

“과학자는 이미 존재하는 것에 파고들지만, 공학자는 아직 보지 못한 것을 창조한다.” – Albert Einstein

지식과 상상력을 세상에 구현하는 AI 서비스 공학자, **장하림**입니다.

기본 정보 (Profile)

이름: 장하림 (Jang Haryeom)

성별/나이: 남성 / 1997 년생

이메일: gkfua00@khu.ac.kr

개인 홈페이지: hrletsgo.me

GitHub: github.com/CocoRoF

Hugging Face: huggingface.co/CocoRoF



학력 (Education)

- 석사학위: 경희대학교 빅데이터응용학과 공학전공 (2022.09 ~ 2024.08) GPA: 4.075 / 4.3
- 학사학위: 삼육대학교 상담심리학과 (2016.03 ~ 2020.02) GPA: 3.89 / 4.5

경력 (Career)

(주)플래티어 (Plateer Inc.) · AI R&D 팀 제품개발파트 · 파트리더

2024.10 ~ 현재

- LLM Training 연구
 - 도메인 LLM(POLAR, Gemma 계열) 학습 파이프라인 설계·운영 – SFT / DPO / GRPO 등 Post-Training 전 주기
 - POLAR Gemma-3 1B / 4B / 12B / 27B SFT + DPO + LoRA-KV 변형 라인업 구축
 - Korean ModernBERT base/large 패밀리 (v1~v22) MLM → SimCSE → GIST distill → 도메인 fine-tune 전체 파이프라인
- LLM Inference Optimization
 - vLLM 기반 멀티모델 서빙 환경 구성·운영 (OpenAI-compatible API)
 - 모델 계열·크기별 서빙 구성 및 throughput·메모리 트레이드오프 튜닝 / 실시간 추론(OmniVoice)에서 추론 스텝(NFE)·캐시·동시성 최적화
- AI Agent Platform 개발 담당
 - Xgen – Plateer 사내·고객사 에이전트 플랫폼 설계 책임
 - 5 종 LLM Provider (Anthropic / OpenAI / Google / vLLM / Claude Code CLI) 통합, MCP-first 아키텍처
 - Node-Edge 기반 Canvas 인터페이스로 비개발 사용자 워크플로 설계 지원

전문 영역 (Areas of Expertise)

AI 시대의 엔지니어링은 “쓸 줄 아는 도구의 목록” 으로 설명하기 어렵습니다. 도구는 Agent 가 자동으로 쓰는 시대가 됐고, 차별점은 어떤 문제에 깊이 들어가 봤는가 입니다. 아래는 제가 실제로 운영해 본 4 가지 영역과 그 깊이입니다.

1. LLM Training Operations · 학습 전 사이클 운영

Plateer 도메인 LLM(POLAR, Gemma 계열) 라인업과 Korean ModernBERT 임베딩 패밀리를 직접 운영했습니다. 단순 SFT 가 아니라 모델 계열·크기·도메인별로 SFT / DPO / LoRA / QLoRA / Unsloth 4-bit 를 조합하며, 학습이 ‘왜 되고 왜 안 되는지’ 를 손으로 확인해 왔습니다.

주요 도구 · PyTorch · TRL · Unsloth · LoRA / QLoRA · GGUF · Hugging Face Hub · MTEB

2. Inference / Serving Engineering · 서빙과 실시간 추론 최적화

vLLM 위에서 여러 모델을 동시에 서빙하는 환경을 운영하면서, 추론 비용은 모델을 더 키우기 전에 짜낼 여지가 많다는 걸 체감했습니다. LLM 의 다양한 혼합 정밀도로 vLLM 을 여러 인프라 환경에서 처리하는 것을 경험했으며, 개인 사이드프로젝트에서는 TTS(omni-voice 기반) Inference 를 최적화 하는 등 다양한 시도를 한 경험이 있습니다

주요 도구 · vLLM · OpenAI-compatible API · OmniVoice · HuBERT · asyncio · CUDA · GGUF

3. Agent Architecture · Harness / Skills / Memory

Xgen 플랫폼의 Agent Harness 를 설계하며 ‘LLM 호출 한 번 = 1 함수’ 라는 단순 모델이 production 에서 왜 무너지는지를 몸으로 겪었습니다.

