

1 서비스 개요 및 목적

현재 스마트워치 시장은 밀레니엄 세대와 X세대를 중심으로 지속적인 성장을 보이며, AI 및 데이터 분석 기술의 발전과 함께 더욱 정교한 건강 모니터링 기능을 제공하고 있습니다. 스마트워치는 심박수 추적, 혈중 산소 수치 측정, ECG(심전도) 기능, 혈압 모니터링 등 다양한 생체신호 분석 기능을 갖추고 있으며, 최근에는 AI를 활용한 건강 이상 예측 기능까지 도입되고 있습니다. 스마트워치의 실용적 활용 사례로는 운동 중 심박수 모니터링 및 건강 정보 제공, 넘어짐 감지 및 알림 기능, 수분 섭취 알림 및 휴식 시간 제안 등이 있으며, 이는 단순한 건강 관리 도구를 넘어 응급 상황 대처에도 활용될 수 있습니다.

스마트워치 시장은 2023-2028년 동안 14.94%의 연평균 성장률(CAGR)을 기록할 것으로 예상되며, 이는 건강 관리 기능에 대한 수요 증가를 반영합니다. 특히, 고령화 사회로 인해 만성질환 관리 및 원격 의료 모니터링의 중요성이 커지고 있으며, 웨어러블 기기의 실시간 건강 모니터링 기능은 노약자 및 만성질환자들에게 필수적인 도구로 자리 잡고 있습니다.

저희 서비스는 스마트워치의 생체신호 정보를 실시간으로 수집하여 AI 기반 응급상황 감지 및 대응 기능을 제공합니다. 심박수 급변, 산소포화도 저하 등의 이상 징후를 감지하면, 즉각적으로 사용자의 상태를 분석하고 적절한 응급 대처법을 안내합니다. 또한, 사용자가 설정한 비상 연락망 또는 응급 구조 기관에 자동으로 구조 요청을 전송하는 기능을 포함하고 있어, 혼자 있는 상황에서도 신속한 대응이 가능합니다.

이러한 기능은 웨어러블 의료기기 시장의 성장 트렌드와 일치하며, 특히 고령자 및 외딴 지역 거주자의 원격 모니터링에 유용하게 활용될 수 있습니다. 더불어, 본 서비스는 의료 데이터 보호 규정을 준수하여 GDPR, HIPAA 등의 국제 표준을 반영한 보안 체계를 갖추고 있으며, 사용자의 개인 건강 데이터를 안전하게 관리하는 것을 최우선으로 합니다.

