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 3346-3351). IEEE.

[7] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). “Convolutional LSTM network: A machine learning approach for precipitation nowcasting.” In Advances in neural information processing systems. (pp. 802-810).

[8] 박준태,장일준,손의영,이수범. (2011). "토지이용 및 교통특성을 반영한 교통사고 예측모형 개발 연구." 대한교통학회지 29.6 :39-56.

[9] View-T. (2021). “View Transport 통행지표 설명자료 중 평균속도”. pp.5

[10] View-T. 시공간 혼접 분석 평균속도 기준표. [Online] Available : https://viewt.ktdb.go.kr/cong/map/second_map/do

[11] 전승배, 오행렬, 이태영, 김건, 정명훈. (2010). “LSTM 순환 신경망을 이용한 도로 교통 속도 예측”. 대한토목학회 학술대회:15-16.

스마트 시티 실현을 위한 전국 졸음운전 다발 구역과 졸음 쉼터 위치 비교 분석

이채원 ○, 강종구

성신여자대학교 AI 융합학부

20221402@sungshin.ac.kr, jonggu.kang@sungshin.ac.kr

Comparative analysis of nationwide drowsy driving areas and drowsy rest areas to realize a smart city

Chaewon Lee○, Jonggu Kang

School of AI Convergence, Sungshin Women's University

요 약

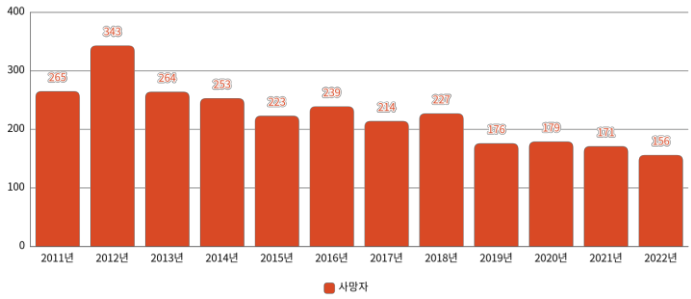
첨단 정보통신기술을 통해 시민들에게 쾌적한 삶을 제공하는 스마트시티를 도입하기 위해서는 교통 문제 해결이 중요하다. 여러 교통 문제 중에서 졸음운전 사고는 높은 치사율과 고속도로 사망사고의 주원인으로, 아주 심각한 문제이다. 본 연구에서는 졸음운전 사고를 줄이기 위해 졸음운전 다발 구역과 졸음 쉼터 데이터를 활용해 현재 우리나라의 졸음 쉼터가 적절한 위치에 설치되어 있는지를 검증하고자 한다. QGIS와 Python을 활용해 WSG84 좌표계로 변환한 표준 링크 데이터와 졸음운전 다발 구역 데이터를 통합해 졸음운전이 자주 발생하는 지역을 파악하고 이를 구글 지도에 시각화하여 데이터 분석을 진행한 결과 대부분의 지역에서 졸음운전 다발 구역과 졸음 쉼터의 위치가 일치하지 않는 것으로 확인되었다. 이러한 연구 결과를 통해 더 적절한 위치에 졸음 쉼터들을 설치하고 졸음운전 사고를 감소시킬 수 있을 것으로 기대된다.

1. 서론

스마트시티란 첨단 정보통신기술을 통해 도시의 여러 문제를 해결함으로써 시민들이 편리하고 쾌적한 삶을 누릴 수 있도록 하는 것을 일컫는다. 이러한 스마트시티를 도입하기 위해서 해결해야 할 문제 중 하나에는 교통 문제가 있다. 본 논문에서는 현재 우리나라에 존재하는 다양한 교통 문제 중 졸음운전에 대해 다루고자 한다.

졸음운전은 고속도로 사망사고에서 큰 비중을 차지하고 있다. 한국도로공사에 따르면 2016년부터 2020년 5년간 고속도로 교통사고 사망자 중 70%는 졸음운전으로 인해 발생했다. 게다가 졸음운전 교통사고 치사율은 2.6명으로, 전체 교통사고 치사율 평균인 1.4명의 거의 2배 수준에 달한다 [1]. 이처럼 현재 우리 사회에서는 졸음운전으로 인한 심각한 피해가 발생하고 있다.

이를 해결하기 위해 한국도로공사에서는 2011년부터 휴게소 간 간격이 먼 구간에 졸음 쉼터를 설치하기 시작했다. 그림 1은 졸음운전 사망자 수를 그래프로 나타낸 것으로, 졸음 쉼터 설치 이후인 2012년부터 2022년까지 사망자 수가 꾸준히 감소하는 추세인 것을 확인할 수 있다. 또한 졸음쉼터를 이용한 실험집단이 이용하지 않은 집단보다 졸음운전 감소 효과가 더 크게 나타난다는 사실을 통해 졸음 쉼터가 졸음운전 사고 감소에 효과적인 것을 알 수 있다 [2].



(그림 1) 졸음운전 사망자 수

현재 졸음 쉼터는 졸음운전 다발 구역이 아닌 휴게소 간 간격을 기준으로 설치되어 있기 때문에 실제 졸음운전이 많이 발생하는 구간과 졸음 쉼터 설치 위치가 일치하는지에 대해서는 알 수 없다. 본 논문에서는 한국교통안전공단에서 제시한 위험 물질 운송 차량 졸음운전 다발 구역 데이터와 졸음 쉼터의 위치를 비교 분석하여 졸음 쉼터가 적절한 위치에 설치되어 있는지를 판단하고, 그렇지 않은 경우에 적합한 졸음 쉼터 설치 위치를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고 3장에서는 활용 데이터와 데이터 전처리 방식, 시각화를 통한 분석 결과를 제시한다. 4장에서는 토의를 통해 추가 연구가 필요한 부분을 기술하고 5장에서는 위협요소, 6장에서는 결론 및 향후 연구 계획을 제시한다.

2. 관련 연구

본 연구에서는 졸음운전과 졸음 쉼터와 관련된 선행 연구들의 결과를 분석하였다. 분석 결과, 졸음 쉼터의 위치와 졸음운전 다발 구역을 비교 분석한 연구는 존재하지 않았지만 졸음 쉼터가 졸음운전을 예방하는데 큰 역할을 한다는 점과 졸음운전이 교통류 안정성에 미치는 영향을 확인할 수 있었다.

2011년 고속도로에 설치된 졸음쉼터를 대상으로 설치 전과 후 각 18개월의 사고이력자료를 수집하여 분석한 결과 졸음쉼터를 설치하지 않았을 경우에 비해 설치함으로써 53.06%의 사고 감소 효과가 있는 것으로 나타났다 [3][4]. 돌발 상황 영향권에서는 졸음운전을 하는 경우 안정성이 악화되는 것으로 나타났다 [5].

3. 본론

3.1 활용 데이터

3.1.1 졸음운전 다발 구역 데이터

졸음운전 다발 구역을 분석하기 위해서 한국교통안전공단의 위험 물질 운송 차량 졸음운전 다발 구역 데이터를 활용했다 [6]. 해당 데이터는 2022년 1월부터 위험 물질 운송차량 1100대에 졸음운전 방지 장치를 설치해 졸음운전과 전방 주시 태만이 발생한 도로의 위치를 표준 노드 링크로 표시한 자료이다. 데이터는 그림 2에서 확인할 수 있듯이 링크 아이디와 기준 연월, 발생 횟수, 발생 등급 코드, 도로명, 링크 길이로 구성되어 있으며, 개수는 총 130229개이다. 본 연구에서는 링크 아이디와 발생 횟수를 활용해 졸음운전 다발 구역을 파악했다. 발생 횟수의 범위는 1~509 까지로 폭넓게 구성되어 있기 때문에, 50번 이상 졸음운전 사고가 발생한 지역만 추출해 436개의 데이터로 연구를 진행했다.

A	B	C	D	E	F
링크아이디	기준년월	발생횟수	발생등급코드	도로명	링크길이(미터)
2340266400	2023-01-01	2	E15	성남이천로	840.6934
3560576300	2023-01-01	2	E15	중앙고속도로	3758.9268
1220002902	2023-01-01	1	E15	동부간선도로	678.9876
2220229702	2023-01-01	1	E15	비룡로	271.0529
2333264800	2023-01-01	1	E15	수도권제2순환	956.9483
1400200700	2023-01-01	1	E15	부산외곽순환고	344.8245
2420006100	2023-01-01	1	E16	순환서로	345.7377

(그림 2) 위험물질운송차량 졸음 운전 다발 구역 데이터

3.1.2 표준 노드 링크 데이터

졸음운전 다발 구역의 위치가 위경도가 아닌 링크 아이디로 제공되었기 때문에, 정확한 위치를 파악하기 위해서는 ITS 국가교통정보센터에서 제공하는 표준 노드 링크 데이터를 활용해야 한다 [7]. 그림 3은 데이터의 구성을 나타낸 것으로, 본 연구에서는 여기에 위경도 좌표를 나타내는 wkt_geom 속성을 추가해 연구를 진행했다.

필드명	LINK_ID	F_NODE	T_NODE	LANES	ROAD_RANK	ROAD_TYPE	ROAD_NO
속성명	링크ID	시작노드ID	종료노드ID	차로수	도로등급	도로유형	도로번호

필드명	ROAD_NAME	MULTI_LINK	CONNECT	MAX_SPD	REST_VEH	REST_W	REST_H	REMARK
속성명	도로명	중용구간여부	연결로코드	최고제한속도	통과제한차량	통과제한하중	통과제한높이	비고

(그림 3) 표준 링크 데이터 구성

3.1.3 졸음 쉼터 데이터

전국에 설치된 졸음 쉼터의 위치를 파악하기 위해 한국도로공사에서 제공한 전국 졸음 쉼터 표준데이터를 활용했다 [8]. 그림 4 는 데이터의 구성을 나타낸 것이며, 데이터의 개수는 총 232 개이다. 본 연구에서는 졸음 쉼터 설치 위치를 필요로 하기 때문에 졸음쉼터명과 소재지 지번 주소, 위도, 경도 속성만 활용했다.

A	B	C	D	E	F	G	H	I	J	K
졸음쉼터명	시도명	시군구명	도로종류	도로노선명	도로노선번호	도로노선방향	소재지도로	소재지지번주소	위도	경도
김포	경기도	김포시	고속국도	서울외곽선	100	판교기점 + 일산종점		경기도 김포시 고촌읍 신	37.5923	126.7684
김포	경기도	김포시	고속국도	서울외곽선	100	일산기점 + 판교종점		경기도 김포시 고촌읍 신	37.587	126.7607
시흥	경기도	시흥시	고속국도	서울외곽선	100	판교기점 + 일산종점		경기도 시흥시 계수동 9(	37.45071	126.8038
시흥	경기도	시흥시	고속국도	서울외곽선	100	일산기점 + 판교종점		경기도 시흥시 계수동 46	37.45048	126.8032
서서울	경기도	안산시	고속국도	서해안선	15	목포기점 + 서울종점		경기도 안산시 상록구 장	37.35664	126.8648
서서울	경기도	안산시	고속국도	서해안선	15	서울기점 + 목포종점		경기도 안산시 상록구 장	37.35493	126.862
군자	경기도	시흥시	고속국도	영동선	50	강릉기점 + 인천종점	경기도 시흥시	군자로335번길 36-2	37.35876	126.7808
이목	경기도	수원시	고속국도	영동선	50	인천기점 + 강릉종점		경기도 수원시 장안구 이	37.31955	126.9858
용인	경기도	용인시	고속국도	영동선	50	인천기점 + 강릉종점		경기도 용인시 기흥구 청	37.28862	127.1387
향남	경기도	화성시	고속국도	서해안선	15	서울기점 + 목포종점		경기도 화성시 향남읍 구	37.07012	126.8884
송탄	경기도	평택시	고속국도	평택제천선	40	제천기점 + 평택종점		경기도 평택시 모곡동 22	37.03098	127.0803
송탄	경기도	평택시	고속국도	평택제천선	40	평택기점 + 제천종점		경기도 평택시 모곡동 22	37.03232	127.0814

(그림 4) 졸음 쉼터 표준 데이터 구성

3.2 데이터 좌표계 변환

3.2.1 평면직각좌표계

지구상의 3 차원적인 위치 좌표를 2 차원의 두 개의 좌표계(x 축, y 축)를 사용하여 지도에 표현하는 방법으로, 일정한 지역 범위로 통일되게 사용하기 위하여 정한 좌표계를 말한다. TM 좌표계는 우리나라에서 우리나라에 맞게 여러 가지 기준을

정하여 투영된 좌표계로, 한국에만 적용되는 고유의 평면직각좌표계이다 [9].

3.2.2 지리좌표계

지표면상의 임의의 점을 기준 타원체로 투영하고 그 위치를 경도, 위도 및 평균 수해면으로부터의 높이로 표시한 좌표 체계이다 [10].

3.2.3 QGIS 와 Python 을 활용한 좌표계 변환

표준 노드 링크 데이터는 EPSG:5186(TM 좌표)으로 설정되어 있어 QGIS 와 Python 을 활용해 WSG84 좌표로 변환 후 위경도를 파악해야 한다.

QGIS 는 오픈소스 GIS 로, 데이터 시각화와 좌표계 변환을 용이하게 할 수 있다. 그림 5 는 변환 전 표준 링크 데이터의 좌표 정보를 나타낸 것으로, wkt_geom 속성이 TM 좌표계로 설정되어 있는 것을 확인할 수 있다. 그림 6 은 QGIS 에서 표준 링크 정보가 포함된 MOCT_LINK_shp 파일을 열어 레이어 좌표계를 WSG84 로 변환한 것으로, wkt_geom 속성이 위경도로 설정되어 있는 것을 확인할 수 있다.

본 연구에서는 QGIS 를 통한 1 차 변환 후, 데이터가 제대로 변환되었는지를 확인하기 위해 Python 을 이용해 2 차 확인 과정을 거쳤다. 그림 7 은 TM 좌표계를 WSG84 좌표계로 변환하기 위한 파이썬 코드이다.

A	B
wkt_geom	LINK_ID
MultiLineString ((198102.16129999980330467 338397.071200000122185	3090459100
MultiLineString ((169015.71949999965727329 541448.15210000053048	1620006602
MultiLineString ((169272.91320000030100346 543301.169700000435113	1620013606
MultiLineString ((171046.61010000016540289 540272.391100000590085	1630010000
MultiLineString ((412194.69869999960064888 341942.946100000292062	1950070600
MultiLineString ((189931.39859999995678663 246164.025599999353289	3350031200
MultiLineString ((227629.39780000038444996 507297.242900000885128	2430116700
MultiLineString ((252201.71399999968707561 263393.827600000426173	3280144500
MultiLineString ((136219.18499999959021807 91928.1508000000876188	4050456700
MultiLineString ((264069.43460000026971102 587141.078999999910593	2500101600
MultiLineString ((375438.21250000037252903 481540.312000000849366	3710315400

(그림 5) TM 좌표계로 설정된 표준 링크 데이터

A	B
wkt_geom	LINK_ID
MultiLineString ((126.97904530865868367 35.6426701405365165,	3090459100
MultiLineString ((126.64973151763788906 37.4719462966090191,	1620006602
MultiLineString ((126.65256167635241979 37.48865042959941718	1620013606
MultiLineString ((126.6727359854608892 37.46141855647888264,	1630010000
MultiLineString ((129.34297111919941667 35.65183142608255906,	1950070600
MultiLineString ((126.88995737631690019 34.81128002453494474	3350031200
MultiLineString ((127.31107232939415042 37.16434593148009213	2430116700
MultiLineString ((127.57158986104897735 34.965294116911920248	3280144500
MultiLineString ((126.31425778584106467 33.4189763370831443,	4050456700
MultiLineString ((127.72827921623780867 37.88189634778137105	2500101600
MultiLineString ((128.96869382705304474 36.91634801653314923	3710315400

(그림 6) WSG84 로 변환한 표준 링크 데이터

```python
from pyproj import Proj, transform

# 주어진 좌표
x, y = 198102.16129999980330467,338397.07120000001221895

# WGS84 좌표 체계로 변환
in_proj = Proj(init='epsg:5186')  # 주어진 좌표 체계
out_proj = Proj(init='epsg:4326')  # WGS84 좌표 체계

lon, lat = transform(in_proj, out_proj, x, y)

print(f'위도: {lat}, 경도: {lon}')
```

(그림 7) 좌표계 변환을 위한 Python 코드

3.3 졸음운전 다발 구역과 졸음 쉼터 위치 시각화

3.3.1 졸음운전 다발 구역과 표준 링크 데이터 통합

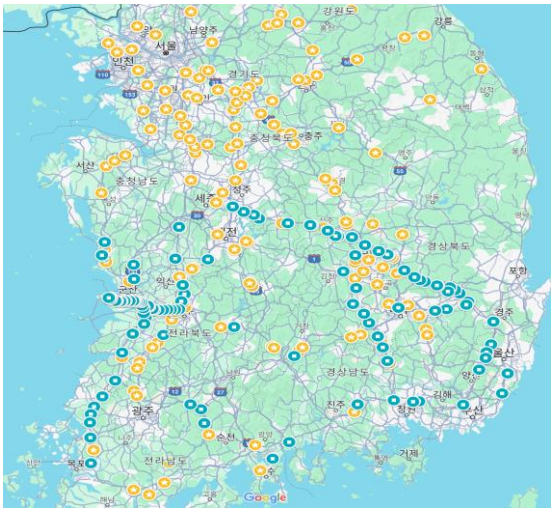
VLOOKUP 함수를 통해 링크 아이디를 기준으로 졸음운전 다발 구역 데이터에 표준 링크의 위경도 좌표를 연결한다. 그림 8 은 두 데이터를 통합한 모습이다.

A	B	C	D	E	F	G	H	I	J	K	L
링크아이디	기준년월	발생횟수	발생등급	도로명	링크길이(미터)	위경도 좌표					
2.56E+09	#######	68	E15	강원남부5	1560.8176	MultiLineString ((129.032150394222810	52	37.18340727505940			
3.19E+09	#######	76	E15	서해안고4	15589.1144	MultiLineString ((126.696327705199180	97	35.59976552307627			
2.54E+09	#######	66	E15	동탄백로	995.2103	MultiLineString ((129.036506785440280	96	37.10904371440820			
2.54E+09	#######	77	E15	동탄백로	1103.2182	MultiLineString ((129.034012051011728	77	37.15789326272066			
3.61E+09	#######	55	E15	상주영천고	4458.0466	MultiLineString ((128.480563709974944	69	36.24586098373047			
2.24E+09	#######	56	E15	서울외곽순	4457.9629	MultiLineString ((126.819500017443488	81	37.434016356942791			
2.54E+09	#######	52	E15	동탄백로	1007.2212	MultiLineString ((129.045399159483054	06	37.10610477458730			
2.54E+09	#######	71	E15	동탄백로	956.1117	MultiLineString ((129.045264217563982	36	37.10610536266650			
3.57E+09	#######	153	E15	상주영천고	9661.5786	MultiLineString ((128.712340840013609	98	36.07082559233226			
2.33E+09	#######	60	E15	서해안고4	3620.1326	MultiLineString ((126.896630954872037	48	37.27593968226648			
2.33E+09	#######	82	E15	서해안고4	6098.2514	MultiLineString ((126.883298016938297	79	37.12074993098508			
3.57E+09	#######	51	E15	상주영천고	3198.7158	MultiLineString ((128.929019524447681	99	36.01401527916321			
1.96E+09	#######	88	E15	울산포항고	5723.169	MultiLineString ((129.282713640622120	01	35.46219555768328			

(그림 8) 졸음운전 다발 구역에 위경도 통합

3.3.2 구글 지도에 시각화

436 개의 졸음운전 다발 구역과 232 개의 졸음 쉼터를 구글 지도에 마커로 표시한 후, 지역별로 나누어 분석을 수행한다. 그림 9 에서 파란색 마커는 졸음운전 발생 다발 구역을, 노란색 마커는 졸음 쉼터 설치 위치를 나타낸다.

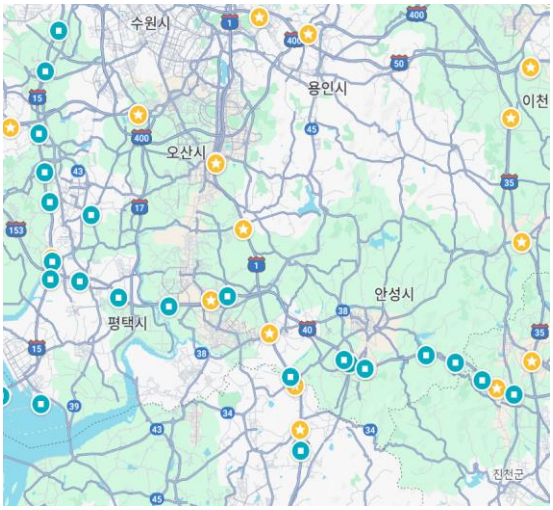


(그림 9) 졸음운전 다발 구역과 졸음 쉼터 시각화

3.4 지역 별 분석

3.4.1 경기도 평택시, 오산시, 화성시 분석

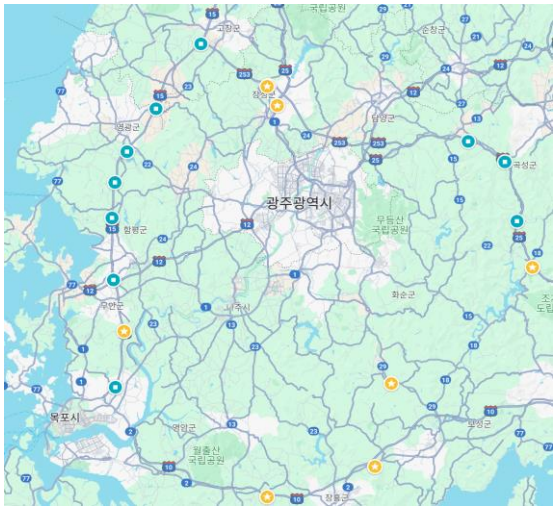
그림 10 의 왼쪽 구역을 보면 평택시의 평택시흥고속도로와 평택제천고속도로에서 졸음운전 사고가 많이 발생한 것을 확인할 수 있다. 그러나 졸음 쉼터는 이천시의 영동고속도로와 성남이천로에 주로 설치되어 있다.



(그림 10) 평택시, 오산시, 화성시 분석

3.4.2 전라도 광주광역시 분석

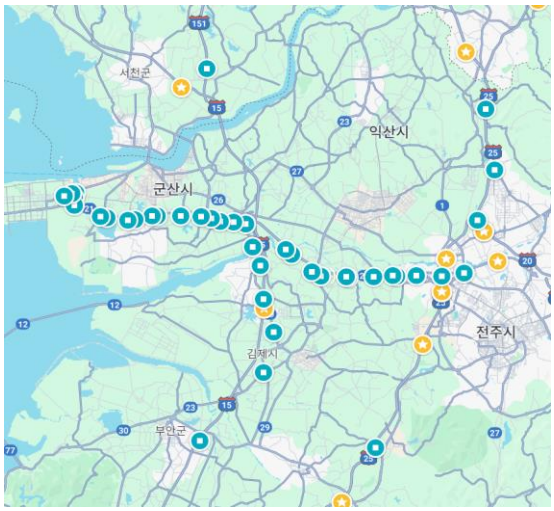
그림 11 을 보면 졸음운전 다발 구역을 나타내는 파란색 마커는 주로 왼쪽 구역인 서해안 고속도로에 분포된 것을 확인할 수 있다. 그러나 졸음 쉼터는 남해고속도로와 고창담양고속도로에 주로 설치되어 있어 졸음운전 다발 구역과 졸음 쉼터 설치 위치가 불일치하는 것을 확인할 수 있다.



(그림 11) 광주광역시 분석

3.4.3 전라도 군산시, 전주시 분석

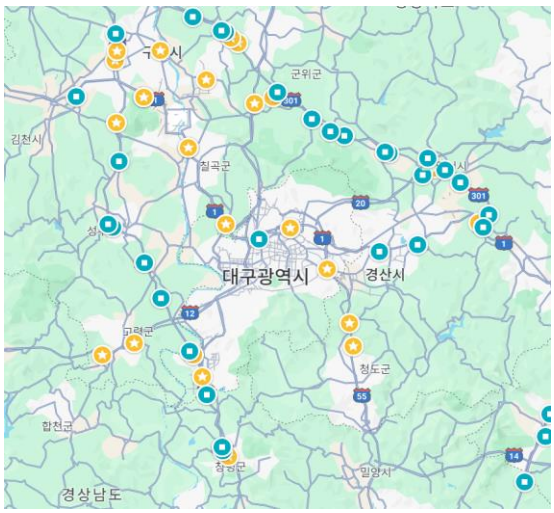
그림 12 을 보면 졸음운전 다발 구역은 주로 군산시의 새만금북로에서 발생한다. 그러나 졸음 쉼터의 위치는 전주시의 호남고속도로에 주로 설치되어 있어 졸음운전 다발 구역과 졸음 쉼터 설티 위치가 불일치하는 것을 확인할 수 있다.



(그림 12) 군산시, 전주시 분석

3.4.4 경상도 대구광역시 분석

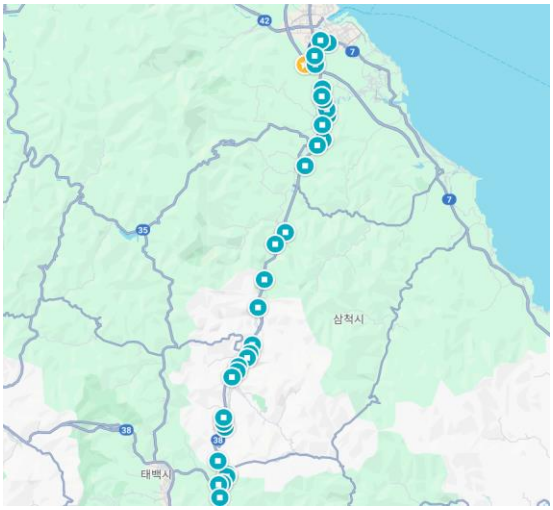
그림 13 의 왼쪽 구역을 보면 대구광역시의 중부내륙고속도로가 졸음운전 다발 구역인 것을 확인할 수 있다. 해당 구역과 근접한 위치에는 현풍 졸음쉼터와 창녕 졸음쉼터, 개진 졸음쉼터가 설치되어 있어 적절한 위치에 졸음 쉼터가 설치되어 있는 것으로 판단할 수 있다. 그러나 대구광역시의 또 다른 졸음운전 다발 구역인 상주영천고속도로에는 장미졸음쉼터를 제외하고는 적절한 졸음 쉼터가 설치되어 있지 않고, 대구부산고속도로에 졸음 쉼터가 다수 분포하는 것을 확인할 수 있다.



(그림 13) 대구광역시 분석

3.4.5 강원도 삼척시 분석

그림 14 를 보면 삼척시 강원남부로에서 졸음운전 발생 빈도가 매우 높은 반면, 졸음 쉼터는 단봉 졸음 쉼터 1 개로 현저히 적은 것을 알 수 있다.



(그림 14) 삼척시 분석

4. 토의

4.1 연구 결과

졸음운전 다발 구역인 경기도, 전라도, 강원도, 경상도의 졸음 쉼터 위치를 분석했을 때, 대구광역시의 중부내륙고속도로를 제외하고는 사고 발생 지역과 졸음 쉼터의 위치가 불일치하는 것으로 나타났다.

4.2 연구 결과를 통해 졸음 쉼터의 새로운 위치를 결정할 수 있는가?

졸음운전 다발 구역만으로 졸음 쉼터의 새로운 위치를 제안하기에는 휴게소 사이의 간격, 도로의 기하적 구조가 미치는 영향을 무시할 수 없다. 특히, 졸음쉼터 진입도로의 기하적 구조는 주행 속도에 영향을 미쳐 졸음 쉼터에서 발생하는 사고와 직접적인 연관이 있기 때문에 졸음 쉼터 설치 시 반드시 고려해야 하는 요소 중 하나이다 [11]. 따라서 졸음 쉼터의 새로운 위치를 제안하기 위해서는 모든 중요 요소들을 고려한 추가 연구가 필요하다.

4.3 졸음 쉼터 설치 위치를 결정할 때, 졸음 운전 다발 구역을 반영하면 졸음운전이 감소할 것으로 예상되는가?

운전자의 피로도가 높은 구간에 졸음 쉼터를 설치하면 자연스럽게 졸음 쉼터 이용률이 상승하고 졸음운전이 어느 정도 감소할 것으로 예상된다. 그러나, 고속도로 운전자의 졸음 쉼터 이용 기피 현상 발생 원인에는 졸음 쉼터에 질 높은 휴식을 취할 수 있는 시설이 갖추어져 있지 않다는 점도 크게 차지하고 있다 [12]. 따라서 확실한 졸음운전 감소를 위해서는 졸음 쉼터들이 적절한 면적과 시설물을 갖추어 운전자들의 졸음 쉼터 이용률이 올라가는 것이 중요하다.

5. 위협요소

본 연구에서는 위험 물질 운송차량의 졸음운전 다발 구역 데이터를 기반으로 연구를 진행했기 때문에, 다른 종류의 차량 데이터를 사용해서 연구를 진행하게 된다면 결과에 차이가 있을 수 있다. 또한, 졸음운전이 50 회 이상 발생한 구역을 추출했기 때문에 전체 데이터를 사용하게 된다면 더 많은 지역이 졸음운전 다발 구역으로 판단되어 시각화 결과가 다르게 나올 수 있다.

6. 결론 및 향후 연구

본 논문에서는 졸음운전 다발 구역과 졸음 쉼터의 위치를 비교 분석하여 졸음 쉼터가 적절한 위치에 설치되어 있는지를 판단하였다. 분석 결과, 대부분의 지역에서 졸음운전 다발 구역과 졸음 쉼터 설치 위치가 불일치하는 것으로 확인되었다. 이러한 분석 결과를 통해 더 적합한 졸음 쉼터 설치 위치를 선정할 수 있고 그로 인한 추가적인 졸음운전 사고 감소 효과를 기대할 수 있다. 그러나, 졸음 쉼터를 설치할 때는 휴게시설 간 배치 간격, 도로의 기하구조도 함께 고려해야 하기 때문에 졸음 쉼터 다발 구역만으로 위치를 정하는 것에는 어려움이 있다. 또한, 본 연구는 시각적인 방법으로만 비교 분석이 진행되었기 때문에 다른 방안을 활용해서 더 정확한 비교 분석을 할 필요가 있다. 따라서 모든 조건을 고려해 최적의 졸음 쉼터 설치 위치를 찾는 모델을 구축하는 추가 연구가 필요하다.

Acknowledgement

이 논문은 성신여자대학교 2023 년도 동계 학부생 연구 참여 프로그램(UROP)의 지원을 받아 수행된 연구임

참고문헌

[1] 컨슈머타임스, 도로교통공단, 단 3 초만 졸아도…'졸음운전사고' 전체比 치사율 86%↑. [Online] Available: https://www.cstimes.com/news/articleView.html?idxno=537213 (Accessed: Jan. 1, 2024)

[2] 정래엽, “휴게시설 설치가 교통사고 예방에 미치는 효과: 졸음쉼터를 중심으로”, 서울대학교, pp. 69-72, 2014

[3] 이명환, “고속도로 졸음쉼터 설치가 교통 안정성에 미치는 효과 분석”, 아주대학교, pp. 59, 2016.

[4] 이명환, 오인섭, 홍두표, 최기주, 오영태, 윤일수, “비교그룹방법을 이용한 고속도로 졸음쉼터 도입 효과 분석”, 대한교통학회 학술대회지, pp. 78-84, 2014.

[5] 도혜원, 민건규, 가동주, 이청원, “졸음운전이 교통류 안정성에 미치는 영향 분석”, 대한교통학회 학술대회지, pp. 193-198, 2022.
[6] 공공데이터포털, 한국교통안전공단 위험물질운송차량 졸음운전다발구역. [Online] Available: https://www.data.go.kr/data/15104639/fileData.do (Accessed: Jan. 1, 2024)
[7] ITS 국가교통정보센터, 표준노드링크. [Online] Available: https://www.its.go.kr/nodelink/nodelinkRef (Accessed: Jan. 1, 2024)
[8] 공공데이터포털, 한국도로공사, 전국졸음쉼터표준데이터. [Online] Available: https://www.data.go.kr/data/15028203/standard.do (Accessed: Jan. 1, 2024)
[9] 평면직각좌표계, [Online] Available: https://terms.naver.com/entry.naver?docId=3480877&cid=58439&categoryId=58439 (Accessed: Jan. 1, 2024)
[10] 지리좌표계, [Online] Available: https://terms.naver.com/entry.naver?docId=2762975&cid=50307&categoryId=50307 (Accessed: Jan. 1, 2024)
[11] 한다정, 김응철, “고속도로 졸음쉼터 진 · 출입 차량 주행속도 분석 및 적정 가 · 감속차로 길이 산정 연구”, 대한교통학회 학술대회지, pp. 222-225, 2018.
[12] 최수임, 유재춘, “고속도로 졸음운전자의 쉼터에 관한 연구”, 한국기초조형학회, pp. 147-150, 2019.