신뢰성은 모델 자체가 아니라 단계 분리와 결정론적 가드에서 나옵니다. 그래서 Xgen 의 Harness 는 Prompt → Memory → Tools → ContentGuard → Execution → Judge → Finalize 처럼 단계를 쪼개고, ContentGuard 는 결정론적 패턴 가드, Judge 는 확률적 사후 검증으로 역할을 나눴습니다. 도구가 늘어나는 문제는 Skills SDK 의 단일 dispatch 패턴으로 토큰을 크게 줄였습니다.

주요 도구 · Anthropic / OpenAI / Google / vLLM provider · MCP · Skills SDK · FastAPI · WebSocket / SSE streaming

4. Document → Context Pipeline · RAG 의 진짜 difficulty

RAG 의 quality 는 embedding 보다 chunk 가 의미를 보존하는가에서 결정된다고 봅니다. 80+ 포맷을 다루는 사내 파서 xgen-doc2chunk (=오픈소스 Contextifier) 를 설계하며 PDF 의 표 / Word 의 코드 블록 / HWP / 이미지 OCR 까지 구조 보존형 chunking 을 구현했습니다. 포맷별 handler 12 종 + Vision-LLM OCR 5 backend (OpenAI / Anthropic / Google / AWS / vLLM) 를 모두 끼울 수 있는 plug-in 아키텍처가 핵심입니다.

주요 도구 · Contextifier · xgen-doc2chunk · 5 Vision-LLM OCR · pgvector · BM25 · RAG

사내 개발 / 제품 (Developments)

Xgen — Plateer 의 강력한 AI Agent 설계 플랫폼

Xgen 은 비개발자도 Node-Edge Canvas 위에서 에이전트 워크플로를 직접 설계할 수 있는 Plateer 의 플래그십 AI Agent 플랫폼입니다. 4 종 LLM Provider (Anthropic / OpenAI / Google / vLLM) 와 직접 개발한 Xgen-Harness 를 바탕으로, 사내·고객사 (롯데홈쇼핑 · 제주은행 등) 환경에 동일한 코어로 배포됩니다.

아래 네 가지는 사내에서 실제로 부딪힌 난제들이고, 모두 2024–2026 LLM Agent 연구의 active frontier 이기도 합니다. 학계·산업이 같은 문제를 어떻게 보고 있는지와 함께, 그 위에서 제가 시도한 방향을 적었습니다.

1. 문서 처리 (Document Parsing for RAG)

Agent 가 다루는 외부 자료는 PDF·Word·Excel·PPT·HWP·이미지·코드까지 형태가 극단적으로 다양한데, 단일 파서로는 표가 깨지고 코드 블록이 토막 나고 이미지 속 의미가 사라집니다. RAG 품질이 임베딩 모델보다 ‘칭크가 의미를 얼마나 보존하느냐’ 에서 갈린다는 건 Late Chunking(Jina, 2024)·Contextual Retrieval(Anthropic, 2024)이 거의 동시에 짚은 지점이기도 합니다. 그래서 사내 파서 xgen-doc2chunk(이후 오픈소스 Contextifier 로 일반화)를 직접 설계했습니다 — 포맷 감지 → 포맷별 handler → 정규화 → 구조 보존 chunking → 메타데이터 추출의 단일 파이프라인에, 포맷별 handler 12 종과 Vision-LLM OCR 5 개 백엔드를 plug-in 으로 끼우고, 표·코드는 분할 금지 영역으로 보호하며 page / line offset 까지 칭크 메타데이터에 박아 LLM 이 ‘P.3, line 12’ 이 인용할 수 있는 구조를 설계했습니다

2. Agent Memory — 짧은 기억과 긴 기억의 분리, 그리고 hallucination 누적

턴이 길어질수록 context window 가 폭발하고, 단순 truncation 은 사용자의 초기 의도를 잃습니다. RAG 만으로는 시간·순서 관계가 사라지고, append-only 메모리는 틀린 추론까지 그대로 누적합니다. 이 문제는 MemoryBank(2023)의 시간 가중 retrieval, Reflexion(NeurIPS 2023)의 self-reflection 처럼 학계도 여러 각도로 풀고 있는 주제입니다. 제가 택한 건 Hybrid Memory — 런타임 대화 상태(state)와 PostgreSQL 장기 기억(db_chat_memory)을 이중 레이어로 두고, MemoryScorer 가 각 메시지에 recency-quality 등의 가중치를 매겨 무엇을 남기고 무엇을 잊을지 결정하게 했습니다.