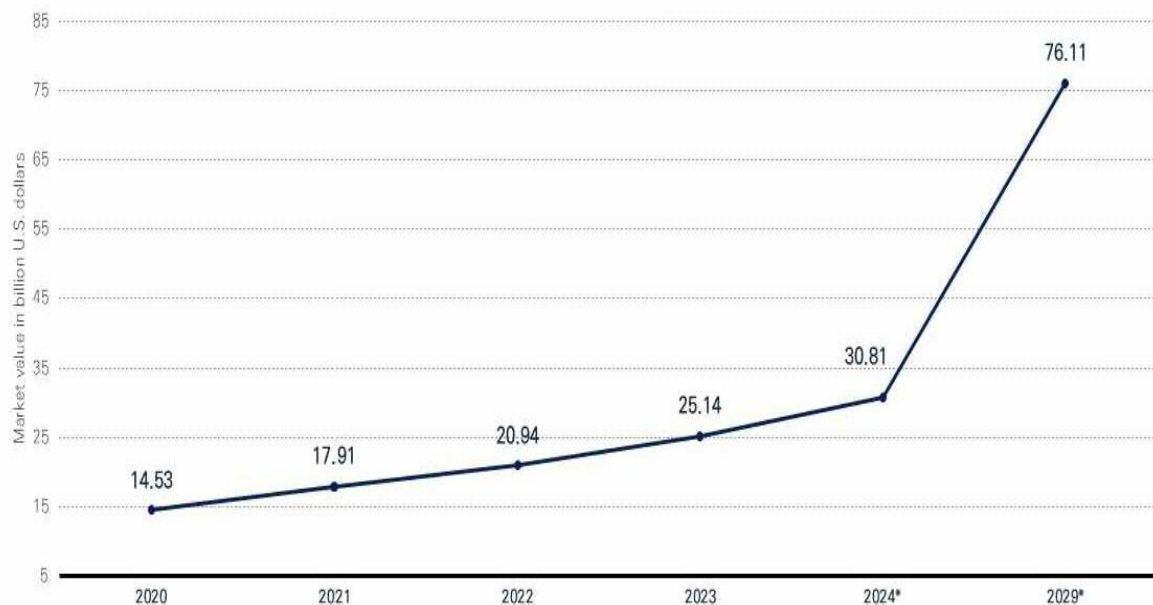


그림 1. 글로벌 웨어러블 의료기기 시장 동향 분석 자료¹

¹ 출처: <http://m.yakup.com/news/index.html?mode=view&nid=302443>

2 서비스 기획

BM(비즈니스 모델) 설계

1. 문제	2. 고객 세그먼트	3. 가치 제안
<ul style="list-style-type: none"> • 응급 상황 시 즉각적인 도움을 요청하기 어려움 (특히 고령자) • 응급 구조 요청을 위해 사용자가 직접 조작해야 하는 불편함 • 혼자 있는 경우, 의식이 없을 시 응급 조치가 어려움 	<ul style="list-style-type: none"> • 고령자 및 만성 질환자 • 스마트워치를 착용하는 건강 관리 관심층 • 원격 모니터링이 필요한 보호자(부모·가족) 	<ul style="list-style-type: none"> • AI 기반 응급 감지 및 사용자 맞춤형 응급 대응 제공 • 보호자가 고령자의 건강 상태를 실시간 모니터링 가능 • 신속한 응급 구조 요청 및 최적의 대응 가이드 제공
4. 솔루션	5. 경쟁 우위	6. 수익 모델
<ul style="list-style-type: none"> • AI가 응급 상황을 판단하고 사용자 맞춤형 조치를 안내 • 보호자가 실시간으로 고령자의 상태를 확인 가능 • 사용자 데이터를 기반으로 지속적인 알고리즘 개선 	<ul style="list-style-type: none"> • 고도화된 AI 기반 최적화 기술 • 사용자 데이터 학습을 통한 맞춤형 응급 대응 시스템 • 다양한 스마트워치 브랜드와 연동 가능 	<ul style="list-style-type: none"> • B2C: 구독 기반 서비스 (예: 응급 대응 AI 프리미엄 서비스) • B2B: 스마트워치 제조업체 및 보험사와의 협업 • 데이터 기반 수익: 건강 데이터 분석 서비스 제공
7. 비용 구조	8. 핵심 지표	9. 채널
<ul style="list-style-type: none"> • 클라우드 비용 • 개발 비용 	<ul style="list-style-type: none"> • 사용자 수 (서비스 가입자 수, 월간 활성 사용자 수) • 응급 감지 성공률 (정확도, 오탐/미탐 비율) • 긴급 구조 요청 전송 횟수 • 파트너십 체결 수 	<ul style="list-style-type: none"> • 디지털 마케팅 (앱스토어, 웹사이트, SNS 광고) • 헬스케어 및 보험사 제휴 • 병원 및 복지 기관과 협업

본 서비스는 스마트워치를 활용한 AI 기반 응급 감지 및 대응 시스템으로, 사용자의 생체 데이터를 실시간으로 분석하여 응급 상황을 판단하고 즉각적인 조치를 안내하는 기능을 제공합니다. 주요 기능으로는 AI 기반 응급 감지, 사용자 맞춤형 응급 대응, 비상 호출, 실시간 건강 모니터링, AI 학습을 통한 지속적인 개선이 포함됩니다.

AI는 심박수, 호흡률, 혈중 산소 포화도(SPO2), 피부 온도 등의 데이터를 실시간으로 분석하여 이상 징후를 감지하고, 사용자에게 즉각적인 알림을 전송합니다. 응급 상황이 감지되면 보호자가 실시간으로 사용자 상태를 확인할 수 있으며, 사용자가 직접 호출 버튼을 눌러 보호자에게 연락할 수도 있습니다. 이를 통해 사용자가 혼자 있는 상황에서도 신속한 대응이 가능하며, 보호자는 원격으로 가족의 건강 상태를 지속적으로 모니터링할 수 있습니다.

예를 들어, 고령자가 집에서 휴식을 취하는 동안 심박수와 혈중 산소 포화도가 급격히 변하면 스마트워치가 이를 감지하고 사용자에게 경고를 보냅니다. 사용자가 위험을 느낄 경우 직접 호출 버튼을 눌러 보호자에게 즉시 연락할 수 있으며, 보호자는 앱을 통해 실시간으로 상태를 확인한 후 적절한 조치를 취할 수 있습니다. 또한, 운동을 즐기는 사용자가 조깅 중 심박수가 비정상적으로 높아지면 AI가 이를 감지하여 운동 강도를 줄이도록 안내하고, 지속적으로 이상 징후가 감지될 경우 보호자에게 추가 알림을 보냅니다. 보호자는 모바일 앱을 통해 부모님의 건강 상태를 주기적으로 확인할 수 있으며, 스트레스 지수 상승이나 수면 질 저하 등의 변화가 감지되면 미리 대응할 수 있습니다.

