

320025-03

블록체인 기술기반의 농업기계 관리기술 개발

2022

농림축산식품부  
농림식품기술기획평가원

보안 과제( ), 일반 과제( O ) / 공개( O ), 비공개( )발간등록번호( O )  
첨단농기계산업화기술개발사업 2022년도 최종보고서

발간등록번호

11-1543000-004317-01

# 블록체인 기술기반의 농업기계 관리기술 개발

2023.05.26.

주관연구기관 / 주식회사 긴트  
공동연구기관 / 경상국립대학교  
공동연구기관 / 한국농기계공업협동조합

농림축산식품부  
(전문기관)농림식품기술기획평가원

## 제 출 문

농림축산식품부 장관 귀하

본 보고서를 “블록체인 기술기반의 농업기계 관리기술 개발”(개발기간 : 2020. 04. 29  
~ 2022. 12. 31)과제의 최종보고서로 제출합니다.

2023. 05. 26.

주관연구기관명 : 주식회사 긴트 (대표자) 김용현 (인)

공동연구기관명 : 경상국립대학교산학협력단 (대표자) 정재우 (인)

공동연구기관명 : 한국농기계공업협동조합 (대표자) 김신길 (인)

주관연구책임자 : 김용현

공동연구책임자 : 최경문

공동연구책임자 : 장길수

국가연구개발사업의 관리 등에 관한 규정 제18조에 따라 보고서 열람에 동의합니다.

최종보고서							보안등급			
							일반[√], 보안[ ]			
중앙행정기관명	농림축산식품부			사업명	사업명		첨단농기계산업화 기술개발사업			
전문기관명 (해당 시 작성)	농림식품기술기획평가원			내역사업명 (해당 시 작성)			농기계산업혁신기술			
공고번호	농축2020-63호			총괄연구개발 식별번호 (해당 시 작성)			-			
				연구개발과제번호			320025-03			
기술분류	국가과학기술 표준분류	LB0801	40%	LB0804	30%	EA0506	30%			
	농림식품과학기술 분류	RC0102	100%	-	%	-	%			
총괄연구개발명 (해당 시 작성)	국문	-								
	영문	-								
연구개발과제명	국문	블록체인 기술기반의 농업기계 관리기술 개발								
	영문	Development of Agricultural machinery management system based on block chain technology								
주관연구개발기관	기관명	주식회사 긴트		사업자등록번호	266-88-00215					
	주소	(우)16227, 경기 수원시 영통구 대학3로 7, 5층		법인등록번호	134111-0422646					
연구책임자	성명	김용현		직위	대표이사					
	연락처	직장전화	031-212-4994		휴대전화	010-2885-3230				
		전자우편	yonghyeon.kim@gintlab.com		국가연구자번호	11521961				
연구개발기간	전체	2020. 04. 29 - 2022. 12. 31 (2년 8개월)								
	단계	1단계	2020. 04. 29 - 2021. 12. 31 (1년 8개월)							
		2단계	2022. 01. 01 - 2022. 12. 31 (1년)							
연구개발비 (단위: 천원)	정부지원 연구개발비	기관부담 연구개발비		그 외 기관 등의 지원금 지방자치단체 기타( )				합계	연구개발 비의 지원금	
	현금	현금	현물	현금	현물	현금	현물			현금
총계	1,460,000	0	444,000	-	-	-	-	1,460,000	444,000	1,904,000
1단계	1년차	400,000	0	134,000	-	-	-	400,000	134,000	534,000
	2년차	530,000	0	177,000	-	-	-	530,000	177,000	707,000
2단계	3년차	530,000	0	133,000	-	-	-	530,000	133,000	663,000
공동연구개발기관 등	기관명	책임자	직위	휴대전화	전자우편	비고		역할	기관유형	
공동연구개발기관	경상대학교 산학협력단	최경문	교수	010-3871-9529	gmchoi23@hanmail.net	공동	대학			
	한국농기계공업 협동조합	장길수	팀장	010-4173-7436	jks7436@hanmail.net	공동	기타			
위탁연구개발기관	충남대학교 산학협력단	김용주	교수	010-4123-5290	babina@cnu.ac.kr	위탁	대학			
연구개발과제 실무담당자	성명	손승락		직위	이사					
	연락처	직장전화	070-4225-4122		휴대전화	010-9428-2023				
		전자우편	seungrak.son@gintlab.com		국가연구자번호	10915715				

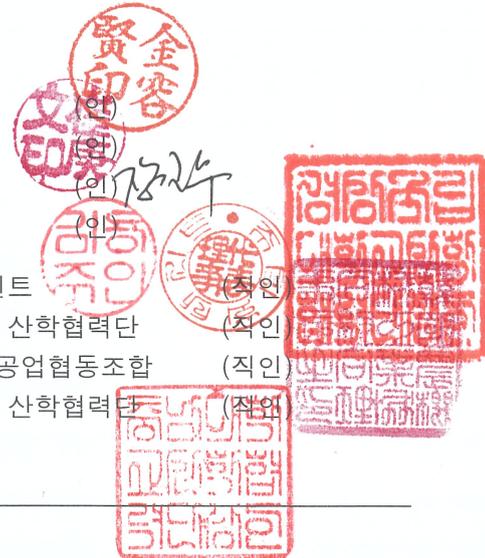
이 최종보고서에 기재된 내용이 사실임을 확인하며, 만약 사실이 아닌 경우 관련 법령 및 규정에 따라 제재처분 등의 불이익도 감수하겠습니다.

2023년 2월 17일

주관연구책임자: 김용현  
 공동연구책임자: 최경문  
 공동연구책임자: 장길수  
 위탁연구책임자: 김용주

주관연구개발기관의 장: 주식회사 긴트 (직인)  
 공동연구개발기관의 장: 경상대학교 산학협력단 (직인)  
 공동연구개발기관의 장: 한국농기계공업협동조합 (직인)  
 위탁연구개발기관의 장: 충남대학교 산학협력단 (직인)

농림축산식품부장관·농림식품기술기획평가원장 귀하



## < 요약 문 >

사업명		첨단농기계산업화기술개발사업			총괄연구개발 식별번호 (해당 시 작성)		-	
내역사업명 (해당 시 작성)		농기계산업혁신기술			연구개발과제번호		320025-03	
기술 분류	국가과학기술 표준분류	LB0801	40%	LB0804	30%	EA0506	30%	
	농림식품 과학기술분류	RC0102	100%	-	%	-	%	
총괄연구개발명 (해당 시 작성)		-						
연구개발과제명		블록체인 기술기반의 농업기계 관리기술 개발						
전체 연구개발기간		2020. 04. 29 - 2022. 12. 31 (2년 8개월)						
총 연구개발비		총 1,904,000천원 (정부지원연구개발비: 1,460,000천원, 기관부담연구개발비: 444,000천원, 지방자치단체: 0천원, 그 외 지원금: 0천원)						
연구개발단계		기초[ ] 응용[ ] 개발[ <input checked="" type="checkbox"/> ]		기술성숙도 (해당 시 기재)		착수시점 기준( ) 종료시점 목표( )		
연구개발과제 유형 (해당 시 작성)		-						
연구개발과제 특성 (해당 시 작성)		-						
연구개발 목표 및 내용	최종 목표	농업기계 이력관리를 위한 블록체인 시스템 개발 및 데이터 확보 체계 구축						
	전체 내용	<ul style="list-style-type: none"> <li>○ 농업기계 생애주기별 이력데이터 블록체인 시스템 설계                             <ul style="list-style-type: none"> <li>- 블록체인 시스템이 적용된 농업기계 생애주기별 이력 데이터 관리 시스템 구축</li> <li>- 공공 데이터 상용화를 위한 공공 클라우드 플랫폼 적용</li> </ul> </li> <li>○ 농업기계 이력 및 운행 데이터 수집 시스템 개발                             <ul style="list-style-type: none"> <li>- 데이터 수집을 위한 농업기계 보급형 단말 HW와 SW 환경구축</li> </ul> </li> <li>○ 농업기계 이력 및 운행 데이터 기반 농업인/제조사 서비스 개발 및 상용화                             <ul style="list-style-type: none"> <li>- 농업 생산성, 편의성 향상을 위한 농업인 서비스 개발</li> <li>- 품질 향상을 위한 제조사 서비스 개발</li> </ul> </li> <li>○ 농업기계 이력 및 운행 데이터 입력 시스템 개발                             <ul style="list-style-type: none"> <li>- 이력 데이터 입력 및 정부 정책 지원을 위한 시스템 구축 (면세유, 금융 지원정책 지원시스템)</li> </ul> </li> <li>○ 수집된 농업기계 이력데이터 분석 및 활용 AI 알고리즘 개발                             <ul style="list-style-type: none"> <li>- 정부 정책 지원을 위한 농기계 가동률, 유휴 농업기계 추적 AI 알고리즘 개발</li> <li>- 농업 환경 지원을 위한 사용패턴, 최적 유지보수 AI 알고리즘 개발</li> </ul> </li> </ul>						

		○ 농작업 기계 내구수명 확보를 위한 고장 예지 기술 개발 - 고장 예지를 위한 필요 데이터 선정 및 계측 기술 개발 - 계측 시험 수행 및 데이터 분석 및 성능 평가
	1단계 (해당 시 작성)	목표 내용
	n단계 (해당 시 작성)	목표 내용

연구개발성과	구분	논문	특허	보고서 원문	연구 시설 · 장비	기술 요약 정보	소프트 웨어	화합물	생명자원		신품종	
	예상성과 (N/Y)	Y	Y	Y	Y	Y	Y	N	생명 정보	생물 자원	정보	실물

연구개발성과 활용계획 및 기대 효과	<p>&lt;활용계획&gt;</p> <ul style="list-style-type: none"> <li>- 자율주행 트랙터와 연계하여 정밀 농업 서비스로 활용한다.</li> <li>- 조기 사업화를 위한 시군구 실증화 기반을 확보한다.</li> </ul> <p>&lt;기대효과&gt;</p> <ul style="list-style-type: none"> <li>○ 기술적 측면 <ul style="list-style-type: none"> <li>- 블록체인 도입으로 인해 농업기계 이력 데이터의 투명성이 보장되며, 별도의 중간 관리자 없이 생애주기별 이력 데이터의 추적이 용이해짐.</li> </ul> </li> <li>○ 경제적, 사회적 측면 <ul style="list-style-type: none"> <li>- 국내 뿐만 아니라 해외 시장으로 진출하여 국가 경쟁력이 이바지함.</li> </ul> </li> <li>○ 사회적 측면 <ul style="list-style-type: none"> <li>- 고령층, 여성 등 정보 취약계층으로 신뢰성있는 정보 습득이 용이해짐.</li> <li>- 정확한 이력관리를 통해 농작업이 효율 증대되고 고장 예측으로 비용 절감.</li> </ul> </li> </ul>											
	연구개발성과의 비공개여부 및 사유	없음										

연구개발성과의 등록·기탁 건수	논문	특허	보고서 원문	연구 시설 · 장비	기술 요약 정보	소프트 웨어	표준	생명자원		화합물	신품종	
	3	5						생명 정보	생물 자원		정보	실물

연구시설·장비 종합정보시스템 등록 현황	구입 기관	연구시설 · 장비명	규격 (모델명)	수량	구입 연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	ZEUS 등록번호

국문핵심어 (5개 이내)	블록체인	AI 알고리즘	농기계	이력데이터	추적관리
영문핵심어 (5개 이내)	Blockchain	AI Algorithm	Agricultural machinery	History data	Tracking management

## < 목 차 >

1. 연구개발과제의 개요	7
2. 연구개발과제의 수행 과정 및 수행내용	
- 주관연구기관(킨트)	27
- 공동연구기관(경상국립대학교)	74
- 공동연구기관(한국농기계공업협동조합)	132
3. 연구개발과제의 수행 결과 및 목표 달성 정도	171
4. 목표 미달 시 원인분석(해당 시 작성)	197
5. 연구개발성과 및 관련 분야에 대한 기여 정도	198
6. 연구개발성과의 관리 및 활용 계획	199
별첨 자료	200