3. Agent Harness — 같은 프롬프트인데 매번 결과가 다른 비결정성문제

Production 에서 가장 까다로운 건 ‘같은 프롬프트·같은 모델인데 턴마다 결과가 흔들리는’ 비결정성입니다. 도구 호출 순서·종료 조건·검증 통과가 매번 다르고, 파일 삭제나 외부 호출 같은 destructive 도구는 한 번 잘못 불리면 복구 비용이 큼니다. NeMo Guardrails(NVIDIA, 2023)나 Anthropic 의 Building Effective Agents(2024)가 공통적으로 말하는 결론은 ‘heavy framework 보다 명시적 단계 분리와 결정론적 가드가 안정적’ 이라는 것이고, Xgen 의 Harness 도 같은 방향입니다 — Prompt → Memory → SystemRole → Tools → ContentGuard → Context/RAG → Execution → Judge → Finalize 로 단계를 쪼개고, ContentGuard 는 결정론적 패턴 가드, Judge 는 확률적 사후

검증으로 역할을 분리했습니다. 신뢰성은 모델을 믿는 게 아니라 단계 사이에 가드를 끼우는 데서 나온다고 봅니다.

4. Tool 폭증 처리 — 도구가 늘 때 context 가 먼저 무너진다 (tool haystack)

use case 가 늘면 도구가 수십~수백 개로 불어나는데, 이걸 한꺼번에 context 에 넣으면 토큰이 폭발(대략 도구 37 개 × 700 자 ≈ 25k 토큰/턴)하고 선택 정확도가 haystack 문제로 떨어집니다. ToolLLM(2023)이 16,000+ API 로 이 스케일 문제를 벤치마크화했고, RAG-MCP(2025)는 MCP tool index 에 semantic retrieval 을 더해 프롬프트 토큰을 줄였습니다. 저는 3 단계 계층화로 접근했습니다 — Level 1 은 개별 tool, Level 2 는 Skills SDK 로 N 개 tool 을 단일 dispatch tool 로 묶어(action 파라미터로 라우팅) 노출 표면을 줄이고, Level 3 은 도구를 의미적으로 그래프화해 현재 task 와 가까운 것만 context 에 띄우고 나머지는 manifest 한 줄로만 두는 방식입니다.

Polar Trainer — Plateer 사내 LLM 학습기

- POLAR 도메인 LLM (POLAR Gemma-3 1B~27B, Qwen3 등) 라인업 학습 운영
- SFT / DPO / LoRA / QLoRA / Unsloth 4-bit / GGUF 양자화 통합 지원
- Korean ModernBERT 패밀리 학습 — MLM 사전학습 → SimCSE → GIST distill → e-commerce 도메인 fine-tune 일괄 운영

사업 수행 (Project)

[롯데홈쇼핑] Xgen 플랫폼 구축 사업

- Xgen 에이전트 플랫폼을 롯데홈쇼핑 환경에 도입 — 비개발 사용자가 Node-Edge Canvas 로 에이전트 워크플로를 직접 설계
- 5 종 LLM Provider 통합 + MCP-first 아키텍처로 사내 RAG / 검색 / 문서 처리 도구와 연동

[롯데홈쇼핑] 사내 에이전트 구축 사업

- Xgen 플랫폼 위에 롯데홈쇼핑 도메인 특화 에이전트 다수 구축 — 사내 업무 자동화 / 정보 검색 / 문서 작성 보조
- QA 워크플로우 설계 (상품 기술서 기반 진위 검증, 내부 문서 대조 후 검토 등 작업)
- 각 use case 별 에이전트 설계·배포 및 운영 모니터링