모빌리티 데이터를 활용한 MTGNN, AGCRN 모델의 하이퍼파라미터 최적화를 위한 예비 연구

양하늘O, 김효은, 정민서, 강종구
성신여자대학교 AI융합학부
{20221380, 20221347, 20221424, jonggu.kang}@sungshin.ac.kr

A pilot study on hyperparameter optimization in MTGNN and AGCRN models using mobility data

Yang HaneulO, Kim Hyoeun, Chung Minseo, Kang Jonggu
School of AI Convergence, Sungshin Women's University

요 약

모빌리티가 진화하고 보급이 확대되면서 도로 교통 혼잡도가 증가하고 있다. 도시 교통 예측의 정확도 향상을 위해서 확장 가능한 통합 라이브러리인 Libcity에서 제공하는 모델의 하이퍼파라미터를 변경하여 실험을 수행하였다. 본 논문에서는 기본 정의된 하이퍼파라미터와 매개변수 튜닝을 이용하여 교통 예측 모델 MTGNN, AGCRN의 정확도 변화를 분석한다. 그 결과, 최적의 값을 도출하기 위해서는 하이퍼파라미터 튜닝에 따른 값의 변동을 확인하며 적당한 값을 찾는 과정이 필요함을 확인하였다. 더 나아가 메타 휴리스틱을 적용한 모델을 사용함으로써 최적의 하이퍼파라미터 값을 도출하고, 교통 혼잡도 예측 모델의 성능을 향상시킬 것으로 기대된다.

1. 서 론

스마트 시티를 실현하기 위해서는 지능형 교통 시스템이 필수적이다. 또한, 모빌리티가 진화하고 보급이 확대되면서 도로 교통 혼잡도가 증가하는 추세이다. 이러한 상황에서 교통 흐름을 분산시키기 위해 교통 혼잡도를 사전에 예측하는 모델의 필요성이 증가하고 있다[1].

Libcity[2]는 현재 연구되고 있는 60개의 교통 예측 모델과 35개의 다양한 시공간 데이터 세트를 지원하여 종합적인 실험과 새로운 모델의 구축을 수행할 수 있게 하는 프레임워크이다. 따라서, 교통 혼잡도 예측 모델을 통한 효과적인 대응을 하기 위해 Libcity에 구축된 모델과 데이터세트를 활용하여 실험을 수행할 수 있다.

기계학습을 통한 예측 모델의 성능은 하이퍼파라미터 설정에 의해 달라진다고 알려져 있다 [3]. 최적의 하이퍼파라미터를 구하는 작업은 모델을 학습하는 데에 있어 많은 시간을 소요하는 작업이며, 다양한 시도들이 이루어지고 있다. 본 연구에서는 교통 예측 모델의 성능도 하이퍼파라미터의 영향을 받는지 실험을 수행하고, 교통 혼잡도 예측 성능의 향상을 위한 향후 방향을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 분석하고, 3장에서는 배경지식에 대해 기술한다. 4장에서는 데이터 세트와 하이퍼파라미터를 통하여 모델을 학습시켜보며 최적의 값을 도출한다. 5장은 4장에서 언급한 연구 질문에 대한 실험 결과를 기술하며, 6장에서는 토의를 통해 메타 휴리스틱 적용으로 본 모델을 개선할 수 있는 방향에 대해 설명한다. 7장은 위협요소에 대해, 8장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

박송희 et al.[4]에서는 CNN 모델과 교통 사고 정보 데이터를 활용하여 도로 혼잡도를 연구하였다. 유입되는 차량의 수를 통해 교통의 흐름을 파악하였고, 차량의 속도 상태로 교통 혼잡도를 분석하였다. 또한 예측하지 못한 다양한 이벤트들에 대한 값을 예측 값과 사건 값에 가중치를 반영하여 연구를 진행하였다.

박재성 et al.[5]에서는 다양한 기계학습 모델을 분석하고 알고리즘과 특성에 대해 연구하여 예측 시스템을 구현하기 위한 최적의 환경을 제안하였다. 선형 회귀를 적용하여 최적의 학습 속도를 구하고 반복 학습을 통하여 최적의 데이터 모델을 산출해 내고자 하였다. 그러나 모델에 직접적으로 영향을 주는 파라미터들의 값에 따라 정확도가 달라지므로 이에 대한 적절한 값의 설정이 필요하다.

본 논문에서는 Libcity에 내재되어 있는 자료들을 분석하여 성과 순위표의 상위 5개 순위에 있는 모델 중 교통 속도 모델로는 MTGNN, 교통 흐름 모델로 AGCRN을 활용하였다. 해당 모델들에 직접적으로 영향을 주는 학습률과 배치 크기를 조절하여 최적화된 값을 구하고자 한다.

3. 배경지식

Libcity[2]는 도시 시공간 예측을 위한 포괄적이고 확장 가능한 통합 라이브러리로 연구자들에게 교통 예측 분야에서 신뢰할 수 있는 실험 도구와 개발 프레임워크를 제공한다. 이는 PyTorch를 기반으로 구현되어 있으며 교통 예측과 관련된 모든 요소를 포함하여 연구자가 포괄적인 실험을 수행할 수 있도록 한다. 이는 9개의 교통 예측 작업을 포함하는 60개의 모델을 재현하여 교통 예측 분야 연구의 표준화와 재현성에 기여하였다.

아래와 같은 교통 예측 모델들을 지원한다.

- 트래픽 상태 예측
  - 트래픽 흐름 예측
  - 교통 속도 예측
  - 온디맨드 서비스 예측
  - 출발지-목적지 행렬 예측
  - 교통사고 예측
- 궤적 다음 위치 예측
- 도착 예정 시간
- 맵 매칭
- 도로망 표현 학습

4. 데이터 및 실험

본 장에서는 Libcity를 이용하여 4.1절의 연구 질문에 대한 답을 구하기 위해서 수행한 실험에 대해 설명한다.

4.1 연구 질문

RQ1 : 교통 속도 예측 성능에 하이퍼파라미터가 영향을 주는가?

RQ2 : 교통 흐름 예측 성능에 하이퍼파라미터가 영향을 주는가?

4.2 데이터 세트

35개의 데이터 세트 중에서 교통 혼잡도를 분석하기 위한 데이터 세트로 이번 실험에서는 PEMSD4와 PEMSD8를 사용한다. 실험에 사용되는 데이터 세트는 아래의 표 1과 같다. 학습 세트와 검증 세트, 그리고 테스트 세트를 각각 80%, 10%, 10%로 분할하여 학습시켰다. 실험 환경은 인공지능 산업 융합 사업단에서 제공하는 국가 클라우드 환경을 사용하였다.

표 1. 레퍼런스의 데이터 구성 요소

	PEMSD4	PEMSD8
노드(Node)	307	170
# 샘플	16,992	17,865
수집 기간/간격	2개월 / 5분	2개월 / 5분
특징	샌프란시스코 베이 지역, 미국 고속도로의 속도 흐름 데이터	미국 샌버나디노 지역 고속도로의 속도 흐름데이터

4.3 평가 척도

교통 흐름과 속도 예측 모델을 학습시켜 도출된 결과의 평가 척도로 MAE, RMSE, R^2을 선정하였다. 다음은 평가 척도의 수식을 나타낸다.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}i - yi| \quad (1)$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}i - yi)^2} \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(yi-\hat{y}i)^2}{\sum_{i=1}^{n}(yi-\bar{y})^2} \quad (3)$$

수식 (1)은 예측 값과 실제 값 간의 절대적인 오차의 평균으로 오차의 크기를 나타내며, 값이 작을수록 모델의 예측이 정확하다. 수식 (2)는 예측 값과 실제 값 간의 제곱 오차의 평균의 제곱근이다. 이는 오차에 큰 가중치를 부여하며, 값이 작을수록 모델의 예측이 정확함을 의미한다. 수식 (3)은 결정 계수로써 예측 값이 실제 값을 얼마나 잘 설명하는지 나타내는 지표이다. 이때 최댓값은 1이며, 높을수록 모델이 데이터를 잘 설명하고 있다고 판단한다.

4.4 비교 모델

60개의 교통 예측 모델 중 선행 연구[2]에 의하면 각 모델들끼리 성능 순위를 매긴 결과, 교통 속도와 교통 흐름의 모델에 각각 적응형 그래프 인접 행렬 방법을 도입한 MTGNN(Multi-Task Graph Neural Network), AGCRN(Adaptive Graph Convolutional

Recurrent Network)이 상위 모델을 차지함을 알 수 있다. MTGNN은 시계열 간의 관계를 마이닝 하는데 효과적인 예측 방법임이 입증되었다[6]. AGCRN은 단기 예측과 장기 예측의 균형을 잘 맞추며 거의 모든 과정에서 최고의 성능을 달성한다. 이에 기존 MAE 및 MAPE의 결과를 개선한다[7]. 따라서 본 연구에서는 두 모델을 활용하여 연구를 진행하였다. 다음은 두 모델에 대한 설명이다.

MTGNN[8]: 그래프 학습 계층, 그래프 컨볼루션 모듈, 출력 모듈로 구성된다. 노드 간 숨겨진 연관성을 발견하기 위해 그래프 학습 계층은 그래프 인접 행렬을 계산하고, 이 행렬은 나중에 모든 그래프 컨볼루션 모듈의 입력으로 사용된다.

AGCRN[7]: 트래픽 계열에서 노드 별 공간적, 시간적 상관관계를 세밀하게 파악할 수 있다. AGCRN을 훈련하면 각 트래픽 계열 소스에 대해 의미 있는 노드 표현 벡터를 얻을 수 있다.

4.5 하이퍼파라미터

하이퍼파라미터는 기계학습에서 모델의 성능을 결정하는 값으로, 좋은 성능을 위해서는 적절한 하이퍼파라미터 값을 설정해야 한다[9][10]. 하이퍼파라미터의 종류에는 학습률, 배치 사이즈, 은닉층의 개수 등이 있다. 본 연구에서는 여러 하이퍼파라미터 중 모델에 영향을 가장 크게 미치는 학습률과 배치 크기를 선정하였다. 특히, 신경망의 가중치 계산에 직접적으로 사용되는 학습률은 학습의 정확도에 큰 영향을 미쳐 최적화할 대상으로 선정하였다[9]. 학습률은 0.1, 0.01, 0.001로 조절하며 학습시켰다. 그 외에 주요한 하이퍼파라미터인 배치 크기[11]는 64, 128, 256, 512로 조절해 학습시켰다.

5. 실험 결과

5.1 (RQ1) : 교통 속도 예측 성능에 하이퍼파라미터가 영향을 주는가?

교통 속도 예측 모델인 MTGNN으로 실험하였을 때, 그림 1에서 데이터 세트 PEMSD4로 학습한 결과를 배치 사이즈 기준으로 비교한다. 배치 사이즈를 64로 설정했을 때, MAE 값은 최솟값 69.08, 중앙값 69.19, 최댓값 86.93으로 다른 배치 사이즈의 값 중 제일 작으며 RMSE 값 또한 최솟값 101.49, 중앙값 101.63, 최댓값 130.70으로 가장 작아 배치 사이즈가 64일 때 모델의 성능이 가장 뛰어나다. R^2 값은 최솟값 -0.013, 중앙값 0.2453, 최댓값 0.2456으로 통합 수치가 1에 가장 가까워 배치 사이즈가 64일 때 모델이 데이터를 잘 설명한다.

이러한 이유로 배치 사이즈를 64로 설정했을 때 최고 성능으로 나타나며, 배치 크기가 평가 지표 값을 좌우한다는 것을 알 수 있다.

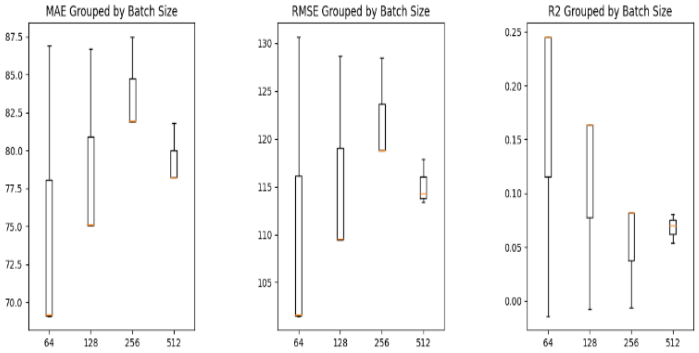


그림 1. MTGNN 모델 + PEMSD4 배치 사이즈 조절

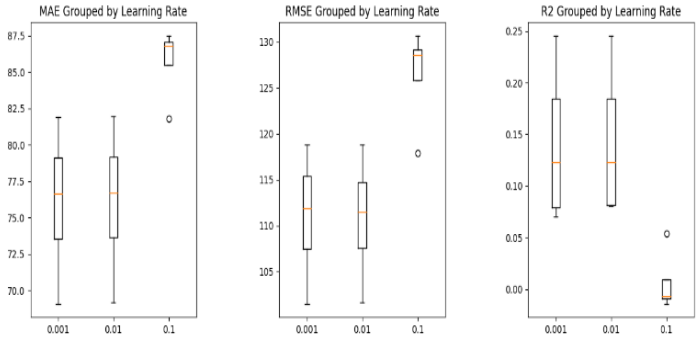


그림 2. MTGNN 모델 + PEMSD4 학습률 조절

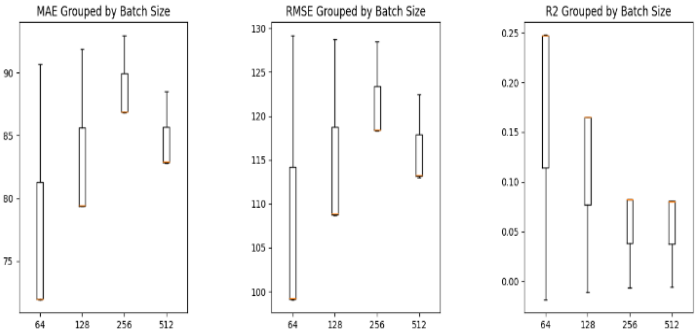


그림 3. MTGNN 모델 + PEMSD8 배치 사이즈 조절

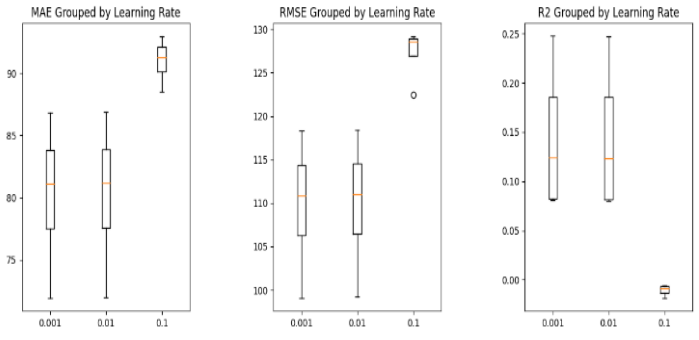


그림 4. MTGNN 모델 + PEMSD8 학습률 조절

그림 2에서 데이터 세트 PEMSD4의 경우, 학습률을 기준으로 값을 비교한다. 학습률이 0.001과 0.01일 때의 MAE 값, RMSE 값, R^2 값 모두 유사하게 나타난다. 하지만, 학습률을 0.1로 설정했을 때, MAE 값은 최솟값 81.80, 중앙값 86.83, 최댓값 87.52로 다른 학습률과 비교했을 때 현격하게 차이가 난다. RMSE 값 또한 최솟값 101.49, 중앙값 111.87, 최댓값 118.77로 가장 큰 값을 가져 학습률이 0.1일 때 모델의 성능이 가장 떨어진다. R^2 값은 최솟값 −0.013, 중앙값 −0.0068, 최댓값 0.054로 5개의 실험 결과 중 4개가 음수 값을 가지며 학습률이 0.1일 때 데이터에 대한 모델의 설명력이 떨어진다. 이러한 결과를 보면, 학습률의 크기에 따라 평가 지표 값이 크게 변함을 확인할 수 있다.

그림 3과 4에서 MTGNN에 데이터 세트 PEMSD8을 학습한 결과를 보면, 배치 사이즈가 64일 때 가장 좋은 성과를 내며 학습률은 0.1일 때 가장 낮은 성능을 보인다. 따라서 데이터 세트 PEMSD8에서 또한 학습률과 배치 사이즈와 같은 하이퍼파라미터가 교통 속도 예측 성능에 많은 영향을 준다는 것을 알 수 있다.

5.2 (RQ2) : 교통 흐름 예측 성능에 하이퍼파라미터가 영향을 주는가?

실험 결과, 그림 5에서 AGCRN 모델에 데이터 세트 PEMSD4의 학습 결과를 배치 사이즈 기준으로 비교한다. 배치 크기가 512일 때는 메모리 부족으로 인해 학습이 어려웠다. 배치 사이즈를 64로 설정했을 때, MAE 값은 최솟값 6.96, 중앙값 8.03, 최댓값 25.55로 다른 비교 배치 사이즈의 값 중 제일 작으며 RMSE 값 또한 최솟값 17.48, 중앙값 19.57, 최댓값 40.67로 가장 작은 값을 가져 배치 사이즈가 64일 때 모델의 성능이 우수하다. R^2 값은 최솟값 0.90, 중앙값 0.976, 최댓값 0.98로 통합 수치가 1에 가장 가까워 배치 사이즈가 64일 때 모델이 데이터를 잘 설명한다.

이러한 이유로 배치 사이즈를 64로 설정했을 때 최고 성능으로 나타나며, 배치 크기로 평가 지표 값들을 좌우한다는 것을 알 수 있다.

그림 6에서 데이터 세트 PEMSD4의 경우, 학습률을 기준으로 값을 비교한다. 학습률이 0.01일 때 MAE 값, RMSE 값 모두 가장 작은 값을 가져 성능이 가장 높으며 R^2 값 또한 가장 1에 가까운 값을 가져 모델의 설명력이 높다. 하지만, MTGNN과 같이, 학습률을 0.1로 설정했을 때 MAE 값과 RMSE 값이 다른 학습률과 비교했을 때 현격하게 차이가 난다. R^2 값 또한 다른 비교 값들에 비해 낮은 값을 가져 학습률이 0.1일 때 모델의 성능이 가장 떨어진다는 것을 알 수 있다. 결과적으로 학습률의 크기에 따라 평가 지표 값의 변화가 크게 드러난다.

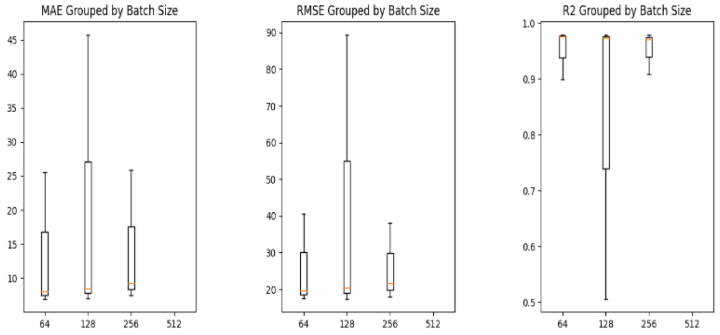


그림 5. AGCRN 모델 + PEMSD4 배치 사이즈 조절

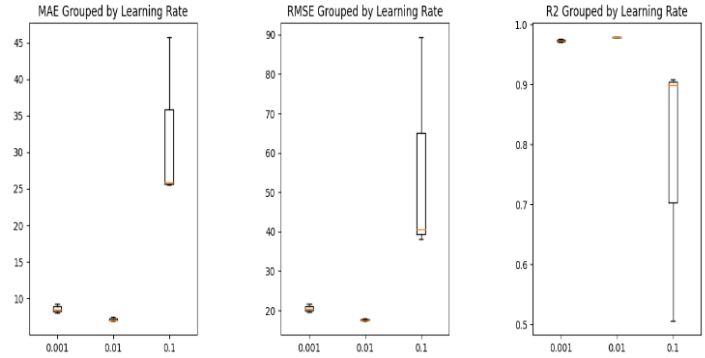


그림 6. AGCRN 모델 + PEMSD4 학습률 조절

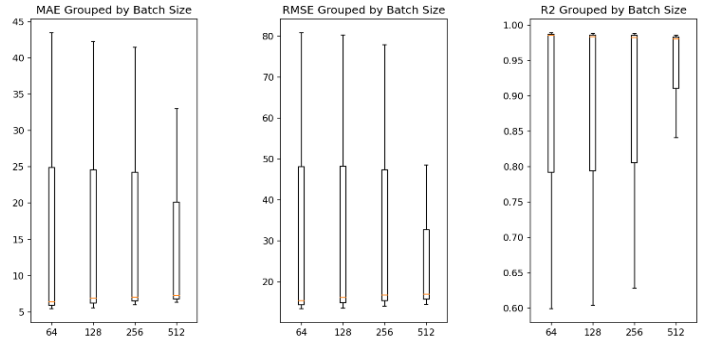


그림 7. AGCRN 모델 + PEMSD8 배치 사이즈 조절

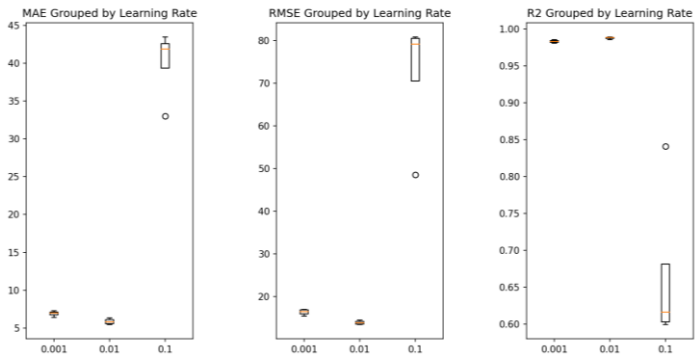


그림 8. AGCRN 모델 + PEMSD8 학습률 조절

그림 7과 8에서 AGCRN에 데이터 세트 PEMSD8을 학습한 결과를 보면, 대체로 배치 사이즈가 64일 때 가장 좋은 성과를 내며 학습률은 0.1일 때 가장 낮은 성능을 보인다. 따라서 데이터 세트 PEMSD8에서 또한 학습률과 배치 사이즈와 같은 하이퍼파라미터가 교통 흐름 예측 성능에 많은 영향을 준다는 것을 알 수 있다.

6. 토의

하이퍼파라미터 최적화를 위해 메타 휴리스틱을 고려할 수 있다[12]. 메타 휴리스틱의 목표는 최적의 솔루션을 찾기 위해 검색 공간을 효율적으로 탐색하는 것이다[13]. 메타 휴리스틱 알고리즘에는 Grid search, Random search, Genetic algorithm, Particle Swarm Optimization, Harmony Search Algorithm 등 다양한 알고리즘이 있다. 최적의 값을 도출하여 더 나은 모델을 개발하기 위해서는 해 공간의 여러 지역을 동시에 탐색하여 적절한 해 주위를 집중적으로 모색할 수 있다. 이를 도입한다면, 기존의 모델 수렴 속도 면에서 보다 더 뛰어나며 더 빠르게 근사치의 값을 찾아낼 수 있다[13][14]. 또한 안정성과 정확성 측면에서도 우수하다는 것을 찾아볼 수 있다[15].

7. 위협 요소

교통 흐름 예측과 교통 속도 예측을 위해 2개의 모델을 사용하였고 이에 적용한 데이터 세트와 변경한 하이퍼파라미터 개수 또한 2개였다. 이들은 모두 Libcity에 내재하고 있는 모델과 데이터들이라는 것을 간주하고 실험을 진행하여 외부 데이터 및 모델 활용에 한계가 있다. 또한 하이퍼파라미터는 모델의 출력 값에 큰 영향을 주기 때문에 하이퍼파라미터를 추가하거나 변경할 경우 다른 결과값이 나올 수 있다. 이에 추후에 메타 휴리스틱을 활용하여 최적의 하이퍼파라미터를 도출할 계획이다.

8. 결론 및 향후 계획

본 연구는 모델에 직접적인 영향을 주는 하이퍼파라미터를 조절하여 최적의 값을 구하기 위한 예비 연구를 하였다. 교통 흐름 예측 모델과 교통 속도 예측 모델을 활용하여 기본 하이퍼파라미터와 매개변수 튜닝을 통하여 교통 예측 모델 MTGNN과 AGCRN의 정확도 변화를 분석했다. 그 결과, 하이퍼파라미터가 성능 지표에 영향을 주는 것을 확인하였으며 최적의 성능을 도출하기 위해서는 적당한 하이퍼파라미터의 조절이 필요함을 알 수 있었다.

향후 연구에는 적절한 값을 도출하기 위하여 메타 휴리스틱을 적용한 모델을 사용하여 최적의 하이퍼파라미터 값을 도출하고, 교통 혼잡도 예측 모델을 향상시킬 계획이다.

Acknowledgement

이 논문은 성신여자대학교 2023년도 동계 학부생 연구 참여 프로그램(UROP)의 지원을 받아 수행된 연구임

참고 문헌

[1] 강동묵, 윤상훈, 정한균, 임기택, 김주영, 장수현. 국내 고속도로의 교통 혼잡도 분석을 위한 교통 데이터 및 예측 모델 적용성 연구. 대한전자공학회, 97, 2,438-2,440 (3 pages). (2020).

[2] Wang, J., Jiang, J., Jiang, W., Li, C., & Zhao, W. X. LibCity: An Open Library for Traffic Prediction. SIGSPATIAL '21: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, 145-148, (2021).

[3] Jiang, J., Han, C., Jiang, W., Zhao, W.X., &Wang, J. LibCity: A Unified Library Towards Efficient and Comprehensive Urban Spatial-Temporal Prediction. arXiv. (2023).

[4] 박송희, 최도진, 복경수, 유재수. 안전 교통 서비스를 위한 스마트 교통 혼잡도 예측 모델 설계. 한국통신학회 학술대회논문집, 466-467. (2019).

[5] 박재성, 박성수. 기계학습 모델을 적용한 대중교통 혼잡도 예측 시스템 구축 방안에 관한 연구. 한국통신학회 학술대회논문집, 304-305. (2015).

[6] Chen, D., Tang, T., & Yao, Y. Research on prediction algorithm of ship equipment heath condition. Ocean Engineering, 249, 110750, (2022).

[7] Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems, 33, 17804-17815, (2020).

[8] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. "Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20), 753-763. (2020).

[9] 원종현, 신종민, 김재호, 이장원. 기계학습의 하이퍼파라미터 최적화 연구 동향. 한국통신학회논문지,

48(6), 733-747, (2023).
[10] 김도영, 김태훈, 윤준희. 딥러닝 기술 활용을 위한 데이터셋 구축방법 연구 - CNN을 활용한 데이터셋과 하이퍼파라미터 조정을 통한 정확도 개선 -. 디지털콘텐츠학회논문지, 24(2), 343-351, (2023).
[11] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," ICT Express, vol. 6, no.4, pp. 312-315, (2020).
[12] Kang, J., Kwon, S., Ryu, D., & Baik, J. (2021). HASPO: Harmony search-based parameter optimization for just-in-time software defect prediction in maritime software. Applied Sciences, 11(5), (2002).
[13] Desale, S., Rasool, A., Andhale, S., & Rane, P. Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. Int. J. Comput. Eng. Res. Trends, 351(5), 2349-7084. (2015).
[14] Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. Metaheuristic research: a comprehensive survey. Artificial intelligence review, 52, 2191-2233. (2019).
[15] 이세정. 공학 설계에서 매개변수 설정이 필요 없는 메타휴리스틱 전역 탐색 방법. 한국CDE학회 논문집, 22(3), 223-233. (2017).

부록 – 하이퍼파라미터에 따른 성능 비교 결과

부록 1. MTGNN 모델에 PEMSD4 학습 결과

학습률	배치	MAE	RMSE	R2
0.001	64	69.089472	101.494805	0.245593
	128	75.053120	109.469456	0.163874
	256	81.903462	118.767487	0.081961
	512	78.200338	114.270180	0.070284
0.01	64	69.186822	101.631590	0.245300
	128	75.096122	109.531347	0.163750
	256	81.939042	118.796389	0.081900
	512	78.248121	113.382040	0.080760
0.1	64	86.929124	130.697773	-0.013794
	128	86.715416	128.616020	-0.007743
	256	87.516520	128.482002	-0.005769
	512	81.804466	117.908273	0.054284

부록 2. MTGNN 모델에 PEMSD8 학습 결과

학습률	배치	MAE	RMSE	R2
0.001	64	71.962269	99.098333	0.247588
	128	79.372500	108.710774	0.165103
	256	86.820387	118.349519	0.082582
	512	82.811473	112.996975	0.080841
0.01	64	72.021227	99.267393	0.247381
	128	79.424709	108.825368	0.164910
	256	86.894838	118.431182	0.082459
	512	82.904245	113.254964	0.080381
0.1	64	90.639623	129.212431	-0.018352
	128	91.855494	128.776720	-0.011493
	256	92.980679	128.441025	-0.006236
	512	88.478243	122.492517	-0.006088

부록 3. AGCRN 모델에 PEMSD4 학습 결과

학습률	배치	MAE	RMSE	R2
0.001	64	8.0294867	19.573249	0.976308
	128	8.4552544	20.325169	0.973872
	256	9.242226	21.690233	0.970286
	512	-	-	-
0.01	64	6.960206	17.477514	0.979320
	128	7.029431	17.429347	0.979535
	256	7.445844	17.841615	0.979461
	512	-	-	-
0.1	64	25.551473	40.666470	0.899663
	128	45.802344	89.467829	0.506721
	256	25.847410	38.082116	0.908134
	512	-	-	-

부록 4. AGCRN 모델에 PEMSD8 학습 결과

학습률	배치	MAE	RMSE	R2
0.001	64	6.367183	15.398937	0.985248
	128	6.868793	16.180413	0.983691
	256	7.057851	16.795498	0.982397
	512	7.260056	16.945249	0.980299
0.01	64	5.449163	13.441953	0.988800
	128	5.587084	13.579751	0.988572
	256	5.986530	14.090578	0.987699
	512	6.353796	14.531059	0.985581
0.1	64	43.470825	80.840788	0.599645
	128	42.328130	80.392418	0.604356
	256	41.462144	77.944852	0.628306
	512	33.021064	48.519374	0.841176

ARM-Net 기반의 소프트웨어 결함 예측

이정화○, 주은정, 류덕산*

전북대학교 소프트웨어공학과

{dlwjdghk133, jeju3146, duksan.ryu}@jbnu.ac.kr

Software Defect Prediction based on ARM-Net

Jeonghwa Lee○, Enjeong Ju, Duksan Ryu*

Department of Software Engineering, Jeonbuk National University

요 약

소프트웨어 결함 예측(SDP)은 결함이 발생하기 쉬운 모듈을 식별하여 소프트웨어 품질을 향상시키고 제한된 자원을 효과적으로 사용하는 데 도움을 준다. 최근 연구 사례를 살펴보면 SDP 성능 향상을 위해 딥러닝 기법을 적용한 사례가 많다. 특히 정형데이터 분석에 있어 높은 성능을 보인 딥러닝 기법들이 SDP에서도 우수한 성능을 보이고 있다. ARM-Net은 적응적인 방식으로 특성 간의 관계를 학습하므로 다양한 데이터 셋에 대응하는 능력이 우수하다. 본 연구에서는 ARM-Net을 SDP에 적용하여 타 모델 대비 성능의 우수함을 확인하고자 한다. 성능 비교를 위해 CatBoost, XGBoost, RandomForest와 비교 실험한다. 실험 결과 ARM-Net은 PF 지표에서 XGBoost 대비 좋은 성능을 보였고 FIR 지표에서 CatBoost 대비 좋은 성능을 보였으며, PD와 Balance 지표에서는 비교 모델보다 낮은 성능을 보였다. ARM-Net의 SDP 적용을 위해서는 성능 향상을 위한 연구가 필요함을 확인하였다.

1. 서 론

산업에서 소프트웨어의 중요성이 증가함에 따라 품질 향상에 대한 요구가 커지고 있다. 품질 향상을 위해서는 결함 식별이 중요하며 결함은 컴퓨터 프로그램에서 발생한 부정확한 단계, 프로세스 또는 데이터 정의로 인해 프로그램이 예상치 못한 방식으로 작동하는 현상으로 컴파일 오류는 발생하지 않지만 예상치 못한 결과를 초래한다. 이에 소프트웨어 결함 예측(SDP)은 품질 향상에 기여하는 중요한 연구 분야로 각광 받고 있다.

SDP에서 중점적인 부분은 결함 예측 성능을 높이기 위한 결함 예측 기법을 찾는 것이다. 이를 위해 일반적으로 Machine Learning(ML) 모델을 사용하며 ML 모델이 안정적인 예측 성능을 보였다. 최근에는 Deep Learning(DL) 모델을 SDP에 적용하는 연구 사례가 증가하고 있다.