# 1. 연구개발과제의 개요

## 가. 일반 개요

### ○ 추진 배경 및 필요성

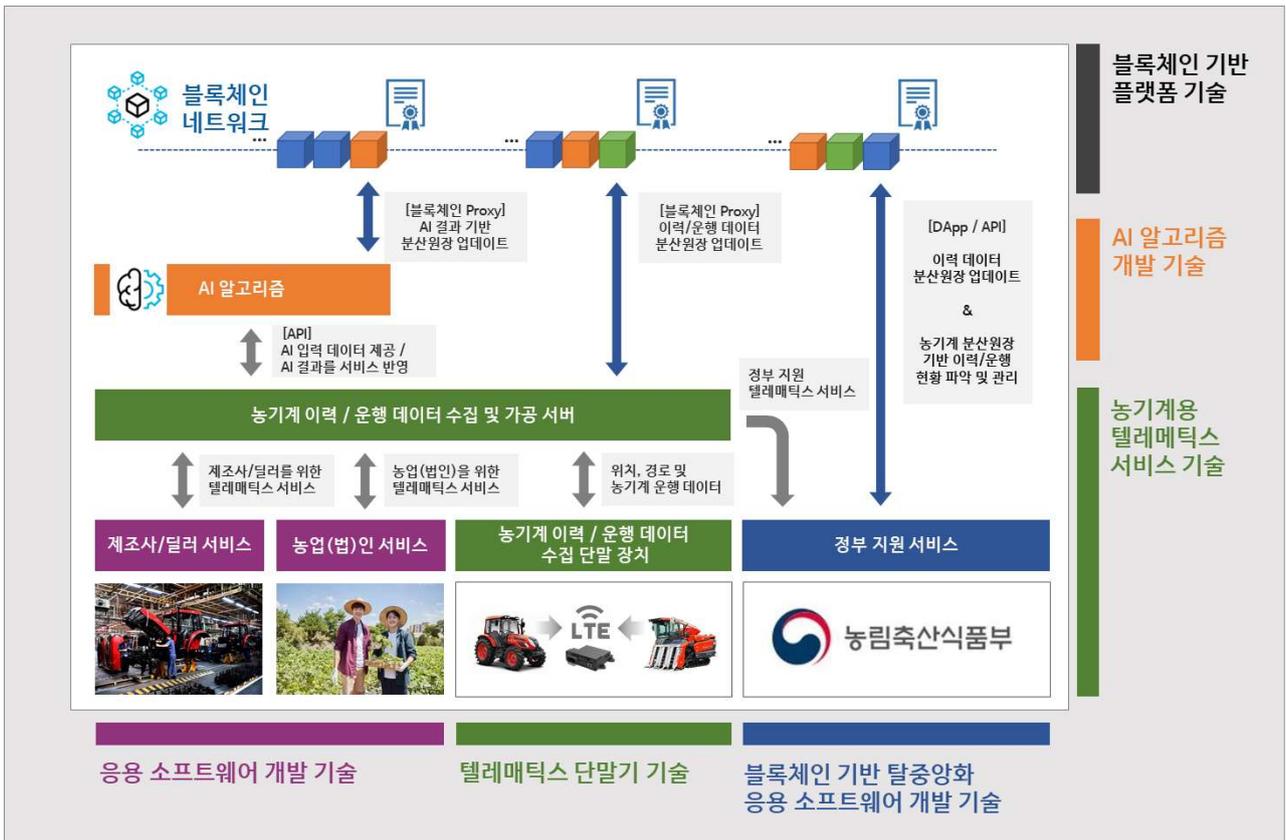
- 4차 산업혁명시대로 진입하며 기존의 전통적인 농업기계는 ICT 기술이 접목된 자율주행 및 정밀농업기계 등으로 발전하고 있음
- 그러나 국내 농업기계는 매년 각 지자체의 읍면동 담당자가 전수조사로 이루어지고 있는 실정임.
- 농업기계를 등록하여 관리하게 되는 경우 농업인의 재산권 행사가능, 손실 방지, 소유자 증명이 가능해지며 정확한 농업기계 사용의 추적이 용이하여 정책 추진시 효율적이며 면세유 절감 효과를 얻을 수 있음.
- 농협 면세유는 농협이 발급한 전자카드로 농민이 유류를 면세가격으로 구입 후 주유소가 국세청에 환급하는 신청하는 프로세스로 이루어짐
- 농기계에 사용에 대한 이력관리가 제대로 이루어지지 않기 때문에 농기계에 배정받은 면세유를 농업용 이외의 용도로 사용하는 일이 발생하고 있음.
- 블록체인을 통해 농기계 사용이력에 대한 정확한 관리가 이루어진다면 사용량에 따른 면세유판매가 가능해짐.
- 농림식품부는 농기계 제조·수리업체의 원자재·부품 구매 및 농업인의 농기계 구매, 농기계 임대시스템 구축·운영 등 농기계의 생산·수리(공급 측면), 구매·임대(수요 측면) 분야별로 보조금 및 융자금을 지원하고 있음.
- 이력관리가 정확하게 이루어지지 않아 용자액과 틀리게 장비를 구입하는 등 지원 목적과 다르게 이루어지는 경우가 있음. 블록체인을 이용하려 이력관리가 가능해지면 제조부터 용자까지 투명하게 관리가 가능함.
- 국내 농업기계는 각 제조사별로 개발에 대한 관리가 이루어짐
- 이력관리의 필요성은 농기계자금 용자, 면세유 구매, 중고기계 구매 등 다양한 방면에서 요구되고 있지만 신뢰성을 확보할 수 있는 시스템이 존재하지 않아 농기계 관리에 많은 어려움이 있음.
- 만약 블록체인을 통한 신뢰받을 수 있는 농기계 이력관리 시스템이 구축되면 다양한 형태로 농기계가 유통될 수 있는 새로운 4차 산업혁명의 시작이 될 수 있음.

### ○ 연구개발 대상 및 기술 제품의 개요

- 개발 기술의 정의

#### 【 블록체인 기반 농기계 운용 데이터 관리 및 응용 기술 】

탈중앙성과 높은 신뢰도를 가진 **블록체인 플랫폼을 기반으로**  
다양한 **농기계 운용(이력 및 운행) 데이터**를 수집 및 응용하여,  
정부 및 민간 수요에 맞는 **농산업 서비스를 제공하는 기술**



[블록체인 기반 농기계 운영 데이터 관리 및 응용 기술 개념도]

○ 개발 기술의 용도 및 적용 분야

- 정부 농기계 지원 서비스 관리에 적용
  - 농기계와 관련된 정부기관(농림수산물부, 지방자치단체, 농업진흥청), 농업협동조합, 제조사 개별적으로 보유하고 관리하는 시스템을 블록체인 기반 농기계 운영 데이터 관리 기술을 기반으로 통합하여 정보의 신뢰성을 높이고 농민과 정부기관이 동시에 사용할 수 있는 통합 시스템을 통해 농기계의 구매, 수리, 면세유 혜택 서비스를 제공
- 국내 농기계 제조사의 수출경쟁력 확보
  - 농업은 세계 모든 국가가 전략산업으로 지원 및 통합적으로 관리하고 있고 다양한 지원정책을 추진하고 있음. 그러나 사업비용에 집행에 대한 절차나 관리 감독이 이루어지고 있지 않기 때문에 본 과제의 기술을 적용한 블록체인 기반 농기계 운영 데이터 관리시스템을 농기계와 같이 수출한다면 타 국가의 농기계 제품보다 경쟁력을 확보할 수 있음
- 농기계 관련 중고거래 및 공유경제를 활용한 4차산업으로의 발전
  - 농기계 관련 중고시장은 급격하게 성장하고 있음. 하지만 자동차와 다르게 운행, 사고, 수리 등의 정보를 정확하게 확인할 수 없음. 하지만 블록체인 기반 농기계 운영 데이터 관리시스템을 활용하면 투명한 중고거래 시스템과 운행 내역을 실시간으로 확인하여 유휴 농기계를 파악하여 공유할 수 있는 4차산업으로 발전할 수 있음

○ 핵심 기술

- 블록체인 기반 플랫폼 기술
  - 블록체인은 데이터 분산 처리 기술의 일종으로, 네트워크에 참여하는 모든 사용자가 모든 거래 내역

등의 데이터를 분산, 저장하는 기술을 지칭. 블록들을 체인 형태로 묶은 형태이기 때문에 블록체인으로 명명함. 블록체인에서 '블록'은 개인과 개인의 거래 (P2P)의 데이터가 기록되는 장부를 의미함. 특히 최근에는 개인과 개인이 아닌 산산업용 블록체인의 형태로 발전하여, 본 과제에서 적용하고자 하는 기관과 기관, 개인과 기관과의 거래 또한 동일한 설계 철학을 적용할 수 있음

- 이러한 블록들은 형성된 후 시간의 흐름에 따라 순차적으로 연결된 사슬(체인)의 구조를 가지게 되며 모든 참여자가 거래내역을 보유하고 있어 거래 내역을 확인할 때는 모든 사용자가 보유한 장부를 대조하고 확인해야 함. 이 때문에 블록체인은 '공공 원장(거래장부)'또는 분산 원장(거래장부)'로 명명  
- 블록체인의 특징은 분산저장을 한다는 것이며, 이는 기존 거래 방식에서 데이터를 위변조하기 위해서 중앙 서버를 공격하거나 해킹하는 상황이 이루어지기 힘들. 블록체인은 여러 명이 데이터를 저장하기 때문에 이를 위변조하기 위해서는 참여자의 거래 데이터를 모두 공격해야하기 때문이며 사실상의 해킹은 불가능에 가까움

- 또한 블록체인은 중앙 관리자가 필요 없다는 특징을 가지고 있음. 은행이나 정부 등 중앙 기관이나 중앙관리자가 필요했던 것은 공식적인 증명, 등기, 인증 등이 필요했기 때문이며 블록체인은 다수가 데이터를 저장, 증명하기 때문에 중앙 관리자나 서버가 필요없어 신뢰도와 구축 운영 비용 절감 측면에서 매우 용이함

- 블록체인 기반 탈중앙화 응용 소프트웨어 개발 기술

- DApp(Decentralized Application)은 탈중앙화된 응용 소프트웨어로 블록체인을 기반으로 한 하이퍼레저, 이더리움, 퀀텀 등 플랫폼의 바탕 위에서 스마트 컨트랙트(스마트 계약)로 작동하는 응용 소프트웨어

- Dapp은 스마트 컨트랙트가 가진 장점을 이용해 네트워크 참여자들 간에 분산네트워크 구조를 만들고, 여기서 실행되는 새로운 방식의 응용 소프트웨어를 만들어보자는 목적으로 시작됨. 각각 비즈니스 모델, 서비스 성격에 적합한 스마트 컨트랙트를 기반으로 만들어지며, 의도에 맞게 다양하고 여러 형태의 계약을 구현할 수 있음

- Dapp의 특징으로는 탈중앙화된 응용 소프트웨어로써 중계자 및 중앙서브를 거치지 않아 투명성이 보장되고, 복잡한 설정이 감소되어 비용이 절감되며, 블록체인 기반인 스마트 컨트랙트로 신뢰성을 확보할 수 있음. 뿐만아니라 대량 정보 유출을 막는 보안성 등의 특징을 가짐

- AI 알고리즘 개발 기술

- 인공 지능(AI)은 학습, 문제 해결, 패턴 인식 등과 같이 주로 인간 지능과 연결된 인지 문제를 해결하는 데 주력하는 컴퓨터 공학 분야로 다양한 방식의 AI 알고리즘이 있지만, 본 개발 기술에서는 주로 ML(Machine Learning)을 적용하고자 함

- ML은 패턴 인식 및 학습에 사용되는 몇 가지 베이지안 기법에 주로 적용되는 이름으로 ML은 기록된 데이터에서 학습하고 이를 기반으로 예측하며, 불확실성 하에서 기본 유틸리티 기능을 최적화하고, 데이터에서 숨겨진 구조를 추출하고, 데이터를 간결한 설명으로 분류할 수 있는 알고리즘의 모음임. 소프트웨어 개발자가 주어진 입력에 따라 프로그램 코드별로 출력을 생성하기 위해 개발하는 일반 컴퓨터 코드와는 달리, ML은 데이터를 사용하여 통계 코드(ML 모델)를 생성. 이 통계 코드는 이전의 입력 예제에서 인식한 패턴을 기반으로 '적절한 결과'를 출력함

- 본 개발 기술에서는 농기계의 이력 / 운행 데이터를 기반으로 ML 모델의 기록 데이터를 입력하고, 이를 통해 농기계의 작업 패턴을 세분화하여 파악하거나, 농기계의 실 가동률이나 유휴 농기계를 추적하는 예상 결과를 출력하고자 함. ML 모델의 정확성은 대부분 기록 데이터의 양과 질에 달려 있으므로, 이력 / 운행 데이터 수집 시스템에서 양질의 데이터를 추출할 수 있게 설계를 반영하며, 사용자 서비스를 통해 추가 적인 데이터를 취득하여 ML 모델 강화 시킬 계획임

- 농기계용 텔레메틱스 단말기 & 서비스 기술

- 농기계용 텔레메틱스 서비스 기술은 농기계에 장착된 텔레메틱스 단말 장치를 통해 농기계의 위치와 각종 운행 정보를 수집 가공하여 사용자에게 서비스하는 기술로 선진국에서 시작되어 세