[제주은행] GenAI 플랫폼 구축 사업

- 제주은행 환경에 Plateer GenAI 스택 (POLAR LLM + RAG + Agent) 통합 구축
- 금융 도메인 특수성 (보안 / 컴플라이언스 / 데이터 격리) 고려한 on-prem 설계

[IM 캐피탈 (대구은행)] AI 플랫폼 구축 사업

- Plateer AI Agent 스택을 IM 캐피탈(대구은행) 환경에 도입 — 금융 도메인 요건에 맞춘 AI 플랫폼 구축

사이드 프로젝트 — Geny · 미래 Agent 시대를 위한 연구장

Geny 는 단순한 사이드 프로젝트가 아니라, 제가 생각하는 Agent 시대의 필수 요소들을 종합적으로 실험·검증하는 연구장 입니다. 사내 production(Xgen)이 “지금 동작해야 하는 것” 을 다룬다면, Geny 는 “1~2 년 뒤에 필요할 것” 을 먼저 깨쳐 보는 공간입니다. 멀티 에이전트 vTuber + 2D Live Avatar(Live2D / Spine) + 실시간 TTS 라는 외형은 이 실험들을 눈으로 볼 수 있게 만들어 준 인터페이스입니다.

아래 다섯 주제는 현재 학계·산업이 동시에 풀고 있는 frontier 입니다. 각 주제에서 어떤 접근들이 오가는지와, 그 위에서 Geny 로 세운 가설과 시도를 함께 적었습니다.

1. Long-horizon Memory - Graph + Embedding 결합

vector RAG 는 시간 순서를 잃습니다 — 어제 한 말과 오늘 한 말의 관계, 결정의 인과 사슬이 평탄화됩니다. 에이전트가 ‘하루를 살게’ 하려면 [언제, 누구와, 왜]라는 관계 그래프가 필요합니다. Generative Agents(UIST 2023)의 memory stream, Zep 의 temporal knowledge graph 가 이 방향의 대표적 시도입니다. Geny 에서는 사건·실체·사용자 의도를 노드로, 시간·인과·언급을 엮지로 저장하는 Graph + Embedding hybrid memory 를 실험했습니다. query embedding 으로 후보 노드를 뽑고 graph traversal 로 인과, 시간 맥락을 따라가는 방식입니다.

2. Multi-tool / Multi-MCP Optimization - Context 의 진짜 비용

MCP server 가 늘수록 tool catalog 가 매 턴 비대해지고 토큰 비용과 응답 속도가 같이 무너집니다. '필요한 tool 만 동적으로 로드' 는 어느 tool 을 모르는 문제가 존재합니다. MCP(Anthropic, 2024)가 agent ↔ tool 의 $N \times M$ 문제를 표준화했고 RAG-MCP 는 거기에 retrieval 을 더했습니다. Geny 에서는 tool 들을 의미적 거리로 그래프화해 현재 task 근처 cluster 만 노출하고, 먼 tool 은 manifest 한 줄로만 두었다가 필요할 때 LLM 이 직접 펼치게 하는 graph-search 기반 context 압축을 시도했습니다.

3. Embodiment - 에이전트가 사는 공간이 정체성에 미치는 영향

같은 모델, 같은 프롬프트라도 에이전트가 ‘어디서 어떻게 사는지’ 가 응답의 결을 바꿀 수 있다는 가설을 검증해보고 싶었습니다. Voyager(NVIDIA, 2023)나 Generative Agents 가 보여준 embodied agent 의 연장선입니다. 그래서 에이전트에게 거주 공간과 몸을 주는 실험을 했습니다. Pixi.js v8 + Live2D / Cubism 기반 2D Live Avatar 와 생성형 텍스처 파이프라인으로 ‘말하고 표정 짓는’ 아바타를 만들고, 위치·이웃·시간대 같은 context 를 system prompt 에 주입해 동일 task 의 응답이 어떻게 달라지는지 비교하는 구조를 두었습니다.