이러한 서비스가 구현되기 위해서는 실시간 데이터 수집 및 분석, 사용자 맞춤형 응급 대응, 보호자의 원격 모니터링 기능이 필수적입니다. 따라서 스마트워치의 생체 데이터를 클라우드 기반으로 저장하고 분석할 수 있는 인프라를 구축하며, AI 모델을 활용하여 응급 감지 기능을 지속적으로 개선합니다. 사용자의 개인정보 보호를 위해 데이터를 암호화하고, 실시간 데이터 분석 및 알림 전송 속도를 최적화하여 응급 대응의 실효성을 극대화합니다. 또한, 다양한 스마트워치 브랜드와 연동 가능하도록 시스템을 확장성 있게 설계하며, 고령자도 쉽게 사용할 수 있도록 직관적인 UI를 제공합니다.

이 시스템은 클라우드 기반 데이터 저장소(Azure DataLake), 분석 엔진(Apache Spark), 데이터베이스(Azure MySQL)를 활용하여 안정적인 데이터 관리 및 분석을 수행하며, AI 모델을 기반으로 응급 감지 알고리즘을 지속적으로 개선합니다. 모바일 및 웹 애플리케이션을 통해 사용자와 보호자가 데이터를 쉽게 확인하고, 실시간으로 응급 대응을 할 수 있도록 지원합니다. 이를 통해 본 서비스는 웨어러블 기기를 활용한 건강 관리 및 응급 대응의 새로운 패러다임을 제시하며, 사용자의 안전과 건강을 효과적으로 보호하는 역할을 수행할 것입니다.

3 API 정의서 및 서비스 개발

3.1 API 설계 및 데이터 흐름

3.1.1 사용자 인증 API

요구사항명	Method	URI	설명
회원가입	POST	/api/signup	클라이언트에서 입력받은 사용자 정보를 바탕으로 새로운 사용자 계정을 생성합니다.
로그인	POST	/api/login	클라이언트에서 입력받은 ID와 비밀번호 정보를 바탕으로 사용자 인증을 수행하고 세션을 생성합니다.
로그아웃	POST	/api/logout	클라이언트로부터 로그아웃 요청을 받으면 세션을 종료하고 로그아웃을 처리합니다.

3.1.2 생체 데이터 수집 및 분석 API

요구사항명	Method	URI	설명
사용자 상태 갱신	POST	/api/update_status	생체 데이터를 로드하여 모델을 통해 예측하고, 예측 결과를 사용자 상태로 저장합니다. State(사용자 상태) 값이 1 이상일 때 /api/predict_model2 (응급 상황 감지) API를 호출한다.
사용자 상태 조회	GET	/api/get_status/	서버에서 최근에 갱신된 사용자 상태 데이터 10개 행을 조회하여 클라이언트로 전송하고, 클라이언트에서는 10개의 사용자 자세를 renderStateChart 함수로 사용자 상태 차트를 구현합니다
사용자 자세 갱신	POST	/api/update_posture/	센서 데이터를 로드하여 Mobile VIT v3 모델을 통해 사용자 자세(0:Downstair, 1:Upstair, 2:Running, 3:Sitdown, 4:StandUp, 5:Walking, 6:Lying, 7:Fall)를 예측하고, 예측 결과를 저장합니다.
사용자 자세 조회	GET	/api/get_posture/	서버에서 최근에 갱신된 사용자 자세 데이터 10개 행을 조회하여 클라이언트로 전송하고, 클라이언트에서는 10개의 사용자 자세를 renderPostureChart 함수로 사용자 자세 차트를 구현합니다.

3.1.3 응급 상황 감지 및 대응 API

요구사항명	Method	URI	설명
응급상황 감지	POST	/api/predict_model2	주의 또는 위급 상태의 생체 데이터를 기반으로 심박, 호흡, 피부 온도, 혈중 산소 농도 이상 여부를 확인하고 openai api로 대치 요령 문구를 클라이언트로 전송합니다.
응급 구조 요청	POST	/api/emergency_call/	라이언트에서 받은 응급 구조 요청을 처리하고, 결과를 클라이언트로 전송합니다.

3.2 백엔드 개발

본 프로젝트의 백엔드는 Django 프레임워크를 기반으로 구현되었으며, 사용자 인증, 생체 데이터 관리, 응급 상황 감지 및 대응 등의 주요 기능을 제공합니다. 또한, OpenAI API를 활용하여 응급상황 발생 시 사용

자에게 적절한 조치를 안내하는 기능을 포함하고 있습니다.