계적인 커넥티드 농기계 흐름에 핵심 기술로 자리 잡음

- 단말 장치는 기본적으로 농기계의 위치 파악을 위한 GPS기능과 이동통신 기능을 포함하고 있으며, 농기계의 데이터를 취득하기 위한 CAN통신 및 이더넷 통신을 지원함
- 수집 및 서비스 서버는 각 농기계 단말에서 수집한 정보를 가공하고 분석하여 사용자의 수요에 맞는 '서비스'형태를 제공하는 역할을 수행함
- 특히, 단위 기술이 아닌 '서비스'의 형태로 기술이 개발되므로 일회성 개발이 아닌 시장과 호흡을 같이하며 지속적인 연구개발 업데이트가 이루어져야 하는 시장 지배 기술.
- 텔레매틱스 서비스 기술은 크게 농업 생산성과 사용자 편의성을 중점으로 지금도 발전하고 있으며, 미국과 일본의 경우 농업 생산량 및 활용 증대에 필수가 되어가고 있음
- 뿐만 아니라 농기계 텔레매틱스의 특징은 차량 진단/고장 예지, 농작물 관리에 적용되며 특히 자율주행 농기계에 핵심 기술임

나. 국내 기술 수준 및 시장 현황

○ 기술현황

- 농업과 ICT를 결합하여 디지털 농업, 지능형 농업을 구현하는 스마트 농업은 전세계적인 추세임. 농업에서는 4차 산업혁명 핵심기술인 사물인터넷, 빅데이터, 클라우드 컴퓨팅, 인공지능, 로봇을 이용하여 정밀농업의 외연이 확대되고 기술적으로 고도화되는 농업의 최적화가 가속화될 것으로 예측되고 있음



<한국농업기술의 변천사>

[출처 : 4차 산업혁명시대 지능정보기술동향과 농업 R&D 추진방향, 농림식품기술기획평가원 2017]

- 정부에서는 1975년부터 농업 기계 보유 현황 조사 분석을 시작하였으며 각 지자체의 읍면동 담당자가 농림정보시스템으로 주요 농업기계8종 및 폐농업기계 6종 보유 대수 전수조사하여 통계에 활용함. 그러나 농업기계 조사대상이 확대시 조사 기간 장기화와 많은 인력이 필요한 상태이나 수동으로 입력을 해야하여 조사의 한계가 존재함



[농기계 보유현황 통계 수집 방법 2016, 통계청]

- 최근 텔레매틱스를 이용하여 농기계의 작업 이력을 관리할 수 있는 연구가 활발히 이루어지고 있음
- 2018년 그린맥스는 사전고장진단 모니터링시스템을 개발하여 로터베이터의 기름잔량이나 문제발생 시 앱을 통해 알람을 제공함
- 2018년~2020년 대동공업, 동양물산, 긴트는 농기계전용 텔레매틱스를 개발하여 장비-사용자-생산자 간 차량 정보 및 작업 지령 상태 정보를 교환할 수 있게 개발중임
- 2020년~2023년 동양물산은 자율주행형 트랙터를 위한 고정밀 위치 정보 모듈을 개발하여 트랙터 위치를 실시간으로 모니터링 할 수 있게 개발중
- 블록체인은 단순한 디지털 화폐의 형태에서 스마트 컨트랙트를 활용한 2세대 블록체인을 거쳐 속도·확장성·상호운용성에 중점을 둔 3세대로 진화되고 있음

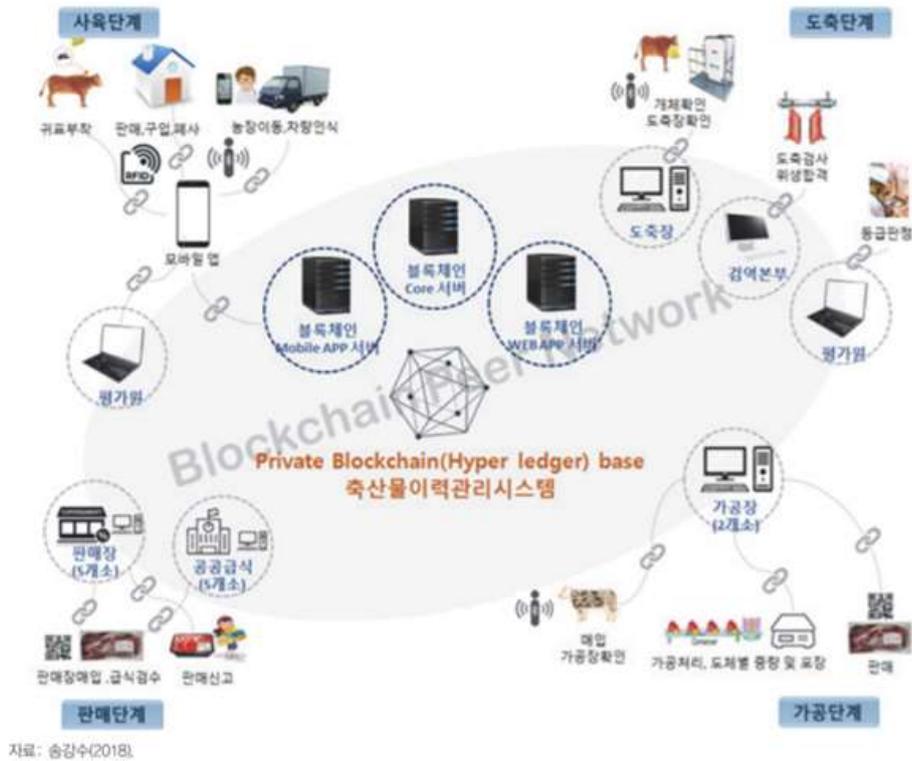
세대	내용
1세대 블록체인	P2P기반 분산원장 등과 같은 송금 및 암호화폐 중심의 기술·서비스
2세대 블록체인	분산원장 기술의 고도화를 통한 스마트 컨트랙트(Smart Contract) 기반의 자동 지급 결제 시스템
3세대 블록체인	다양한 유·무형 자산의 공유·유통·거래 플랫폼을 기반으로 스마트 컨트랙트 고도화를 통한 타 산업과의 융합 발전 및 활용분야 확대

- 대기업은 core business기반 성장동력을 확보하기 위해 새로운 비즈니스 탐색 및 블록체인을 변화요소로 활용하고 있음

세대	기업	변화된 비즈니스	블록체인 활용사례	변화 방향성
대기업 SI사		클라우드 기반의 기업용 블록체인 플랫폼 사업	관세청 블록체인 기반 수출통관 물류서비스 플랫폼	IT솔루션 중심 서비스에서 요소 기술 중심의 서비스로 확장 클라우드 기반 기업용 블록체인 플랫폼 사업 진출
			조폐공사 블록체인 오픈 플랫폼 구축사업	
제조사		DApp 개발 및 모바일 디바이스 탑재	갤럭시 10에 탑재된 삼성 블록체인 월렛 신규 DApp 10종 탑재	스마트폰 시장 중심으로 블록체인 플랫폼 생태계 구축 얼라이언스 참여를 통한 타산업으로 확장
			그라운드X 블록체인 플랫폼 클레이튼 참여	
통신사	 	기업용 블록체인 플랫폼 사업을 확장	DID 사업 통합 전자 증명앱 개발 - SKT, KT, LG유플러스, 삼성전자, 하나은행, 우리은행, 코스콤 참여	플랫폼 레벨의 기업용 블록체인 플랫폼 사업으로 확장 어플리케이션 레벨의 블록체인 결합 서비스로 확장
				

- 최근 농림축산식품부는 ‘블록체인 기반 축산물 이력 관리시스템’을 통해 사육·도축·포장·판매 등 단계별로 단절되어 있는 이력정보들을 블록체인으로 투명하게 공휴함으로써 기존 5일 이내 신고대상 정보뿐만 아니라 더욱 세분된 정보(소 개체 단위, 실물 포장단위)를 실시간으로 수집·저장하여 문제

발생 시 유통과정의 추적시간을 10분 이내로 단축할 수 있음



[블록체인 기반 축산물 이력 관리 시스템]

○ 시장현황

- 최근 7년간 농업기계 내수시장(정부용자기준)은 연 평균 8,937억 원 규모로 12년 기준으로 18년은 83.7%로 연 평균 3.26% 하락세이며, 추세는 아래 표와 같음
- 국내 트랙터 시장에서 교체 수요 수준으로 다소 침체되어 있던 시장이 내구년한에 의한 새로운 기계의 도입과 최근 환경규제에 의한 다량의 신제품 판매 및 구매 효과로 상승세로 돌아설 것으로 전망

표 최근 내수시장 농업기계 시장규모 (구성비:2012=100%)

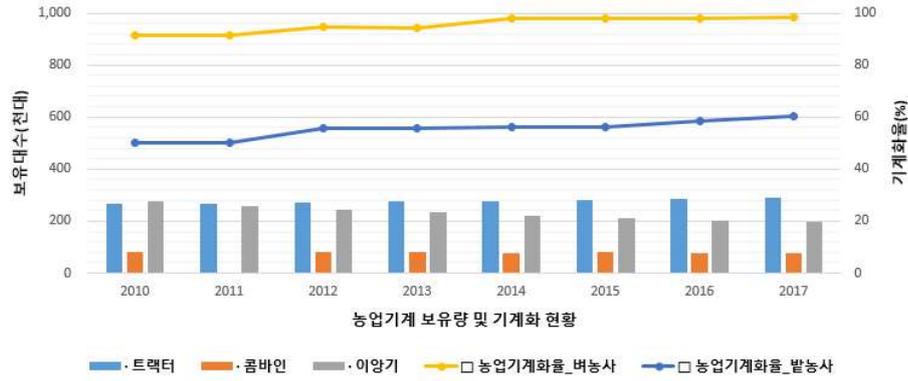
(단위 : 억 KRW, %)

구분	2012	2013	2014	2015	2016	2017	2018	평균
시장규모	9,118	9,252	9,004	10,420	9,846	7,286	7,632	8,937
구성비	100.0	101.5	98.7	114.3	108.0	79.9	83.7	98.0

[출처: 농림식품부, 2017]

- 국내 벼농사 기계화율은 최근 30년(1987~2017) 동안 연 평균 2% 성장하여 2017년 98.4%로 완료 상태로 벼농사 노동 투하시간은 86.3% 단축하였고, 미국 생산량은 123.6% 증가
- 반면, 밭농사 기계화율은 2017년 60.2%로 벼농사 기계화율에 비해 미진한 실정
- 농업기계 중 트랙터 비중은 42.6%로 가장 높으며, 트랙터 보유수는 2010년부터 연평균 1.3% 성장하여 2017년 29만대로 2010년 대비 8.6% 상승

### 농업기계 보유량 및 기계화 현황



[국내 농업기계 보유량 및 기계화 현황[출처: 농림축산식품부]

- 국내 블록체인 시장은 '22년까지 연평균 약61.5% 성장하며 3,500억 수준에 도달할 것으로 전망  
국내 블록체인 시장규모

(단위 : 억 KRW, %)

구분	2016	2017	2018	2019	2020	2021	2022
시장규모	201	324	524	846	1366	2206	3562

#### ○ 경쟁기관현황

- 국내 경쟁 기관 중 블록체인을 이용하여 농기계이력관리를 제공하는 회사는 없음
- 2019.06 LG유플러스에서 미국의 PTC(증강현실 및 IOT솔루션 업체)와 연계하여 '트랙터 원격진단' 서비스를 개발 시작

경쟁기관	국가	보유시스템 그림 및 명칭	주요기능	특징
 LG유플러스	대한민국		트랙터 원격진단	증강현실 원격진단

#### ○ 지식재산권현황

- 한국특허정보원의 특허 데이터베이스를 활용하였고, 농기계 작업관리와 이력관리의 키워드 검색을 2000년도부터 2019년까지 등록된 특허를 분석
- 기술분류 체계에 따른 최종 검색식을 다음과 같이 한 이후, 추출한 유효데이터는 다음과 같음
- 농기계 작업관리 및 이력관리 특허 중 국내 특허 298건 중 일본의 종합형 농기계 업체(구보다, 안마)가 국내 특허 약 16건을 보유하고 있음
- 특허의 내용은 대부분 운행이력 관리와 전송 및 저장장치가 대부분을 이루고 있으며 나머지 관리 시스템과 부품 관련으로 등록된 것으로 분석됨
- 주요 특허로는 2015년 등록된 일본 구보다사의 농작관리시스템 및 농작물 수확기(출원번호에 관한 특허와 대지작업차량, 대지작업차량 관리 시스템으로 등록된 특허가 있음
- 국내에서 동양정밀, 아세아테크, 국제 등이 관련 특허를 등록하고 있음

대분류	검색식	검색 건수		
		한국	일본	합계
농기계작업관리	((농기계*작업관리)+(농기계*이력관리) (farming*machine*management))*PD=[200 00101~20191231])	284	14	298