4. Realtime Streaming + Tool Use 의 동시성 문제

LLM 이 응답을 생성하면서 동시에 도구 호출·결과 수신·WebSocket 푸시·TTS 변환을 처리하는 streaming 환경에서는 race condition·partial chunk·중복 호출이 가장 흔한 버그입니다. OpenAI Realtime API(2024)가 persistent session + async function calling 으로 푼 문제와 같은 걸입니다. Geny 에서는 21-stage 파이프라인의 각 stage 를 idempotent 하게 설계하고 chunk-level checkpoint 를 뒤서 같은 input 이 두 번 들어와도 side effect 가 한

번만 일어나게 했고(외부 호출에 deterministic key 부여), 실시간 TTS(OmniVoice)에서는 문장 단위로 텍스트를 쪼개 합성·재생을 겹치고 asyncio Semaphore 와 GPU Lock 을 분리해 동시성과 단일 GPU 직렬성을 동시에 잡았습니다.

오픈소스 프로젝트 (Open Source)

저는 AI Agent Engineer 로, 사내 프로젝트를 진행하는 것도 중요하지만 관련 학계, 커뮤니티와 소통하며 주요한 기술들에 대해 지속적인 관심을 기울이는 것이 중요하다 생각합니다. 아래의 프로젝트들은 제가 개인적으로 진행하는 오픈소스 프로젝트이며, 여기서의 기술은 곧 제가 회사에서도 더 잘 기능할 수 있는 발판이 되었습니다.

geny-executor v2.1.3 (Python)

github.com/CocoRoF/geny-executor · PyPI

LangChain-LangGraph 의 추상화는 편하지만 production 에서 미들웨어가 어디서 끼어들었는지.왜 그 도구가 불렸는지 역추적이 막히는 순간이 옵니다. 그래서 raw provider API 위에 단계가 전부 보이는 21-stage agent 파이프라인을 직접 구현했습니다. Anthropic 의 Building Effective Agents(2024)가 말한 'heavy framework 보다 명시적 워크플로가 안정적' 이라는 결론과 같은 방향이고, 5 종 provider 를 하나의 메시지·도구 spec 으로 정규화하는 dual-abstraction 이 핵심입니다.

Contextifier v0.2.5 (Python)

github.com/CocoRoF/Contextifier · PyPI

RAG 품질은 임베딩 모델이 아니라 청크가 의미를 보존하느냐에서 갈립니다. PDF 표가 깨지거나 코드 블록이 토막 나면 그 뒤 LLM 호출은 손상된 정보를 받게 됩니다. Late Chunking(Jina, 2024)·Contextual Retrieval(Anthropic, 2024)이 같은 결론을 쫓았습니다. Contextifier 는 모든 포맷을 단일 5-stage 파이프라인으로 처리하고, 포맷별 handler 와 Vision-LLM OCR 5 개 백엔드를 plug-in 으로 끼우며, 표·코드를 분할 금지 영역으로 보호해 의미 단위를 지킵니다.

graphier v1.3.3 (TypeScript)

github.com/CocoRoF/graphier · npm

Geny 의 Agent 관계 그래프를 브라우저에서 그리려는데, 기존 3D 그래프 라이브러리는 노드가 1k 만 넘어도 프레임이 무너졌습니다. 노드마다 draw call 을 하나씩 부르는 구조가 가장 큰 원인입니다. Three.js + d3-force-3d 위에 instancing 으로 draw call 을 묶어 대규모 노드도 부드럽게 렌더링하도록 만들어 라이브러리로 만들어, Graph 구조가 필요한 모든 부분에 적용하고 있습니다

f2a v1.1.0 (Python)

github.com/CocoRoF/f2a · PyPI

데이터를 받으면 가장 먼저 하는 일이 '어떤 데이터인지 보는' 것인데, 그 단계가 매번 코드 기반의 작업이라 반복 비용이 컸고 비개발자에게 공유할 수단도 없었습니다. f2a 는 데이터 형태의 포맷을 받아 통계 분석부터 단일 HTML 리포트 생성까지 한 줄로 끝내주는 라이브러리입니다.