본 논문에서는 SDP를 효과적으로 수행하기 위해 최신 DL 기법인 ARM-Net 기법[1]을 SDP에 적용하고자 한다. ARM-Net은 관계형 데이터에 적합하게 설계된 DL 기법으로 데이터의 특성과 관계를 적응적으로 파악하여 효율적이고 쉽게 해석이 가능한 예측을 제공한다. ARM-Net은 정형데이터에 적용되어 큰 효과를 보인 모델로 아직 SDP에 적용된 사례가 없다. 이에 이 모델의 효용성을 확인하고자 예측 성능을 실험한다. 더 나아가 SDP에 적합한 ARM-Net의 하이퍼파라미터를 탐색 및 분석한다. 본 연구를 통해 SDP에서 ARM-Net이 적용되려면 추가적인 성능 개선 연구가 필요함을 확인했다.

이 논문은 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF- 2022R1I1A3069233)과 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과임. (2023RIS-008)

*교신저자임

2. 관련 연구

많은 DL이 다양한 도메인에 적용되어 좋은 결과를 얻음에 따라 SDP에도 DL이 적용되고 있다.

Qiao et al.[2]은 소프트웨어 모듈의 결함 수를 예측하기 위해 딥러닝 모델을 제안했다. 제안한 DPNN 모델은 ML 기법인 SVR(Support Vector Regression), DTR(Decision Tree Regression) 대비 평균 제곱 오차를 14% 이상 감소시켰으며 제곱 상관 계수를 8% 이상 증가시켰다.

Batool et al.[3]은 세 가지 딥러닝 방법, 즉, LSTM(long short-term memory), BILSTM(bidirectional LSTM), RBFN(radial basis function network)을 사용하여 소프트웨어 결함 예측을 진행하였고 기존 ML 모델과 비교하여 LSTM, BILSTM이 우수한 예측 성능을 보였고, RBFN은 필요한 결과를 빠르게 생성하는 데 효과적이었다.

이처럼 두 연구는 모두 DL을 SDP에 적용해 우수한 성능을 보였다. 하지만 Qiao et al.[2]은 기법의 성능 평가를 위해 평균 제곱 오차와 제곱 상관 계수 두 개 밖에 사용하지 않았고 Batool et al.[3]은 Chidamber와 Kemerer (CK) 메트릭스 기반의 데이터 셋 두 가지 종류의 데이터 셋 밖에 이용하지 않았다. 이에 본 연구에서는 DNN,GCN 등과 같은 여러 딥러닝 기법들 대비 높은 정확도를 보인 ARM-Net을 SDP에 적용하여 PD, PF, Balance, FIR 4가지 성능 지표를 통해 예측 성능을 확인하고자 한다. 또한 모델의 성능을 평가하기 위해 오픈 소스 프로젝트인 AEEEM, Relink 외에도 자동차 프로젝트인 AUDI 데이터 셋을 이용한다.

3. 연구 방법

본 연구에서는 ARM-Net을 SDP에 적용한다. 교차검증을 위해 데이터를 학습 데이터와 테스트 데이터로 나누어 학습 데이터를 학습한 모델에 테스트 데이터의 예측 성능을 평가한다. 그림 1은 전체 연구 방법이며, Algorithm 1은 ARM-Net의 학습 과정이다. 피처 스케일링을 위해 정규화 기법인 MIN_MAX를 활용하여 학습 데이터와 테스트 데이터의 범위를 0과 1 사이로 축소한다(1~2행). 이후 합성 소수 샘플링(SMOTE)[4]을 통해 학습

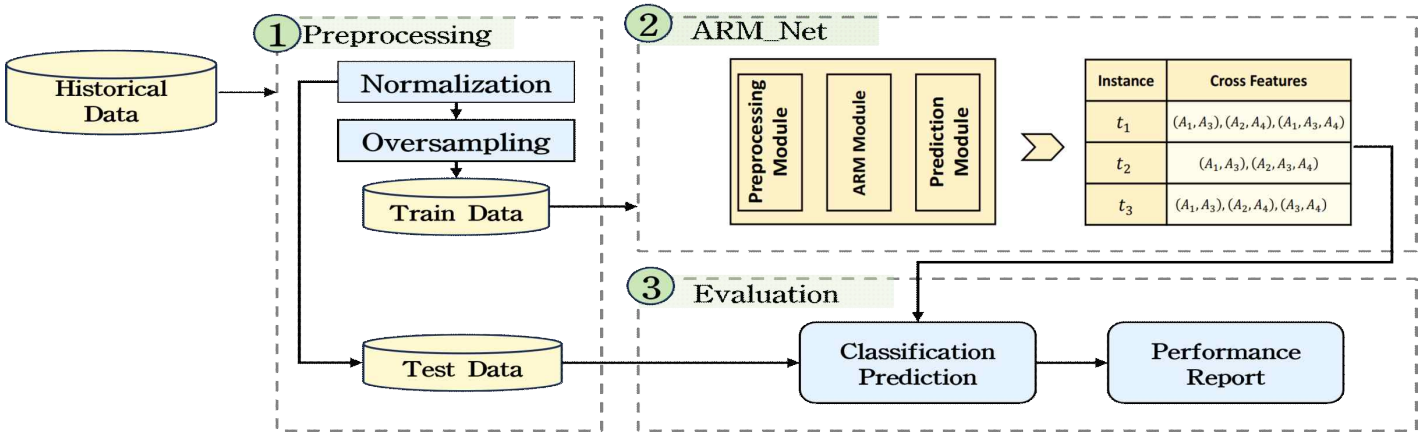


그림 1. 연구 방법

데이터를 1:1 비율로 오버 샘플링 한 뒤 ARM-Net을 학습시킨다(3~4행). ARM-Net은 구조화된 데이터에서 발생하는 복잡한 특성 간의 상호작용을 적응적으로 모델링하여 예측 성능을 향상시키고, 이를 통해 모델의 결과를 쉽게 이해할 수 있게 한다. 입력 데이터 X를 임베딩하고 임베딩한 결과를 X_arm에 저장한다(6~7행).

Algorithm 1. ARM−Net

Input Train data X

Output Data Y predicted for fault

```
1: /* Preprocess X                                   */
2: Min−Max ← X
3: X_Nomalization = X
4: Smote ← X_Nomalization
5: X_Oversampled = X_Nomalization

6: X_Embedding = X_Nomalization
7: X_arm = X_Embedding
8: /* Calculate weights                             */
9: X_arm is Bilinear
10: keys = X_arm
11: Get attention gate value using keys, query
12: Sparsemaxfunction ← attention gate value
13: arm_weight = Sparsemax function

14: attention value = X_arm*arm_weight
15: attention value is Batch Nomalization
16: Exponential function ← attention value
17: last attention value = Exponential function
18: MLP layer ← last attention value
19: Y = MLP layer (Generating final predictions)
20: /* Predict defects                              */
```

X_arm을 Bilinear 연산을 통해 keys를 생성한다. 어텐션 게이트 값을 키와 쿼리를 사용하여 계산하고 이를 Sparsemax 함수에 적용하여 어텐션 가중치를 얻는다(8~13행). 어텐션 가중치 값(attention weight)과 입력값을 곱하여 어텐션 값(attention value)을 계산한다(14행). 어텐션 값(attention value)에 배치 정규화를 수행한다(15행). Exponential 함수를 어텐션 값(attention value)에 적용하여 최종 어텐션 값(last attention value)을 얻는다(16~17행). 최종 어텐션 값(last attention value)에 MLP layer를 적용하여 최종 예측값을 얻는다(18~20행).

$$\widetilde{z_{ij}} = \phi att(q_i, e_j)$$
$$z_i = \alpha - entmax(\widetilde{z_i}), \widetilde{z_i} \in R^m \quad (1)$$

수식 (1)은 어텐션 가중치를 구할 때 사용되는 메커니즘이다. $\widetilde{z_{ij}}$는 어텐션 게이트, ϕatt는 bilinear 함수 q_i는 query 벡터, e_j는 embedding 벡터를 나타낸다. z_i는 어텐션 가중치 값이고 $\alpha - entmax$는 Sparsemax 함수를 의미한다.

4. 실험 설정

이 섹션에서는 연구 질문과 데이터 셋, 평가 척도에 대한 내용을 기술한다.

4.1 연구 질문

RQ1: ARM-Net이 타 기법 대비 결함 예측 성능이 우수한가?
− H_{10}: ARM-Net의 결함 예측 성능이 타 기법과 유사하다.
− H_{1A}: ARM-Net의 결함 예측 성능이 타 기법 대비 우수하다.
RQ2: ARM-Net의 하이퍼파라미터가 결함 예측 성능에 영향이 있는가?
− H_{20}: ARM-Net의 하이퍼파라미터가 결함 예측 성능에 영향이 없다.
− H_{2A}: ARM-Net의 하이퍼파라미터가 결함 예측 성능에 영향이 있다.

본 논문에서는 ARM-Net의 결함 예측 성능을 평가하기 위해 RQ1, RQ2에 대한 귀무 가설(H_0)과 대립 가설(H_A)을 세우고 대립 가설을 채택할 수 있는지 통계적 검정을 통해 증명한다.

4.2 데이터

모델의 성능을 평가하기 위해 여러 데이터 셋(AEEEM, Relink, AUDI)를 이용한다 각 데이터의 세부적인 정보는 표 1과 같다. AEEEM, Relink는 오픈소스 프로젝트이며, AUDI는 자동차 소프트웨어 프로젝트이다.

표 1 실험 데이터 셋

Dataset	Project	#of instance		#of metric	Prediction Granularity
		AII	Buggy		
AEEEM	EQ	324	129(39.81%)	61	class
	JDT	997	206(20.66%)	61	class
	LC	691	64(9.26%)	61	class

Relink	apache	194	98(50.52%)	26	isDefective
	zxing	399	118(29.57%)	26	isDefective
	safe	56	22(39.28%)	26	isDefective
AUDI	ProjectA	1908	77(4.03%)	12	bug
	ProjectK	2515	112(4.45%)	12	bug
	ProjectL	2891	61(2.11%)	12	bug

4.3 전처리

본 논문에서는 모든 모델을 학습시키기 이전, MIN_MAX 정규화 과정을 거쳐 데이터의 범위를 0과 1의 사이로 축소하는 피처 스케일링을 거친다. 이후 SMOTE를 통해 1:1 비율로 학습 데이터를 학습시킨다.

4.4 성능 평가 지표

표 2 혼동 행렬

		Predicted class	
		Defective	Clean
Actual class	Defective	TP(True Positive)	FN(False Negative)
	Clean	FP(False Positive)	TN(True Negative)

본 연구에서는 평가 지표로 혼동 행렬 중 예측을 Positive로 한 대상 중에 예측값과 실제값이 Positive로 일치한 데이터의 비율을 뜻하는 PD, 모든 Negative 중 Positive로 잘못 예측된 Negative 데이터 수를 의미하는 PF, 클래스 불균형 환경에서 적합한 Balance($=1-\frac{\sqrt{(0-PF)^2+(1-PD)^2}}{\sqrt{2}}$)[5]를 사용한다. 또한 코드 검사 노력도 감소 효과를 분석하기 위해 FIR(File Inspection Reduction $=\frac{PD-FI}{PD}$)[6]을 측정하였다. "False Identification (FI)"는 주로 패턴 인식, 분류 및 탐지 시스템에서 사용되는 용어로, 시스템이 실제로 없는 것을 존재하는 것으로 잘못 식별하는 비율이다.

5. 실험 결과

5.1 RQ1:ARM-Net이 타 기법 대비 결함 예측 성능이 우수한가?

본 연구에서는 ARM-Net과 CatBoost, XGBoost, RandomForest 비교 실험을 진행했다. CatBoost, XGBoost, RandomForest는 기존 소프트웨어 결함 예측에서 우수한 성능을 보여준 기법이기에 ARM-Net의 비교 대상으로 선정하였다.

표 3 ARM-Net과 다른 모델과의 성능 비교

Metric	Model			
	ARM_NET	CatBoost	XGBoost	Randomforest
PD	0.6359	0.6726	0.6927	0.6921
PF	0.2141	0.1923	0.2755	0.1781
Balance	0.8636	0.8846	0.896	0.8954
FIR	0.5688	0.5319	0.6109	0.5757

표 3은 그 결과이며 ARM-Net이 PF 지표에서 XGBoost 대비 좋은 성능을 보였지만 CatBoost, Randomforest 대비 성능이 낮았다. 또한, FIR 지표에서 CatBoost 대비 좋은 성능을 보였지만 XGBoost, Randomforest 대비 성능이 낮았다. CatBoost, XGBoost, Randomforest가 과적합 문제 해결 측면에서 뛰어나기 때문에 ARM-Net 기법의 성능이 상대적으로 낮게 나왔다.

또한, 어떤 프로젝트에서 성능이 우수한지에 대해서 자세히 알아보기 위해 데이터 프로젝트 별 네 가지의 지표에서 최고 성능을 가지는 모델과 그 모델이 가지는 수치를 도출했다. 그 결과는 아래 표 4와 같다.

표 4 각 데이터 셋의 최고 성능 모델 및 수치

Dataset	Project	Highest Performance Model by Project							
		PD		PF		Balance		FIR	
AEEEM	EQ	XG Boost	0.807	ARM_ NET	0.340	XG Boost	0.886	Random forest	0.268
	JDT	XG Boost	0.753	Random forest	0.092	Cat Boost	0.956	Random forest	0.666
	LC	XG Boost	0.513	ARM_ NET	0.067	XG Boost	0.872	Random forest	0.710
Relink	safe	Random forest	0.650	Cat Boost	0.400	XG Boost	0.783	Random forest	0.833
	apache	Random forest	0.688	Random forest	0.359	Random forest	0.886	Random forest	0.270
	zxing	XG Boost	0.479	Random forest	0.139	XG Boost	0.804	Random forest	0.399
AUDI	Project A	ARM_ NET	0.836	ARM_ NET	0.002	ARM_ NET	0.986	Cat Boost	0.963
	Project K	Random forest	0.981	Random forest	0.014	Cat Boost	0.100	Random forest	0.850
	Project L	ARM_ NET	0.992	Cat Boost	0.002	Random forest	0.100	Cat Boost	0.978

데이터 프로젝트 별로 성능을 살펴본 결과, 오픈 소스 프로젝트에서는 기존 머신러닝 모델인 CatBoost, XGBoost, RandomForest가 성능에서 우위를 보였으며, 그 중 RandomForest가 모든 지표에서 성능이 우수함을 보였다.

자동차 소프트웨어 프로젝트에서는 ARM-Net이 Project A에서 FIR을 제외한 모든 지표에서 성능이 우수함을 보였다.

표 5 effect_size 비교

ARM_NET	Measure			
	PD	PF	Balance	FIR
CatBoost	0.1498(S)	0.1130(S)	0.1944(S)	0.1047(S)
XGBoost	0.2769(S)	0.2386(S)	0.3400(S)	0.1262(S)
Randomforest	0.2456(S)	0.1757(S)	0.3211(S)	0.0203(S)

표 5는 ARM-Net과 다른 기법들 사이의 효과 크기를 비교한 결과이다. 효과 크기는 *Cohen's D* 공식을 사용한다. ARM-Net은 Balance 지표에서 CatBoost, XGBoost, Randomforest 모델과 Small size 수준의 차이를 보였다.

5.2 RQ2:ARM-Net의 하이퍼파라미터가 결함 예측 성능에 영향이 있는가?

표 6 파라미터 범위와 기본값

Parameter	Range	default
mlp_layer	{1,2,3,4,5}	2
Epochs	{100,500,1000}	100

표 6은 ARM-Net의 파라미터 mlp_layer와 Epochs의 범위와 기본값을 나타낸 것이다.

mlp_layer는 예측 헤드의 층 수로 예측 헤드는 모델의 마지막 부분으로, 주로 모델의 최종 출력을 생성하는 부분이다. 예측 헤드의 층 수를 늘리거나 줄임으로 인해 모델의 성능을 조절할 수 있다. 층이 많을수록 더 복잡한 함수를 학습할 수 있지만, 과적합의 위험이 있을 수 있다.

Epochs는 학습 데이터를 전체로 한 번 훈련하는 주기를 의미한다. DL은 학습 횟수에 따라 예측 성능에 영향을 주기 때문에 최적의 Epochs를 찾는 것이 중요하다. 학습 횟수가 많아지면 성능이 좋아지는 게 일반적이지만, 데이터에 따라 학습 횟수가 많아지면 과적합이 생겨 오히려 성능이 떨어질 수도 있어 데이터별로 최적의 Epochs 값을 찾는 것이 중요하다.

표 7 최적의 파라미터 값

Dataset	Project	Best preforming parameter(PD)	
		mlp_layer	Epochs
AEEM	EQ	4	500
	JDT	1	1000
	LC	1	100
Relink	safe	3	100
	apache	2	500
	zxing	1	100
AUDI	ProjectA	1	500
	ProjectK	5	100
	ProjectL	1	100

표 7은 프로젝트별 PD 지표를 기준으로 최고의 성능을 보이는 ARM-Net의 파라미터값을 나타낸 것이다. PD 지표 기준 mlp_layer가 1, Epochs가 100인 경우에서 가장 좋은 성능을 보였다.

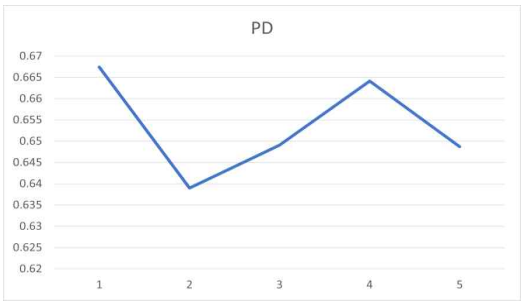


그림 2 mlp_layer 값에 따른 PD 성능 지표

그림 2는 파라미터값에 따른 전체 프로젝트에서 도출된 PD의 평균값을 나타냈다. mlp_layer의 경우 값이 1일 때 0.6674로 가장 높았다.

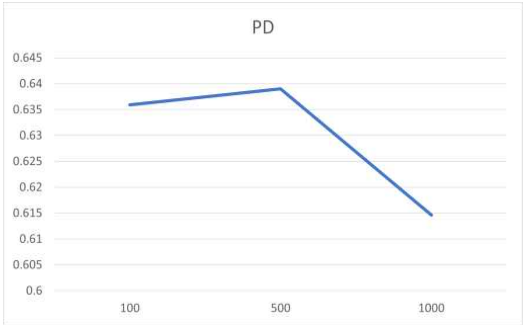


그림 3 Epochs 값에 따른 PD 성능 지표

그림 3은 파라미터값에 따른 전체 프로젝트에서 도출된 PD의 평균값을 나타냈다. Epochs 값의 경우 값이 500일 때 0.6390으로 가장 높았다.

프로젝트 및 하이퍼파라미터별로 성능 차이가 있음을 파악했고 특정 값에 따라 데이터에 적합한 파라미터값을 조정 해줘야 함을 확인했다.

6. 위협 요소

본 연구는 다양한 데이터 셋(AEEEM, Relink, AUDI)을 활용하여 분석을 수행했다. 그러나, 선택한 데이터들이나 프로젝트들은 일부 특정한 범주에 속하는 경우가 있어, 연구의 일반성과 다양성에 한계가 있다.

7. 결론 및 향후 과제

본 연구에서는 소프트웨어 공학 분야에서 중요한 연구 분야인 SDP에 최신 DL 기법 ARM-Net을 적용하였다. 비교 기법으로는 ML 기법인 CatBoost, XGBoost, RandomForest를 사용하였고 프로젝트별 최고의 성능을 보이는 파라미터 값을 도출했다. 실험 결과 ARM-Net은 PF 지표에서 XGBoost 대비 좋은 성능을 보였고 FIR 지표에서 CatBoost 대비 좋은 성능을 보였다. 향후 현재보다 다양한 데이터 셋에서 ARM-Net의 성능을 확인하려 한다. 또한 현재는 ML 기법들과 주로 비교하였는데 이후 연구에서는 최신 DL 기법과의 성능 비교를 수행할 계획이다.

참고 문헌

[1] CAI et al. Arm-net: Adaptive relation modeling network for structured data. In: Proceedings of the 2021 International Conference on Management of Data. 2021. p. 207-220.

[2] QIAO et al. Deep learning based software defect prediction. Neurocomputing, 2020, 385: 100-110.

[3] Iqra Batool and Tamim Ahmed Khan. Software fault prediction using deep learning techniques. Software Quality Journal, 2023, 1-40.

[4] FENG et al. Investigation on the stability of SMOTE-based oversampling techniques in software defect prediction. Information and Software Technology, 2021, 139: 106662.

[5] Shuo WANG and Xin YAO. Using class imbalance learning for software defect prediction. IEEE Transactions on Reliability, 2013, 62.2: 434-443.

[6] KWON et al. eCPDP: Early cross-project defect prediction. In: 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2021. p. 470-481.

[7] AKIMOVA et al. A survey on software defect prediction using deep learning. Mathematics, 2021, 9.11: 1180.

[8] Omri Safa and Sinz Carsten. Deep learning for software defect prediction: A survey. In: Proceedings of the IEEE/ACM 42nd international conference on software engineering workshops. 2020. p. 209-214.

[9] Giray et al. On the use of deep learning in software defect prediction. Journal of Systems and Software, 2023, 195: 111537.

[10] 김 et al. Ft-Transformer 기반의 소프트웨어 결함 예측. 한국정보과학회 학술발표논문집, 2022, 1770-1772.

EAGLE: 고가용 AI모델 서빙을 위한 마이크로 서비스 아키텍쳐 기반 API 게이트웨이 연구

박진우 O, 이원영, 이상정, 이동훈, 윤형화, 최호영

LG 전자 CTO 부문 인공지능연구소

jinwoo35.park@lge.com, wonyoung.lee@lge.com, sangjeong.lee@lge.com,
donghoon.lee@lge.com, hh.yoon@lge.com, hoyoung.choi@lge.com

EAGLE: Event-Driven API Gateway in Microservice Architecture System for AI Services

Jinwoo Park O, Wonyoung Lee, Sangjeong Lee, Donghoon Lee, Hyounghwa Yoon,

Hoyoung Choi

AI Lab., CTO Division, LG Electronics

Abstract

The purpose of this paper is to explore how the performance of API gateway, a core component of a Microservice Architecture (MSA) system for Artificial Intelligence (AI) services, can be improved. The ability to process large amount of data is essential for building reliable commercial AI services, and MSA is known as an architecture suitable for processing large-scale data due to its scalability. API gateway, sits at the front of the MSA, exposes a single entry point to clients and provides common functionality. Since all requests are processed through this gateway, its performance affects the overall quality of services. In this study, we propose an Event-driven API Gateway with Low latency Execution (EAGLE) to process large amounts of data, reliably and efficiently with minimum delay. As a verification step, we compared the performance of our method, using metrics such as throughput and error rates, with the thread-based method. It was confirmed that the event-driven processing method applied to EAGLE improves performance, stability, and resource efficiency. The experiment results show that the proposed structure can significantly optimize the API gateway performance and facilitates a highly reliable and accessible AI services.

1. Introduction

Recently, Artificial Intelligence (AI) technologies are rapidly spreading in various fields such as chatbots, search agents, development tools, and enterprise platforms. And new services are emerging by combining advancing AI technologies. Most AI services are operated on public clouds such as AWS (Amazon Web Services), Azure, and GCP (Google Cloud Platform) [1],[2]. Through the public cloud, it is possible to provide AI services that can be used simultaneously by many users in various regions. However, the increasing connectivity and interaction between services creates scalability and dependency problems, making maintenance difficult. Microservice Architecture (MSA), which makes it possible to build a maintainable and scalable system by reducing system complexity, has been proposed as a solution to this problem [3]. MSA refers to a structure that connects services with small functions that can be independently deployed, and due to this characteristic, it is possible to distribute only necessary updates without interrupting the entire system. MSA-based systems allow service providers to respond flexibly to diverse and complex requirements, giving them strengths in services such as AI that respond to real-time user requests. Therefore, many companies are adopting and applying this method to their fields [4].

An API gateway is needed to effectively provide services to users in the MSA. It can reduce the number of requests by merging few requests into one and support optimal protocol for specific service and clients. To deliver fast responses, various methods have been proposed to improve the performance of gateway. Representative processing methods include event-driven method, thread-based method, and hybrid method that combines the two [5].

In this paper, we propose EAGLE *(Event-driven API Gateway with Low latency Execution)*, an API gateway based on the event-driven method, to enhance the performance of the MSA-based platform. We aim to offer a practical guide on how the proposed method should be applied in a large data

processing environment. In our work, an in-depth analysis was performed on how the proposed method can improve the overall performance and efficiency of the system. The main contributions of this study are as follows: First, it presented an API gateway structure that guarantees high reliability in MSA. Second, it introduced a structure that can provide high performance yet cost-effective for providing AI services such as voice recognition. Third, we verify our method by comparing and evaluating the performance of EAGLE and thread-based API gateway in a simulated environment that involves realistic interactions within AI service systems.

Section 2 explains the existing studies related to MSA, API gateway, and data processing methods. Section 3 introduces EAGLE proposed in this study. Section 4 describes performance comparison experiments for the proposed structure, and finally presents conclusions and future tasks.

2. Background and Related Work

2.1 Microservice Architecture

MSA, which has recently been in the spotlight as a software architecture for complex service development, is a system of many small services that operate independently, unlike the monolithic architecture in which all modules are developed as a single service. It has a structure in which the entire system operates through interconnection between services [3],[6]. These MSA-based services, i.e. microservices, which interact with each other through interfaces like HTTP REST and gRPC and operate independently, facilitate development maintenance and horizontal expansion according to each service's load [7],[8]. Further efforts to reduce coupling and enhance scalability by processing internal communication between services through Event-Driven Architecture (EDA) are ongoing [9],[10]. Although there are difficulties in service monitoring, failure diagnosis and tracing as the number of microservices and service connection complexity grows, it is widely used in AI and big data service.

2.2 API Gateway

API gateway, a core component of MSA, integrates and coordinates various APIs. It performs common functions such as authentication, authorization, and logging, and manages and optimizes communication between services [8]. These functions help stable processing of large-scale data and make predictions and inferences in AI services more reliable [11]. Moreover, in cloud-based AI platforms, the API gateway plays a crucial link between the cloud and client devices, providing tools and services that efficiently perform various AI functionalities, such as voice recognition, device control, and conversation processing [1],[2],[4]. Collaboration between AI researchers and cloud engineers facilitated by API gateway helps achieve higher productivity and efficiency throughout the software lifecycle. All these elements make MSA and API gateway play a vital role in large-scale data processing and AI platforms [12].

2.3 Thread-based vs Event-driven Server Architecture

In a thread-based server architecture, a request received by the server is assigned to a specific worker thread for processing. Fig. 1 shows the operation flow of the thread-based API gateway. This method is simple and intuitive as it has a flow identical to procedural execution so it is relatively easy to construct server. However, it is difficult to utilize resources efficiently with this method and has limitations in handling high traffic. In this synchronous operation method, the worker thread assigned to a request can process another request only after the processing of the current request is completed. In the event of blocking I/O processing, the corresponding worker thread remains waiting and cannot perform any other tasks. This leads to inefficient use of resources, which can impact the overall performance of the system [13],[14],[15]. As the number of threads in a waiting state increases, the system's load increases, which can cause performance degradation and delays [5]. A large number of threads are required for heavy load with large concurrent requests, but the number of threads that can be maintained is limited due to HW constraints, and there is a cost to creating and terminating threads. Thread pools are used to minimize the overhead of thread creation and termination, but loads exceeding the size of the pool cannot be processed at once, delaying real-time processing. Even with sufficient threads, performance can decline due to multithreading overhead like context switching. There are also difficulties in finding an appropriate thread pool size that matches the HW specifications and a target service. When the pool is smaller than the optimal size, HW resources cannot be utilized appropriately, and if it is too large, optimal performance cannot be achieved due to thread maintenance costs and multithreading overhead.

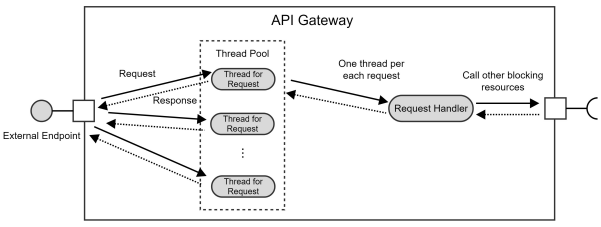


Fig. 1 Structure of thread-based API gateway

The limitations and disadvantages of the thread-based server architecture are known to be overcome through an event-based asynchronous approach [16],[17]. This approach enhances the

robustness and performance of the system by minimizing multithreading overhead as it utilizes a fixed number of threads. The non-blocking I/O also helps efficient use of resources. The asynchronous event-driven server architecture is suitable for today's HW with multiple core and therefore plays a vital role in modern systems handling large-scale data in complex scenarios. This method improves the resiliency, flexibility, and response time of the system, providing better service to users [18]. However, despite these advantages, there are many challenges in implementing and optimizing the event-driven API gateway. Additional work required for non-blocking operation can increase execution time, and unfamiliar asynchronous flows can make application development difficult. When mixed with the blocking method, performance could worse than thread-based approach. Performance degradation can also occur due to incorrect design or specific request/response patterns [5]. Therefore, a clear understanding of how this approach should be integrated with various system components and how optimal performance can be achieved through it is necessary. A process to verify in advance whether it is a suitable solution for the domain to be applied is also required

3. System Design

AI MSA services, including the API gateway, have a structure as shown in Fig. 2. External endpoints are provided to clients, and they either connect to back-end microservices directly or provide services by orchestrating several AI microservices through mediation functionality. The API gateway exchanges text or binary data based on JSON through REST API with the endpoint.

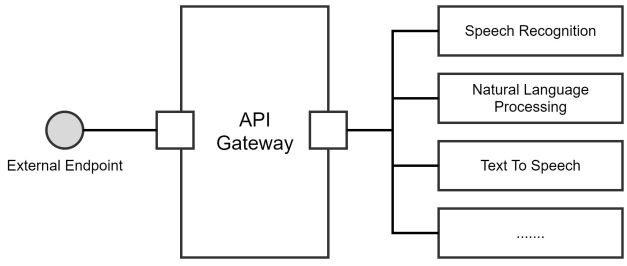


Fig. 2 Overview of AI MSA service structure with API gateway

We propose an AI service gateway EAGLE *(Event-driven API Gateway with Low latency Execution)* that maintains high performance and reliability. EAGLE is a gateway that can efficiently manage resources in an event-driven manner.

Fig. 3 represents the functions supported by EAGLE. It performs routing and mediation functions for AI back-end services such as vision, voice and other AI services. It also provides common functions such as authentication and logging. It resolves authentication functions by connecting to the authorization service and conducts logging for information requested to EAGLE through database and logging services. Each of the AI services thus receives common functions through EAGLE, making additional service implementation unnecessary.

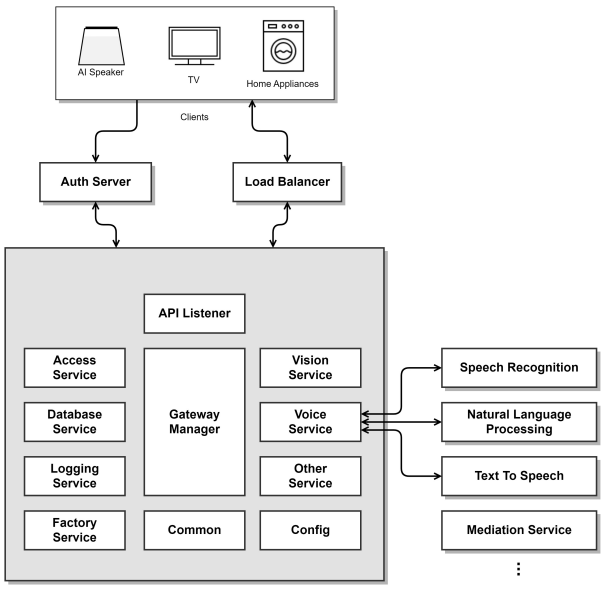


Fig. 3 EAGLE functions and roles

Fig. 4 explains the processing structure of EAGLE in its event-driven method. When a client makes an API request through an external endpoint, an I/O event for the request is registered in the event queue within EAGLE. Events registered in the queue are processed by the event loop, and since each event loop is processed by a single thread, the event-driven method processes many requests through fewer working threads compared to the thread-based method. Since there are a small number of worker threads, usually equal to the number of CPU cores, multithreading overhead such as context switching can be greatly reduced. Unlike the thread-based method, it processes API requests in a non-blocking manner based on event, so even if the number of requests increases, the delay does not increase rapidly. Fig. 5 shows pseudo-code of how EAGLE processes requests.