3.2.1 데이터베이스 설정

MySQL을 사용하여 사용자 및 생체 데이터를 저장하며, SQLAlchemy를 활용하여 데이터 처리 성능을 최적화하였습니다. 데이터베이스 설정은 settings.py에서 정의되며, Django ORM을 통해 모델을 관리합니다

3.2.2 모델 정의

- User 모델 : 사용자 기본 정보를 저장하는 모델

Column name	Type	Default	설명
unique_num	BigAutoField	PK, NotNull, AUTO_INCREMENT,INVISIBLE	회원 고유 번호
id	CharField	Unique, NotNull, max_length=255	사용자 ID
password	CharField	NotNull, max_length=255	해싱된 비밀번호
name	CharField	NotNull, max_length=255	사용자 이름
age	IntegerField	NotNull	나이
address	CharField	NotNull, max_length=255	주소
detailed_address	CharField	max_length=255	상세 주소
phone_num	CharField	NotNull,max_length=20	사용자 전화번호
guard_name	CharField	max_length=255	보호자 이름
guard_phone_num	CharField	max_length=20	보호자 전화번호
danger_degree	IntegerField		위험도
user_posture	IntegerField		사용자 자세

- UserStatus : 사용자의 건강 상태(정상, 주의, 응급)를 저장하는 모델

Column name	Type	Default	설명
user	BigAutoField	NotNull , ForeignKey	User 모델과 연결된 외래키
status	IntegerField	NotNull	사용자의 상태(0:정상, 1:주의, 2:위험)
updated_at	DateTimeField	NotNull, default=timezone.now):	상태 갱신 시간

- UserPosture : 사용자의 자세 정보를 저장하는 모델로, 다양한 활동 상태(걷기, 앉기, 넘어짐 등)를 기록

Column name	Type	Default	설명
user	BigAutoField	NotNull , ForeignKey	User 모델과 연결된 외래키
posture	IntegerField	NotNull	사용자의 자세
updated_at	DateTimeField	NotNull, default=timezone.now):	자세 갱신 시간

3.3 프론트엔드 개발

프론트엔드는 React.js와 TypeScript를 기반으로 개발되었으며, 실시간 생체 데이터 모니터링 및 응급 상황 감지를 직관적으로 제공하는 UI/UX 구현에 중점을 두었습니다.

- 메인 페이지 (Main.tsx)

메인 화면에서 로그인 상태가 인증되면 사용자의 실시간 상태를 모니터링할 수 있는 대시보드 형태로 구성

하였습니다. ECharts 라이브러리를 활용하였는데, 상단에는 시계열 Line Chart를 통해 사용자의 상태(정상/주의/위험)를 색상별로 구분하여 표시하고, 하단에는 Stack Horizontal Bar 차트를 통해 사용자의 자세 변화(걷기/앉기/넘어짐 등)를 시간대별로 확인할 수 있도록 구현했습니다. 사용자의 상태와 자세 데이터는 10분 간격으로 자동 갱신되며, 응급 상황이 감지될 경우 openai api로부터 받아온 대처 요령 문구를 경고 메시지로 표시하고, 119 신고 기능을 포함하였습니다.

- 로그인 페이지 (Login.tsx)

사용자 ID 및 비밀번호 입력을 받아 로그인 요청을 수행하며, ID 유효성 검사와 CSRF 토큰 기반의 보안 처리를 통해 안전한 인증을 구현하였습니다.

- 회원가입 페이지 (Register.tsx)

회원 가입에 필요한 사용자 정보에 보호자의 이름과 번호를 함께 등록할 수 있도록 설계했으며, ID 중복 검사 및 입력 필드에 대한 유효성 검사를 통해 데이터 무결성을 확보하였습니다.

- 데이터 통신

데이터 통신은 Axios를 활용한 RESTful API 방식을 통해 서버와 데이터를 주고받으며, 폴링 방식을 적용하여 실시간 모니터링을 구현하였습니다. React의 useState 및 useEffect를 활용하여 상태를 관리하고 비동기 작업을 효율적으로 처리하며, 에러 발생 시 사용자에게 적절한 피드백을 제공하도록 설계하였습니다. TypeScript를 도입하여 코드의 안정성을 높였으며, 이를 통해 사용자는 실시간으로 건강 상태를 모니터링하고, 보호자는 원격으로 상태를 확인하며 필요 시 즉각적인 조치를 취할 수 있습니다.