an-web (Python)

github.com/CocoRoF/an-web

Agent 에게 사람용 브라우저(Playwright / Selenium)는 너무 무겁습니다. Node 의존, 큰 메모리, headless 안정성 이슈가 따릅니다. 게다가 에이전트는 픽셀이 아니라 '구조'가 필요합니다. 그래서 v8 엔진을 임베드한 Python-native 브라우저를 만들어, 페이지를 사람이 보는 화면이 아니라 Agent 가 추론·조작할 수 있는 의미 그래프(PageSemantics)로 환원했습니다. React-webpack SPA 를 실행하면서도 메모리는 ~30MB 수준입니다.

edit2ppt (Python)

github.com/CocoRoF/edit2ppt

문서 파싱 경험을 거꾸로 뒤집어, 이번엔 Agent 가 결과물(PPT)을 '생성' 하게 만든 도구입니다. SVG → OOXML 변환 엔진 위에 LLM 오케스트레이션(구조화 → 이미지 → 템플릿 → PPTX)을 얹고, 로컬 파일 접근 없이도 외부 Agent 가 MCP 로 접속해 편집할 수 있게 설계했습니다. PPTX 구조의 placeholder 를 AI 가 이해하기 쉬운 형태로 추상화한 것이 핵심입니다.

github-ai-network (Python · React)

github.com/CocoRoF/github-ai-network

파편화된 GitHub 데이터를 수집·가공해 Repository·Author·Topic 을 Entity 로, 관계를 Edge 로 저장하는 Graph Database 로 만든 프로젝트입니다. 단순 임베딩 검색을 넘어 Entity 기반 3-hop 연결까지 탐색하는 Graph Search Engine 을 설계했고, 3D 그래프 렌더링에는 직접 만든 라이브러리(graphier)를 썼습니다. '컨텍스트가 늘어나는 Agent 에게 정보를 어떻게 줄까' 라는 고민의 답을 실험한 일종의 테스트용 라이브러리입니다

geny-adapted-project-toolkit (GAPT) (Python · TypeScript)

github.com/CocoRoF/geny-adapted-project-toolkit

AI 프로젝트 서빙에서 가장 까다로운 부분 중 하나가 '인프라를 맞추는 것' 입니다. GAPT 는 외부 Git 레포를 컨테이너로 격리하고, 웹 에디터에서 편집·테스트·빌드·배포까지 한 콘솔에서 끝내는 self-hosted AI DevOps 플랫폼입니다(Cursor × Coolify × OpenHands 를 목표로). Geny 의 Harness 엔진을 그대로 얹어 Agent 와 협업하며 개발·배포하는 흐름을 만들었고, Cloudflare 터널 기반 실시간 미리보기·배포를 붙였습니다.

저서 / 기술 감수 (Authorship)

『랭체인으로 구현하는 AI 서비스 & 에이전트 개발 입문』

- 출판사: 영진닷컴, 2026. ISBN: 978-89-314-8285-0
- 역할: 기술 감수자 — LangChain · LangGraph 핵심 챕터 재작성

논문 및 연구 (Publications)

Journal — 게재 / 투고

- 건강추천시스템 (HRS) 연구 동향: 인용네트워크 분석과 GraphSAGE 를 활용하여 (장하림, 유지수, 양성병, 2023, 지능정보연구) [게재됨]
- 재생에너지 발전량 예측제도 기반 집합전력자원 구성모델 개발 (강은경, 장하림, 양선욱, 양성병, 2023, 지능정보연구) [게재됨]
- Exploring consumer behavior in the travel experience sector through real-time live commerce chat: a text mining and machine learning approach (Kang, Jang, Yang, 2024, Journal of Hospitality Marketing & Management) [투고]
- Leveraging stock discussion forum posts for stock price predictions: A focus on the secondary battery sector (Yoo, Jang, Kang, Yang, 2024, IEEE Access) [투고]

Journal — 진행중

- HormAS, Hotel Review Management Assistant Service — Review Analysis and Response Using Generative AI: Application of Design Science Methodology (장하림, 윤상혁, 양성병, 2024)
- FedKDA: The Decentralized Federated Learning Methods Using Knowledge Distillation and Aggregation in Graph (장하림, 양성병, 2024)