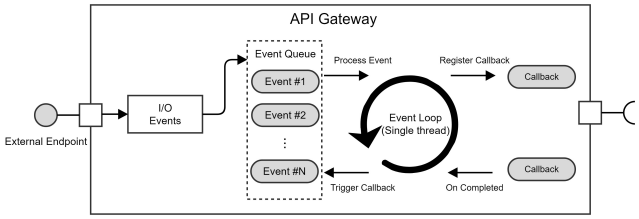


Fig. 4 EAGLE event processing structure

```
procedure onConnect ()
   channel ← created
   event_loop ← event_loop_group.select()
   channel.register_event_loop(event_loop)
   channel.activate()
end procedure

procedure onEventFired(channel)
   event ← channel.emit()
   channel.event_loop().execute(event)
end procedure

procedure EventLoop.execute(event)
   if current_thread() is this.get_thread() then
      this.handle(event)
   else
      this.queue().enqueue(event)
   end if
end procedure

procedure EventLoop.run ()
   while server is running do
      events ← this.queue().block_until_ready()
      for each event from events do
         this.handle(event)
      endfor
   endwhile
end procedure

procedure onDisconnect(channel)
   channel.deactivate()
   channel.deregister_event_loop()
end procedure
```

Fig. 5 EAGLE event loop execution example code

4. Performance Evaluation / Experiments

4.1 Test Environment

To verify the impact of the EAGLE on AI service operations, this study used a system similar to the actual service operating environment. AWS cloud environment was utilized in system construction to minimize the impact of external networks and enhance the reliability of the results. Two types of instances were used to verify operation in a resource-limited environment. (Table I).

TABLE I. Server Specification

Instance	Instance Size	HW Specification
Small Instance	c6g.medium	1vCPU 2GiB
Big Instance	c6g.xlarge	4vCPUs 8GiB

Spring Web MVC and Spring WebFlux framework operating in the Java environment were used for thread-based method test and event-driven method test, respectively. Any possible blocking operations in event-driven server code were transformed to non-blocking operations before testing to avoid unnecessary performance degradation. As HTTP clients that generate load, thread-based OkHttp and event-driven Spring WebClient were used. The details of the used SW are as in Table II.

TABLE II. Software Informations

Category	SW	Version
OS / Kernel	Amazon Linux 2 AMI (64bits-ARM) 6.1.34-59.116.amzn2023.aarch64	
Runtime	Java	Amazon Corretto-17.0.7.7.1
Build Tool	Gradle	8.1.1
Web Framework	Spring Web MVC	6.0.10
	Spring WebFlux	6.0.10
Web Server	Tomcat	10.1.10 (Servlet 6.0)
	Reactor Netty	1.1.8
HTTP Client	OkHttp	4.11.0
	WebClient	6.0.10

4.2 Test Method

For the test, an API transmitting binary data was used to reproduce the same load situation as the operating voice recognition service for home appliance/TV. The load test was conducted by increasing the number of concurrent requests until the server was reached to its processing limit. The thread pool size of the thread-based server was set to be the same as the number of concurrent requests to accommodate all requests. To prevent cold start errors due to sudden loads applied to the server in an idle state and to obtain stable results, a warm-up and ramp-up stage were set at the beginning of each test, gradually increasing the number of concurrent requests over 10 iterations. The warm-up stage, which could reflect the impact of cold start, was excluded from result collection. In addition, instead of fixing the test time, the total number of test requests was fixed to prevent requests being processed at the end of the

test from being treated as failures. The detailed test conditions are listed in Table III. The total number of requests for each test condition and an example of the test execution pattern can be found in Table IV and Fig. 6.

TABLE III. Test Conditions

Element	Value
Used API	HTTP API transmitting binary data Round trip traffic size: approx. 773 KiB
Concurrent Requests	50, 100, 200, 300, 400, … (up to server processing limit)
Number of Iterations for Each Test	Warm-up(10) + ramp-up(10) + test iteration(30)
Timeout for Each Test	Warm-up(1minute) + Remaining 4minute

TABLE IV. Total Number Of Requests For Each Test

Concurrent Requests	50	100	200	300	400
Total Requests	1,775	3,550	7,100	10,650	14,200

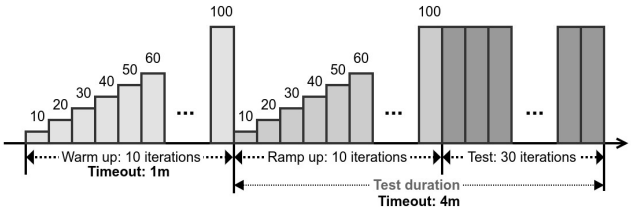


Fig. 6 Test execution pattern when concurrent requests are 100

The test proceeded in the following order. First, assuming a situation where the API gateway receives requests from clients, the performance of thread-based server and event-driven server was compared. To check the operation pattern in a high-load situation, a server was operated on the small instance and a client was operated on the big instance. In this test, the client instance generates load using a thread-based approach commonly used for REST API calls (Thread-per-request). Next, to test the situation where the event-driven API gateway sends requests to the backend server, the same test was conducted again after the client's operation method was changed to an event-driven approach. For analysis, the following metrics were collected: test duration (time for all requests to be processed), latency for each request, error rate, memory usage, and the number of thread context switches that occurred during test. Memory usage was derived through the difference between the system free memory value before starting the test (with the server running in idle) and the lowest system free memory value recorded during the test.

4.3 Test Results

4.3.1 Client to API Gateway

In the first test, the server performance was checked when the API gateway is under heavy external load. For load generation, a thread-based client (OkHttp), a common REST API call method, was used. Changes in the total test duration, average throughput, average latency for each request, failure rate, and memory usage according to the number of concurrent users are presented in Table V.

TABLE V. Test Results (Client to API Gateway)

Metrics	Server	Concurrent Requests				
		50	***100***	***200***	***300***	***400***
Test duration (ms)	Thread	6,254	10,069	17,717	23,035	Timeout [a]
	Event	4,943	8,067	13,351	18,614	26,727
Average throughput (req/s)	Thread	284	353	401	462	3
	Event	359	440	532	572	531
	Improvement	26.53%	24.82%	32.70%	23.75%	17068.90%
Average latency[b] (ms)	Thread	162	263	463	605	967,058
	Event	117	197	335	463	669
	Improvement	27.91%	25.13%	27.62%	23.34%	99.93%
Failure rate (%)	Thread	0%	0%	0%	0%	94.77%
	Event	0%	0%	0%	0%	0%
Max. Mem Used (MiB)	Thread	90.43	114.06	214.87	294.95	339.51
	Event	76.44	87.30	142.88	207.98	229.39
	Improvement	15.47%	23.46%	33.50%	29.48%	32.43%
Total # Context Switches	Thread	28,719	48,923	89,311	122,388	345,516
	Event	13,743	15,681	19,768	24,831	30,872
	Ratio	47.85%	32.05%	22.13%	20.29%	8.94%

[a.] Timeout after 4 minutes

[b.] Only completed requests collected

The event-driven server demonstrated superior performance in all metrics compared to the thread-based server. Notably, while the thread-based server reached its processing limit at 400 concurrent requests, failing to properly process 94.77% of the requests, the event-driven server managed to process all

requests successfully although it exhibit performance degradation with increased load. When all requests were processed normally, the maximum difference between two servers are like following: event-driven server processed approximately 130 more requests per second, showed a shorter latency of approximately 150ms, and used approximately 100MiB less memory than the thread-based server. In terms of ratios, the event-driven server showed up to 32.7% higher average throughput, up to 27.9% lower latency, and up to 33.5% less memory usage. This confirmed that the event-driven server could process higher loads more stably and quickly, even with less memory resources. It can be seen that these results were influenced by the decrease in thread switching overhead. While the thread-based server showed a proportional increase in the total number of context switches as the number of concurrent requests increased, the event-driven server showed a significantly reduced increase, resulting in a growing difference in the total number of switches as the load increased. At 300 concurrent requests, only about 80% of the context switches compared to thread-based occurred. Fig. 7 shows a graph of the total test execution time.

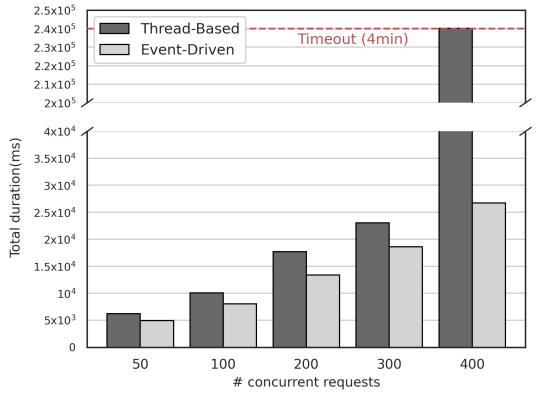


Fig. 7 Total test execution time
(Client to API gateway)

4.3.2 API Gateway to AI Service Server

For testing the situation where the event-driven API gateway sends requests to the backend AI service, the client's operation method was changed to event-driven and the same test was conducted. It's because the thread-based HTTP client operates in a blocking manner and cannot be used in the event-driven server due to performance issues. The results are shown in Table VI.

TABLE VI. Test Results (API Gateway to AI Service)

Metrics	Server	Concurrent Requests				
		50	*100*	*200*	*300*	*400*
Test duration (ms)	Thread	6,278	9,826	17,077	25,174	Timeout [a]
	Event	4,840	7,990	12,997	18,175	25,328
Average throughput (req/s)	Thread	283	361	416	423	24
	Event	367	444	546	586	561
	Improvement	29.72%	22.97%	31.40%	38.51%	2269.41%
Average latency[b] (ms)	Thread	164	256	440	663	2,755
	Event	122	195	321	446	627
	Improvement	25.42%	23.57%	27.11%	32.63%	77.23%
Failure rate (%)	Thread	0%	0%	0%	0%	60.01%
	Event	0%	0%	0%	0%	0%
Max. Mem Used (MiB)	Thread	88.46	120.46	218.26	301.09	342.20
	Event	77.02	92.87	158.88	205.56	297.29
	Improvement	12.93%	22.91%	27.21%	31.73%	13.12%
Total # Context Switches	Thread	28,571	44,847	88,717	130,345	208,901
	Event	13,062	15,283	19,325	23,857	28,866
	Ratio	42.72%	34.08%	21.78%	18.30%	13.82%

[a] Timeout after 4 minutes

[b] Only completed requests collected

The results again showed that the event-driven server outperformed the thread-based server. Looking at the maximum performance difference when all requests were processed successfully, throughput shows 38.51% (about 160 req/s), average request latency shows 32.63% (about 220ms), memory usage shows 31.73% (about 100MiB), and the total number of thread context switches shows 91.7% (about 100k). A somewhat different point from the previous test, as seen in Fig. 8, is that the performance difference gap widened as the number of concurrent requests increases.

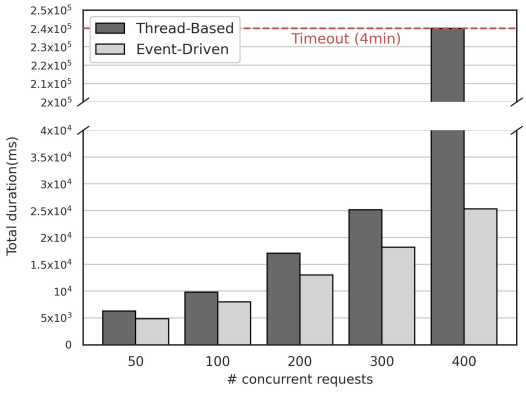


Fig. 8 Total test execution time
(API gateway to AI service)

Previous test also has shown that most requests failed (94.8%) when the thread-based server reached the 400 concurrent users limit. However, a lower failure rate (60%) was shown in the current test, and slightly higher throughput and lower latency were shown than before in all but few exceptional cases (thread-based server test result in 300 concurrent requests). Through these results, it was confirmed that there were no performance issues when the event-driven API gateway communicates with a thread-based backend server.

5. Conclusion

In this paper, we proposed EAGLE, a highly available API gateway that facilitates MSA systems in a cloud computing environment. To verify the performance of the proposed event-driven method, a comparison between the proposed method and the thread-based method was conducted in an environment similar to the actual service operating environment. Changes in performance were compared and verified by measuring metrics such as average throughput, latency, and error rate while increasing the number of concurrent users.

Test result showed that the event-driven server achieved higher throughput and lower latency compared to thread-based server as load increased. The reduced number of context switches in the event-driven server suggests that the decrease in multithreading overhead contributed to this result. While using less memory, the event-driven server stably processing all requests even when the thread-based server reached the server limit and showed an error rate exceeding 70%. As a result, it was confirmed that the event-driven method is suitable for the API gateway of the AI MSA system, which needs to process large data, such as voice recognition and large language model Inference engines.

In future work, we plan to conduct performance comparison and optimization in more diverse environments. Through this, we expect to further enhance the performance and scalability of the API gateway and MSA system for AI services.

References

[1] Amazon Web Services. "Build, Tune, and Deploy an End-to-End Churn Prediction Model Using Amazon SageMaker Pipelines", Oct. 2021. [accessed on 3 October 2023]. [Online]. Available: https://aws.amazon.com/blogs/machine-learning/build-tune-and-deploy-an-end-to-end-churn-prediction-model-using-amazon-sagemaker-pipelines/

[2] Microsoft, "Azure Machine Learning Architecture". [accessed on 3 October 2023]. [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/ai-ml/idea/azure-machine-learning-solution-architecture

[3] L. De Lauretis, "From monolithic architecture to microservices architecture," In 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 93-96, October 2019.

[4] Google Cloud, " Tabular Workflow for End-to-End AutoML". [accessed on 3 October 2023]. [Online]. Available: https://cloud.google.com/vertex-ai/docs/tabular-data/tabular-workflows/e2e-automl

[5] S. Zhang, Q. Wang, Y. Kanemasa, H. Shan and L. Hu, "The Impact of Event Processing Flow on Asynchronous Server Efficiency," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 3, pp. 565-579, 1 March 2020.

[6] G. Mazlami, J. Cito, and P. Leitner, "Extraction of microservices from monolithic software architectures," In 2017 IEEE International Conference on Web Services (ICWS), pp. 524-531, June 2017.

[7] R. Faradj, "The run-time impact of business functionality when decomposing and adopting the microservice architecture," 2018.

[8] K. Indrasiri, Microservices in Practice—Key Architectural Concepts of An MSA, Jan 2021.

[9] RAHMATULLOH, Alam, et al. Event-Driven Architecture to Improve Performance and Scalability in Microservices-Based Systems. In: 2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS). IEEE, 2022. p. 01-06.

[10] E. D. Giovanni and I. B. K. Manuaba, Event-driven approach in microservices architecture for flight booking simulation, ICIC Express Letters, vol.16, no.5, pp.545-553, 2022.

[11] Zhelev, S.; Rozeva, A. Using microservices and event driven architecture for big data stream processing. In AIP Conference Proceedings; AIP Publishing LLC: Melville, NY, USA, 2019; Volume 2172, p. 090010.

[12] N. Mungoli, "Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency," arXiv preprint arXiv:2304.13738, 2023.

[13] S. S. Prakash and B. C. Kovoor, “Performance optimisation of web applications using in-memory caching and asynchronous job queues,” in Proc. Int. Conf. Inventive Comput. Technol., 2016, vol. 3, pp. 1–5.

[14] A. Aytekin, H. R. Feyzmahdavian, and M. Johansson, “Analysis and implementation of an asynchronous optimization algorithm for the parameter server,” arXiv preprint, 2016. [Online]. Available: https://arxiv.org/abs/1610.05507

[15] J. Davis, A. Thekumparampil, and D. Lee, “Node.fz: Fuzzing the server-side event-driven architecture,” in Proc. 12th Eur. Conf. Comput. Syst., 2017, pp. 145–160

[16] PRAKASH, P.; BIJU, R.; KAMATH, MohanSowmya. Performance analysis of process driven and event driven web servers. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). IEEE, 2015. p. 1-7.

[17] Y. Su, D. Feng, Y. Hua and Z. Shi, "Predicting Response Latency Percentiles for Cloud Object Storage Systems", 2017 46th International Conference on Parallel Processing (lCPP), pp. 241-250, 2017.

[18] K. Elmeleegy, A. Chanda, A. Cox, and W. Zwaenepoel, "Lazy Asynchronous I/O for Event-Driven Servers," In USENIX Annual Technical Conference, General Track, pp. 241-254, June 2004.

API Management를 위한 LLM 기반 Test Case 자동 생성 기법

정하늘, 백두산, 전하영, 박성준
KT
[hn.jeong, dusan.baek, hayeong.jeong, sung.park]@kt.com

LLM-Based Test Case Automatic Generation for API Management

Haneul Jeong, Dusan Baek, Hayeong Jeon, Sungjune Park
KT

요 약

다양한 API를 수용하여 관리하는 API Management(APIM)의 중요성이 증가함에도 불구하고, API에 대한 제한된 지식과 정보 접근의 어려움 등으로 인해 APIM 관리자가 APIM 시스템 내 수용된 모든 API에 대한 품질을 보장하는데 어려움을 가지고 있다. 이러한 문제를 해결하기 위해, 본 논문에서는 Large Language Model(LLM) 기반 Test Case(TC) 자동 생성 기법을 제안한다. 제안하는 방법은 개별 API 정보를 담고 있는 Open API Specification(OAS) 문서와 APIM의 공통된 테스트 시나리오를 통해 프롬프트로 생성하고, LLM에 질의하여 TC를 자동으로 생성한다. 이를 통해 APIM 개발자는 API에 대한 통합 테스트를 자동으로 수행하여, APIM 시스템의 품질을 향상시킨다.

1. 서 론

API란 Application Programming Interface의 줄임말로, 애플리케이션이라는 고유 기능을 가진 소프트웨어 사이의 서비스 계약이라고 할 수 있다. 이 계약은 요청과 응답을 사용하여 두 애플리케이션이 서로 통신하는 방법을 정의하며, API 문서 산출물에는 개발자가 이러한 요청과 응답을 구성하는 방법에 대한 정보가 들어 있다.

근래에는 더욱 많은 시스템이 API로 연결되면서, API에 대한 종속성이 확산되고, 증가하고 있다. 이러한 흐름에 맞춰 API의 종합적인 관리를 목적으로 하는 API Management(APIM)의 필요성이 대두되고 있다. APIM 시장은 2023 년 67.7 억달러에서 2032 년 말에는 943 억 달러로 매년 34%의 평균 성장률이 전망된다[1].

APIM의 주요한 역할 중 하나는 수용된 다양한 API의 품질을 보장하는 것이다. 물론, 개별 API 개발 주체가 API의 품질을 보장하는 노력을 수행하지만, 이에 더하여 APIM 시스템의 품질을 보장하기 위해서는 APIM 관리자 또한 APIM에 수용된 API들의 품질을 보장하는 활동을 수행해야 한다. 하지만, APIM에 수용된 모든 API를 관리하는 APIM 관리자는 단일 API를 직접 개발한 API 개발자에 비해 API에 대한 지식이 부족하고, API 관련 정보에 대한 접근성도 낮다. 이로 인해 APIM 관리자는 APIM 품질 보장 활동에 어려움을 가진다.

이러한 문제점을 해결하고자, 본 논문은 APIM 관리자가 적은 노력으로 API의 품질을 보장하기 위해 Large Language Model (LLM)을 활용하여 Test Case (TC)를 자동으로 생성하는 방법을 제안한다. 제안하는 TC 자동 생성 방법은 다음과 같다.

1. 개별 API 정보를 담고 있는 OpenAPI Spec (OAS) 및 APIM의 공통된 테스트 시나리오 수집
2. 상기 정보로 구성된 프롬프트를 생성하고, LLM에

질의하여 API 호출을 위한 (Oracle을 포함한) TC 자동 생성

3. 생성된 TC를 바탕으로 API를 호출하여 테스트하고, 실패 케이스에 대한 수동 검수 및 회귀 테스트 수행

제안된 방법을 사용하여 생성된 TC를 활용하여 API Management 서비스 중 하나인 KT GenieLabs 내 16 개의 API를 대상으로 총 96 개의 TC를 생성하여 품질을 검증하였다. 수동 검수를 통해 테스트 실패 케이스 중 잘못된 TC로 인해 발생된 테스트 실패를 식별하였으며, 프롬프트 템플릿을 정교화 한 뒤 회귀 테스트를 통해 이러한 과정을 반복하였다. 최종적으로 생성된 모든 TC에 결함이 없는 것을 확인하였으며, 발생한 테스트 실패 케이스를 리포트하여 API의 결함을 해결 할 수 있었다.

2. Background 및 관련 연구

2.1 APIM Background

좁은 의미에서 APIM의 역할은 인프라를 제공하고, API를 배포하는 역할을 의미한다 (예: Huggingface [2], RapidAPI [3] 등). 그러나, 보다 넓은 의미에서 APIM의 기능을 나열하자면, API lifecycle 관리, 공통 에러 처리, 보안, 사용자 호출 권한 및 제한 정책 관리, 자원 관리, API 간 연동 및 endpoint 관리, 문서화, 분석 등으로 개별 API의 기능적 요소를 관리함과 동시에 보안과 같은 API의 비기능적 요소에 대한 지원도 포함된다. 이러한 차원에서 APIM의 역할은 수용된 다양한 API의 품질을 보장하기 위한 적극적인 활동을 포함하며, Google [4], Naver [5], Kakao [6] 등의 APIM 서비스는 이러한 역할을 수행하고 있다.

APIM 시스템 내 API의 품질을 검증하기 위한 프로세스는 일반적인 S/W 개발 프로세스 내 인수 테스트와는 다른 성격을 가진다. 인수 테스트는 시스템이 인수 기준을 만족시키는 가를 기준으로 삼고 있으며, 이러한 인수 기준은 고객(발주사) 요구사항에 기반하여 구체화된다. 인수 테스트에서는 이해당사자들이 개발된 S/W형상이 인수 기준에 부합하는지를 검증한다.

반면, APIM의 경우, 앞서 언급한 바와 같이 APIM 정책에 따라 별도의 품질 검증 활동 자체를 수행하지 않기도 한다. 만약 검증 활동을 수행한다 하여도, API를 위한 요구사항에 더하여 APIM 시스템에 수용되기 위한 추가적인 요구사항을 반영할 필요가 있는데, 이에 대한 검증 주체가 모호하다는 어려움이 있다. 더불어, 활동의 주체가 APIM 관리자인 경우에는 API에 대한 지식이 부족하고, API 개발자에 비해 검증 대상(API)에 대한 접근성이 떨어진다는 어려움을 가진다. 그림 1 은 이와 같은 차이를 도식화한 그림이다. 만약, APIM 관리자와 API 개발자가 이해 관계가 적은 제 3 자 관계일 경우에는 API 품질 확보에 더욱 어려움을 가진다. 그럼에도 불구하고, 보다 높은 품질의 APIM을 위해서는 API에 대한 품질 검증 활동이 요구된다.

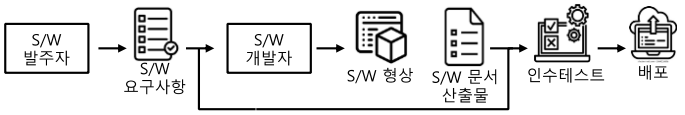


그림 1-1. 일반적인 S/W 개발 lifecycle 내 인수테스트 과정

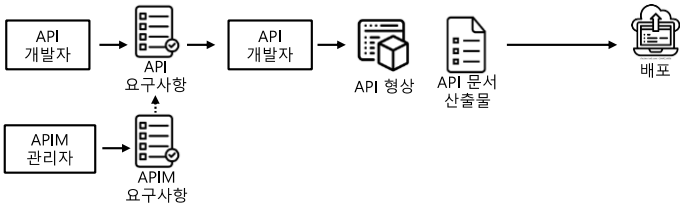


그림 1-2. 별도 검증을 수행하지 않는 API Management 환경에서의 API 배포 과정

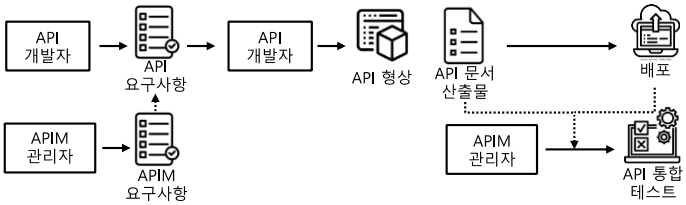


그림 1-3. APIM 관리자에 의해 검증을 수행하는 API Management 환경에서의 API 배포 과정

그림 1. 일반적인 S/W 개발 인수테스트 과정과 APIM 환경에서의 API 배포 과정 차이

APIM 환경에서 API의 품질 검증 활동은 다음의 두 이해당사자에 의해 수행된다.

㉮ 개발자에 의한 검증: API에 대한 높은 이해를 바탕으로 Positive TC에 대한 테스트를 적절히 수행할 수 있음. 하지만, Negative TC에 대한 Test Coverage가 낮을 수 있음

② APIM 관리자에 의한 검증: 다양한 API 수용

경험을 가지고 있어, Negative Test 시나리오에 대한 이해도는 높음. 그러나 검증 대상 API를 파악하기 위한 추가적인 노력이 필요하므로 특정 API에 대한 TC를 생성하는 데 어려움이 있음.

본 논문에서는 ② APIM 관리자에 의한 검증 어려움을 해결하고자, LLM을 사용하여 TC를 자동 생성하는 방법을 제안한다. 이를 통해, 보다 높은 Test Coverage를 확보하여 APIM의 품질을 향상할 수 있다.

2.2 관련 연구

생성 능력이 뛰어난 ChatGPT[7]와 같은 LLM의 출현과 더불어, 소프트웨어 공학 분야에도 LLM이 활발하게 사용되고 있다. Github의 Copilot[8]과 같이 LLM의 생성 능력을 활용한 소프트웨어 개발 분야 외에도, 소프트웨어 유지보수와 품질 보장을 위한 연구도 활발히 진행되고 있다[9].

특히, 프롬프트 엔지니어링 연구[10]는 적절한 지시사항을 지닌 프롬프트의 입력 만으로 별도의 학습 없이 다양한 분야에 LLM 활용이 가능함을 보여주었다. 프롬프트를 LLM이 소프트웨어 공학을 수행하도록 지시하는 프로그래밍 형태로의 관점을 제시한 이전 연구[11]에서는, 소프트웨어 패턴처럼 재사용성이 가능한 소프트웨어 공학을 위한 프롬프트 패턴 카탈로그를 제공한다. 해당 카탈로그를 이용하면 특정 개발 요구사항이나 프로그래밍 가이드라인 준수 등과 같은 소프트웨어 문제를 LLM과 상호작용하면서 해결할 수 있다. 해당 방법은 서론에서 언급한 API 개발자에 의한 검증을 위해 LLM을 활용하는 좋은 수단이 될 수 있다.

위와 같은 프롬프트 엔지니어링 기법을 활용하여 LLM 기반의 TC를 자동으로 생성하는 연구들도 진행되었다[12]. 그 중 다수는 Unit Test을 위해 LLM을 활용하는 방법을 제시하였다[13,14]. 이들은 LLM을 통해 TC를 생성하고, 생성된 TC를 작성된 함수에 입력하여 정상 동작하는지를 확인한다. 만약, 에러가 발생한다면 에러를 포함한 프롬프트를 재생성하여 상기 과정을 반복하는 피드백 루프를 구성한다. 버그 리포트 기반 소프트웨어 테스터를 생성하는 프롬프트 엔지니어링 기법[15]도 시도되었다. 에러 재현의 어려움을 해결하고자, 보고된 버그 리포트를 기반으로 LLM을 이용하여 TC를 생성한다.

이러한 방법들은 LLM을 사용하여 TC를 자동으로 생성해준다는 점에서 본 논문과 유사한 점이 있다. 하지만, 상기 방법들은 코드, API 버그 리포트와 같이 API 형상에 대한 구체적인 정보를 기반으로 하는 반면, 본 논문에서 제시하는 방법은 Black-Box에도 적용이 가능하다는 차이를 가진다.

본 논문에서는 API Management의 요구사항 검증을 위한 테스트 케이스 생성을 위해 API의 입출력 정보를 가지는 API Open API Specification (OAS) 문서와 함께 TC 생성 요구사항을 기반으로 프롬프트를 생성한다.

3. 본 론

본 논문은 LLM을 활용하여 TC를 자동으로 생성하는 방법을 제안하고, 이를 통해 API 테스트를 수행한다. 이를 위한 과정은 요구사항 명세(3-1), 프롬프트 생성 및 TC 생성(3-2), API 테스트 및 검수(3-3)의 세 단계로 구성된다. 그림 2 는 이러한 단계를 명세한 그림이다.

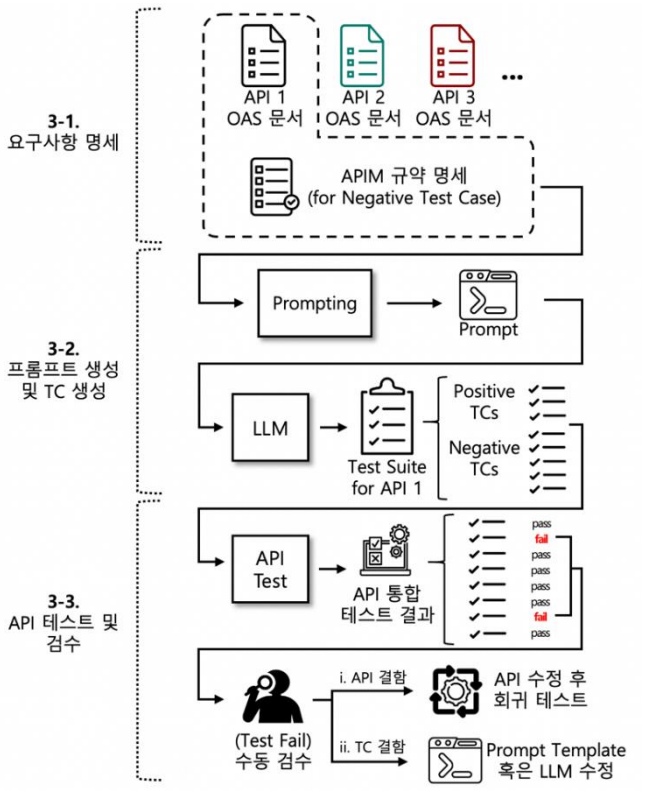


그림 2. LLM 기반 Test Case 자동 생성 및 API 테스트 수행 과정

요구사항 명세 단계(3-1)에서는 TC생성을 위해 필요한 OAS 문서와 테스트 시나리오를 획득하는 단계이다. 프롬프트 생성 및 TC 생성 단계(3-2)에서는 이러한 Spec을 프롬프트 템플릿에 맞춰 프롬프트를 생성하고 LLM에 입력하여 TC를 생성한다. 마지막으로 API 테스트 및 검수 단계(3-3)에서는 생성된 TC를 통해 API를 테스트하고 테스트 결과에 따라 API를 수정한 후 회귀테스트를 수행한다.

3-1. 요구사항 명세

TC를 생성하기 위해 API의 사양과 테스트 시나리오가 필요하다. 먼저 API의 사양은 엔드 포인트, 입출력 파라미터, 에러 코드, 버전 등을 포함한다. 이러한 정보는 API 별로 다르기 때문에 APIM 내 각 API 별로 수집되어야 한다.

반면, 테스트 시나리오는 API의 사양을 참조하여 구성될 수 있기 때문에 하나의 공통 테스트 시나리오를 APIM 내 여러 API를 위한 테스트에 활용할 수 있다. 예를 들어, Positive TC를 위한 테스트 시나리오는 “Valid Request”로 정의하여 다양한 API에 공통적으로 활용할 수 있다.

3-1-1. API 사양

API 사양은 문서, 코드 샘플, 다이어그램, 인터랙티브 콘솔 등의 양식으로 명세 될 수 있다. 이 중 본 논문에서는 OAS 문서를 통해 API 사양을 기술한다.

OAS란 HTTP API의 사양을 명세하는 표준화된 언어로, APIM의 API 중 다수인 HTTP API의 사양을 표현할 수 있다. 또한, 기 배포된 다양한 문서화 도구를 사용하여 추가적인 작업 없이도 자동으로 OAS 기반의 API 사양 문서(OAS 문서)를 획득할 수 있다. 따라서, OAS 문서를 활용하면 내부 로직이나 소스코드에 대한 정보 없이도 LLM을 이용하여 TC 자동 생성이 가능하다. 그림 3 은 OAS 문서 내 각 요소를 설명한다.

OAS 문서는 요청 방법, 요청 파라미터 사양, 응답 사양, 데이터 스키마 등으로 구성된다. 각각의 요청 파라미터 사양은 파라미터 명, 데이터 타입, 필수 포함 여부 등으로 구성된다. 응답 사양은 다양한 HTTP 상태 코드에 따라 명세 되며, 이를 기반으로 정상 응답과 오류 응답으로 구분된다.