4 생체 신호 기반 응급 판단 모델 개발 및 데이터 셋

4.1 데이터 출처 및 구성

- 환경 센서 및 생체신호 데이터: SME20SD, MVPC10, Vital Ring 장비에서 실시간 수집.
- 상담 기록 데이터: 돌봄 기관의 사례 기록 및 일정 메모 활용.
- 고위험 대상 데이터: 요양병원의 응급환자 데이터 및 돌봄 기관에서 선별한 대상 포함.

데이터 유형	형식	사용 여부	설명
통합 라벨링	JSON	사용	응급 판단 상태 코드(Estimation)와 원인(Reason) 포함
생체신호	JSON	사용	심박수, 호흡수, 산소포화도 등
환경 측정	JSON	미사용	응급 감지와 직접적 관련 적음
적외선 이미지	PNG	미사용	단순한 동작 감지만 가능
응급 요청 기록	JSON	미사용	원인 파악 어려움
상담 기록	JSON	사용	운동 여부 등 추가 정보 활용

4.2 데이터 전처리 과정

- 데이터 평탄화: JSON 배열 데이터를 개별 행으로 변환하여 분석 가능하도록 처리.
- 불필요한 변수 제거: 타임스탬프 데이터 삭제하여 일관성 유지.
- 응급 발생 데이터 필터링: 응급 버튼 사례(267건) 제외.
- 라벨링 변환: 정상(2), 주의(1), 응급(0)으로 변환하여 모델 학습 최적화.
- 이상치 탐지 및 제거
 - IQR 기반 탐지 적용:
 - 심박수, 호흡수, 칼로리 소비량 → 3사분위수(Q3) 이상 값 제거.
 - 산소포화도 → 1사분위수(Q1) 이하 값 제거.

- Feature Engineering:

ANOVA 검정 결과, 걸음 수·칼로리 소비량이 상태 분류에 유의미한 영향을 미침($p\text{-value} < 0.05$).
차원 축소(PCA) 적용 → exercise_pca 생성.
상담 기록에서 운동 여부 반영 → 운동 외 이상(high energy outlier) 변수 추가

4.3 모델 설계 및 학습

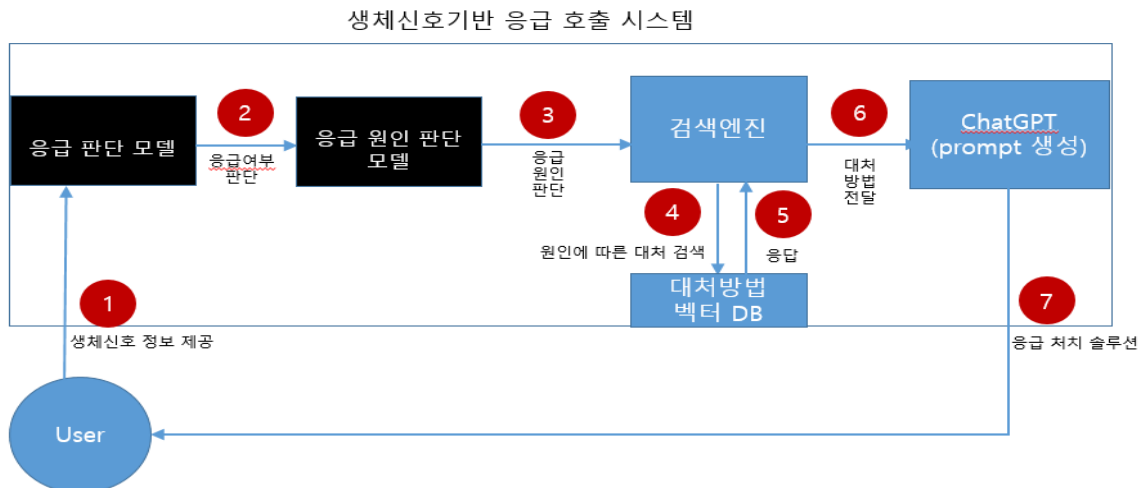


그림 2. 생체신호기반 응급 호출 시스템 아키텍처

응급 판단 및 응급 원인 판단 모델은 Tableau 데이터에 대해 성능이 우수하다고 알려진 트리 기반 앙상블 모델인 LightGBM과 XGBoost를 사용하였으며 추가적인 모델 학습을 위해 딥러닝 기반 MLP 모델과 트리 기반 모델의 장점을 결합한 TabNet 모델을 활용하였다