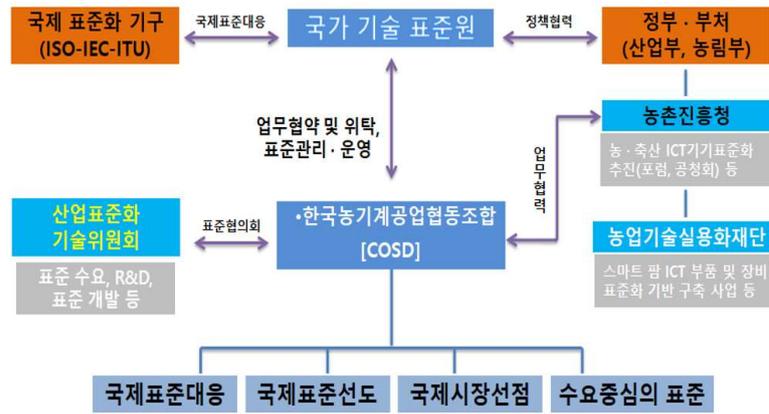
- 한국특허정보원의 특허 데이터베이스를 활용하였고, ‘블록체인’과 ‘Block Chain’의 키워드 검색을 2015년도부터 2019년까지 등록된 특허를 분석
- 기술분류 체계에 따른 최종 검색식을 다음과 같이 한 이후, 추출한 유효데이터는 다음과 같음
- 특허의 내용은 대부분 블록체인의 관리방법 및 연결관리가 대부분을 이루고 있으며 나머지 검증 프로세스 관련으로 등록된 것으로 분석됨

대분류	검색식	검색 건수
		국내특허
블록체인	((블록체인)+(Block Chain) (farming*machine*management))*PD=[201 50101~20191231])	13,765

○ 표준화현황

- 2019년 한국농기계공업협동조합은 농업기계 관리 관련하여 국제 표준을 제정하였음

표준번호	표준명	비고
KS B ISO 11783-1	농림업용 트랙터 및 기계 — 직렬 제어 및 통신 데이터 네트워크 — 제1부: 모바일 데이터 통신표준	2019 제정
KS B ISO 12188-1	농림업용 트랙터 및 기계 — 농업용 위치인식 및 유도장치에 대한 시험 절차 — 제 1부: 위성기반 위치인식장치의 동적 시험)	2019 제정
KS B ISO 12188-2	농림업용 트랙터 및 기계 — 농업용 위치인식 및 유도장치에 대한 시험 절차 — 제 2부: 직진 및 수평 이동 중 위성기반의 자동유도장치 시험	2019 제정



- 정부 부처는 스마트팜 로드맵에 따라 포럼 혹은 단체 표준 기구를 운영하고 있음

관련 부처	포럼 혹은 단체 표준 기구	주요 활동
 <b>농림축산식품부</b>	스마트팜 ICT융합표준화포럼	2017년, 신규 설립 이후, 시설원에 센서/구동기 인터페이스 표준을 국가 표준으로 추진 중
 <b>과학기술정보통신부</b>	스마트농업 ICT융복합표준포럼	2014년에 신규 설립 이후, 2015년부터 시설원에, 유통 및 축산 분과 운영을 통해 시설원에 인터페이스와 관련된 표준 개발
	TTA 스마트농업 PG	시설원에 장비와 연동을 위한, 유스케이스를 개발한 바 있으며, 스마트팜 서비스 프레임워크 및 시그널링에 관한 표준을 개발한 바 있음

- AI를 이용한 농기계 이력데이터 분석 및 활용 절차나 블록체인을 이용한 생애주기별 이력데이터 절차에 관한 표준 재정 활동은 없음

○ 기타현황

- 본 기술과 관련 있는 수행 기관 표준화 활동 현황

참여기관	활동 현황
한국농기계공업협동조합	<ul style="list-style-type: none"> <li>• 한국농기계공업협동조합은 2009년부터 표준개발협력기관(COSD)으로 지정되어 농기계류 (TC23)와 관련 된 국가표준을 제·개정 관리하고 있음</li> <li>• 현재 농기계·자재 분야의 표준은 190종으로 산업계 수요를 반영하여 표준화 관련 활동을 수행하고 있음</li> <li>• 농기계조합 최근 3년간 표준화 활동 실적</li> </ul>

참여기관	활동 현황					
	구 분	2017	2018	2019	2020(계획)	비 고
	고유표준 제정	-	2	1	2	
	고유표준 개정	1	3	-	3	
	부합화 제정	2	2	9	10	
	부합화 개정	12	7	9	6	
	표준확인	45	11	41	21	
	표준폐지	1	-	5	-	
	합 계	60	25	65	42	

- 2019년에는 농업용기계 관리개발 관련 국제표준 4종을 국가표준으로 제정하였으며 관련표준을 지속 개발 예정에 있음
- 한국농기계공업협동조합의 국제표준화 활동으로는 ISO/TC23분야에 전문가로 등록되어 활동하고 있으며 국제표준 2종을 개정 제안하여 ISO 개정 발간 완료 하였고, 현재 국제표준 2종을 서울대학교와 함께 제정하는데 참여하고 있음

다. 국외 기술 수준 및 시장 현황

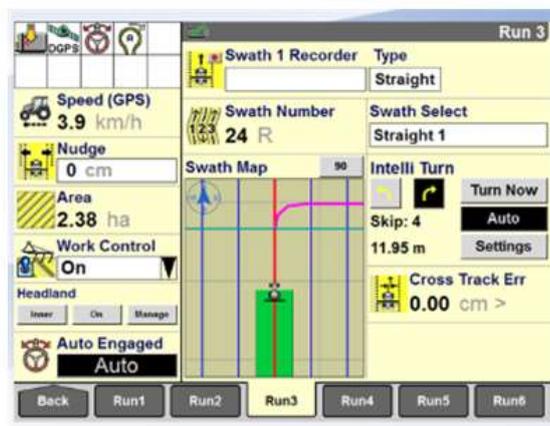
○ 기술현황

- 해외에서는 트랙터 관련 글로벌 대기업이 기술개발을 주도하고 있음
- 미국의 John deere社は JDLink 시스템을 개발하고 운영하며 시장을 선도하고있음. JDLink 시스템은 농기계의 위치정보, 운행지역과 운행시간의 제한 서비스 그리고 관제센터로 통보되는 DTCs(Diagnstic Trouble Codes)를 이용한 차량진단 서비스를 제공. 이벤트 우선 순위를 색으로 구분하여 스마트폰으로 통보하는 알림 서비스와 정기점검 및 소모품 관리 서비스 그리고 연료 소비량의 분석 서비스를 제공. 또한 텔레매틱스 시스템과 연계하여 수확량, 작업량, 원격 고장 진단 등의 데이터 모니터링 기술이 접목된 정밀농업 솔루션(AMS, Agricultural Management Solutions)을 제공함



[ewholland社 상품 구성 및 서비스에서 : John deere 홈페이지]

- 네덜란드의 New Holland는 PLM(Precision Land Management) Connect Essential과 Professional 패키지를 이용하여 농기계를 관리하며, 다양한 센서를 부착하여 생산성 향상을 지원. 연료 사용량 및 운행 정보가 성능분석 리포트로 변화되어 생산성과 효율성을 개선. 웹페이지를 통하여 농기계의 현재 위치와 동작 상황을 확인할 수 있음. 설정된 지역을 벗어나거나 설정된 시간 이외에 작동하면 관리자에게 실시간으로 통보



[ewholland社 상품 구성 및 서비스예시 : Newholland 홈페이지]

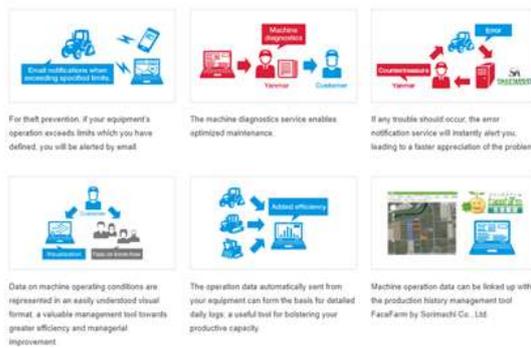
- 일본의 KUBOTA社は 2017년도에 KSAS(Kubota Smart Agri System) 솔루션을 소개 하였음. KSAS는 기본서비스와 전문서비스로 제공되고 있으며 기본서비스는 농기계와 연동하여 운행정보, 상태관리와 이력관리를 지원함. 유료서비스로 제공되는 전문서비스는 모든 농사에 활용할 수 있는 농장의 관리, 작업계획과 지시 그리고 작업일지와 같은 농업 경영의 가시화로 효율성을 높이고 생산성 향상을 지원. KSAS에 추가로 센서가 연동되는 경우는 추수 시에 쌀 수확량과 맛 그리고 수분 함량의 측정이 가능. 또한 작물의 수분 함량에 따라 선별 건조하여 비용 절감할 수 있음



[Kubota KSAS 서비스 구성 및 기능 출처 : Kubota 홈페이지]

- 일본의 Yanmar社는 GPS안테나 및 통신 단말기를 탑재한 농기계의 작업 시간, 주행거리, 연료 소비율, 연비 소비량, 엔진 회전수, 동력 부하율 등의 가동 상황을 시각화하여 관리할 수 있는 Smart Assist를 제공함. Smart Assist는 실시간으로 농기계의 오류, 위치를 확인할 수 있는 서비스, 그리고 3개월마다 기계의 수리 기록을 제출하는 진단 서비스를 제공

SmartAssist Features and Main Technologies

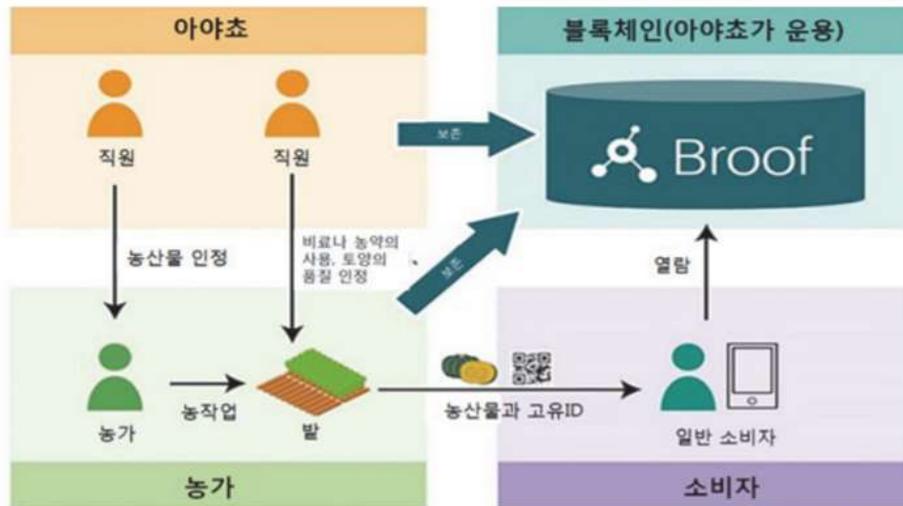


[Yanmar SmartAssist 서비스 구성 및 기능 출처 : Yanmar 홈페이지]

- 이탈리아의 Bolzano 대학은 트랙터 엔진의 효율성에 대한 프로젝트인 TRATnet. one을 수행함. 프로젝트의 목표는 센서들의 출력을 정확하게 분석 할 수 있는 추론 소프트웨어-엔진 기반의 자동화 시스템. 원격에서 엔진을 모니터링 하는 방법들을 정의하고 개발하기 위해 엔진회전수, 배출되는 가스의 온도와 산소의 양을 측정 및 분석
- 미국 주정부에서는 헬스케어 이외의 산업에 대해서도 블록체인 기술 도입 진행

미국 주 또는 시	내용
뉴욕시	이더리움 컨소시엄 EEA에가입
버먼트 州	합법적 기록 관리 방법 연구 및 주 증거법 따라 블록체인 디지털 기록 이전
델라웨어 州	주식거래명부에 블록체인 활용, 국가기록물보관소 기록물에 블록체인을 활용
일로노이 州	생년월일을 블록체인으로 기록하는 실증 추진
조지아 州	부동산 거래 부문의 소유권 등록·인증 및 계약의 변조방지 등 부정부패 방지
메릴랜드 州	블록체인 기술을 이용한 네트워크와 데이터베이스 회사 주식장부 등 회사기록을 관리할 수 있도록 법 개정
텍사스 州	자유당은 대선 후보 선정에 블록체인 기술을 활용한 온라인 투표를 도입

- 덴쯔 국제정보서비스12)(電通國際情報サービス, ISID)와 미국 GE가 합작하여 설립된 이노랩은 2016년 10월부터 일본 미야자키현(宮崎県) 내 기초 지방자치단체인 아야쵸(綾町)와 제휴하여 블록체인 기술을 활용한 유기 농산물 품질 검증을 시작함



[블록체인을 활용한 덴쯔 국제정보서비스의 유기 농산물 검증시스템]