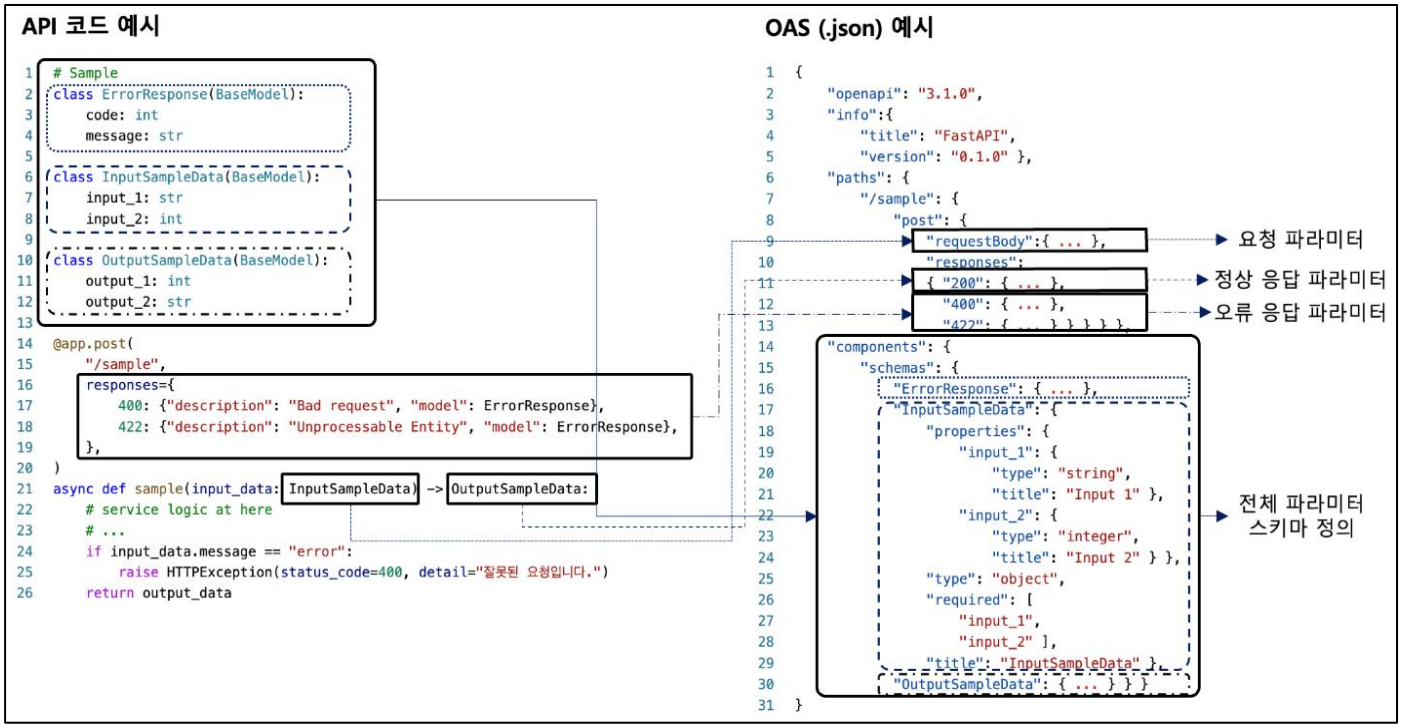


그림 3. API 코드와 OAS 문서 예시

3-1-2. 테스트 시나리오 명세

정상 응답에 해당하는 상태 코드와 달리 오류 응답에 해당하는 상태 코드는 오류 케이스에 따라 다양하게 정의된다. 각각의 오류 응답을 발생시키는 시나리오가 다양하게 존재하기 때문에 정상 응답과는 달리 오류 응답의 경우 다양한 시나리오로 테스트해야 한다. 예를 들어, “Bad Request” 오류 응답은 필수 파라미터 누락, 데이터 타입 오류, 비유효 데이터 입력 등 다양한 케이스를 통해 테스트되어야 한다. 이러한 테스트 시나리오는 다양한 API 수용 경험을 가지고 있는 APIM 관리자에 의해 정의된다. 테스트 시나리오는 자연어 형태로 명세 되며, OAS 문서와 함께 프롬프트를 생성하는 데 사용된다.

3-2. 프롬프트 생성 및 TC 생성

TC 자동 생성을 위해 LLM에 질의되는 프롬프트는 대표적인 프롬프트 작성 기법으로 구성한 프롬프트 템플릿에 기초하여 생성된다[16]. 그림 4 는 본 논문에서 사용한 프롬프트 템플릿을 도식화한 그림이다.

```
You are Negative Testcase Generator

This is OpenAPI Specification
{OAS Document}

With this OpenAPI Specification, make negative testcases
{Test Scenarios}

{Instructions}

{Restrictions}

Example:
{Code Example}
```

그림 4. 프롬프트 템플릿 예시

프롬프트 템플릿의 주요 구성은 다음과 같다.

① 역할 지정
② OAS 문서 정의 및 내용
③ 테스트 시나리오 정의 및 내용
④ LLM이 수행할 Task
⑤ 양질의 TC를 생성하기 위한 제약사항
⑥ (Optional) One-shot 예시

프롬프트 템플릿을 완성한 후, 각 API의 OAS 문서와 테스트 시나리오를 프롬프트 템플릿에 대입하여 프롬프트를 구성한다. APIM 내 API 마다 개별적으로 생성되는 OAS 문서와는 다르게, 테스트 시나리오는 공통으로 적용 가능하기 때문에 프롬프트를 작성할 때 고정할 수 있다. 결과적으로, OAS를 제외한 프롬프트 템플릿은 API별로 동일하다.

위 과정을 통해 생성된 프롬프트를 LLM에 질의하여 TC를 생성한다. 생성된 TC는 입력 값과 원하는 출력(Oracle)으로 구성되며, 프롬프트 템플릿의 ‘④ LLM이 수행할 Task’의 명세 방법에 따라 호출 가능한 코드의 형태를 가질 수 있다. 또한, ‘③ 테스트 시나리오 정의 및 내용’ 작성 방법에 따라 여러 TC를 Test Suite 형태로 얻을 수 있다.

3-3. API 테스트 및 검수

LLM을 통해 생성된 TC를 사용하여 API 테스트를 수행한다. 만약 Oracle과 API 출력이 일치하지 않는다면 수동 검수가 필요하다. 왜냐하면 일반적인 테스트와는 달리 테스트 실패가 반드시 API 결함으로 발생하는 것이 아닌, TC 결함으로 인해 야기된 것일 수 있기 때문이다.

수동 검수를 통해 API 결함으로 테스트가 실패된 경우, API 수정 후 회귀 테스트를 수행하여 API의 품질을 향상시킬 수 있다. 만약 TC 결함으로 테스트가 실패했다면, LLM의 프롬프트를 수정하거나, LLM Fine Tuning을 수행하여 TC 자동 생성 프로세스를 고도화 해야 한다.

4. 실 험

본 논문에서 제안하는 LLM (OpenAPI의 Chat-GPT4) 기반 TC 자동 생성 기법을 활용하여 KT GenieLabs APIM 시스템의 API 품질 검증을 수행하였다. KT GenieLabs APIM 시스템은 일상채팅, 유해 표현 탐지 등의 API를 관리하는 시스템이다.

4-1. OAS 문서 수집

OAS 문서는 기 배포된 문서화 도구를 통해 추가적인 작업 없이도 자동으로 생성할 수 있다. 예를 들어, KT GenieLabs APIM 시스템 내 수용된 API 중 FastAPI 프레임워크 기반의 API는 자동으로 OAS를 생성하기

때문에, 추가 공수 없이 OAS 문서 수집이 가능하다.

OAS 문서의 입력 파라미터는 추후 추가될 테스트 시나리오의 요구사항을 수용하기 위해 다음과 같은 내용을 포함하여 최대한 구체적으로 명세 되어야 한다.

- Data type
- Required / Not Required
- Default Value
- string: minLength, maxLength
- integer: min, max
- Enum (fixed values)
- (optional) 자연어 Description

개발자는 OAS 문서의 파라미터를 표기하는 규칙을 통해 기술하기 어려운 파라미터의 상세한 제약사항들을 자연어를 통해 기술할 수도 있다. 예를 들어, 입력 파라미터의 데이터 타입 중 문자열로 구성된 배열의 경우, 배열의 최소 및 최대 길이는 제한할 수 있다. 하지만 배열 내 각 문자열의 길이를 기존 파라미터 표기 규칙으로 제한하는 것은 불가능하며, 자연어로 명세하는 것은 가능하다. 본 논문에서는 LLM을 사용하여 OAS 문서를 파악하기 때문에 이와 같이 자연어로 기술한 제약사항 및 파라미터의 특성까지 반영한 TC 생성이 가능하다.

상기 파라미터 특성 등의 요청 규격 외에도 OAS 문서는 정상 응답에 대한 규격과 다양한 비정상 요청에 대한 응답 규격을 모두 포함한다.

4-2. 테스트 시나리오

APIM 내 API들의 오류 응답을 발생시키는 시나리오는 다음과 같이 정의된다.

- Missing required parameters
- Empty string in required parameter
- Exceeding maxLength or less than minLength for string parameter
- Wrong data type for integer parameter
- Invalid value entered
- Value out of range (minimum, maximum)

이는 실제 사용자들이 빈번히 경험한 API 오류 케이스로, 사용자들이 예상치 못한 입력 또는 부적절한 요청 시 발생하는 상황을 정의하였다.

4-3. 프롬프트 템플릿 별 API 테스트

3-2 절에서 제안한 프롬프트 템플릿을 기반으로 다양한 프롬프트를 생성하여 TC를 생성했다. 그림 5 는 생성한 프롬프트의 예시이다 (code example은 생략).

```
You are Testcase Generator

This is OpenAPI Specification.
{OAS Document}

With this OpenAPI Specification, make positive testcase and negative testcases.
Case 1: Valid Request
Case 2: Missing required parameters
Case 3: Empty string in required parameter
Case 4: Exceeding maxLength or less than minLength for string parameter
Case 5: Wrong data type for integer parameter
Case 6: Invalid value entered
Case 7: Value out of range(minimum, maximum)

For the seven cases presented above, please create test cases that can test the API without changing the request input format specified in the specification. Create a Python code in the form of a function. If you need any assumptions other than the conditions given when creating a test case, please create it except for the case.

base_url is https://aiapi.genielabs.ai/kt/nlp

I'll give you an example for your reference. I created the sumarize-literature api as follows, and please create it in the same form.
getTimestamp, getHmac, getHeaders functions are predefined and already established in my environment. you should exclude these placeholder definitions and not request pameters used in those functions.

Example:
{Code Example}
```

그림 5. Pii Detection API 프롬프트

3-2 절에서 제시한 프롬프트 템플릿의 Instructions 부분에는 LLM이 수행할 Task를 명세하며, 정의한 테스트 시나리오를 수행할 수 있는 입력 양식의 생성을 지시한다. 다음으로 Restrictions 부분에는 OAS 문서에서 제시한 입력 사양을 변경하지 않아야 한다는 제약 사항과, 호출 가능한 코드의 형태로 결과를 생성해야 함을 요구한다. 높은 품질의 응답 생성을 위해 OAS에 기술된 동일한 어휘로 프롬프트를 작성하며, 응답의 관련성을 향상시키기 위해 세부적인 정보를 포함하여 프롬프트를 작성했다. 마지막으로 예시를 추가하여 One Shot 기법을 사용했다. 각 기법들을 통해 API 테스트를 수행했다.

5. 결 과

본 장에서는 4.3 절에서 제안한 다양한 프롬프트 템플릿을 통해 API 테스트를 수행하고, 그 결과를 수동 검수하였다. 이를 통해 가장 품질이 좋은 프롬프트 템플릿을 확보할 수 있다. 최종적으로 확보된 프롬프트 템플릿을 통해 KT GenieLabs APIM 시스템 내 API를 테스트 하였다.

그림 6 은 프롬프트 템플릿을 통해 생성한 테스트 코드이다. Positive TC와 4-2 절에서 기술한 Negative Test 시나리오를 충족하는 TC를 생성하여, 총 7 개의 TC를 생성했다.

5-1. One shot 프롬프팅 VS Zero shot 프롬프팅

Zero shot 프롬프팅으로 생성한 결과를 이용하여 API를 테스트 하기 위해서는 실제 API 테스트 시 필요한 정보를 추가하기 위해 세부 정보의 수정이 필요했다. 반면, One shot 프롬프팅은 Example을 추가함으로써 헤더에 추가할 Authorization 정보(별도 구현 필요) 및 url 설정이 되어 있는 상태로 TC 및 실행 코드를 생성하였다.

Zero shot 프롬프팅 결과 코드는 실행 시 헤더 설정 실패 및 URL 오류 등의 잘못된 요청으로 API 검증 이전에 테스트에 실패했으나, One shot 프롬프팅 결과 코드는 추가 수정 없이 성공적으로 API 테스트를 수행할 수 있었다.

5-2. 세부 정보 기술

OpenAI에서 제시한 프롬프트 작성 전략을 활용하여 프롬프트를 작성했다, 전략을 활용하지 않은 모호한 프롬프트 사용 시에 비해 LLM이 생성한 TC의 정확도가 상승했다. OAS에 기술된 동일한 어휘로 프롬프트를 작성했다. 예를 들어, 프롬프트에 입력 파라미터의 데이터 타입 기술 시, 'Int'로 기술한 경우 관련된 테스트 시나리오를 정확히 이해하지 못했다. 반면 OAS에서 명세된 'Integer'로 기술 시, 관련된 테스트 시나리오를 충족하는 TC를 생성했다.

또한 응답의 연관성을 향상시키기 위해 세부 정보를 추가했다. 예를 들어, API의 입력 사양을 의미하는 용어로 'input format'을 사용한 경우, LLM은 입력 사양 변경 금지 요구사항을 이해하지 못하여, 요구사항을 반영하지 않은 TC를 생성해 테스트 실패로 이어졌다. 반면, 구체적으로 'request input format' 으로 기술 시, 제약사항을 이해하여 입력 사양 및 시나리오를 준수한 TC를 생성하여 테스트에 성공했다.

```python
import requests
import json
base_url = 'https://aiapi.genielabs.ai/kt/nlp'

def test_pii_detection_api():
    timestamp = getTimestamp()
    signature = getHmac(client_secret,
     f"{client_id}:{timestamp}")
    headers = getHeaders(client_key, signature, timestamp)

    # Valid Data
    valid_data = {
        "user_id": "LETS",
        "data": ["This is a test string."]}

    # Test Cases
    test_cases = [
        {"input": valid_data, "expected_status_code": 200},
        # Case 1: Valid Request
        {"input": {}, "expected_status_code": 400},
        # Case 2: Missing required parameters
        {"input": {"user_id": "", "data": [""]},
        "expected_status_code": 400},
        # Case 3: Empty string in required parameter
        {"input": {"user_id": "LETS", "data": ["A" * 513]},
        "expected_status_code": 400},
        # Case 4: Exceeding maxLength for string parameter
        {"input": {"user_id": 123, "data": ["Valid string"]},
        "expected_status_code": 400},
        # Case 5: Wrong data type for parameter
        {"input": {"user_id": "INVALID", "data": ["Valid
        string"]}, "expected_status_code": 400},
        # Case 6: Invalid value entered
        {"input": {"user_id": "LETS", "data": []},
        "expected_status_code": 400},
        # Case 7: Value out of range (empty array)]

    for i, case in enumerate(test_cases, start=1):
        response = requests.post(f"{base_url}/pii-detection",
        json=case['input'], headers=headers)
        status_code = response.status_code
        expected_status_code = case['expected_status_code']
        print(f"Test Case {i} Status Code: {status_code},
       Expected: {expected_status_code}")
        print(json.loads(response.text))

# Execute the function
test_pii_detection_api()
```

그림 6. LLM으로 생성한 테스트 코드

5-3. API 테스트 및 검수

4-2 절에서 확인한 고품질의 프롬프트를 통해 획득한 TC를 활용하여, KT GenieLabs APIM 시스템 내 API를 대상으로 테스트를 수행하였다. 그림 7 은 테스트 실패 케이스의 예시이다.

수정 전 API 테스트 결과	수정 후 API 테스트 결과
Test Case 1 Status Code: 200, Expected: 200 { 'code': 200, 'message': 'Success', 'resultCnt': 1, 'result': [{'detection': []}]} Test Case 2 Status Code: 500, Expected: 400 { ‘Message’: { ‘code’: 503 'type': 'InternalServerException', ‘message’: “Prediction failed” }} Test Case 3 Status Code: 200, Expected: 400 { 'code': 200, 'message': 'Success', 'resultCnt': 1, 'result': [{'detection': []}]	Test Case 1 Status Code: 200, Expected: 200 { 'code': 200, 'message': 'Success', 'resultCnt': 1, 'result': [{'detection': []}]} Test Case 2 Status Code: 400, Expected: 400 { 'Message': 'mandatory param error'} Test Case 3 Status Code: 400, Expected: 400 { 'Message': 'mandatory param error'}

그림 7. 테스트 실패 케이스의 예시 (일부)

그림 7 의 예시를 보면, HTTP 요청에 대한 응답 결과가 TC 내 Oracle과 다른 것을 확인 할 수 있다.

실험 내 모든 테스트 실패 케이스에 대한 수동 검사한 결과, 생성된 TC는 API 명세 사양과 테스트 시나리오를 충족하는 것으로, 다시 말해, API 결함인 것을 확인하였다. 이에 따라 API의 결함을 수정했으며, 그림 7 의 수정 후 테스트 결과와 같이 Oracle과 일치하는 응답을 얻을 수 있었다.

6. 결 론

APIM의 품질을 보장하기 위해 수용된 다양한 API의 테스트의 TC를 생성하는 활동은 노동집약적이다. 본 논문은 이를 해결하기 위해 LLM 기반 TC 자동 생성 기법을 제안한다. 제안하는 방법은 OAS문서와 테스트 시나리오를 바탕으로 프롬프트를 생성하고, 이를 LLM에 질의하여 자동 생성한다. KT GenieLabs APIM 시스템 내 NLP 관련 API 16 종을 테스트하기 위해 TC 96 개를 생성하였으며, API 테스트 결과 생성된 모든 TC에 대해 TC 결함이 없는 것을 확인하였다. 다만, 제안하는 방법은 vision, voice 등 Multimodal API를 테스트하기 위해 적용되기 어려운 한계를 가지고 있으며, 향후 Multimodal로 확장하기 위한 연구를 진행할 계획이다.

참고문헌

[1] “Global API Management Market 2023-2032.” (2023-05-06) Custom Market Insights.

[2] https://huggingface.co/spaces

[3] https://rapidapi.com/hub

[4] https://cloud.google.com/apis

[5] https://developers.naver.com

[6] https://developers.kakao.com

[7] Y. Bang et al., “A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity”, arXiv preprint arXiv:2302.04023, 2023.

[8] “Github copilot: your ai pair programmer.” [Online]. Available: https://github.com/features/copilot

[9] X. Hou et al., “Large Language Models for Software Engineering: A Systematic Literature Review”, arXiv preprint arXiv:2308.10620, 2023.

[10] Reynolds, Laria, and Kyle McDonell. “Prompt programming for large language models: Beyond the few-shot paradigm.” Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems . 2021.

[11] ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design.

[12] Software Testing with Large Language Model: Survey, Landscape, and Vision.

[13] Xie, Zhuokui, et al. “ChatUniTest: a ChatGPT-based automated unit test generation tool.” arXiv preprint arXiv:2305.04764 (2023).

[14] Wang, Chaozheng, et al. “No more fine-tuning? an experimental evaluation of prompt tuning in code intelligence.” Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering . 2022.

[15] Kang, Sungmin, Juyeon Yoon, and Shin Yoo. “Large language models are few-shot testers: Exploring llm-based general bug reproduction.” 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE) . IEEE, 2023.

[16] https://platform.openai.com/docs/guides/prompt-engineering

SDV 개발을 위한 차량 제어 소프트웨어 리팩토링

구태완 O, 김백준 O, 성병준 O, 조광현 OO, 이승준 OO, 손정호 OO
현대자동차 O, 에어플러그 OO
{taewan.gu, bjkim, blizzard}@hyundai.com, {jun, eric, jerry}@airplug.com

Vehicle Control Software Refactoring for SDV Development

Taewan Gu O, Baekjoon Kim O, Byungjun Sung O, Seung Jun Lee OO, Kwanghyun Cho OO, Jeogho Son OO
Hyundai Motor Company O, AriPlug OO

요 약

"소프트웨어 중심 차량 (SDV, Software Defined Vehicle)"을 지양하는 차량 개발 패러다임의 변화는 다양한 고객 니즈를 반영하기 위해 기존 차량 제어 소프트웨어 아키텍처의 변화를 필요로 한다. 기존 차량 소프트웨어 아키텍처는 제어 기능 (Function) 중심의 아키텍처로 독립적인 단일 제어기간 통신에 기반한 분산형 아키텍처이다. 그러나 SDV에서는 복합적이고 고도화된 고객 니즈를 만족시키기 위해 사용자 중심의 피처 (Feature)에 기반한 중앙 집중형 아키텍처를 필요로 한다. 그러나 아쉽게도 차량 제어 소프트웨어 아키텍처 변화에 대한 연구 또는 사례가 충분히 제시되지 못한 실정이다. 따라서 본 연구에서는 SDV를 지원하기 위한 차량 제어 소프트웨어 리팩토링에 대한 사례 연구를 진행 하였다. 차량 제어 소프트웨어 리팩토링은 다양한 관점에서 다루어질 수 있지만, 가장 먼저 필요한 기존 단일 제어 기능들을 바탕으로 사용자 중심 피처를 식별하는 활동에 집중하였다. 즉, 신호 중심의 파편화된 제어 기능들을 어떻게 사용자 중심의 피처로 재정의 할 것이며, 또한 사용자 중심 피처가 얼마나 올바르게 식별되었는지에 대한 결과를 고찰하였다. 마지막으로 SDV에 필요한 추가적이고 지속적인 차량 제어 소프트웨어 리팩토링 방안에 대해 고찰하였다.

1. 서 론

소프트웨어 기술이 발전함에 따라, 사용자 요구는 점점 더 복잡하고, 다양해 지며, 고도화되는 추세이다. 그래서 자동차 산업에서 소프트웨어의 중요성과 비중은 과거에 비해 기하급수적으로 높아진 것이 사실이다. 실제 최근 출시되는 차량들은 OTA (Over-The-Air)에 기반한 FOD (Feature On Demand) 서비스를 지원하고 있는데, 이를 통해 차량 성능이나 편의 기능을 추가적으로 제공하여 고객의 다양한 니즈를 충족시켜 나가고 있다. 또한 향후 자율 주행을 비롯한 개인화된 인포테인먼트, 그리고 물류 및 운송과 같은 상용차 분야에까지 다양한 서비스로 확대되어 나갈 것으로 기대된다.

이러한 일련의 소프트웨어 중심의 자동차 산업의 변화는 전세계적으로도 확산중인 흐름으로 2023년 현대자동차에서 SDV (Software Defined Vehicle) 전환 선언을 비롯한 2023년 9월 Mercedes-Benz Group의 MMA (Mercedes Modular Architecture) 플랫폼 및 MB.OS (Mercedes Benz Operating System) 구축 선언, Volkswagen Group의 Software Architecture E3 2.0, 그리고 Toyota의 Arene OS 확산 등 여러 완성차 업체에서도 소프트웨어 중심 차량 아키텍처 전환을 시도하고 있다.

일반적으로 차량 제어 소프트웨어는 차량에 탑재된 수많은 센서들을 이용하여 데이터를 수신하여, 운전자의 의지 및 사용자 요구 서비스를 처리 하는 다양한 제어 연산을 수행한 후, 이를 다시 차량 네트워크를 통해 각종 액츄에이터를 제어하는 역할을 수행한다. 따라서 신호 처리 및 제어 연산이 하나의 독립적인 기능으로 구분되어 신호 송/수신 기반의 아키텍처를 구성하고, 물리적으로는 제어기 (ECU, Electronic Control Unit) 기반의 분산형 아키텍처를 가지는 것이 특징이다. 그러나 최근 SDV 중심의 아키텍처에서는 다양하고 복합적인 고객 니즈를 반영하고 효율적인 FOD 제공을 위해 여러 기능들이 복합된 제어 수행이 가능하도록 피처 (Feature)가 중심이 된 중앙 집중형 아키텍처로 리팩토링 할 필요성이 대두되었다. 이때 독립적 기능 (Function)은 사용자 또는 운전자가 특정 목적을 가지고 수행되도록 요구하는 기능 또는 기능들의 연결을 의미하며, 복합 제어를 위한 피처 (Feature)는 운전자가 식별할 수 있는 기능 수행 환경으로 사용자 또는 운전자가 인지 할 수 있는 기능 단위로 정의할 수 있다. 따라서 복잡해진 고객 니즈를 처리하기 위해서는 제어기 기반의 파편화된 독립 기능 중심 아키텍처 보다는 사용자 또는 운전자 중심의 기능 단위인 피처

기반의 중앙 집중형 아키텍처를 이용할 경우, 보다 효과적인 FOD 지원 및 SDV 기반 여러 서비스 지원 확대, 그리고 차량 제어 소프트웨어 개발 및 유지보수 측면에서 유리하다는 장점이 있다.

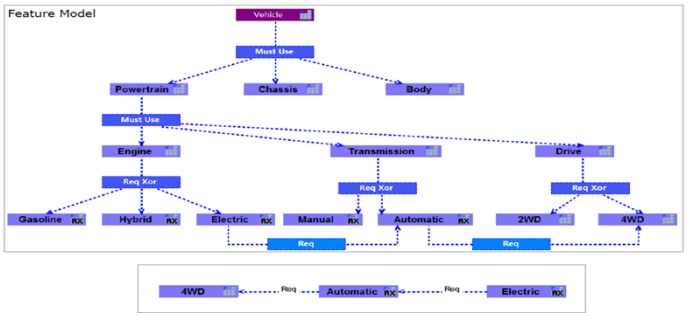
본 연구에서는 차량 제어 소프트웨어 아키텍처 변화에 대응하기 위해 전통적 아키텍처 리팩토링 관점에서 사례 연구를 수행하며, 기존 아키텍처 자산을 최대한 재사용 하는 것을 목표로 한다. 다만 통상 아키텍처 리팩토링은 다양한 아키텍처 관점에서 수행되어야 하지만, 본 연구에서는 기존 제어기 중심의 제어 기능에 대한 운전자 및 사용자 중심의 피처로 재정의 하는 것에 집중한다.

정리하면, (1) 소프트웨어 기술 발전에 따른 소프트웨어 중심 차량 개발 패처다임의 변화가 대두되었고, (2) SDV를 지원하기 위해 기존 차량 제어 소프트웨어가 분산형 아키텍처에서 복합 기능 즉, 피처 중심의 중앙 집중형 아키텍처로의 변화가 필요함을 인식하고, (3) 마지막으로 피처 식별에 집중한 아키텍처 리팩토링을 수행 및 결과 분석을 통해 리팩토링 효과를 살펴본다.

본 사례 연구는 상용 수소전기 트럭에 탑재되는 VCU (Vehicle Control Unit) 소프트웨어와 해외 유사 상용차의 제어 소프트웨어를 대상으로 진행되었다.

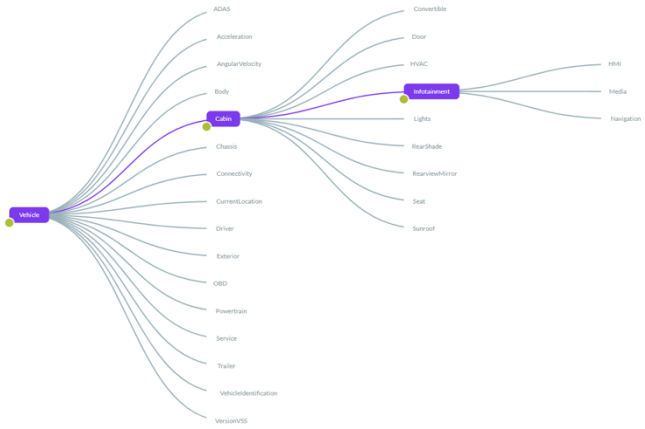
2. 관련 연구

소프트웨어 개발에 있어 피처를 중심에 놓는 FDD (Feature Driven Development)의 개념이 애자일 방법론의 일종으로 처음 소개되었고, 이후 보다 일반적인 설명을 Java 모델링과 분리하여 제시하였다. 또한 실무적으로 보면 차량 전기/전자 시스템 및 소프트웨어 기술 표준화를 위한 자동차 업계 협력의 일환으로 피처를 기술하는 데이터 모델의 표준화 작업이 있었다. 다음 [그림 1]은 표준 차량 플랫폼인 AUTOSAR (Automotive System and Software Architecture)의 피처 모델을 이용한 피처 모델링 사례를 나타낸다.



[그림 1] AUTOSAR Feature 모델을 활용한 Feature 모델링 사례

내용적 측면에서는, 차량 전기/전자와 관련한 기능을 구조적으로 분석하여 더 작은 단위의 요소로 분할하고 인터페이스를 표준화하려는 시도가 있었고[4], 최근에는 다른 단체에서 기능의 분할 구조보다는 차량 데이터의 표준화에 집중하여 연구개발을 진행하고 있다[5] (그림 2).



[그림 2] COVESA Vehicle Signal Specification

3. 피처 식별 (Feature Identification)

기존 차량 제어 소프트웨어는 요구사항 개발 단계부터 제어기별 기능 구현을 고려하여 개발이 이루어진다. 따라서 제어기별로 운전자 및 사용자 요구를 만족시키기 위해 필요한 정보들을 각종 센서 또는 스위치로부터 입력을 받아 차량의 구동, 제동, 조향, 공조 등의 기능이 정의되기 때문에 이들의 조합으로 궁극적으로는 초기 요구사항을 만족시킬 수는 있으나, SDV 관점에서 볼 때 운전자 및 사용자는 눈에 보이지 않는 이들 제어 기능 하나하나를 식별하는 것이 아니라, 사용자가 직접 인지하고, 성능/편의성을 제공하는 피처 단위로 소프트웨어를 간접적으로 인식하기 때문에 SDV가 급변하는 자동차 시장 환경에서 향상된 사용자 경험과 가치를 신속하게 제공하기 위한 시도라고 한다면, 그 사용자 경험과 가치는 피처 형태로 표현할 수 있다.

피처와 요구사항은 관점의 차이여서 공급자 입장에서는 요구사항이고 사용자 입장에서는 피처이다. 그러나 개발자 관점의, 개발자의 언어로 표현된 요구사항은 사용자와 소통하고 사용자의 요구를 담는 그릇으로 적당하지 않다. 사용자의 요구를 온전히 담은, 더 나은 경험과 가치를 제공하는 자동차를 만들기 위해서는 종래의 요구사항이 아니라 사용자의 입장에서 사용자의 언어로 기술된 피처를 중심에 두어야 한다.

3.1 피처 식별 템플릿 (Feature Identification Template)

차량 소프트웨어 피처를 사용자가 이해하는 언어로

기술하고 있는 문서는 바로 자동차 사용자 매뉴얼(Owner's Manual)이다. 본 연구에서는 상용차의 사용자 매뉴얼을 기초 자료로 아래와 같은 템플릿을 추출하였으며, 템플릿 구성은 다음과 같다.

- 사용자 관점의 피처 설명, 작동 방법, 작동 조건, 작동 결과 정리
- 해외 차종과 피처 비교 정보
- 피처 리스트 작성 및 분석에 유용한 형태로 정리
  - 피처 ID
  - 피처 개선 방향
  - 피처 연관 태그
  - 유사 피처 그룹

예를 들어 작업등(Work Light) 기능의 경우, 그림 2와 같이 정리할 수 있다.

[그림 3] 피처 자료화 예시

이와 같은 과정을 통해 총 113개의 상용차(트럭) 피처를 식별하였다. 주 분석 대상인 국내 차종으로부터 식별된 피처가 109개이고 (해외 차종에도 공통으로 존재), 해외 차종에서만 식별된 피처가 4개이다

3.2 피처 식별 분석 (Analysis of Feature Identification)

3.2.1 피처 분류 및 피처 뷰 포인트 정의

수많은 피처를 단순히 나열하면 내용 탐색이 용이하지 않고 정보를 가공, 활용하기에도 효율적이지 않다. 피처를 탐색, 검색하기 쉽도록 트리구조를 갖는 피처 분류 체계를 마련하였다. 대분류, 중분류 두 단계로 구성하였고, 중분류 아래에는 선택적으로 소분류를 둘 수 있다. 가장 상위의 대분류는 다음과 같다.