1 응급 판단 모델

응급 판단 모델은 Multi Class 분류 문제로 접근하였으며 상태에 따라 정상은 0 주의는 1 위험은 2로 구분하여 예측을 수행하였다

2 응급 원인 판단 모델

응급 원인 판단 모델은 Multi Label 분류 문제로 접근하였으며 응급 상태인 주의와 위험으로 분류된 데이터에 대해 심박 호흡 피부 온도 산소포화도를 각각 이진화하여 예측하였다

4.4 모델 예측 결과 (Multi Class 분류)

모델	응급 판단 정확도 (%)	원인 판단 정확도 (%)
LightGBM	98.3	98.5
XGBoost	97.8	97.9
TabNet	98.2	98.3
MLP	95.6	87.7

- LightGBM, XGBoost, TabNet을 사용하여 상태(정상-주의-위험) 분류.

- 응급 상태(주의, 위험)인 데이터에서 심박, 호흡, 피부 온도, 산소포화도 이상 여부를 개별 이진 분류로 예측.

- MLP 모델(3개 은닉층, Softmax 활성화) 추가 실험.

- LightGBM이 가장 높은 정확도를 보이며, 빠른 예측 속도와 해석 가능성을 제공.

- TabNet은 딥러닝 기반 모델로, 트리 기반 모델 대비 성능 차이는 적지만 데이터 양이 많아질수록 개선 가

등.

- MLP 모델은 추가적인 하이퍼파라미터 튜닝이 필요.

4.5 검색 엔진 및 GPT 적용

```
# 프롬프트 템플릿 정의
template = """
현재 상태: {status}
원인: {cause}
대처 방법: {method}

너는 응급상황 대응 전문가다.
고객의 현재 상태를 단호하게 설명하고, 즉각적인 대처 방법을 안내하라.
'만약' 같은 불확실한 표현을 사용하지 말고, 필요한 말만 전달하라.
위급한 경우 119에 신고해야 한다는 경각심을 심어줘라.
출력은 다음 형식으로 하라:

1. 현재 상태 설명 (단호한 문장)
2. 원인 분석 (간결하게)
3. 즉각적인 대처 방법 (실용적이고 구체적으로)
4. 경고 문구 (119 신고 필요 여부 강조)

"""
```

그림 3, GPT 프롬프트 템플릿

- 검색 엔진 (Llama-Index): WHO의 "생체신호 기반 안전 지침" 문서를 활용하여 응급 원인과 대응 방법 검색.
- LangChain Prompt Template 활용: 응급 판단 결과(status) + 응급 원인(cause) + 대처 방법(method)을 결합하여 최적 대응책 제공.

5 센서 신호 기반 사용자 행동 판단 모델 개발 및 데이터 셋

5.1 데이터 수집 및 전처리

오픈소스 IMU 센서(손목, 허리 위주-> 주로 센서가 위치하는 곳)

HIFD² FallAID³ HHAR⁴ RealWorld⁵ PAMAP2⁶ MotionSense⁷ UCI⁸

7가지의 일상 상황과 1가지의 넘어짐 상황으로 라벨링 진행

Downstair, Upstair, Running, Sit down, Stand up, Walking, Lying, Fall

전체 데이터 중 Accelerometer와 Gyroscope만 추출

Train : validation : test = 7(169732) : 1(24186) : 2(48363)의 비율로 분리된 데이터 셋을 사용

² HIFD - https://github.com/nhoyh/HR_IMU_falldetection_dataset

³ FallAID - <https://ieee-dataport.org/open-access/fallalld-comprehensive-dataset-human-falls-and-activities-daily-living>

⁴ HHAR - <http://archive.ics.uci.edu/ml/datasets/Heterogeneity+Activity+Recognition>

⁵ RealWorld - https://www.uni-mannheim.de/dws/research/projects/activity-recognition/#dataset_dailylog

⁶ PAMAP2 - <https://archive.ics.uci.edu/dataset/231/pamap2+physical+activity+monitoring>