Conference

- 라이브커머스 실시간 채팅을 활용한 여행/체험분야 소비자 행태 분석: 텍스트 마이닝 및 머신러닝을 적용한 탐색적 연구 (강은경, 장하림, 양성병, 2023.06, 경영정보관련 춘계통합학술대회)
- 재생에너지 발전량 예측제도 기반 집합전력자원 구성모델 개발 (강은경, 장하림, 양선욱, 양성병, 2023.10, 한국지능정보시스템학회)
- 전국 기상 관측 데이터를 활용한 전력수요 예측 모델 구현 (김진수, 장하림, 강민석, 양성병, 2023.10, 한국지능정보시스템학회)
- 머신러닝 및 딥러닝을 활용한 식품 라이브커머스 방송 매출의 주요 영향 요인 분석 (조윤주, 장하림, 양성병, 2023.11, 한국인터넷전자상거래학회)
- 라이브커머스 카테고리별 소비자 행태 분석: 실시간 채팅을 활용한 탐색적 연구 (강은경, 김진수, 장하림, 양성병, 2024.05, 한국지능정보시스템학회)
- 생성형 인공지능을 활용한 호텔리뷰 관리지원서비스 개발: 디자인 사이언스 방법론 적용 (장하림, 윤상혁, 양성병, 2024.06, 한국경영정보학회)
- DLinear 모델을 적용한 비용 효율적 전력 수요 예측: 기상 데이터 활용 (김진수, 장하림, 강민석, 양성병, 2024.08, 한국경영학회)

공모 / 외부 프로젝트 (Competitions & External Projects)

- 제 10 회 산업통상자원부 공공데이터 활용 BI 공모전 빅데이터 분석부문 (2022.09)
- 2022-2 KHU-Fellowship (2022.11) — 건강추천시스템에 관련한 리뷰 연구, 네트워크 방법론을 활용하여
- (주)영림원 대학(원)생 ERP 아이디어 공모전 (2023.05) — 미달성 매출목표 이월화 자동시스템 연구

- 제 11 회 산업통상자원부 공공데이터 활용 BI 공모전 빅데이터 분석부문 과제 4-5 (2023.06) — 전력수요 예측 / 재생에너지 집합전력자원
- 2023-1 KHU-Fellowship (2023.07) — 라이브커머스 플랫폼 차이에 따른 판매사업자 전략 연구
- 삼성휴먼테크논문대상 (2023.09) — Factor Analysis 와 Attention 을 응용한 Dimensionality Reduction 연구
- 삼성엔지니어링 리스크 관리 프로젝트 (2023.05 ~ 연기됨)

수상 (Honors & Awards)

학술 / 학회

- 한국지능정보시스템학회 2024 춘계학술대회 최우수논문상 (1 위, 2024.05)
- 한국경영정보학회 2024 춘계통합학술대회 최우수논문상 (1 위, 2024.06)
- 2024 한국경영학회 융합학술대회 우수논문상 (2 위, 2024.08)
- 2023 경희대 Big Data Graduate Student Conference (BDGSC) Best Paper Runner-up (2 위, 2023.12)
- 2023 경영정보관련 춘계통합학술대회 우수논문상 (2 위, 2023.06)

공모전 / 정부 시상

- 산업통상자원부 장관상 — 제 12 회 공공데이터활용 BI 공모전 빅데이터 분석부문 대상 (1 위, 2024.08)
- 산업통상자원부 장관상 — 제 11 회 공공데이터활용 BI 공모전 빅데이터 분석부문 대상 (1 위, 2023.08)
- (주)영림원 대학(원)생 ERP 아이디어 공모전 대상 (1 위, 2023.06)

Reference

- **양성병 교수님** — 연구실 지도교수
경희대학교 빅데이터응용학과 & 경영학과
E-mail: sbyang@khu.ac.kr
- **김태경 교수님** — 연구실 참여교수
경희대학교 빅데이터응용학과
E-mail: tk_kim@khu.ac.kr
- **윤상혁 교수님** — 연구실 참여교수
한국기술교육대학교 산업경영학과
E-mail: yoonsh@koreatech.ac.kr