- Driving (주행)
- Energy (에너지)
- Safety (안전)
- ADAS (주행 보조)
- Body/Cabin (바디/실내)
- Commercial Purpose (상용)
- Infotainment (인포테인먼트)

전체적인 분류 체계는 다음 [표 1]과 같다. 트리 구조를 갖는 분류 체계 내에서 개별 피처, 즉 단위 피처는 분류 체계 내에서 유일하게 존재하고, 위치를 특정할 수 있다. 일종의 피처 주소 체계가 되는 셈이다. 이 분류 체계는 기본적으로 실무에서 쓰이는 전통적인 도메인 기준 기능 분류, AUTOSAR의 Application Interface, COVESA의 Vehicle Signal Specification을 참조하였다. 그리고 최근의 전동화 추세를 반영하여, 기존에 별도의 도메인으로 분류하던 Powertrain과 Chassis를 Driving 대분류로 통합하였고, 에너지를 대분류로 추가하였다.

피처 분류 체계는 피처를 식별하고 정리할 수 있는 도구가 될 수 있지만, 피처 상호 간의 관계를 깊이 이해하고, 기존 피처를 개선하거나 새로운 피처를 발굴하는 데 그다지 유용하지 않다. 즉 현재(AS-IS)를 있는 그대로 정리하는 것일 뿐 새로운 아이디어나 기술의 진화 방향성에 대한 통찰을 얻기 어렵다는 한계가 있다. 특정한 목적을 갖고 피처 또는 피처 집합에서 어떤 결과물을 얻어내기 위해서는 그 목적하는 관점에 맞게 관련 피처를 재구성, 재구조화 하여 보여줄 필요성이 있다. 대표적으로 사용자 중심의 관점에서 피처의 발전 방향을 검토하고 이를 SDV를 지원하는 차량 제어 소프트웨어 아키텍처 설계에 반영하기 위해서는 사용자의 관점에서 피처의 구조를 파악해야 한다.

이것이 관점(View)의 개념 도입이 필요한 지점인데, "User Scenario View"를 활용하여 SDV 지원 차량 제어 소프트웨어 아키텍처의 설계 포인트를 도출하고자 한다.

사용자 관점에서 시스템(차량)을 이용하는 사용자 시나리오 내지 스토리를 상정하고, 그 상황에서 사용되는 피처를 매칭할 수 있다. 이렇게 하면 개별 피처 상호 간의 관계를 더 깊이 이해할 수 있고, 개선이 필요한 피처, 새로운 피처를 찾기도 쉬워진다.

사용자 시나리오는 다양한 형태로 구성할 수 있는데, 본 논문에서는 아래 예와 같이 사용자가 상용차를 사용하는 상황별로 구성하였다.

- 출발: 원격 제어, 차량 접근/승차, 주행 준비/조정
- 주행
  - 차량 운전 및 운전자 보조 시스템 활용
  - 날씨 조건으로 인한 시야 확보
  - 야간 주행을 위한 시야 확보
  - 험로, 언덕길, 내리막길 주행
  - 사고 방지 및 대응 및 경제 운전
- 도착: 주차, 주차 후 하차, 충전, 차내 거주/휴식
- 화물 운송: 화물 적하차, 화물 운송 주행, 운행기록 관리, 화물칸 관리
- 정비: 자가/전문 정비, 폰/태블릿을 이용한 정비
- 차량 점검: 원격 점검, 출발 전 점검, 운행 중 점검

[표 1] 피처 분류 체계

대분류	중분류	설명
Body/Cabin	Accessibility	차량 접근성, 인증, 승하차 관련 피처, 예) 도난 방지, 도어락, 캡 틸팅
	Visibility	• 운전자에게 차량 외부에 대한 시야를 확보, 차량 주행이 가능하도록 돕는 기능 • 승하차 과정에서 승객에게 차량 외부 시야를 제공하여 승하차를 돕는 기능 • 외부에 시각적인 정보를 제공, 도로상의 다른 주체(차량, 보행자, 자전거 등)에게 알리는 기능 • 승객에게 차량 내부의 시야를 제공
	Comfort	운전자 및 사용자에게 안락한 편리한 내부 환경을 제공하는 기술 예) 유틸리티 모드, 온도 조절, 서스펜션 시트
	Warning	워닝 기능이 특정 기능 하위에 포함되어 있지 않은 경우, 예) 혼
Safety	Occupant Safety	사고 또는 위험 상황 직후 차량 내부 운전자 및 승객을 위한 안전 기술 예) 에어백, SOS 시스템
	Pedestrian Safety	사고 또는 위험 상황 시, 보행자 안전 기술, 예) VESS, DRL
ADAS	Driving Assist	• 주행 시 효과적인 경고 또는 적극적인 운전 개입을 통해 주행 유지 또는 사고를 방지하는 기술, 예) FCA, DSW • 운전 피로를 줄이고 운전자의 편의를 도모하기 위해 시스템이 적극적으로 주행을 유지 또는 사고를 방지하는 기술, 예) SCC, Downhill Cruise
	Parking Assist	주차 시 효과적인 경고 또는 적극적인 운전 개입을 통해 사고를 방지하고, 주차를 쉽게 할 있도록 돕는 기술, 예) RVM
Energy	Energy Storage	차량 주행을 비롯한 운영에 사용하기 위한 에너지 발전, 저장과 관련된 기술 예) H2 Tank Status, Air Pressure Gauge
	Energy Efficiency	에너지 효율 관점에서 목적이 차량의 연비/전비를 향상시키는 기술 예) DTE, Battery Discharge Warning
Driving	Powertrain	차량 주행을 위한 동력 생성 기술, 예) Drive Mode Shift, Top Speed Limit
	Transmission	생성된 동력을 전달(체결) 또는 단절(미체결)하는 기술, 변속을 위한 기어 변경 예) Differential Lock, Wheel Lock
	Brake	차량 제동, 감속 또는 정지 상태를 유지, 예) Retarder Brake, Brake Assist, EHS
	Steering	차량의 조향, 예) Rear Axle Steering
	Suspension	노면으로부터의 진동을 저감하여 편안한 승차감을 제공하는 기술로 차량 전복을 방지하는 기술, 예) ECAS
	Axle	차축은 바퀴를 통하여 자동차의 중량을 지지하는 축 제어 기술 예) Axle Up/Down, TPMS

[표 2] 사용자 시나리오 예시

Lv.1	Lv.2	사용자 시나리오	Feature Title (관련 Feature Matching)	기존 Feature 개선점 / 신규 Feature 필요성
출발	원격 제어	원격에서 스마트폰을 이용해 1) 주차 위치 확인 2) 주행 가능 거리 확인 3) 시동을 건다. 4) 공조를 켜고 설정한다. 스티어링 휠, 좌석 열선도 켠다. 5) 창문을 열어 환기 6) 운행 전 점검/진단: 차량 운행 전 차량의 상태가 양호하고 주행이 가능한지 종합적으로 점검	Bluelink	차량과 스마트폰 간의 다양한 사용자 경험 제공
			DTE (Distance To Empty)	
			Vehicle Start Stop	
			Temperature Adjustment	
			Auto Temperature Adjustment	외부 온도에 따라 Target 온도를 시스템이 자동으로 변경
			Steering Wheel Warmer	1) 열선 단계 조절 및 지속시간 도입 2) 원격에서도 미리 작동하도록 개선
			Heated/Ventilation Seat	원격으로 동작
			Window Opening-Closing	
			[NEW] 운행 전 차량 점검	운행 전 점검/진단: 차량 운행 전 차량의 상태가 양호하고 주행이 가능한지 종합적으로 점검
	차량 접근 및 승차	1) 다양한 형태의 키(FOB, 카드, 폰)를 갖고 차에 접근한다. 2) 키 인증 (키 형태에 따라 인증 절차는 명시적 또는 묵시적) 3) 차문을 열고 탑승한다.	Smart key	
			[NEW] Various Key Types	카드 키, 스마트폰 앱 키 지원
			Welcome Light	

[표 1]는 피처 분류 체계를 기반으로 사용자 시나리오 예시를 나타내고 있다. 또한 시나리오 외에도 다양한 뷰를 적용하고 시사점을 도출 할 수 있다. 본 사례 연구에서는 단순 사용자 시나리오에 추가하여 다음과 같은 추가적인 뷰를 정의하였다.

- **Operation Pattern View:** 사용자가 피처를 작동하는 방식
  - 사용자가 직접 작동: On-board, Remote
  - 자동으로 작동: enabled by Default, enabled by User
  - No Operation: Information, Warning, Comfort
- **Object View:** 피처의 목적 대상, 수혜를 받는 주체
  - 차량: 주행, 주차, 물류, 운송
  - 탑승자: 운전자, 승객
  - 외부: 보행자, 다른 차량, 기타 도로 이용 주체
  - 관리: 유지보수, 안전, 보안, 데이터 수집
- **Regulation View:** 규제와 관련된 피처
  - 안전(Safety): 내/외부 안전에 연관된 모든 것을 아우르는 규제
  - 환경(Environment): 엔진 출력, 배기, 연료, 소음 등에 대한 규제
  - 기술(Technical): 기술적 요구사항 규제
  - 기타: 다양한 요구사항 및 기타 분류 규제

3.2.2 피처 리스트 작성 및 피처 개선 방안

검색, 분석이 용이하고 향후에도 계속 피처를 추가하거나 쉽게 활용할 수 있도록 하려면 피처 리스트를 데이터베이스 형태로 자료화 할 필요가 있다. 그러기 위해서는 먼저 각 피처의 특징 및 유용하게 활용될 만한 제반 피처 관련 정보를 담을 수 있는 틀이 필요하다. [표 3]은 피처 속성을 정의하고, 그 속성의 내용을 작성하여 채우는 기준을 나타낸다. 명확성, 가독성을 높이고 피처 간 비교 분류 및 필터링을 쉽게 할 수 있도록 될 수 있으면 선택지에서 고를 수 있게 하였다. 속성을 큰 분류에서 보면 다음 세 가지로 나눌 수 있다.

1) **피처 식별과 관련한 속성:** Title, Description, Marketing Name, ID
2) **사용자와의 상호 작용과 관련한 속성:** User Control, 필요정보, Output
3) **피처 발전 방향과 관련한 속성:** Diagnostics, OTA, New Tech, EOL, FOD

[표 3] 피처 속성 정의

항목	설명
Title	피처 이름
Description	피처 기능적인 설명
Marketing Name	운전자 및 사용자에게 알려진 대중적인 명칭
ID	피처 식별자
User Control	사용자가 어떻게 기능을 조작하는지 (조작 방식)
필요정보	피처의 활성화 및 동작에 대한 조건이 식별된 경우 표기
Output	물리적인 출력 유형
Diagnostics	진단 여부(필요성), 진단 방식
OTA	OTA 소프트웨어 업데이트가 필요한지
New Tech	계속 발전하는 기술인지(New) 오래된 변하지 않는 기술인지(Legacy)
EOL Configuration	EOL 단계에서 설정이 가능해야 하는 피처
FOD	FOD 지원 여부

다음[표 4]는 전방 충돌 방지 (FCS, Forward Collision Avoidance) 피처의 식별 예이다.

[표 4] FCA 피쳐 속성

Title	FCA
Description	전방 충돌 방지 보조 (FCA) 전방 충돌을 피하거나 충돌시 속도를 줄여주는 피처
Marketing Name	전방 충돌방지 보조
ID	1
User Control	Manual
필요정보	전방 차량 거리 작동 조건 (속도, 방향 지시등, 가속)
Output	Vehicle Motion (차량 제동) Display Sound
Diagnostics	Remote
OTA	Yes
New Tech	New
EOL Configuration	Not Necessary
FOD	TBD

[표 5]는 본 사례 연구에서 정의한 피처간 관계 정의 구조이다. 피처 간의 관계를 정의하는 방법은 이미 AUTOSAR에서 제시하고 있는 기준이 있으므로 이를 차용하였다. 그러나 의미가 명확하지 않고 자의적일 수 있는 것은 배제하였고, 향후 차량 구매자가 카탈로그에서 선택하는 사양으로부터 피처 집합을 산출하는데 쓰이는 필수적인 관계 정의만 취사선택하였다.

[표 5] 피처간 관계 정의

Relation 속성	Description
USE	Decomposition/Hierarchy Category 부모-자식의 포함 관계를 가질 때, 즉 부모 피처를 더 작은 피처로 분할할 수 있음 1) MANDATORY,

	2) OPTIONAL, 3) ALTERNATIVE
REQUIRES	어느 한 피처가 다른 피처를 필요로 하는 관계
REQUIRES REF.	어느 한 피처가 필요로 하는 다른 피처들의 참조 목록

예를 들어 [표 6]에서 보는바와 같이 언덕길 발진 보조 (EHS, Easy Hill Start) 피처의 경우, 다음과 같이 정리될 수 있다.

[표 6] 피처간 관계 예시 (EHS)

Title	EHS
Description	언덕길 발진 보조, 경사로에서 정차하였다가 다시 출발 시키려고 할 때, 차량이 뒤로 밀리지 않게 함
Marketing Name	언덕길 발진 보조
ID	98
USE	OPTIONAL
RELATION	REQUIRES
RELATION REF.	Retarder Brake

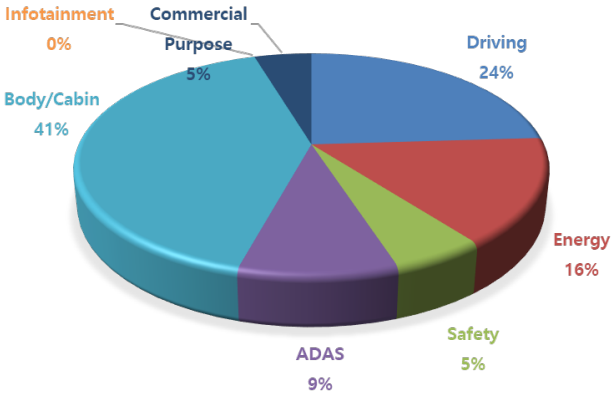
위 방식으로 정리된 피처 리스트를 기반으로 각 피처의 기술적인 발전 방향성을 검토하였다. 단일 피처 또는 연관된 여러 피처를 Use Case Diagram에 그려서 브레인스토밍을 통해 다각도로 살펴보며 개선 가능성 내지 개선 필요성을 찾았다. 이렇게 찾은 피처 개선 포인트를 공통적인 것끼리 분류하면 크게 다음 4개의 큰 범주로 나눌 수 있다.

[표 7] 피처 발전 방향성

개선 방향	분류 기준	사례
입력 다양화	피처 활성화, 동작 방법에 대한 사용자 입력 방식의 다양화 (새로운 사용자 인터페이스)	도어락 스위치 : 터치스크린 : 스마트폰 UI
출력 다양화	사용자가 체험하는 피처 동작 결과물의 다양화 (새로운 Actuator, 추가적인 액션)	수소누출탐지 : 블루링크 알림 DSW 개선 : 공조, Window, Warmer 등
기능 개선	기능 자체의 내재적인 개선을 통한 사용자 경험의 향상	VESS → 주변 환경에 따른 음량 조절 기능 추가
융복합 피처	기존 피처의 융복합	운행 전 사전 점검 알림 제공
		: TPMS : Brake Lining Warning 활용 신규 피처

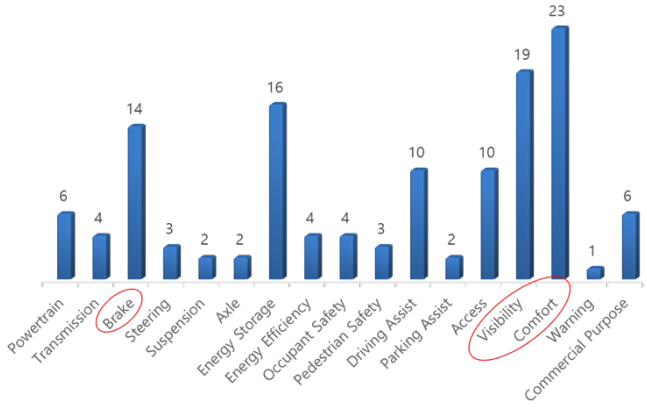
4. Validation of Feature Identification

사용자 매뉴얼로부터 식별한 피처의 통계적인 수치를 산출해 정합성을 살펴본다. 먼저 분류 체계의 대분류로 볼 때, [그림 4]와 같이 바디/실내, 주행처럼 차의 기본적인 기능과 관련된 피처가 많음을 알 수 있다. 또 FCEV 특성상 에너지 관련 피처가 많다.

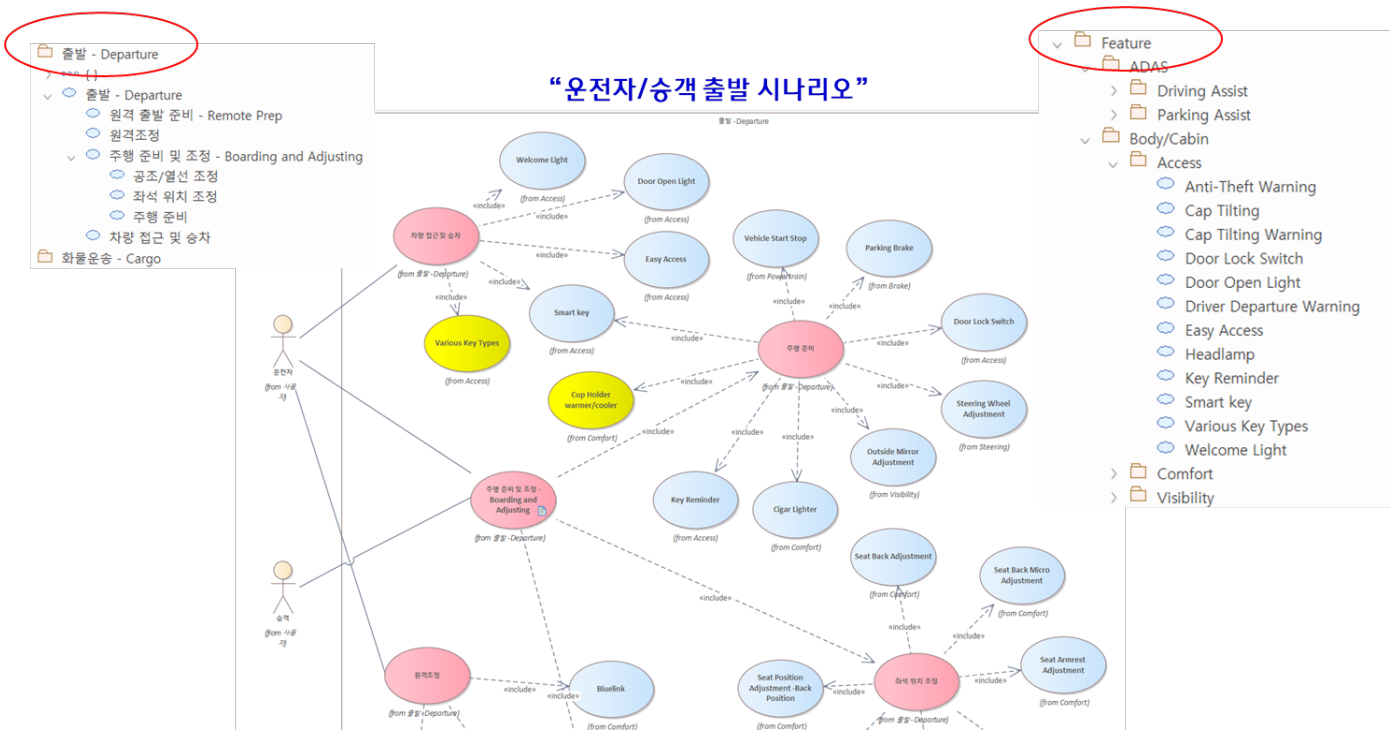


[그림 4] 분류 체계 대분류 분포

[그림 4]와 같이 분류 체계 중, 중분류로 보면, 대형 상용차의 특성상 Brake 관련 피처가 많고, 숫자로는 Comfort, Visibility 관련 피처가 많다. ADAS, 인포테인먼트 등 최근에 관심을 받는 기능은 상대적으로 비중이 크지 않다. 인포테인먼트 분야는 별도의 매뉴얼로 분리되어 있어 집계에 포함하지는 않았지만, 승용차 시장이나 인포테인먼트, 인터넷 연결성이 우수한 특정 해외 사례와 비교하면 양상은 크게 달라지지 않는다.



[그림 5] 분류 체계 중분류 분포



[그림 6] 유즈케이스 다이어그램 사례

피처 리스트와 사용자 시나리오의 관계를 고찰해보면, 사용자 매뉴얼에서 식별하고 양식에 맞춰 정리한 피처 리스트는 피처의 원본을 담고 있고, 사용자 시나리오에서는 특정 상황, 조건에서 개별 피처를 사용, 즉 참조한다. 그렇다면 사용자 시나리오가 실제 사용자가 경험하는 다양한 상황을 반영하고 있는지는, 개별 피처가 얼마나 사용자 시나리오에서 사용(참조)되고 있는지 그 커버리지로 환산할 수 있다. 또 사용자 매뉴얼에는 없는데 차량에는 존재하는 Feature가 있다면 그것은 사용자 매뉴얼에 누락되었음을 의미한다. 그리고 사용자 시나리오를 면밀히 검토하면서 현재는 존재하지 않지만 신규로 필요한 피처를 발굴할 수도 있다.

5. Discussion and Future Work

피처 리스트는 개발 측면에서 볼 때 E/E 아키텍처 설계 요구사항이 된다. 이 맥락에서 기존 피처 개선 또는 미래의 피처를 검토한다는 것은 거기에서 찾은 개선 방향, 변경 사항이 차세대 E/E 아키텍처의 설계 요구사항이 된다는 의미가 있다. 본 사례 연구에서 도출한 기술의 발전 방향성(입력 다양화, 출력 다양화, 기능 개선, 융복합)을 반영하는 차세대 E/E 아키텍처를 개발하는 것이 다음 과제이다.

하드웨어, 소프트웨어를 모두 포함한 통합 E/E 아키텍처를 설계에 들어가기 전에 중간 과정으로 "논리적 기능 아키텍처"를 개발할 예정이다. 기술 발전 방향성을 기능 아키텍처의 설계 주안점으로 삼아 다음과 같은 과정을 거쳐 기능 아키텍처를 완성할 예정이다.

1) 기능 아키텍처 작성 방법 및 Notation 검토
2) 기능 아키텍처 주요 설계 목표 도출
   • 사용자 관점 피처 개선 방향에 따른 설계 목표
3) 기능 아키텍처 설계 대상 피처 선정
   • 개선 방향성 고려한 효용성 있는 피처 선정
   • 아키텍처 설계 포인트 구체화
4) 기능 아키텍처 작성
   • 상위 레벨 기능 정의 및 Decomposition
   • 구체화된 아키텍처 설계 포인트 반영

6. 결론

자동차 업계에서 SDV 전환을 위해 차량 제어 소프트웨어 전반에 변화가 필요한 것은 이미 알려진 과제이지만, 안타깝게도 무엇을 어떻게 변화해야 하는지에 대한 구체적인 사례는 자동차 제조업체별로 개별적으로 시행되다 보니 적절한 사례를 찾아보기 어렵다. 본 사례 연구에서는 SDV 전환에 대응하기 위한 차량 제어 소프트웨어 아키텍처 리팩토링 필요성을 살펴보고, 그 첫번째 단계로 기존 독립 제어기 단위의 기능(Function) 기반에서 복합 기능 형태의 피처(Feature) 기반 중앙 집중형 아키텍처로의 변환에 필요한 피처 식별에 대한 사례를 살펴 보았다. 이를 통해 아키텍처 전반에 걸쳐 필요한 리팩토링 요소를 정의할 수 있는 기반을 확보하고, 향후 SDV 패러타임에

적극적으로 대응할 수 있는 기반을 마련하고자 한다.

참고 문헌

[1] 김주성, “자동차 소프트웨어 생태계 관련 주요 이슈 및 발전방안,” ETRI Journal, 2023.10.01.

[2] 현대자동차그룹, “소프트웨어 혁신으로 SDV 전환을 앞당기는 현대자동차그룹,” 1 June 2023. [온라인]. Available: https://www.hyundai.co.kr/story/CONT00000000 00094656.

[3] Mercedes-Benz Group, “Digital First Production,” Sep. 20. 2023. [온라인]. https://group.mercedes-benz.com/innovation/digitalisation/industry-4-0/digital-first.html

[4] AUTOSAR Feature Model Specification. [온라인]. Available: https://www.autosar.org/.

[5] AUTOSAR Application Interface Specification. [온라인]. Available: https://www.autosar.org/.

[6] AUTOSAR Vehicle Signal Specification. [온라인]. Available: https://www.autosar.org/.

[7] Coad, P., Lefebvre, E. & De Luca, J., “Java Modeling in Color with UML: Enterprise Components and Process,” Prentice Hall International (ISBN 0-13-011510-X), 1999.

[8] Palmer, S. R., Felsing, J. M., “A Practice Guide to Feature-Driven Development,” Prentice Hall (ISBN 0-13-067615-2), 2002

2024 한국 소프트웨어공학 학술대회 (KCSE 2024)

LAMP7 | JISAN (주)지산웨어

1등급 Software Process GOOD Software

로우코드개발플랫폼(LCDP)과 RPA, ML을 융합한 Intelligent Hyper Automation

Low code based
Agile prototyping
Methodology
Platform

소프트웨어,
설계는 사람이 하고 개발은 로봇이 합니다.

[7.1.02.01]

spring eGovFrame SAP

* LCDP(Low Code Development Platform)
* RPA(Robotic Process Automation)
* ML(Machine Learning)

www.swrobot.com

LAMP7 | JISAN (주)지산웨어

목차

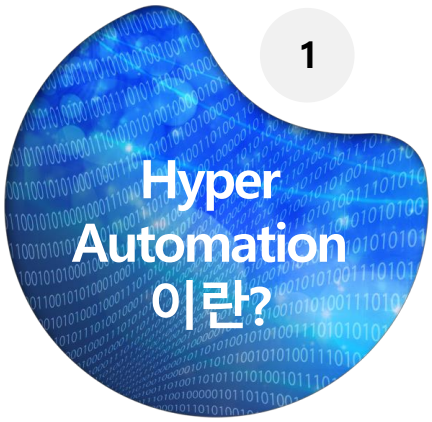
Contents

1 Hyper Automation 이란?
1.1 사례) 입찰관리 AS-IS 업무프로세스
1.2 업무자동화 - RPA적용
1.3 업무자동화 이후 시사점
1.4 개선방안 – Hyper Automation의 출발
1.5 Intelligent Hyper Automation!

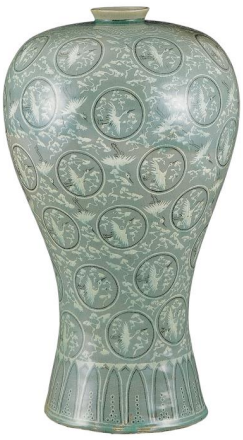
2 Hyper Automation SDLC
2.1 로우코드 개발 플랫폼(LCDP)
2.2 Hyper Automation으로 확장

3 기대효과
3.1 업무적 효과
3.2 기술적, 경제적 효과

* SDLC : Software Development Life Cycle



Intelligent Hyper Automation



고려시대 도공

고려청자는
깨끗한 백토를 빚어 영롱한 비취 빛 불꽃이 나오도록
3일 밤낮으로 불을 때야 만들 수 있는 것이다.

월100만 PCS 자동생산공장

1. 불순물이 제거된 자토(고령토)를 고르게 반죽한다.
2. 반죽을 물레에 올려 형태를 만든다.
3. 표면에 상감 문양을 새겨 넣고 백토(자토)와 적토(도토)를 넣는다.
4. 채색이 끝난 청자는 800도에서 초벌구이를 거친다.
5. 초벌 구이 된 청자에 유약을 바른다.
6. 가마에 넣어 1300도의 온도로 48시간 동안 굽고 식기를 기다려 청자를 꺼낸다.
7. 꺼낸 뒤 결함이 있으면 그 자리에서 쇠망치로 깨 버린다.
8. 생산된 자기는 개별포장을 하고 10개 Box에 담아 100BOX씩 팔레트로 창고에 입고한다.

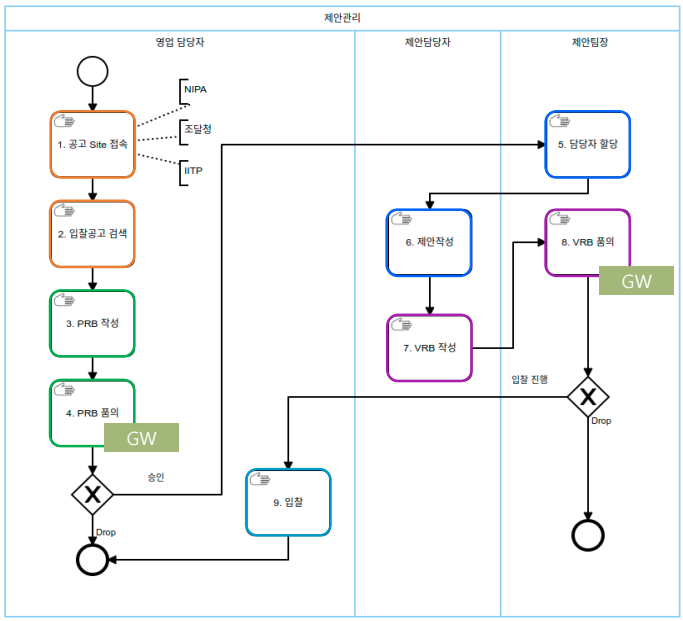
Intelligent Hyper Automation

비즈니스를 최소의 비용으로 지능화 하기 위해서는
비즈니스 영역을 정해 프로세스를 재 디자인하고
절차에 따른 기초 Data를 기반으로 연관업무가 처리 되도록 시스템을 만들어야 합니다.

불특정하거나 정성적인 업무들을 계량화, 절차화 하여 자연스럽게 지능적으로 처리되도록 하고
반복 학습모델로 업무개선과 확장을 해 나갈 수 있도록 해야 합니다.

1.1 사례) 입찰관리 AS-IS 업무 프로세스

입찰관리 업무 사례의 AS-IS 프로세스는 일부 그룹웨어와 수작업으로 진행해 왔습니다.



*PRB : Proposal Review Board * VRB : Value Review Boar

1.2. 입찰공고는 영업담당자가 수시로 NIPA, 조달청, IITP 등의 인터넷 사이트를 방문하여 검색합니다.
- 검색 시점은 주1회, 월1회 등 영업 담당자 별로 개인차가 있습니다.
- NIPA, 조달청, IITP 등 각 사이트별로 검색 키워드나 분류가 다르며 사업공고를 확인하는 영업담당자 별 개인 성향에 의해 사업이 검색됩니다.

3.4. 검색된 입찰공고에 대한 제안 의사결정을 위하여 제안요청검토 (PRB)자료를 만들어 협의하고 그룹웨어로 결재를 득합니다.
- PRB회의는 영업담당자가 준비하며, 회의양식은 기존 회의양식을 참고한 뒤 예상이익, 영업전략, 경쟁 우위 현황 등을 작성합니다.

5.6. 제안담당자가 배정되어 제안을 작성합니다.

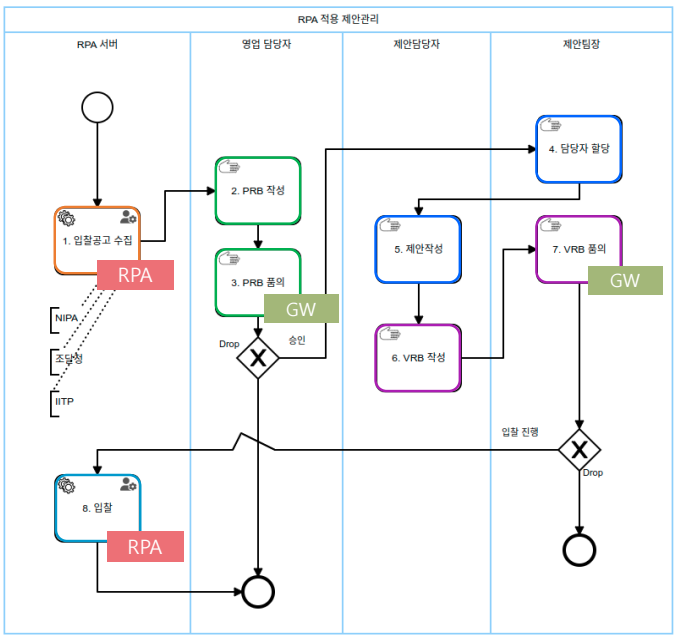
7.8. 제안작성이 완료되면 수주가치평가(VRB)자료를 작성하고 입찰 진행 의사결정을 위하여 VRB 회의를 진행하고 그룹웨어를 통해 승인을 받습니다.
- VRB회의는 제안담당자가 준비하며, 회의 양식은 기존 양식을 참고한 뒤 원가,이익률 등을 산정하여 작성합니다.

9. 입찰이 결정되면 영업담당자가 해당 사이트에서 입찰을 수행합니다.