⁷ MotionSense - <https://github.com/mmalekzadeh/motion-sense/tree/master/data>

⁸ UCI - <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>

5.2 모델 설계 및 학습

5.2.1 사용 모델 아키텍처, MobileViTv1에서 MobileViTv3로의 개선점

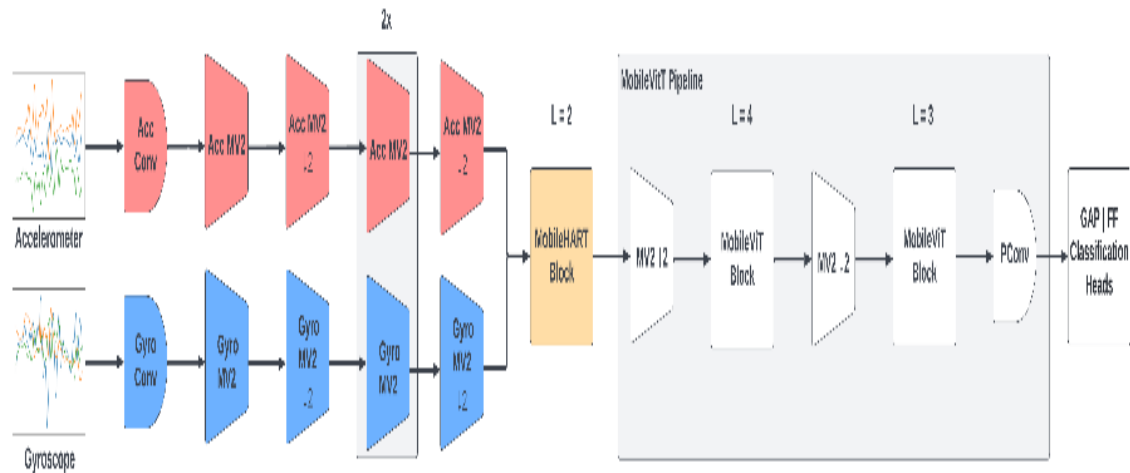


그림 4. 사용 모델 아키텍처

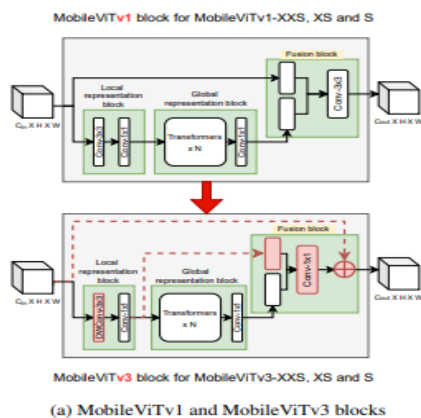


그림 5. MobileViTv1에서 MobileViTv3로의 개선점

5.2.2 Tensorflow 환경에서 사용 가능한 MobileViTv3 모델을 MobileViTv1을 기반으로 제작.

변경사항 : layer들을 2d->1d로 dimension을 감소, transformer block 전 후의 unfolding, folding 삭제(time series sensor data(2d)를 input으로 받기 때문)

block의 input을 DepthwiseConv1d(3x3)로 받음(parameter 감소)

local representation과 global representation을 fuse(input data보다 연관성이 높아 학습에 도움이 됨),

residual connection을 추가

5.2.3 학습 환경

100 epoch, Adam optimizer, cross-entropy loss with 0.1 label smoothing, learning rate: first 10 epoch - $1e-4 \sim 1e-3$ (warmup) after warmup - $1e-4$ annealing cosine scheduler 적용, no additional data augmentation, batch size = 196

Evaluate using Top-1 accuracy

5.2.4 성능 평가(결과)

Model's spec

	parameters	Inference time(avg)	Model size(MB)
V1	2,544,992	0.089초	16.19
V3	2,297,216	0.088초	15.43

F1-score

	FallAIID	UCI	HHAR	MotionSense	RealWorld	PAMAP	HIFD
V1	96.32	99.95	99.7	99.53	99.67	99.49	100
V3	96.22	100	99.32	99.91	99.94	98.61	100

Clustered combined test set using t-SNE

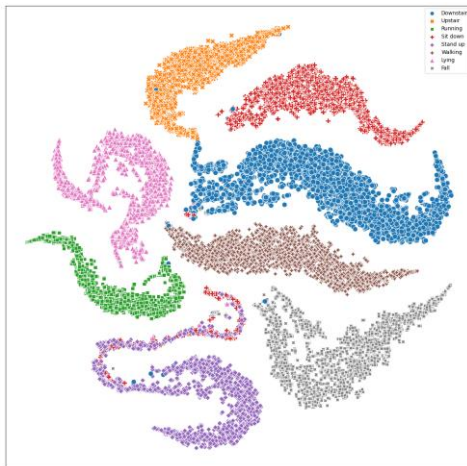


그림 6. 시각화

6 웹 기능

10분마다 생체신호 정보와 자세 위치 정보가 업데이트 되어 응급 판단과 자세 판단에 대한 정보를 제공

6.1 응급 판단 시스템

주요기능

1. 실시간 상태 확인
 - 주의/응급 상태를 감지하여 경고 메시지 제공
 - 응급 발생 시 대처법 자동 안내
2. 신고 시스템
 - “신고하기” 버튼 클릭 시 즉각 응급 신고