- 해외에서는 농업 부문에서 블록체인 기술을 아래와 같이 사용하고 있음
- 식품 공급망 최적화 - 식품 원산지 추적 시간을 몇 초로 단축하여 안전성을 보장하고 효율성향상
- 농작물 보험 -화물, 지리적 웨이 포인트 및 기본 준수 정보를 운송 업체와 통신하고 제품의 품질, 가격, 위치 및 관련 당사자를 등록
- 거래 - 농민이 적정 가격으로 상품을 판매하고 거래 수수료를 낮추어 소규모 농민이 시장에 진입 할 수 있도록 지원
- 추적 성 - 현명한 계약은 농부의 농작물을 보장하고 손해 배상을 청구하여 오래되고 부담스러운 보험 프로세스를 대체하며 최대 몇 달이 걸릴 수 있음

○ 시장현황

- 세계 각국은 식량 부족 등 식량증산 정책을 추진하고 있으며, 농기계 시장도 매년 확장되고 있음. 세계 농기계시장의 규모는 2007년 937억 달러에서 2017년 1,285억 달러로 연평균 6.5% 성장하였으며, 2017년 1,285억 달러에서 2027년 2,060억 달러로 연평균 5.2% 성장할 전망
- 아시아/태평양지역에서는 중국이 2007년 97억 달러에서 2017년 277억 달러로 연평균 성장률이 23.4%로 급성장을 이뤘으며, 2017년부터 2020년 연평균 성장률도 5.8%로 지속적인 상승을 보여 줄 것으로 전망
- 농업인구 비율이 높은 인도는 2007년 기준으로 71억 달러에서 10년 후인 2017년에는 96억 달러로 6.2%성장하였으며, 2017년~2022년 연평균 성장률은 7.1%로 중국과 함께 전 세계에서 가장 큰 농기계 시장을 형성할 것으로 예상
- 또한 중국과 인도의 성장에 따라 베트남, 미얀마와 같은 개발도상국의 농기계시장 역시 성장 예상

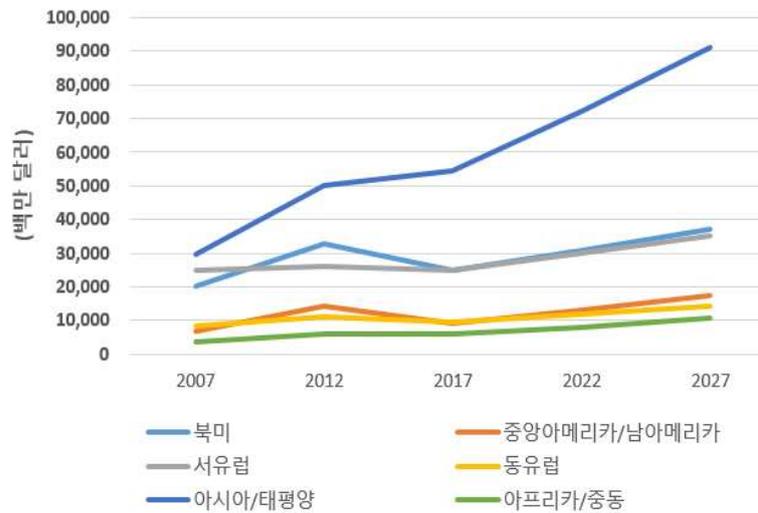
[세계 농기계 수요(권역별)]

(단위 : 백만 달러)

항목	2007	2012	2017	2022	2027	연증가율 ('17/07')	연증가율 ('17/22')
세계 농기계 수요	93,700	140,200	128,500	165,800	206,000	6.5%	5.2%
북미	20,150	32,950	24,700	30,900	36,950	4.2%	4.6%
미국	17,250	28,500	20,950	26,400	31,600	4.0%	4.7%
캐나다/멕시코	2,900	4,450	3,750	4,500	5,350	5.3%	3.7%
중앙/남아메리카	6,785	14,100	9,150	12,950	17,500	6.2%	7.2%
서유럽	24,900	25,950	24,850	30,000	35,300	0.0%	3.8%
동유럽	8,380	11,100	9,490	11,800	14,200	2.5%	4.5%
아시아/태평양	29,800	50,250	54,500	72,050	91,150	12.8%	5.7%
중국	9,700	22,800	27,750	36,850	46,450	23.4%	5.8%
인도	7,110	9,240	9,600	13,500	18,000	6.2%	7.1%
기타	12,990	18,210	17,150	21,700	26,700	5.7%	4.8%
아프리카/중동	3,685	5,850	5,810	8,100	10,900	9.5%	6.9%

[출처: The Freedonia, 2019]

세계 농기계 수요



[세계 농기계 수요]

- 농업기계 전체시장에서 가장 많이 사용되는 트랙터 및 수확기가 큰 비중을 차지하며 특히 다양한 환경에서 다용도로 사용할 수 있는 트랙터는 전체 시장의 35% 이상을 차지하고 있는 대표기종
- 세계 농기계 수요와 함께 2017년 453억 달러에서 2022년 591억 달러로 연간 평균 성장률은 5.5%로 전망

[기종별 세계 농기계 시장 수요]

(단위: 백만 달러)

항목	2007	2012	2017	2022	2027
세계 농기계 수요	93,700	140,200	128,500	165,800	206,000
농용 트랙터	33,180	49,810	45,300	59,100	73,850
수확기	15,500	24,000	21,350	27,900	34,900
이앙기 및 비료살포기	6,330	10,520	9,340	12,500	15,850
건초제조기	6,050	8,770	7,630	9,910	12,350
축산기기	5,260	8,465	7,630	10,350	13,400
경운기	4,480	6,995	6,350	8,500	11,000
기타농업기계	4,200	5,980	5,050	6,640	8,250
농기계관련부품	18,700	25,660	25,850	30,900	36,400

[출처: The Freedonia, 2019]

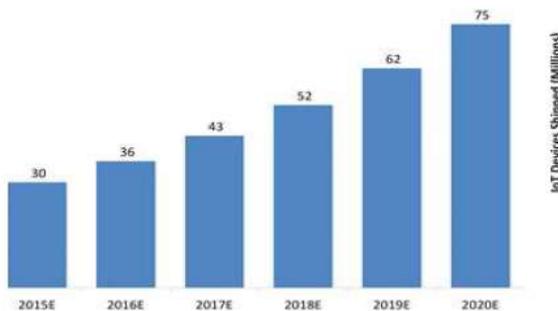
- 보고서에 따르면 현재는 북미 시장이 가장 큰 시장을 형성하고 있으며, 점차 호주, 중국, 인도, 일본, 한국과 같은 아시아-태평양 지역으로 시장 동력이 옮겨갈 것으로 예상
- 최근 ICBM(IoT, Cloud, Big Data, Mobile), 딥러닝, 머신러닝, 드론, 로봇릭스 등이 급격하게 발전하면서 정밀농업은 기술적 한계에서 벗어나 본격적인 성장 시작

관찰	처방	농작업	결과분석
농경지의 토양, 작물, 수확량 상태를 조사하여 기초정보를 만들어 내는 단계	센서기술로 얻은 정보를 기반으로 농약과 비료의 알맞은 양을 결정하는 단계	최적으로 판단된 정보에 따라 필요한 양의 농자재와 비료를 투입하는 단계	산출된 양을 기존 수확량과 비교, 시비방법 등의 적절성을 확인한 후 수정보완하여 데이터 축적
센서, 토양지도, 인공위성, 드론 등	GPS, GIS, 빅데이터, 클라우드, 인공지능 등	드론, 로봇릭스, 모바일기기, VRT 등	빅데이터, 클라우드, 인공지능 등

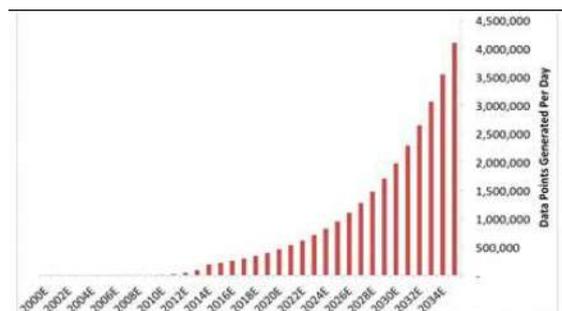
\* GIS(Geographic Information System): 지리정보시스템  
 \* VRT(Variable Rate Technology): 변량살포기술

[정밀농업 단계별 적용 ICT 기술]

- 농업용 IoT 장비에 대한 수요는 2016년 3,600만 달러에서 2020년 7,500만 달러규모로 늘어나고, 농장에서 1일 생산되는 데이터 포인트는 2014년 19만 개에서 2050년 50만개로 급격하게 증가할 것으로 전망



\* (출처) BI Intelligence Estimates, 2015



\* (출처) On Farm, Intelligence Estimates, 2015

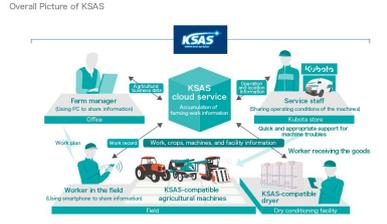
[농업용 IoT 디바이스 출하량 전망]

[농장에서 생산되는 1일 데이터량 전망]

- 농업 시장에서 블록 체인 혁신의 규모는 2017 년 약 4120 만 달러에서 2023 년까지 거의 4 억 6 천만 달러로 증가하여 47.8 %의 복합 연간 성장률 (CAGR)을 나타낼 것으로 예상

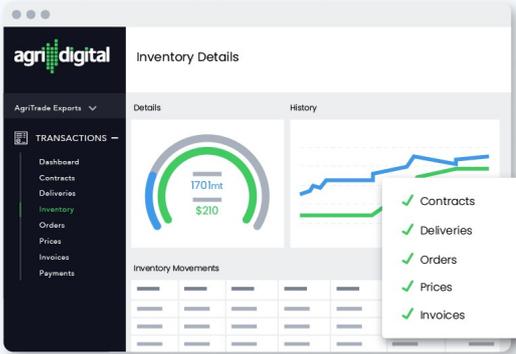
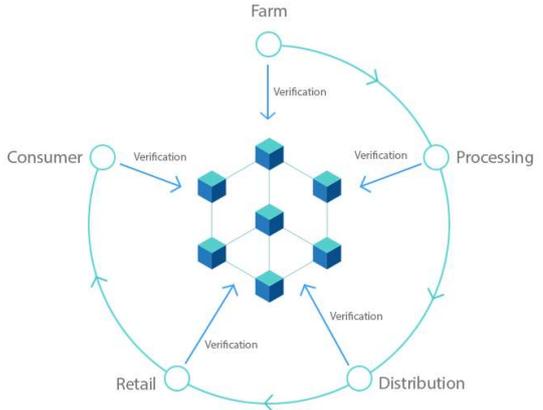
○ 경쟁기관현황

- 해외의 경쟁기업은 거의 유사한 이력관리 서비스를 제공하고 있음
- 하지만 블록체인을 적용하지 않아 데이터신뢰성과 안정성의 문제가 있음

경쟁기관	국가	보유시스템 그림 및 명칭	주요기능	특징
 John Deere	미국	 JDLink	운행이력 관리, 장비관리, 도난방지 등	통합농업관리, 세계최고의 기술력
 New Holland	네덜란드	 Precision Lnad Management	운행이력 관리, 장비관리, 도난방지 등	다양한 외부 디스플레이 제공
 Kobota	일본	 Kubota Smart Agri System	운행이력 관리, 장비관리,	지리정보와 연동한 농장 관리시스템
 Yanmar	일본	 Smart Assist	운행이력 관리, 장비관리, 부품A/S 리포트 제공 등	A/S리포트별 도제공

[농기계 관리 시스템 경쟁기관]

경쟁기관	내용	서비스 그림
	P2P 농업 거래 및 처리에 중점을 둔 블록 체인 회사.	