수작업 그룹웨어 RPA LCDP ML

1.2 업무자동화 – RPA 적용

최근 5년 전부터 RPA를 적용하여 고객접점의 단순 업무, 반복적인 내부 업무를 자동화하여 성과를 얻었습니다.



1. 입찰공고는 RPA에 의해 주기적(일 1회, 시간당 1회 등)으로 각 사이트에서 자동 수집 합니다.
- 사이트별 검색 기준 및 Rule에 의해 자동화 합니다.

2. AS-IS 업무 절차와 동일

3. AS-IS 업무 절차와 동일

4. AS-IS 업무 절차와 동일

5. AS-IS 업무 절차와 동일

6. AS-IS 업무 절차와 동일

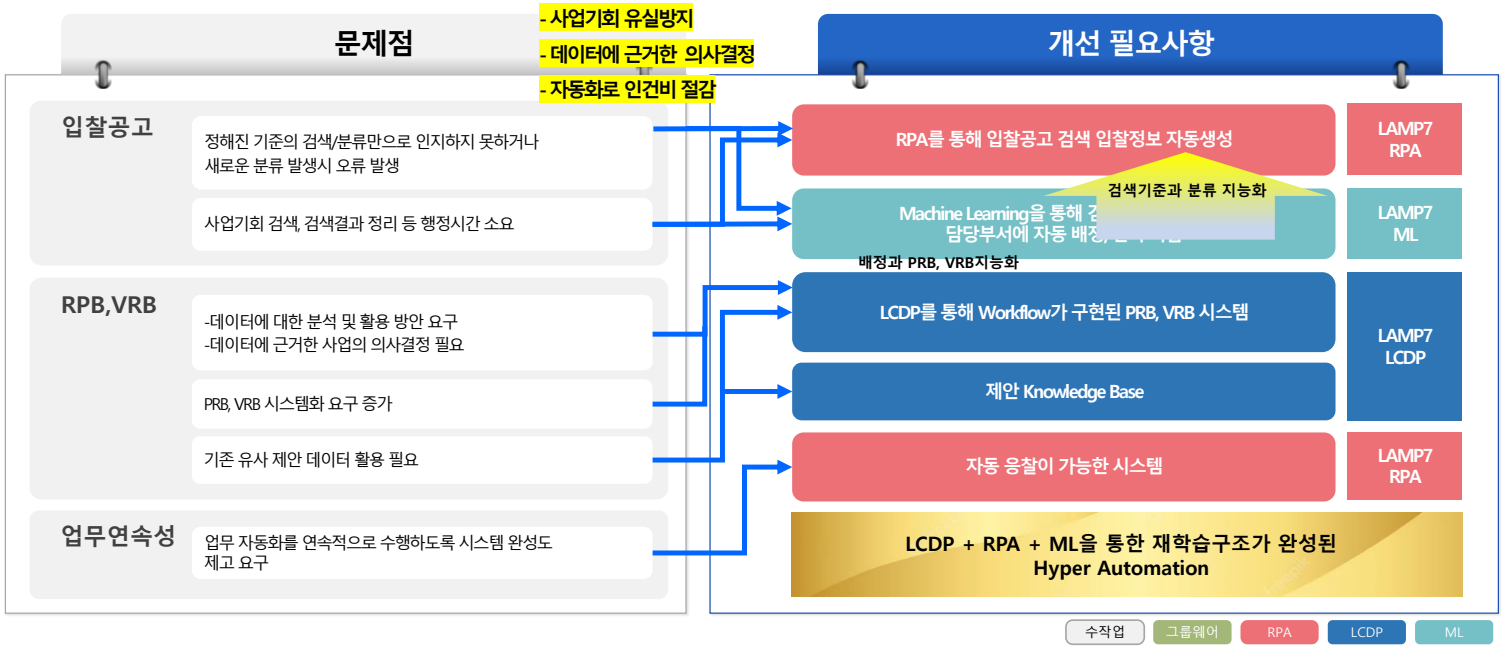
7. AS-IS 업무 절차와 동일

8. 입찰을 위한 리뷰보드가 완료되면, 제안 마감 시간에 맞춰 RPA가 VRB 내용대로 자동 입찰합니다.

수작업 그룹웨어 RPA LCDP ML

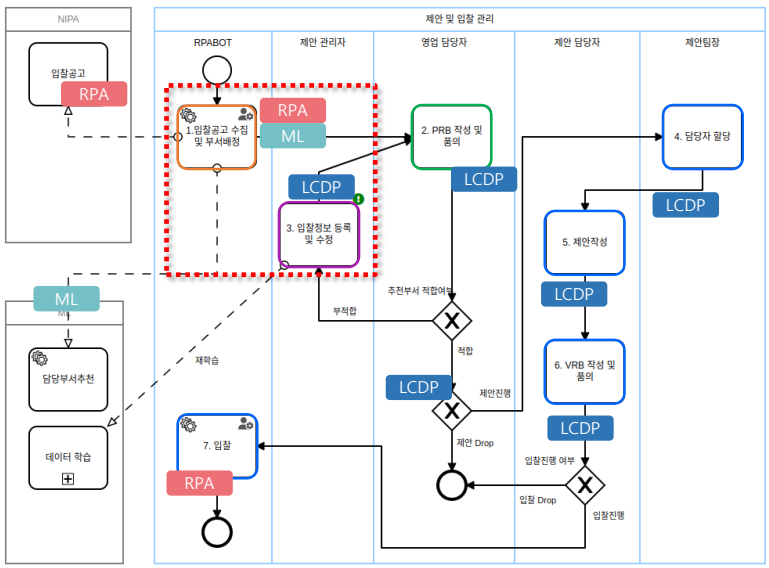
1.3 업무자동화 이후 시사점

AI시대가 도래하면서 기존 시스템을 활용하여 내,외부를 넘나드는 Operation중심의 자동화 뿐만 아니라 디지털화, 지능화 요구가 증대되면서
업무가 통합/분할/ 수정 되어 더 자동화,지능화 되길 요구하고 있습니다. 이는 업무 재설계를 통해 시스템을 혁신적으로 재구성하는 것을 필요로 하고 있습니다.



1.4 개선방안 – Hyper Automation의 출발

업무와 정보를 부서중심이 아닌 업무처리 중심으로 전환하고 병합하거나 분리하여 최적화합니다.
미분류, 미정의, 부정형을 수용하도록 시스템을 구성하고 모든 정보를 디지털화 하여 사람 개입을 최소화하고 지능화, 조자동화 하도록 시스템을 재구성합니다.

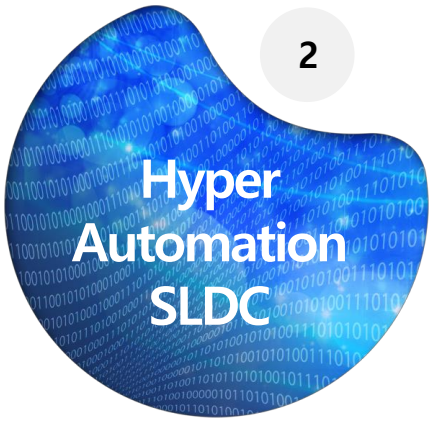


1. RPA에 의해 주기적으로 수집된 입찰공고를 Machine Learning 기능을 활용하여 각 부서로 자동분류하고 생성합니다.
   - 수집정보로 어느 부서에서 수행하는지를 포함하여 입찰정보 생성
2. PRB 작성자는 내용을 분석하고 품의합니다.
3. 부서가 적절하지 않은 경우 부서를 조정합니다.
4.5.6.7. 제안검토(PRB), 담당자지정, 제안, 수주가치평가(VRB)를 시스템으로 수행할 수 있도록 LCDP로 빠르게 구현, 적용합니다.
   - 리뷰보드의 모든 양식과 프로세스는 표준화되어 운영됩니다.
   - 과거 데이터 축적, 분석을 통해 합리적이고 체계적인 리뷰보드 운영이 가능합니다.
   - 과거 제안 내용을 통해 제안 작성이 용이해 집니다.
8. 입찰 리뷰보드가 완료되면, 승인된 VRB결과를 기준으로 RPA가 자동으로 입찰합니다.

수작업 그룹웨어 RPA LCDP ML

1.5 Intelligent Hyper Automation!

RPA의 적용은 **수작업, 외부시스템, Legacy시스템을 변형없이 활용하여 연계**, 자동화하는 부분에는 저비용으로 큰 역할을 했지만,
업무절차를 **개선 또는 통합** 하거나, **지능화 및 초자동화하기 위하여 LCDP와 Machine Learning 기술이 추가로 필요**합니다.



2.1 로우코드 개발 플랫폼 LAMP7

LCDP는 빠르고 쉽게 고품질의 시스템을 만들도록 지원합니다.

01

Drag & Drop, GUI를 통해 프로세스와 화면을 설계하여 쉬운 개발을 지원하고
보안(Secure Code)을 적용한
Spring framework, 전자정부 프레임워크 기반의 소스를 생성하거나
SAP GUI ABAP 소스 또는 Web + SAP ABAP 소스로 생성해 줍니다.

02

요구/설계/개발 등 **단계별 설계정보와 소스의 버전 및 이력을 통합 관리**하고,
한번의 설계로 **테스트와 매뉴얼까지 자동으로 실행하고 생성**하여
산출물을 따로 만들 필요가 없습니다.

LAMP 7
(Ver 7.1 Since 2002)

03

설계과정에서 소스가 자동생성 되고 즉시 빌드함으로써
실시간으로 설계한 시스템을 추가 개발 없이 작동해볼 수 있습니다.

04

프로세스 설계 시 **자동 실행할 프로세스를 식별**하여
RPA를 활용할 수 있도록 지원하며,
기계학습(ML) 과정을 거쳐 **업무를 지능화 하도록 지원**합니다

LAMP7은 2002년 BPMN기반의 Business Process Modeling 도구 개발을 시작하여 EAMS(2005), NCDP(2010), PMS(2013)를 거쳐 Agile-Prototype 방식의 SDLC를 지원하는 LCDP(2022)로 발전해 왔고 RPA, ML(2023)이 추가되었습니다.
*BPMN(Business Process Model & Notation) * EAMS(Enterprise Architecture Management System) * NCDP(No Code Development Platform) * PMS(Project Management System)

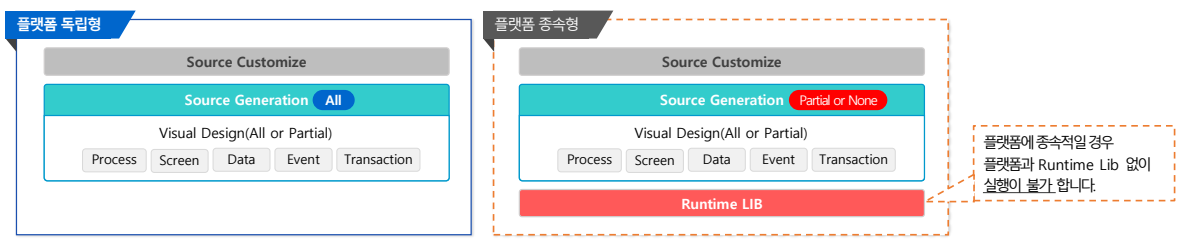
2.2 LCDP 서비스 수준별 분류

3Level, 플랫폼 독립형!

LCDP는 SDLC지원 범위에 따라 **부분적인 설계로 부분적인 소스가 생성되는 1 Level,** **전체 설계로 전체 소스가 생성되며 설계와 소스에 대한 버전, 이력, 배포관리까지 지원 2 Level,** **요구분석, 프로세스설계, 자동테스트, 자동실행까지 가능 3 Level로 분류**할 수 있습니다.
플랫폼의 독립성에 따라 전체소스가 생성되어 독립실행되는 플랫폼독립형과 플랫폼기반과 RunTime LIB가 있어야 동작되는 플랫폼종속형으로 구분됩니다.
LAMP7은 3 Level, 풀스택을 지원하는 플랫폼 독립형입니다.

SDLC 지원 범위에 따른 개발플랫폼 Level 분류

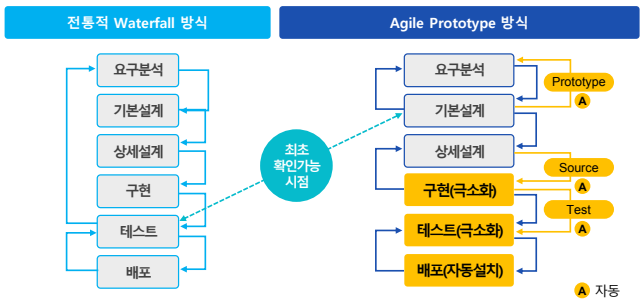
설계와 소스생성 범위에 따른 실행플랫폼 방식 분류



2.3 Agile Prototyping LCDP

정보공학 방법론에서는 요구사항이 적용된 시스템을 확인할 수 있는 시점이 늦어지고, 요구사항 변경의 어려움을 극복하기 프로토타이핑방법론에서는 기본설계만으로 CRUD가 가능한 시스템을 생성하여 요구자와 설계자가 상호검증을 통해 요구를 확정하고, 상세설계 수행을 통해 코딩, 테스트, 매뉴얼 작성 등을 **자동화 하여 생산성을 극대화** 합니다.

전통적 SDLC와 Agile-Prototype LCDP 비교



- 기본설계 후 Prototype System을 자동생성하여 요구사항 적용결과 확인 및 확정
- 상세설계 후 Source를 자동으로 생성하여 공수 절감 극대화
- 테스트는 비즈니스 케이스 기반의 Robot-Framework로 자동실행
- **요구사항이 반영된 정확한 설계, 개발 생산성 향상, 일정 단축의 일거삼득 효과**

LAMP7의 자동화 범위

- 분석단계의 60%, 설계단계의 40%, 개발과 시험단계의 90% 공수 절감 가능
- 프로세스, 데이터, 화면 설계로 Web이나 SAP 프로그램을 자동 생성해주고, 시험을 자동화
- **업무, 설계정보와 프로그램 자동화/동기화로 Biz-Dev-Ops의 IT Governance 확보**

등록번호	출원일	등록일	특허명
10-23-77607	2020.06	2022.03	정보시스템을 설계하여 프로그램과 데이터베이스를 자동으로 생성하고 테스트를 자동 수행하는 소프트웨어 공학플랫폼

2.4 서비스 지원 범위

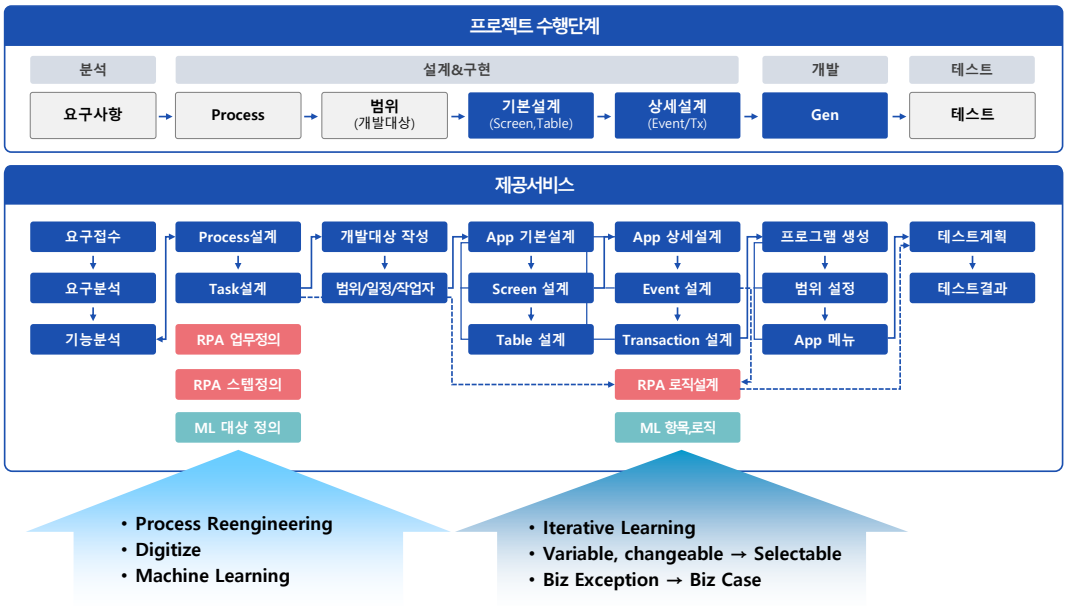
프로젝트 시 2가지 수행방식을 지원합니다. ①선후행 업무를 순서대로 수행(Tightly Coupled)하는 방식, ②각 단계를 개별적으로 수행(Loosely Coupled)하는 방식입니다. 소규모 프로젝트는 기본 및 상세 설계, Gen 등 Mandatory영역만을 사용할 수도 있고, 여러가지의 개별도구를 사용함으로써 분산 관리되던 **업무정보, 설계정보, 시스템정보 등 정보화 자산을 LAMP7에서 통합관리** 합니다.

- Enterprise Service: Option + Mandatory
- Standard Service: Mandatory

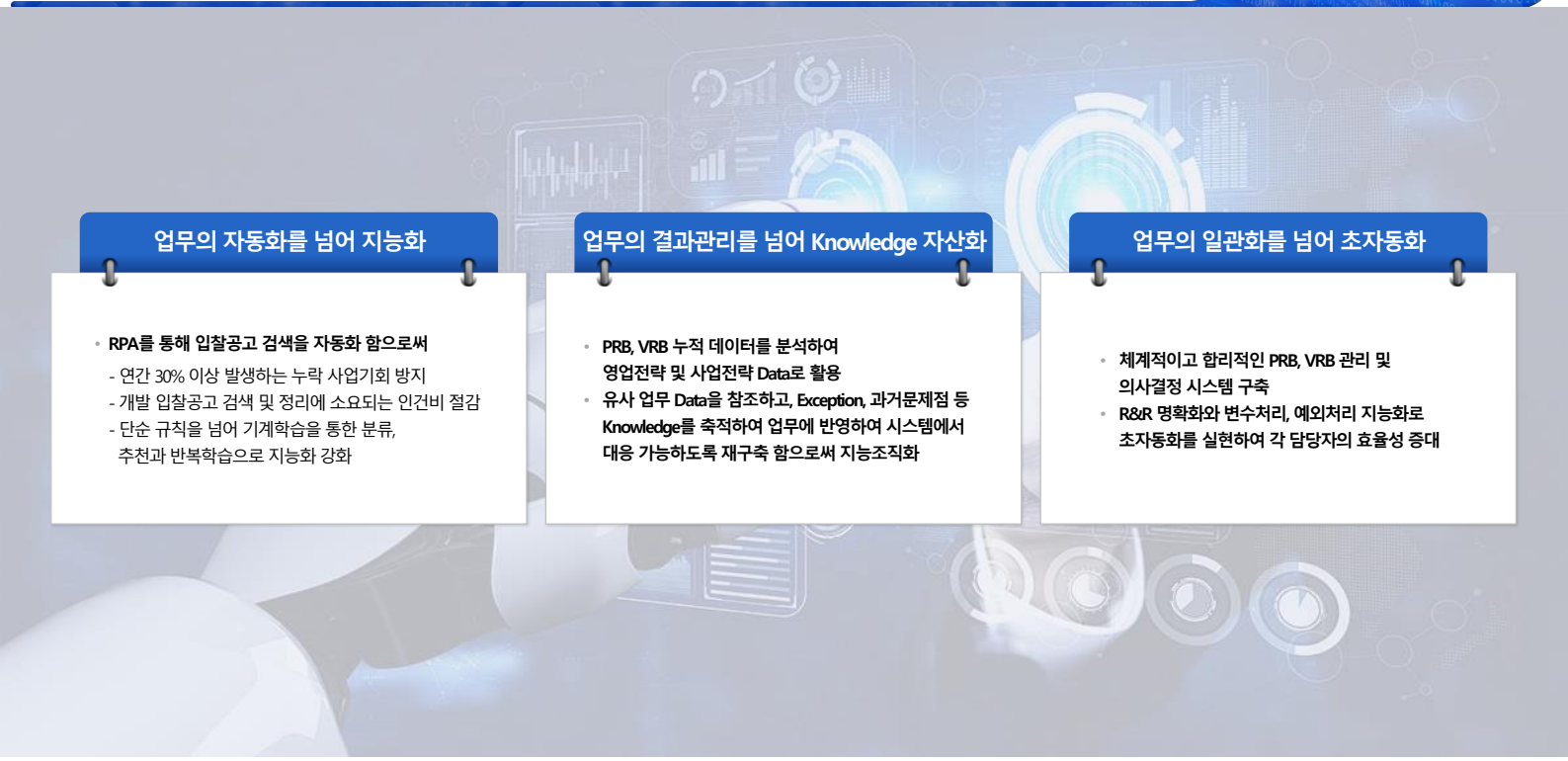
등록번호	출원일	등록일	특허명
10-17-27186	2015.08	2017.04	분할발주 및 원격지 개발 지원을 위한 **소프트웨어 공학 시스템**
10-17-42157	2015.08	2017.05	분할발주 및 원격지 개발 설계 진척관리를 위한 **프로젝트 관리 시스템**

2.5 Hyper Automation으로 확장

LCDP, RPA, Machine Learning이 융합된 소프트웨어 로봇으로 기업의 모든 업무를 설계만으로 디지털 전환할 수 있습니다.
프로세스를 Reengineering, 디지털화하고 반복 기계학습으로 자동화, 지능화 합니다.
또한 절차와 데이터의 변수를 상수화하는 매커니즘으로 조직 전체를 Rebuild하여 Hyper Automation으로 확장 가능하도록 지원합니다.



3.1 업무적 효과



업무의 자동화를 넘어 지능화

- **RPA를 통해 입찰공고 검색을 자동화 함으로써**
  - 연간 30% 이상 발생하는 누락 사업기회 방지
  - 개발 입찰공고 검색 및 정리에 소요되는 인건비 절감
  - 단순 규칙을 넘어 기계학습을 통한 분류, 추천과 반복학습으로 지능화 강화

업무의 결과관리를 넘어 Knowledge 자산화

- **PRB, VRB 누적 데이터를 분석하여 영업전략 및 사업전략 Data로 활용**
- **유사 업무 Data을 참조하고, Exception, 과거문제점 등 Knowledge를 축적하여 업무에 반영하여 시스템에서 대응 가능하도록 재구축 함으로써 지능조직화**

업무의 일관화를 넘어 초자동화

- **체계적이고 합리적인 PRB, VRB 관리 및 의사결정 시스템 구축**
- **R&R 명확화와 변수처리, 예외처리 지능화로 초자동화를 실현하여 각 담당자의 효율성 증대**

3.2 기대 효과

	공급 기업	설계·개발자	수요 기업·기관
설계	• **설계, 개발, 완료시점의 설계산출물 중복작성 배제** • 매뉴얼 등 부속 산출물 작성 배제 • 설계공수 40% 절감	• **요구사항 확정이 용이** • 반복 중복 작업 배제 • 설계 형상관리 용이	• **정확한 코드수준 설계정보 확보** (사람으로부터 독립) • 시스템과 동기화된 매뉴얼 확보 • 현업 참여로 사전검증 가능
개발	• 개발공수 90% 이상 감소	• 일정 획기적 축소로 야근 회피 가능	• 정확한 일정관리 가능, 개발자 암묵적 지식의존 배제
시험	• 데이터기반 시험 자동화로 빠짐없는 시험 진행 • **반복시험에 소요자원 급 감소로 비용,일정 감소**	• 테스트 데이터와 시나리오 확보시 추가 부담 없음	• 고객시험, 인수시험 부담 감소 (데이터, 케이스 검증 위주)
운영	• 사업자 변경시에도 안정된 서비스 제공 가능	• **기 설계/개발된 내용 파악 용이**	• **개발 사업자 종속성 배제 가능**
품질	• 설계, 생성단계에서 Syntax 오류발생 배제 • **논리오류 검증에 집중함으로써 전체 품질 향상** • **품질 기능 내재화로 품질향상 유도**	• 논리적 오류를 막아 고품질 확보 가능	• **개발능력 개인차 축소, 초급자도 고품질 개발효과**
인력	• 인력 수요 감소(50%이상) • 설계 수요 증가로 시니어 인력 활용가능	• **동료의 개발 생산성 걱정이 없음**	• 인력 수요 감소(50% 이상)
비용	• **설계공수 40%, 개발공수 90% 이상 절감** • **개발인력 유지, 관리비용 감소**	• 부가가치 창출 능력 2~3배 증가	• **총사업비 50%이상 절감, 차세대 시 75%이상 절감** - 개발 공수 감소로 전체 비용의 50% 절감 - 설계 재사용으로 설계비용의 25%추가 절감

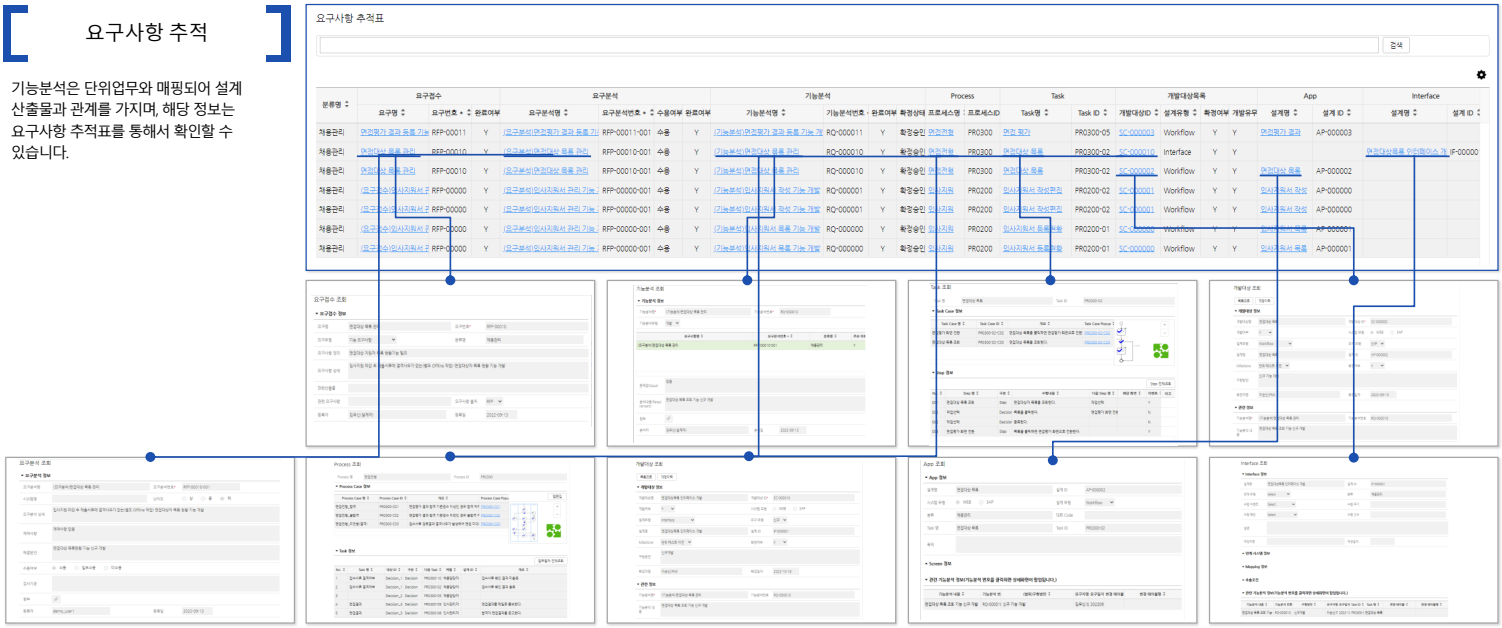
요구 관리(2/2)

★ 별첨

요구접수, 요구분석, 기능분석이 완료되면 Process/Task, 개발대상, 기능명세, Interface, Test 등 프로젝트 수행 절차에 따라 요구사항 추적관리가 자동으로 제공됩니다.

요구사항 추적

기능분석은 단위업무와 매핑되어 설계 산출물과 관계를 가지며, 해당 정보는 요구사항 추적표를 통해서 확인할 수 있습니다.



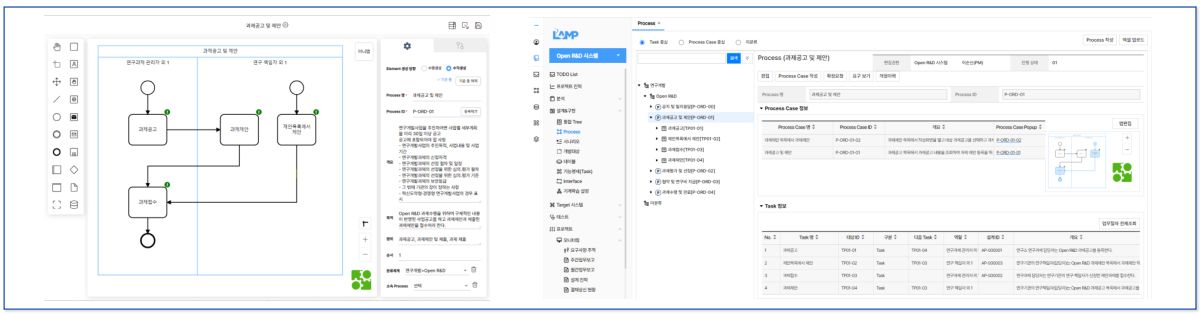
Process Design

★ 별첨

업무의 분류체계를 정의하고 흐름을 Process, Task, Step으로 구분하여 BPMN 기반으로 Process Diagram과 Task Diagram으로 작성합니다. On-Line Task는 개발 대상으로 정의하고, Step은 단위 프로그램의 Event로 연결되어 업무와 시스템이 동기화됩니다.

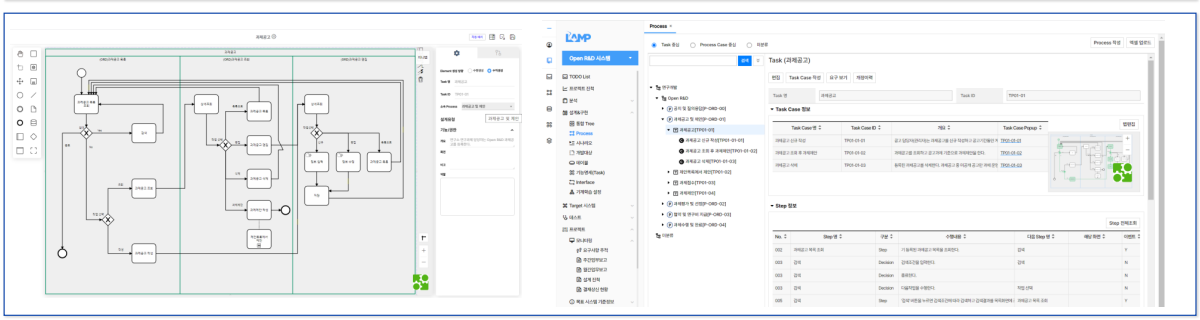
Process

Process Designer는 단위업무를 Task로 정의하고 스위밍 레인에 수행 역할(Role)을 정의하여 누가 어떤 업무를 어떤 단계로 수행하는지 쉽게 식별하도록 합니다.



Task

Task내의 업무절차를 Step으로 정의하여 단위업무내의 처리 절차와 수행해야 할 구체적인 처리 내용을 정의합니다.

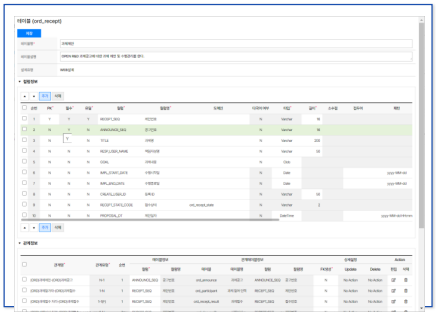


Data Design

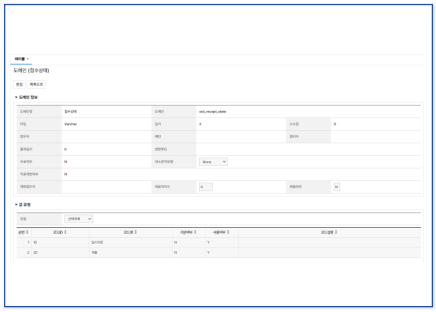
★ 별첨

테이블 Designer에서 테이블, 컬럼, 관계, 인덱스 정보를 정의하고, 컬럼에 사용되는 도메인을 정의하여 설계합니다. 관계 정보는 논리관계, 물리관계로 구분하여 정의할 수 있으며, 화면 설계 시 정의한 관계를 바로 사용할 수 있습니다. 일시적으로 사용하는 데이터 관계는 Event/Transaction 설계 시 SQL 편집기로 Join해서 사용할 수 있습니다.