6.2 자세 판단 시스템

주요기능

1. 실시간 행동 확인
 - 시간 별로 Downstair, Upstair, Running, Sitdown, StandUp, Walking, Lying, Fall 중 한 가지를 표시

7 기획 및 배포

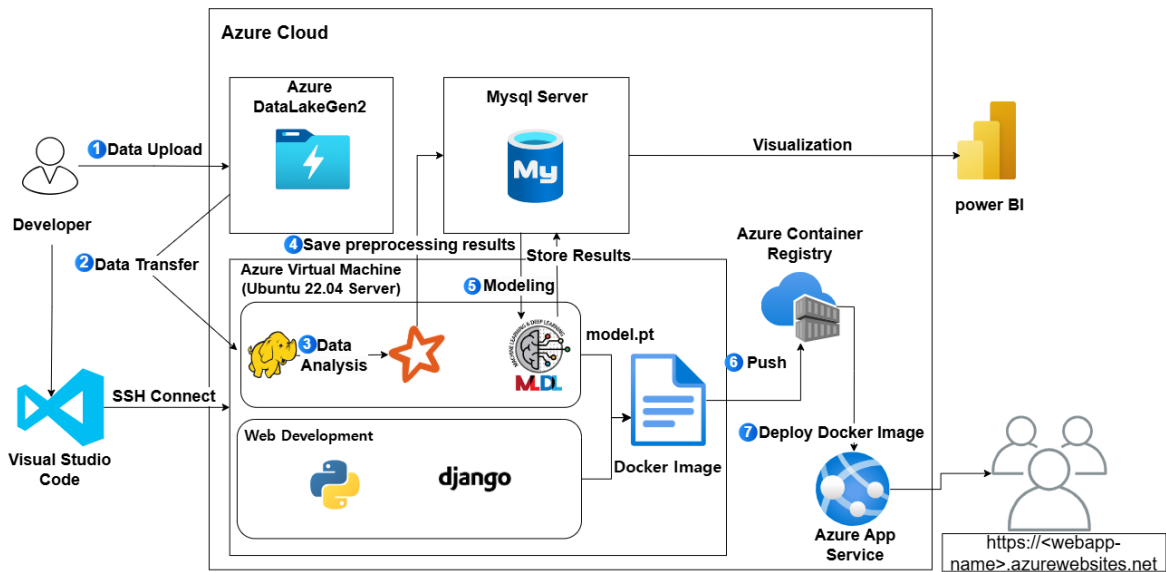


그림 7. 서비스 시스템 아키텍처

본 서비스는 Azure 클라우드 플랫폼을 기반으로 데이터 수집, 분석, 모델 학습, 배포까지의 전 과정을 통합적으로 지원하는 End-to-End 데이터 처리 아키텍처입니다. 본 문서에서는 전체 시스템의 구조 및 데이터 흐름을 설명하며, 서비스 확장성과 효율성을 극대화하기 위한 핵심 인프라 설계를 제시합니다. 본 시스템은 데이터의 효율적 관리와 분석을 위한 분산 처리 기술과 클라우드 네이티브 환경을 활용하여 설계되었습니다.

Azure DataLake Gen2를 활용하여 대용량 데이터를 효율적으로 저장 및 관리하며, 저장된 데이터는 실시간 분석 및 배치 처리를 위한 HDFS(Hadoop Distributed File System)로 전송됩니다. 이후 Apache Spark를 활용하여 데이터 전처리 및 분석을 수행하며, Spark의 인메모리 컴퓨팅을 통해 기존 하둡 기반 처리보다 빠른 데이터 처리가 가능합니다. 분석된 데이터는 Azure MySQL Server에 저장되며, 이후 모델 학습을 위한 데이터 전처리가 진행됩니다. 머신러닝 및 딥러닝 모델 학습 프로세스를 적용하여 서비스의 핵심 기능을 담당하는 모델이 개발됩니다.

개발된 트랜스포머 모델(모델명 변경 예정) 및 생체신호 기반 응급 분류 시스템을 컨테이너화하여 Azure Container Registry에 등록한 후, Azure App Service를 통해 배포하여 확장성과 유지보수성을 보장합니다. 이러한 아키텍처를 통해 클라우드 기반 분산 처리 기술을 활용하여 대용량 데이터 처리 속도를 향상시키고 확장성을 극대화할 수 있으며, 컨테이너화된 모델 배포를 통해 서비스 운영의 안정성과 유지보수의 용이성을 확보할 수 있습니다. 데이터 분석부터 모델 배포까지의 효율적인 파이프라인을 구축함으로써 서비스의 경쟁력을 강화할 수 있습니다.