	<p>글로벌 곡물 산업을 위한 블록체인 기반의 통합 상품 관리 솔루션. 이 플랫폼은 스마트 계약을 통해 복잡한 농업 거래를 처리하는 데 도움을 줌.</p>	
	<p>식량 원산지 추적, 금융 지원 및 거래 데이터 저장에 있어 농민을 지원하는 영국의 사회적 기업 프로젝트.</p>	
	<p>가축, 운송 및 신선 식품 포장에 식별 도구를 적용하여 전체 공급망에서 품목을 추적</p>	

[농업 산업을 변화시키는 블록체인 스타트업]

○ 지식재산권현황

- 한국특허정보원의 특허 데이터베이스를 활용하였고, 농기계 작업관리와 이력관리의 키워드 검색을 2000년도부터 2019년까지 등록된 특허를 분석
- 기술분류 체계에 따른 최종 검색식을 다음과 같이 한 이후, 추출한 유효데이터는 다음과 같음
- 2008년부터 2018년까지 특허출원 증가율 분석에 따르면, 이전 5년(2009.12 ~ 2014.12)에 비하여 최근 5년구간(2015.1 ~ 2019.12)에서 증가율이 577%로 대폭 증가하였음
- 특히 한국 및 일본의 특허출원 증가율이 1,000%이상으로 이들 국가에서 본격적으로 운행이력에 관련된 연구를 진행한 것으로 보임
- 일본은 Yanmar 사가 172건으로 가장 활발한 연구를 진행하였으며, 미국 및 유럽 국가에서는 John Deere가 22건으로, 일본과 미국은 이미 기술 성숙도에 도달한 것으로 판단됨
- 주요 장벽 특허들은 운행 내역을 수집하고 전송하는 하드웨어와 소프트웨어에 대한 기초적인 특허로 분석되며, 본 연구의 핵심요소인 블록체인을 이용하여 데이터 공유 및 플랫폼을 구성한다는 차이가 있다는 점에서 기술적 차별점이 높은 것으로 분석되어 지적재산권(IP, Intellectual Property) 장벽도는 낮은 정도로 판단되었음

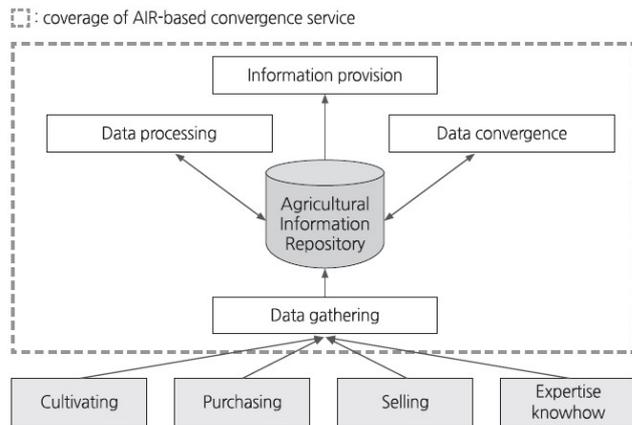
대분류	검색식	검색 건수				
		한국 KIPO	미국 USPTO	일본 JPO	유럽 EPO	합계
농기계작업관리	((농기계*작업관리)+(농기계*이력관리) (farming*machine*management))*PD=[20000101~20191231])	402	37,359	102	3,013	48,076

- 한국특허정보원의 해외 특허 데이터베이스를 활용하였고, Block Chain'의 키워드 검색을 2015년도부터 2019년까지 등록된 특허를 분석
- 기술분류 체계에 따른 최종 검색식을 다음과 같이 한 이후, 추출한 유효데이터는 다음과 같음
- 특허의 내용은 국내와 동일하게 대부분 블록체인의 관리방법 및 연결관리가 대부분을 이루고 있다. 그리고 각 산업 별 적용 관련 내용을 다루고 있음

대분류	검색식	검색 건수				
		미국 USPTO	일본 JPO	유럽 EPO	중국 CPO	합계
Block Chain	block*chain*PD=[20150101~20191231]	252,039	62,413	70,797	16,757	402,006

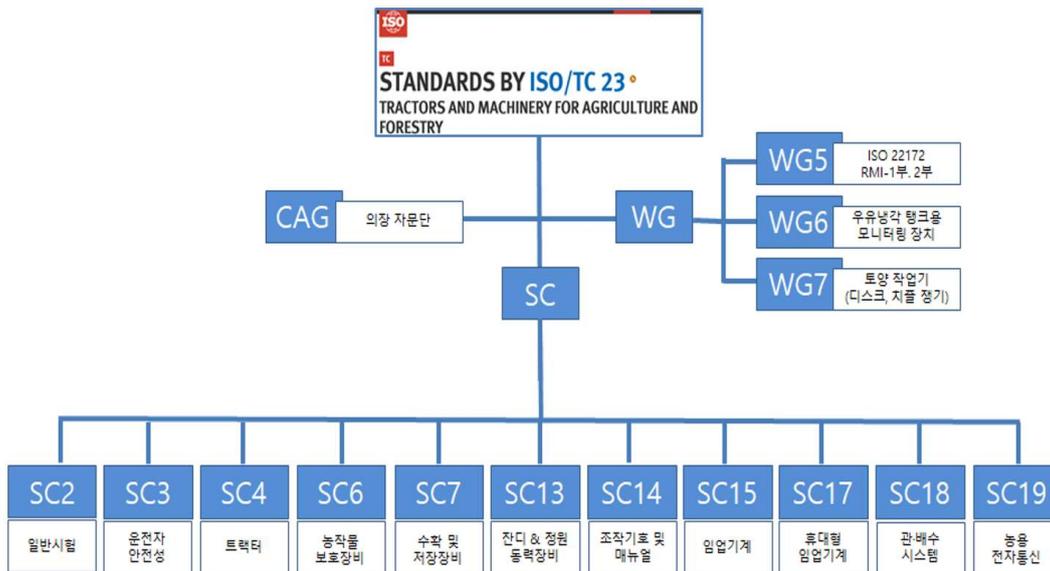
○ 표준화현황

- ITU-T에서 진행중인 스마트팜 관련 국제표준화 작업은 SG13과 SG20에서 진행하고 있음SG13은 스마트농업의 개요에 대한 표준인 Y.4450/Y.2238을 발간한 후, 융합서비스 관점에서 Y.smp, Y.farms, Y.saic, Y.sfes 초안권고안 작업을 추진하고 있으며, SG20은 IoT 응용의 관점에서 Y.ISG-fr(스마트 온실), Y.IoT-SLFG(스마트 축산) 두 문서가 권고안 작업을 추진 중임
- ITU-T SG13의 농업정보기반 융합서비스의 서비스 모델(Y.saic)



[농업정보저장소 기반 융합 서비스 개념]

- UN 산하 기구 국제 표준화 기구 ISO/TC23에서는 농기계에 관한 표준을 담당하고 있음.



[ISO/TC23 조직도]

SC	제목	내용	국제 표준
SC3	농기계 운전자 안전성	농기계 안전성과 운전자 편의성을 담당하는 SC로, 트랙터 부착 작업기 안전에 대한 표준이 다수를 차지하고 있음 무인자율주행 트랙터의 기초가 되는 '운전자가 제어하는 트랙터의 자동안내시스템의 안전요구사항, 트랙터-고전압 전기 및 전자 부품의 시스템 안전성 등 첨단 농기계로 가는 과정의 표준도 일부 제정 됨	ISO18497 - Highly automated machines
SC19	농기계 전자통신	주요 표준은 직렬제어 및 통신 데이터 네트워킹과 관련하여 데이터 링크 계층, 네트워크 관리, 트랙터 ECU, 진단서비스, 환경 조건에 대한 내성시험 등이 개정작업에 있으며, 고전압인터페이스, 제어시스템을 위한 안전요구조건 등이 제정 작업이 진행되고 있음	ISO12188 - 무선통신을 이용하여 농기계의 위치를 인식·결정하고 안내하는 시스템 관련 표준

[ISO/TC23 농기계 관리 관련 SC]

- ISO/TC 307 (Blockchain and distributed ledger technologies)은 2016년 9월 설립되었고, 전 세계에서 가장 활발히 블록체인 및 분산원장 기술 국제 표준을 개발하는 국제표준화기구의 기술 위원회임

담당분과	문서	내용
ISO/TC 307/WG 1	ISO/CD 22739	블록체인 및 분산원장 기술의 용어를 정의하는 표준안. 블록(block), 블록체인(blockchain), 계정(account), 주소(address), 합의(consensus), 포크(fork), 스마트 컨트랙트(smart contract), 노드(node)등 블록체인 및 분산원장 기술 표준화에 필수적인 용어들을 명시
	ISO/CD 23257	블록체인 및 분산원장 기술의 참조모델을 정의하는 표준안. ISO/CD22739이 정의하지 않은 기타 용어들을 정의하고, 블록체인 및 분산원장 기술의 개념을 기술. 나아가 분산원장 시스템상의 합의(consensus), 스마트 컨트랙트(smart contracts) 및 보안(security) 등의 개념 및 분산원장 시스템의 종류도 소개. 마지막으로,블록체인

		및 분산원장 시스템의 주요 특성을 정의하고, 주요 기능적 요소 (functional components)들을 분산원장 기술, API, 사용자 기반의 프레임워크를 정의
	ISO/AWI TS 23258	블록체인 및 분산원장기술 주제별 분류를 명시하는 기술규격. 개념, 블록체인 및 분산원장 기술 및 유즈케이스에 대한 분류 및 특성 (attributes) 및 클래스(classes) 관련 온톨로지도 포함
	ISO/NP TR 23578	블록체인 및 분산원장기술에서 상호운용성 관련한 발견 이슈에 대한 주요 기술적 아이템들을 정리하고 이를 기반으로 표준화 아이템 정리
ISO/TC 307/WG 2	ISO/NP TR 23244	블록체인 및 분산원장기술을 위한 개인식별정보(PII) 및 개인정보보호를 설명하는 기술보고서. 분산원장 기술상 개인정보 프레임워크를 제시하고, 분산원장 및 블록체인의 개인정보 관리를 위한 방법을 제안
	ISO/NP TR 23245	블록체인 및 분산원장기술의 취약성 및 잠재적인 보안 위협요소를 제공하는 기술보고서. 현재의 위협요소를 경감시킬 수 있는 요구사항들을 바탕으로 블록체인 및 분산원장 기술 보안 표준화의 방향성 제시
	ISO/NP TR 23246	블록체인 및 분산원장 기술 시스템의 신원 관리에 대해 설명하는 기술보고서. 신원의 개요, 인터페이스 (interfaces), 증명(verification), 비표준 ID(non-standard ID), 개인정보(privacy), 보안(security) 등의 중요성을 기술
	ISO/NP TR 23576	블록체인 및 분산원장 기술 기반의 디지털 자산 관리 보안에 대한 중 요성이 부각되면서 이에 대한 실무 지침을 제공하기 위해 시스템 모델, 보안 관리, 생명주기 관리, 보안 통제 등에 대한 내용으로 구성
ISO/TC 307/WG 3	ISO/AWI TS 23259	스마트 컨트랙트의 기술적 및 법적 측면도 고려한 기술규격. 스마트 컨트랙트가 법적 효력을 가질 경우 관할권 주장 문제 등 국가 간 장벽 및 경계가 존재하지 않는 블록체인 세상에서 충분히 발생할 수 있는 사례들을 검토하고 분석
	ISO/DTR 23455	블록체인 및 분산원장 기술 시스템에서의 스마트 컨트랙트를 소개하고, 스마트 컨트랙트의 상호작용 및 관계에 대해 설명하는 기술보고서

[ISO/TC 307 표준 문서]

- ITU-T는 현재 여러 연구반(SG) 및 포커스 그룹(FG)에서 블록체인 및 분산원장 기술 표준화를 수행

위원회	내용
ITU-T SG20	IoT 및 스마트시티(Internet of things (IoT) and smart cities and communities (SC&C))
ITU-T FG-DLT	분산원장 기술(Focus Group on Application of Distributed Ledger Technology)
ITU-T FG-DFC	디지털 화폐(Focus Group on Digital Currency including Digital Fiat Currency)
ITU-T FG-DPM	데이터 처리 및 관리(Focus group on data processing and management to support IoT and smart cities and communities)

[블록체인을 연구하는 위원회 목록]

- 국제 공적표준화기구 외에도 W3C에서도 다양한 표준 전문가들이 웹페이먼트, 블록체인 및 분산원장 기술 관련 웹표준을 개발하고 있음

## 2. 연구개발과제의 수행 과정 및 수행 내용

### - 주관연구기관(긴트) :

#### ㉔ 농기계 이력 데이터 관리 블록체인 시스템 설계

##### ㉠ 농기계 생애주기별 이해관계자 생태계 및 작업 요구사항 검토

가. 농기계 생애주기 조사를 위한 다양한 기종의 시기별 이벤트 확인

##### 1. 트랙터의 시기별 이벤트

- 제조 : 매년 연말경에 차년도 물량의 대부분을 생산
- 홍보 : 매년 농한기(10월이후)를 기점으로 홍보
- 판매 : 매해 1~2월에 대부분의 신차 수량을 계약 및 판매
- 사용 : 작물을 직파하고 심는 단계인 3~5월에 최다 빈도 사용 후 지속적 사용
- 수리 : 사용 최다 빈도 직전 기간인 2~4월에 최대 수리 빈도 발생

##### 2. 이앙기의 시기별 이벤트

- 제조 : 매년 연말경에 차년도 물량의 대부분을 생산
- 홍보 : 매년 농한기(10월 이후)를 기점으로 홍보
- 판매 : 매해 1~2월에 대부분의 신차 수량을 계약 및 판매
- 사용 : 작물을 직파하고 심는 단계인 3~4월에 최다 빈도 사용, 트랙터와 달리 연내 지속적 사용은 훨씬 적음
- 수리 : 사용 최다 빈도 직전 기간인 1~3월에 최대 수리 빈도 발생