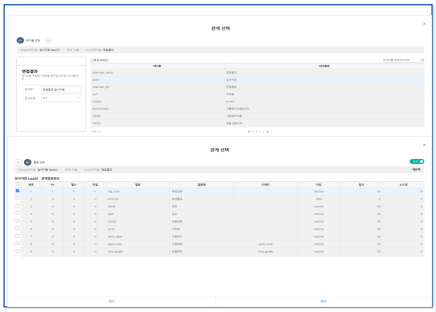
테이블 정의



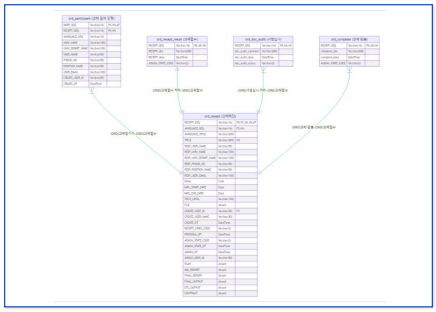
도메인 정의



Data 관계 정의



ERD View



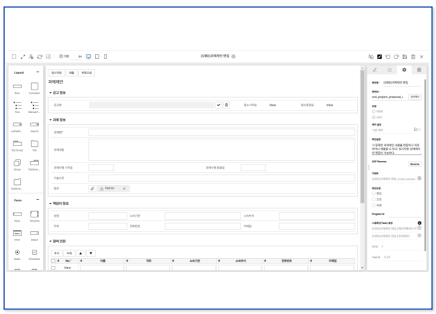
Screen Design(1/2)

★ 별첨

Screen Designer에서 화면설계를 위한 UI Designer, Data Selector, Logic Designer(Event/Transaction), Code Editor, SQL Editor 등 기능별 설계도구를 제공합니다. Drag & Drop으로 Canvas에 배치하고, Block Code로 Algorithm을 만들고, Attribute설정 기능으로 쉽고 빠르게 설계할 수 있습니다.

UI Design

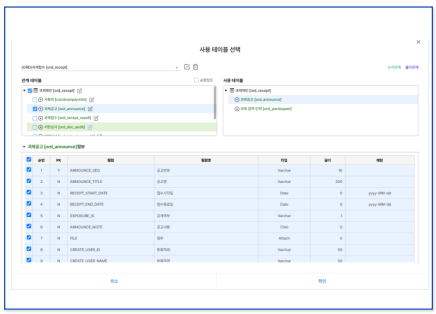
다양한 Layout, Form, Component를 Drag & Drop하여 설계가 가능하고 오브젝트별 다양한 설정(Size, Font, Color, Flex, Layer 등)이 가능합니다.



Data Select

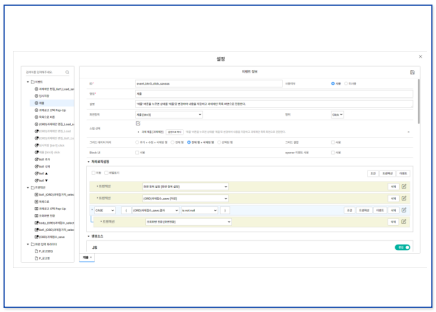
정의한 테이블 간의 관계정보(논리, 물리)를 이용하여 화면에서 사용할 테이블과 컬럼을 선택합니다.

SAP Legacy Table도 등록하여 사용합니다.



Event Design

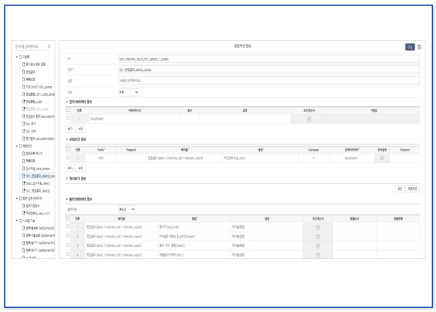
화면의 클릭, Data Loading, On-change 등의 Event를 정의하며, 설계 결과에 따라 자동 생성되는 Event와 사용자 정의 Event로 구분되며, 처리 로직은 Block Code 형식으로 설정이 가능하여 조건식과 Transaction의 입출력 정보 그리고 Customizing 내용을 정의합니다.



Transaction Design

화면 설계로 자동으로 생성되는 CRUD Transaction을 이용하거나 사용자가 별도로 정의하여 설계할 수 있습니다. Event 처리 로직에서 수행되는 Transaction 파라미터, 조회조건 및 처리 로직을 Block Code 형식으로 설정하여, 정의된 Transaction은 다른 화면에서도 Reference가 가능하여 재사용성과 유지보수가 용이합니다.

SAP Table을 사용하는 Transaction의 경우 SAP ABAP 코드가 생성되어 즉시 확인 가능하고, GEN 시 SAP에 등록, 활성화됩니다.



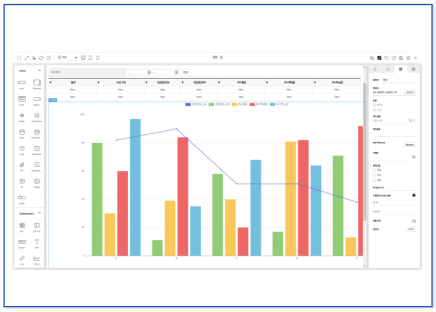
Screen Design(2/2)

★ 별첨

화면설계 시 다양한 Chart Design과 Blockly Library 기반 계산식 로직 설정, 그리고 상용 Library 및 JAR File 등 외부 API를 등록하여 설계 및 커스터마이징 시 사용이 가능하며 CSS 파일 적용 및 수정이 가능하여 전체 또는 화면별 UI 디자인 구현이 용이합니다.

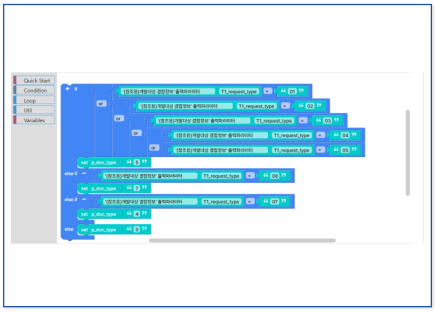
Chart Design

설계가 가능한 **Chart Component 제공** 및 데이터 정의로 다양한 Chart 구현이 가능하고,
상용 시각화 도구 사용 시 외부 API 연계 기능을 활용하여 손쉽게 적용이 가능합니다.



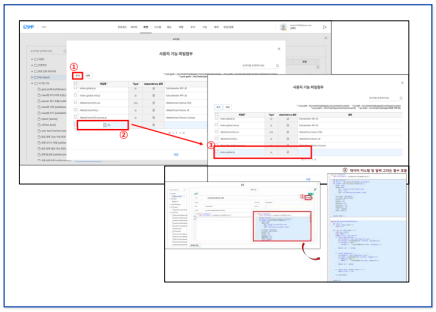
Blockly

Blockly는 인터렉티브 블록으로 코딩 개념을 반영한 Editor를 제공하여 개발자는 물론 일반 사용자도 계산식 로직 정의가 용이합니다.



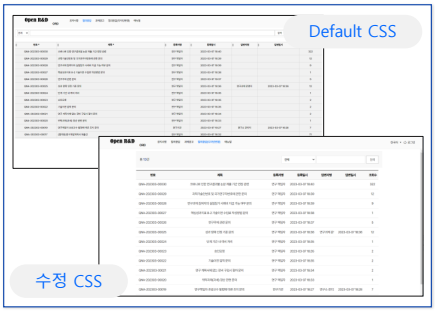
외부 API 연계

기 개발되었거나 **상용 Library, JAR File 등 등록하여 설계 및 커스터마이징** 시 사용이 가능하도록 외부 API 연계기능을 강화하였습니다.



CSS 적용

LAMP7은 Default CSS를 제공하여 UI 디자인을 적용하고 있으며, CSS 파일의 수정이 가능하므로 **전체 또는 화면별 UI 디자인 구현**을 가능하게 합니다.

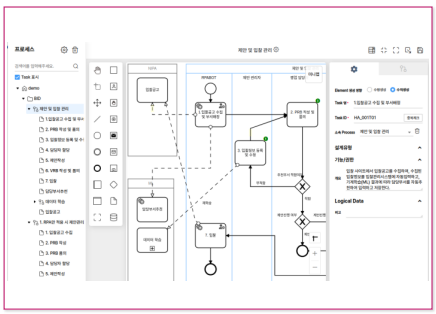


RPA/ML Design

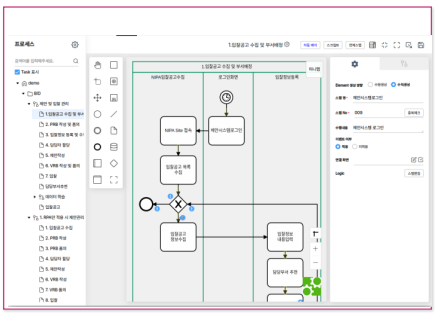
★ 별첨

개발 대상으로 정의된 On-Line Task 중에서 자동처리가 필요한 것은 RPA 적용 대상으로 식별하며 처리할 업무 Step을 정의하고, 인공지능을 업무에 활용하고자 할 경우에는 기계 학습할 항목과 라벨링을 정의하게 됩니다.

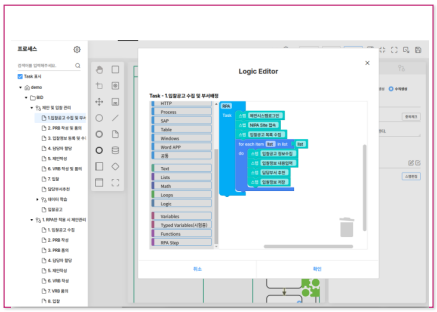
RPA 대상식별



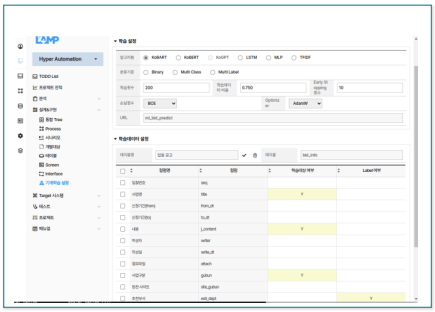
RPA Step정의



Step 로직 전환



기계학습 설계

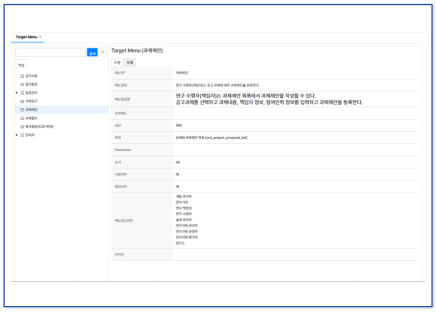


Generation & Deploy

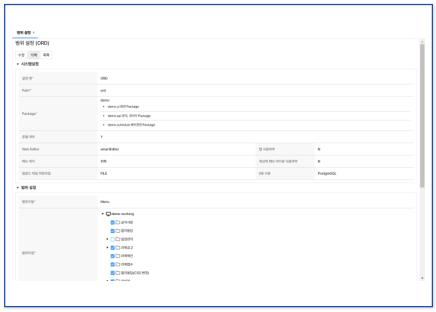
★ 별첨

메뉴 Designer로 메뉴체계를 설계하고 메뉴 단위/Task 단위 등 범위별로 프로그램 생성이 가능하며, 시스템 유형(WEB, SAP)을 선택하면 WEB은 전자정부 프레임워크 기반의 JAVA Source로, SAP는 SAP GUI기반의 ABAP Source 혹은 Web GUI가 아닌 Web화면과 SAP ABAP Source로 Generation되어 일반 Web환경과 같이 동작합니다.
목표시스템을 지정하면 Source 생성부터 Deploy까지 One-Stop으로 자동 수행하게 됩니다.

메뉴 설정

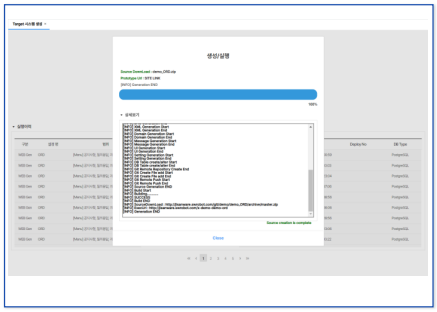


생성 범위 설정



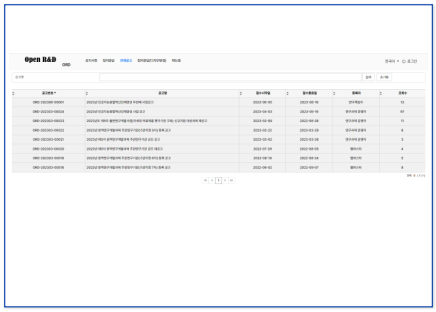
소스코드 자동생성

Web은 보안이 적용된 Spring framework,
전자정부 프레임워크의 JAVA 소스를,
SAP에서는 SAP GUI용 ABAP 소스나
일반 web화면에서 동작하는 HTML, Java, Java script, ABAP를 자동으로 Generation 합니다.



시스템 자동생성

소스 생성 시 생성 범위 설정에서 정의된 Target URL로 자동 Deploy 되며, swrobot.com 클라우드에 Prototype이 생성되는 서비스도 선택적으로 제공합니다.

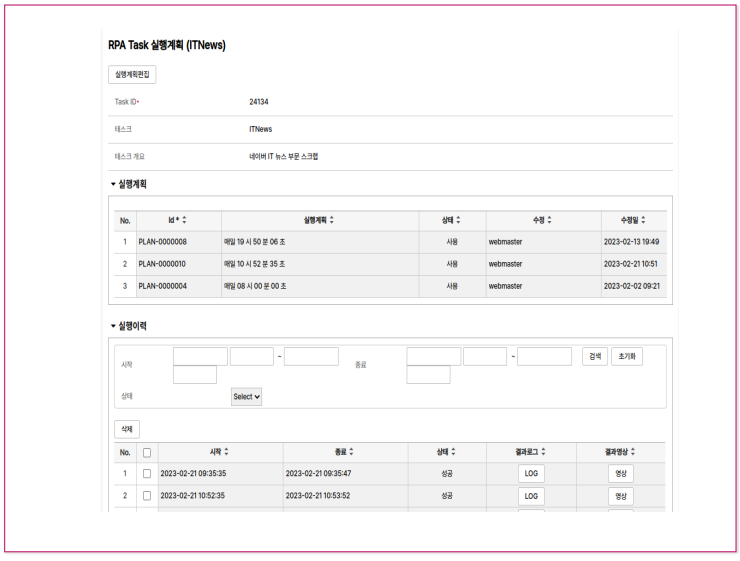


RPA/ML – 실행 및 모니터링

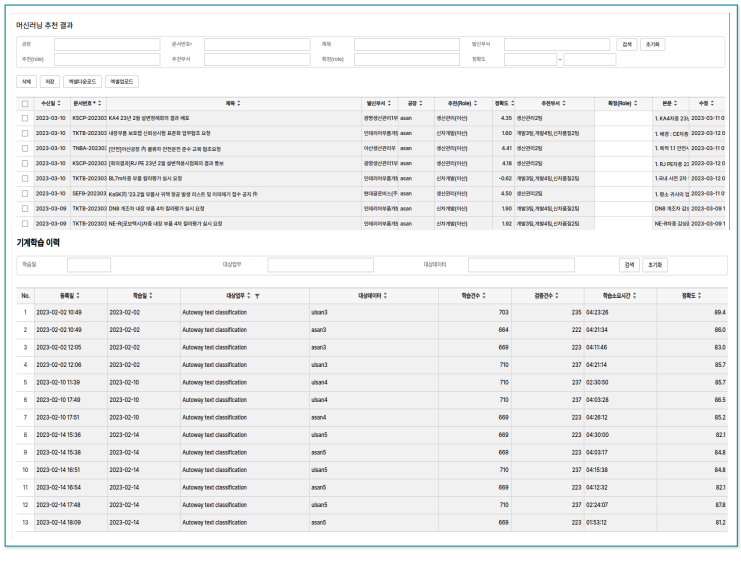
★ 별첨

LCDP 적용 업무와 함께 연결된 RPA Task는 생성과정을 거쳐 시스템으로 배포되고, 스케줄링에 의하여 실행되고 모니터링 됩니다.
ML의 학습결과는 이력으로 관리되며, 실시간 추천 결과도 정확도와 함께 이력으로 남습니다.

실행 및 모니터링



학습 및 추천, 이력

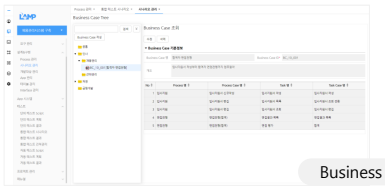


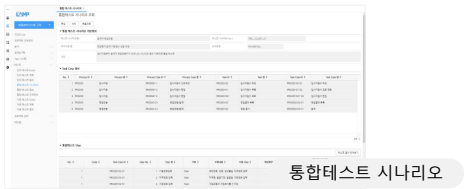
Test Automation(1/2)

★ 별첨

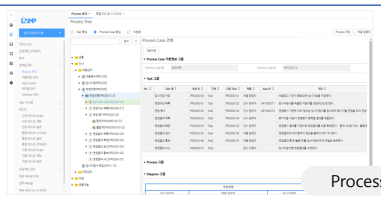
Process에서 정의한 Task Case(단위업무 유형)별 업무절차(Step)를 기반으로 테스트 스크립트를 자동 생성하고, 화면 항목에 대한 테스트 데이터를 입력하여 Robot 프레임워크 기반으로 테스트를 자동 실행합니다. Business Scenario는 통합 테스트, Process Case는 업무 테스트, Task Case는 단위 테스트로 실행됩니다.

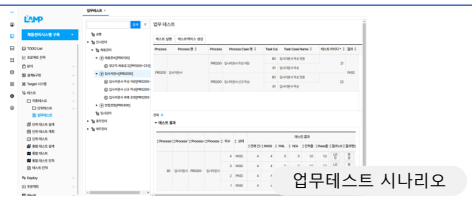
Business Scenario → 통합 테스트 시나리오



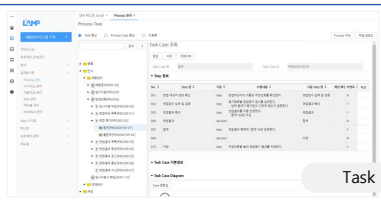


Process Case → 업무 테스트 시나리오





Task Case → 단위테스트 스크립트



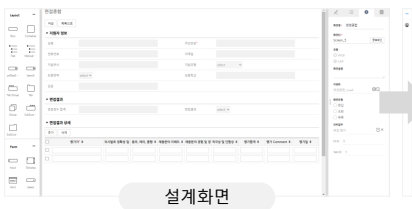
Test Automation(2/2)

★ 별첨

단위업무(Task) 설계화면의 입력항목을 기준으로 테스트 데이터를 생성(입력)할 수 있으며, 대용량 테스트 데이터도 업로드 기능 제공으로 쉽게 생성 가능합니다.
테스트 결과는 테스트리포트와 테스트동영상으로 저장되어 결과를 언제든지 확인할 수 있습니다.

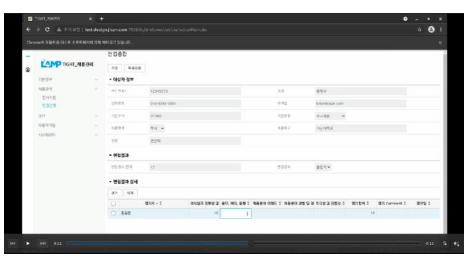
설계화면 → 생성된 화면 → 테스트 데이터 생성

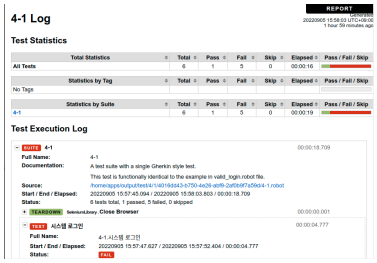
설계화면의 입력항목에 대한 입력값과 검증항목 및 검증값을 입력할 수 있으며 대용량 데이터로 처리도 가능합니다.



자동 테스트 실행 테스트 결과

테스트 스크립트와 테스트 데이터 기준으로 생성된 화면에서 Robot 프레임워크 기반으로 테스트를 자동 실행하고 테스트 결과가 저장되어 언제든지 확인할 수 있습니다.



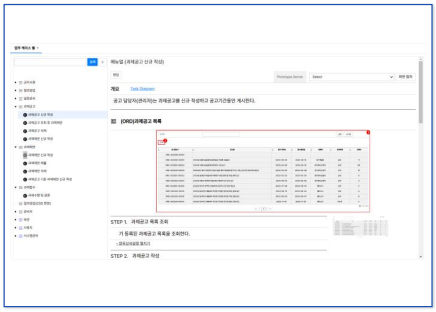


Manual Automation

★ 별첨

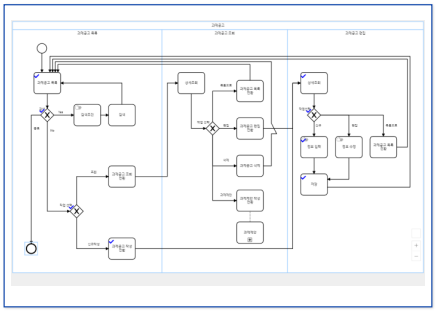
업무와 화면을 설계하면 화면설계 정보(UI, Event, 화면항목)와 업무설계 정보(Task, Step) 기준으로 프로그램 생성과 동시에 업무매뉴얼도 자동 생성되며, 설계를 변경하면 시스템과 매뉴얼에 실시간으로 반영됩니다. 별도의 매뉴얼 작성 작업이 불필요하며 업무 지침이나 시스템 사용 교육자료 등 활용성과 접근성이 높아집니다.

업무 매뉴얼

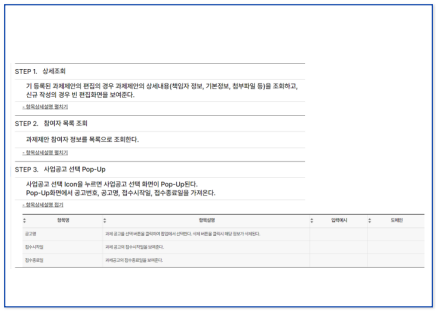


화면 내 Event 대상 구분

업무 흐름도



업무 절차

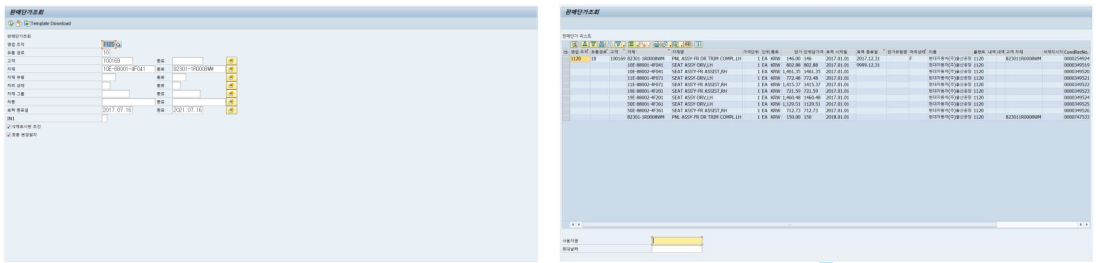


Reverse Engineering

★ 별첨

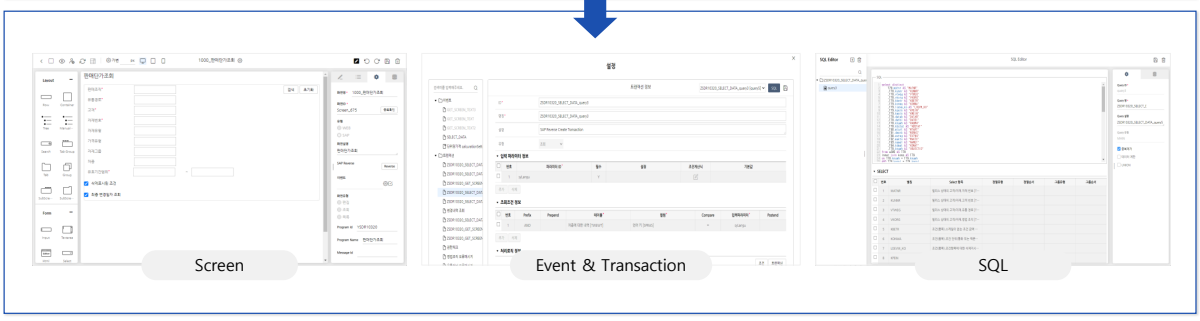
기업이 보유하고 있는 SAP Legacy 시스템을 개선된 업무 프로세스에 맞는 신규 시스템으로 전환하고 고도화하기 위하여 기존 시스템에서 개발된 프로그램을 Reverse Engineering을 통해 설계정보로 변환합니다. (UI, SQL 트랜잭션을 Reverse하고 Biz Logic은 Capsulation하여 가져옵니다.
기업의 차세대 시스템 개발을 위한 설계자료를 확보하고 개발 표준의 일원화 그리고 사용 중인 프로그램의 UI 통합 등 활용가치가 증대됩니다.

기간 시스템
(SAP Legacy System)



설계정보 생성

기간시스템(SAP)의 UI, Event/Transaction 항목, SQL Transaction을 가져와 LAMP7의 설계정보로 생성합니다.

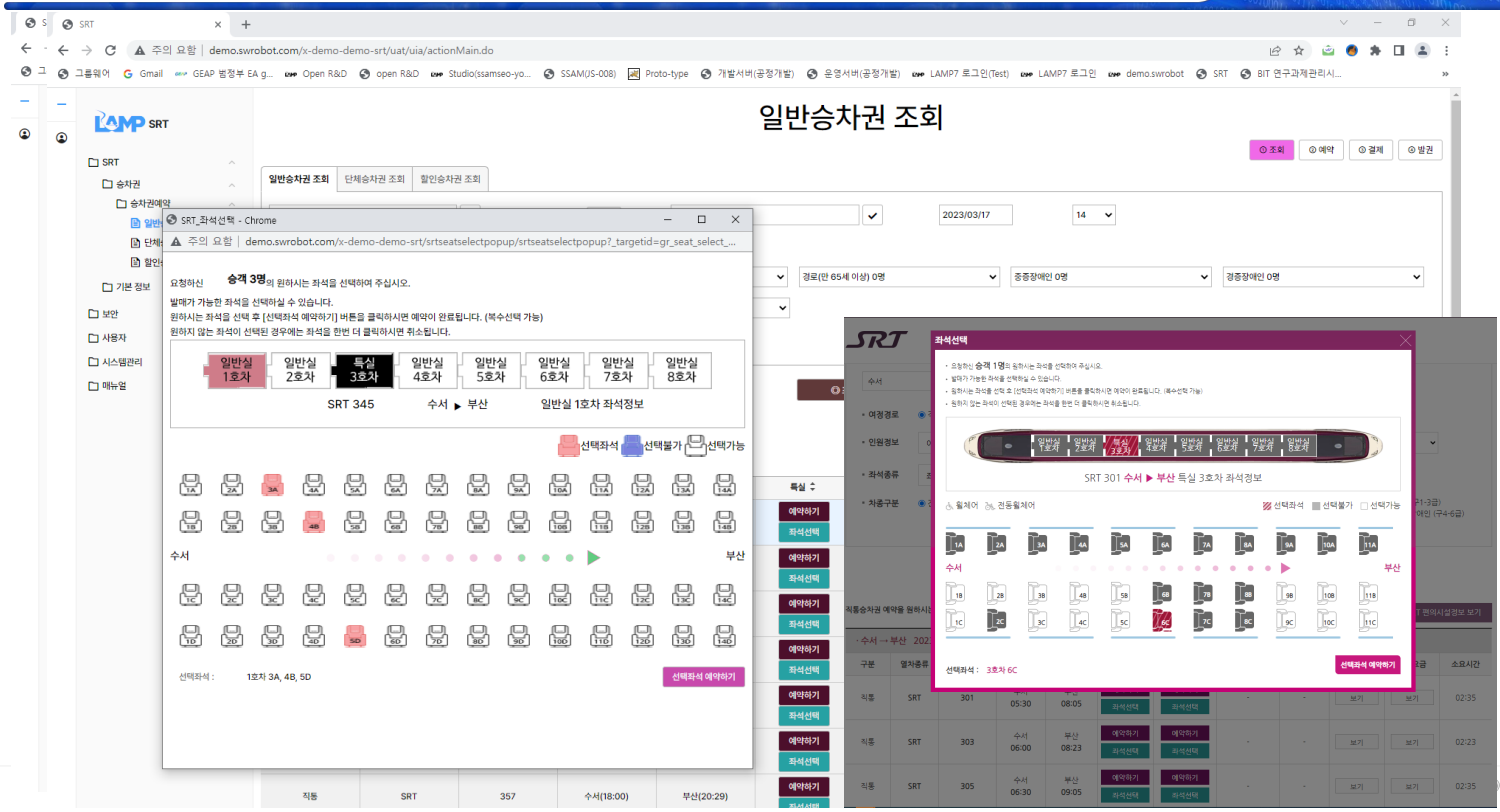


활용방안

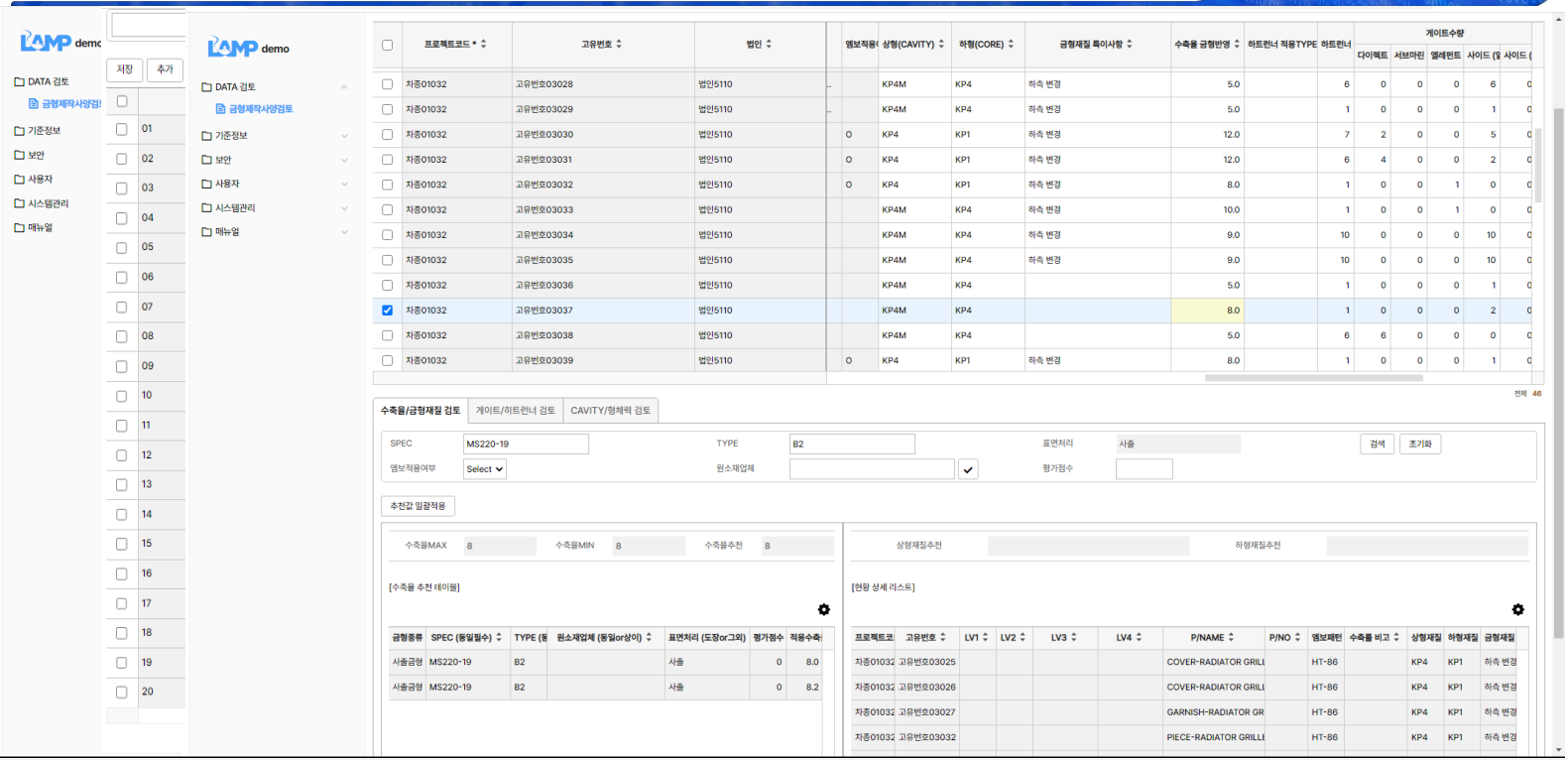
① SAP GUI 기반 로직을 추가 설계하거나
② SAP GUI를 WEB UI로 전환, 통합

시스템 생성 사례(국회도서관)

시스템 생성 사례(SRT)



시스템 생성 사례(금형시스템)



35