##### 3. 콤바인의 시기별 이벤트

- 제조 : 매년 여름경에 당해년도 물량의 대부분을 생산
- 홍보 : 매년 농한기(10월 이후)를 기점으로 홍보
- 판매 : 매해 6~8월에 대부분의 신차 수량을 계약 및 판매
- 사용 : 작물 추수하는 단계인 9~10월에 최다 빈도 사용, 트랙터/이앙기와 달리 연내 지속적 사용은 훨씬 적음
- 수리 : 사용 최다 빈도 직전 기간인 8~10월에 최대 수리 빈도 발생



[주요 농기계의 시기별 이벤트 조사]

#### 나. 연관된 주요 이해관계자 섭외 및 인터뷰

##### 1. 제조사 및 대리점

- 주요 업무 : 농기계의 생산 및 유지 보수 등
- 주요 보유 정보 : 농기계의 생산 정보, 판매 이력, 유지 보수 이력 등
- 필요 정보 : 판매량 및 유지 보수를 위한 정보

##### 2. 농협(금융기관)

- 주요 업무 : 농기계 구매시 금융지원
- 주요 보유 정보 : 금융 지원 농민 정보, 농기계 구매 이력 등
- 필요 정보 : 사용 연한 및 폐기 확실성 정보

### 3. 정부 산하기관

- 주요 업무 : 보급 사업 업무 및 관리 감독
- 주요 보유 정보 : 보급 사업의 기본 정보 및 관리 감독 정책
- 필요 정보 : 관리 감독을 위한 다양한 정보

### 4. 지자체 및 시군구

- 주요 업무 : 지방 농업 활성화를 위한 농기계 구매 및 임대
- 주요 보유 정보 : 해당 지역 농민의 실제 농업 활동 정보
- 필요 정보 : 농기계 임대 및 회수를 위한 다양한 정보

### 5. 농기계 사용자

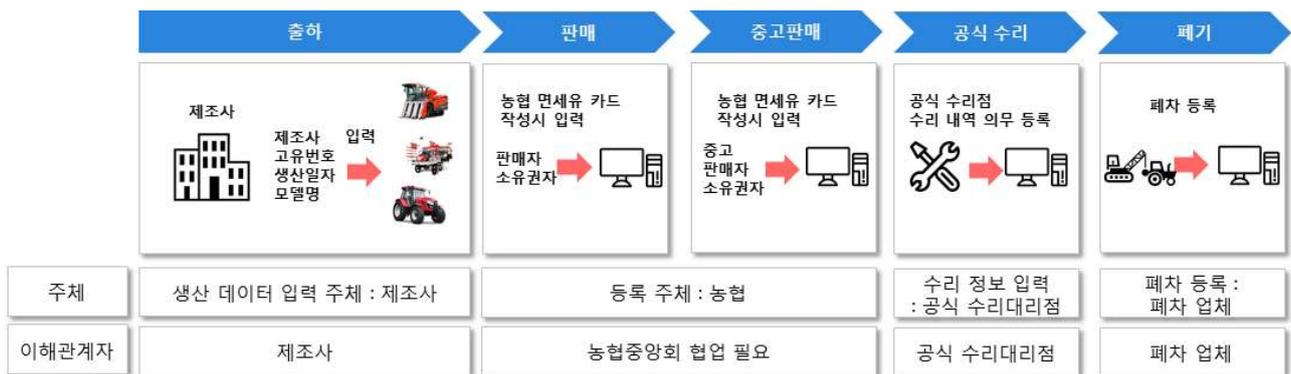
- 주요 업무 : 농기계를 사용하여 실제 농업 활동
- 주요 보유 정보 : 농업 활동을 통한 시기별 농기계 활용 정보
- 필요 정보 : 고장 및 사고 예방

### 6. 농기계 폐차 업체

- 주요 업무 : 농기계를 폐차
- 주요 보유 정보 : 농기계 폐기를 통한 생애주기 종료 정보
- 필요 정보 : 주요 부품 사용시간 정보

### 다. 주체별 이력 데이터 열람 및 입력 과정 시나리오

- 다양한 주체들을 통한 원활한 이력 전환 흐름 고려
- 예상 시나리오의 간략화를 통한 추가 보완성 확보



[인터뷰를 바탕으로한 시나리오 도출]

### ㉔ 농기계 생애주기별 이력 관리를 위한 원장 데이터셋 정의

- 이해관계자별 생성 가능 원장 데이터셋 항목 작성
- 상호관계에 따른 데이터 세부 권한 정의
- 데이터셋은 블록체인 네트워크에 올라간 이후 위변조가 불가하므로, 정의된 데이터셋의 무결성은 추후 교차 검증에 대해 고려하도록 함. 다만, 대부분의 데이터가 ECU 및 타 기관이 연계된 데이터 이므로 무결성에 오류가 생길 확률은 낮을 것으로 판단함

항목	분류	작성 권한		열람 권한
		발현	입력	
제조사	이력 데이터 (초기)	제조사 정책	제조사	농림부/농협 등
고유번호	이력 데이터 (초기)	제조사 정책	제조사	농림부/농협 등
생산일자	이력 데이터 (초기)	제조사 정책	제조사	농림부/농협 등
자대번호	이력 데이터 (초기)	제조사 정책	제조사	농림부/농협 등
세부 모델명	이력 데이터 (초기)	제조사 정책	제조사	농림부/농협 등
판매자	이력 데이터 (초기)	판매자 정책	대리점	농림부/농협 등
소유권자	이력 데이터 (초기)	판매자 정책	대리점	농림부/농협 등
주요 부품 고장 이력	이력 데이터 (주기)	수리점 정책	대리점, 수리점	농림부/농협 등
주요 부품 수리 이력	이력 데이터 (주기)	수리점 정책	대리점, 수리점	농림부/농협 등
총 운행 거리	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
총 운행 시간	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
총 작업 시간	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
총 연료 소모량	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
총 탄소 배출량	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
최근 정차 위치	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
예상 소모품 교체 잔여시간	이력 데이터 (주기)	제조사 정책, 농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
폐기 시기	이력 데이터 (최종)	폐차 업체 정책	폐차업체	농림부/농협 등
임대 시작 시기	이력 데이터 (주기)	임대 정책	임대사업자, 지자체	농림부/농협 등
반납 예상 시기	이력 데이터 (주기)	임대 정책	임대사업자, 지자체	농림부/농협 등
실제 반납시기	이력 데이터 (주기)	임대 정책	임대사업자, 지자체	농림부/농협 등
대출금액	이력 데이터 (초기)	금융기관 정책	금융기관(농협)	농림부/농협 등
대출 대상	이력 데이터 (초기)	금융기관 정책	금융기관(농협)	농림부/농협 등
소유권이전이력	이력 데이터 (이벤트)	판매자 정책	대리점, 중고판매점	농림부/농협 등
농기계 현재 위치	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
농기계 현재 연료소비량	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
농기계 현재 엔진소모동력	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
농기계 움직임감지	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
엔진 냉각수 온도	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
시간당 연료 소모율	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
축매탱크 잔여레벨	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
현재 변속 단수	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
미션 오일 온도	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
PTO 출력	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
키오프 이후 배터리 잔량	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
PTO 총 사용시간	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
엔진 RPM	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
고장 진단 정보	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
소음	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
수평도	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개
무게	운영 데이터 (최근)	농기계	운영데이터 수집 시스템	개인정보법 내에서 공개

[데이터셋 항목 작성 일부 발췌]

© 블록체인 코어 선정 및 용량 설계

가. 블록체인 코어 플랫폼의 비교 분석

1. 하이퍼레저

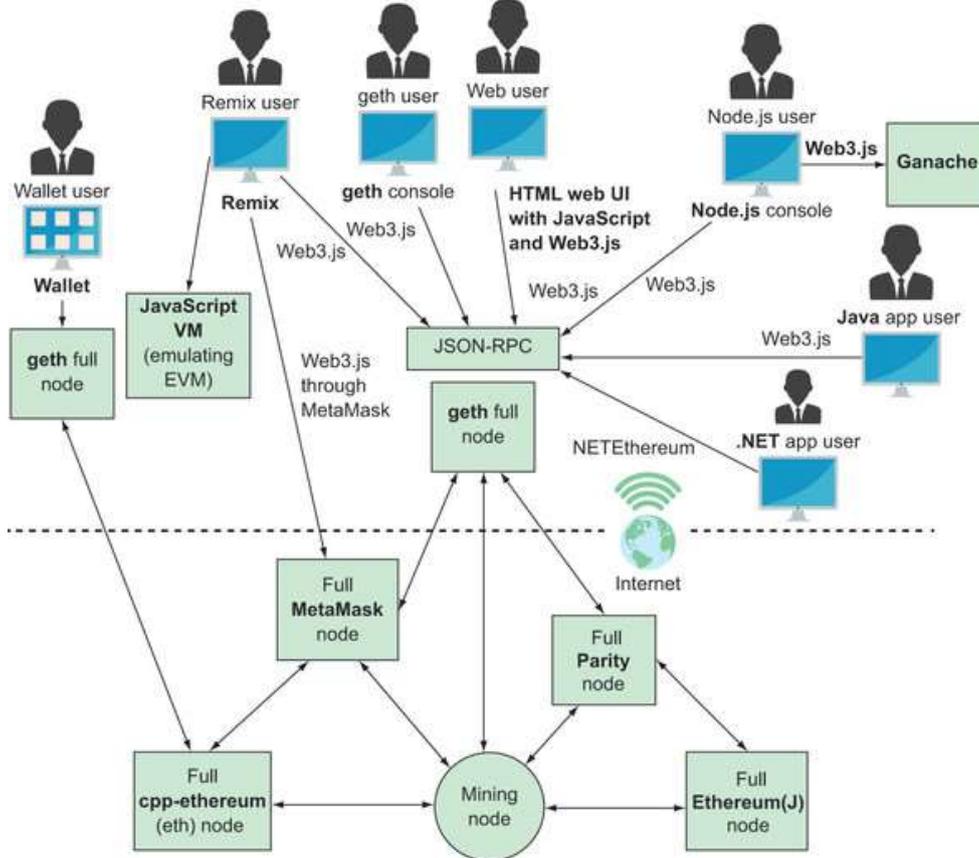
- 현재 다양한 산업군의 적용을 목표로 프로젝트 진행
- 다른 플랫폼과 달리 금융에 특화되지 않음
- 다양한 유스케이스 배출을 위한 제품군의 구조 확립



[하이퍼레저의 제품군 및 프레임워크 아키텍처]

## 2. 이더리움 엔터프라이즈

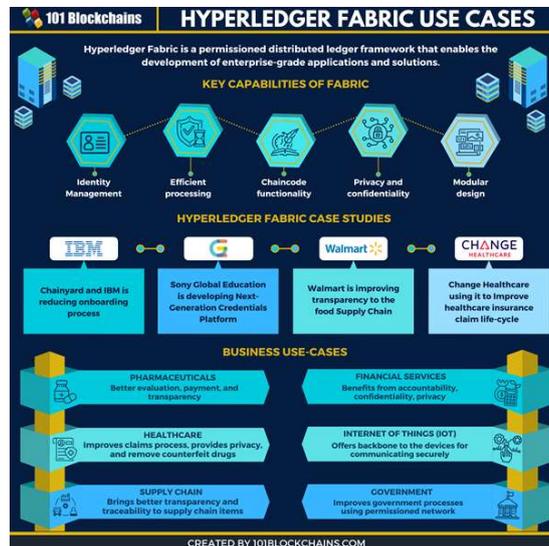
- 다양한 산업군의 적용을 목표로 하고 있으나, 아직까진 계약, 금융에 특화
- 하이퍼레저 대비 엔터프라이즈 기능 및 툴링 제품군 미비



[Billing에 특화된 예시]

## 나. 이해관계자 및 데이터셋 정의를 바탕으로 선정

- 농업기계 이력 관리에 다양한 산업군(제조, IoT, 금융 등)으로 이루어진 생태계
- 금융에 특화된 이더리움 엔터프라이즈는 다양한 분야에 적용 역부족 예상
- 최종적으로 유스케이스가 다양한 산업군에 걸쳐있는 하이퍼레저로 선택



[하이퍼레저의 다양한 유스케이스]

- ㉔ 농기계 이력 및 운행 데이터 수집을 위한 H/W와 S/W 환경 구축
  - ㉠ 농기계의 이력 및 운행 데이터 수집을 위한 보급형 농기계 단말(H/W&S/W) 개발
    - 가. 장비 입출력 프로토콜 표준화 작업
      - 1. 선진사 시스템 분석을 통한 장비 필수 데이터 선별화 작업

[단말기 기반 서비스 구축을 위한 필수 데이터 선별]

- 유사 농기계 기반 데이터 서비스 데이터 분석을 통한 필수 데이터 선별
- J1939 기반의 취득 가능한 정보 및 단말기 자체의 특수 정보를 포함한 데이터 선별

2. 선별 데이터 기반 농기계 데이터 프로토콜 협의 및 1차 문서화

		Name of document		Page																																																	
Issuer: GINT Co., Ltd.		Date: 15.03.2018		Approval: Name Issue Date: 01 Reg. No:																																																	
GINT Telematics System - Equipment Data Protocol (EDP)																																																					
<b>1. Information Parameters for Telematics Gateway</b>																																																					
1.1. Information Parameters for Agricultural and Construction Equipments (SAE J1939)																																																					
1.1.1. Fuel Consumption: LFC (SPN 250)																																																					
<table border="1" style="width: 100%;"> <tr> <td colspan="8" style="text-align: center;">PGN Data: 00001120</td> </tr> <tr> <td colspan="8" style="text-align: center;">PGN Data: 01217</td> </tr> <tr> <td colspan="8" style="text-align: center;">Req. Rate: 100ms</td> </tr> <tr> <td>Data Byte1</td> <td>Data Byte2</td> <td>Data Byte3</td> <td>Data Byte4</td> <td>Data Byte5</td> <td>Data Byte6</td> <td>Data Byte7</td> <td>Data Byte8</td> </tr> <tr> <td style="background-color: #cccccc;">Not used for EDP Standard</td> <td>Engine total fuel used 0.1 L / 100 pass 0.1, off-on</td> <td>Engine total fuel used 0.1 L / 100 pass 0.1, off-on</td> <td>Engine total fuel used 0.1 L / 100 pass 0.1, off-on</td> <td>Engine total fuel used 0.1 L / 100 pass 0.1, off-on</td> </tr> <tr> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> <td>SPN 256</td> <td>SPN 256</td> <td>SPN 256</td> <td>SPN 256</td> </tr> </table>						PGN Data: 00001120								PGN Data: 01217								Req. Rate: 100ms								Data Byte1	Data Byte2	Data Byte3	Data Byte4	Data Byte5	Data Byte6	Data Byte7	Data Byte8	Not used for EDP Standard	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on					SPN 256	SPN 256	SPN 256	SPN 256			
PGN Data: 00001120																																																					
PGN Data: 01217																																																					
Req. Rate: 100ms																																																					
Data Byte1	Data Byte2	Data Byte3	Data Byte4	Data Byte5	Data Byte6	Data Byte7	Data Byte8																																														
Not used for EDP Standard	Not used for EDP Standard	Not used for EDP Standard	Not used for EDP Standard	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on	Engine total fuel used 0.1 L / 100 pass 0.1, off-on																																														
				SPN 256	SPN 256	SPN 256	SPN 256																																														
<p><b>Description of SAE J 1939:</b>  <b>Total Fuel Used:</b> Accumulated amount of fuel used during vehicle operation.</p> <p><b>Additional comment:</b>          Calculated values given as indications, not as contractual values.          Might be set to "not available" if the High Resolution Fuel Consumption is available.          The mentioned resolution is not related to the accuracy of the signal.</p>																																																					

[단말기와 대상 장비간 데이터 프로토콜 문서 작성]

- 향후 단말기 데이터 스펙 공유를 위한 표준 문서 작성
- 국내 Tier4 이상 농기계에 모두 적용 가능한 범용 형태의 데이터 선별

나. 단말기 프로토 타입 개발

1. 데이터 서버 송신을 위한 SW 드라이버 개발



[개발 SW 프로토 타입 개요]

- 리눅스 기반 플랫폼 구축 완료
- 이더넷 기반 데이터 서버 송수신 드라이버 V1 개발 완료
- EDP 프로토콜 기반 CAN 통신 드라이버 V1 개발 완료
- 단말기를 위한 서버 인증 기능 구현

2. 단말기 프로토타입 HW 개발



[개발 HW 단말기 사양 개요]

기능 고려 항목	용도	우선 순위
LTE 기능 모듈	- 무선 이동 통신 기반 인터넷 이용	필수
GPS 기능 모듈	- 농기계 위치 추적	필수
CAN 기능 모듈	- J1939 및 국내 농기계 개별 CAN 프로토콜 대응	필수
Ethernet 기능 모듈	- 국내 통신사 인증 및 A/S	조건부 필수
Emergency 기능 모듈	- 사고 /도난으로 인한 배터리 단락, 장기간 미가동 농기계에서 자체 배터리로 더 오랜 시간 위치 파악 가능	선택 사양
FOTA 기능 모듈	- 향후 무선 기능 업데이트를 위해 권장	필수
Key-Off Mode 기능 모듈	- 원격시동 시 아날로그 입력을 통한 시동 조건 판별 및 릴레이 출력을 통한 Engine 시동	선택 사양
자세 센싱 기능 모듈	- 기울기 및 충격량 감지를 통한 전복 사고 감지	선택 사양

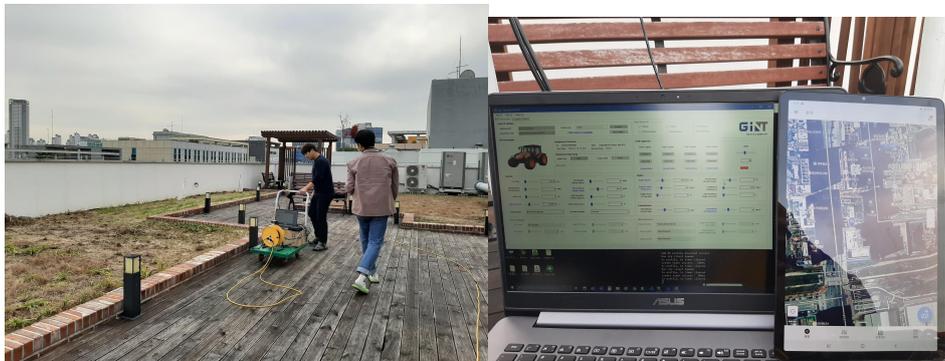
[단말기 주요 사양표]



[HW 단말기 개발품 실사진]

- HW 사양 및 설계 완료
- HW 사양 설계 기반 단말기 SW 개발용 위한 EV 단말기 환경 구축 완료
- HW 단말기 프로토타입 개발

다. GPS 및 온보드 센서를 활용한 단말기 SW 드라이버 개발



[GPS 단말기 개발 및 시험]

1. Geo Fence / Time Fence 기능을 위한 측위 정보 수집 드라이버 개발
  - 농기계 정보 수집을 위한 Dual 채널 CAN 드라이버 개발 완료
  - 서버 통신 및 서비스를 위한 서버 인증 기능 구현
    - A. AWS IoT 사물 디바이스 인증서 발급 및 배포 로직 구현 완료
    - B. 디바이스 권한 부여 및 관리 로직 구현 완료
    - C. 인증키 관리 정책 구현 완료
2. 콘텐츠 개발 최적화를 위한 원격 데이터 필터 업데이트 기능 개발 완료



[TRACTOR 데이터 CAN 데이터 시뮬레이터를 통한 개발 검증]

- 송/수신 데이터를 원격에서 선택하도록 하는 데이터 필터 개발 완료
- 차량 정보 수집을 위한 Dual 채널 CAN 드라이버의 필터 갱신 기능 개발
- A. CAN 송신 필터 갱신 기능: 다양한 농기계의 신호에 대한 기능 자유도 확보
- B. CAN 수신 필터 갱신 기능: J1939 DM1 및 각종 고장 진단 시 요청기능

3. 데이터 암호화 모듈 개발

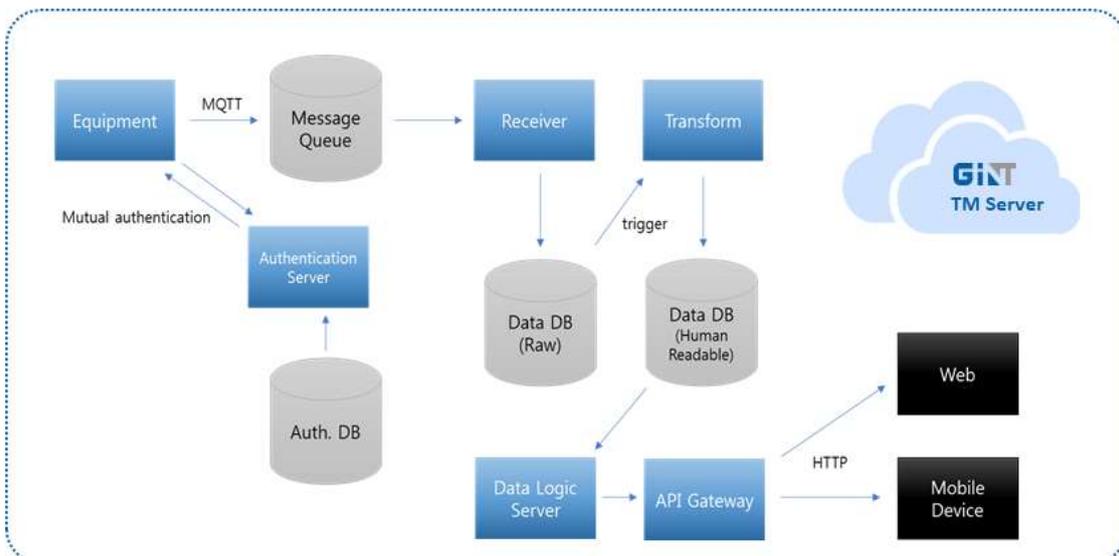
- 단말기와 서버 간 송/수신 데이터의 암호화 모듈 개발
- 단말기의 HW 성능 및 단가를 고려하여 최적의 암호 알고리즘 선정
- A. 비대칭 key를 활용한 장비, 사용자 인증 기능 구현
- B. 단말 및 통신 성능에 최적화된 암호화 기능 구현

③ 농기계 운용 데이터 수집 및 가공을 위한 서버 시스템 개발

가. 데이터 서버 개발 환경 및 테스트 서버 구축



[농기계 및 클라우드 서버 개요]



[클라우드 서버 구성도]

1. AWS 환경에 테스트 서버 구축
2. AWS IoT, DynamoDB를 사용하여 데이터 입/출력 서버 구현
3. AWS Lambda를 이용하여 인증 기능 구현
4. 테스트를 위한 더미 데이터 조회 가능 API 개발

나. 대용량 데이터 처리를 위한 서버 설계 및 서비스 용량 산정



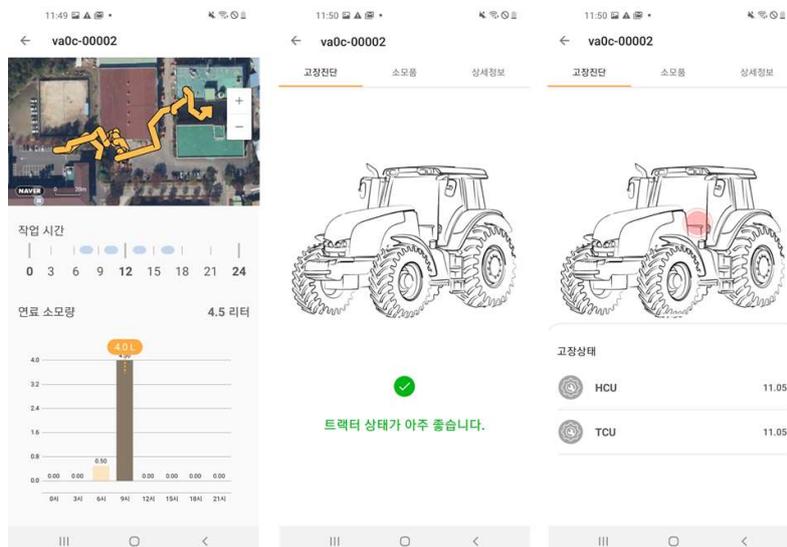
[서버 구현 및 시험 환경]

1. MQ 고가용성/고성능 확보를 위한 클러스터 및 index 설계
  - 메시지 수신부 MQ 클라이언트 개발
  - 2차년도 각종 콘텐츠의 원활한 상용 서비스 제공을 위한 MQ 기능 확보
  - MQ가 적용된 서버 기반 각종 콘텐츠의 정량적 시험 수행 완료
2. 데이터 릴레이 및 통합 물리 변환 엔진 개발
  - CAN data를 조합/가공하여 콘텐츠에 제공하는 통합 물리 변환 엔진 개발
  - 데이터 송/수신부와 독립적으로 작동하여 대용량 처리 지원
  - 1차년도 각종 콘텐츠의 원활한 상용 서비스 제공을 위한 물리 변환 기능 확보
  - 물리 변환 기능이 적용된 서버 기반 각종 콘텐츠의 정량적 시험 수행 완료
3. 1차년도 데이터 및 콘텐츠 사양에 따른 용량 산정
  - 서버 HW 사양 및 규모 확정
  - 가용성 수준에 따라 이중화/다중화 설계 적용
  - 1차년도의 수용 가입자와 서버 응답속도, 제공 콘텐츠 달성을 위한 용량 선정
  - 용량 선정 및 다중화 설계 적용으로 수용가입자 및 서버 응답 속도 시험 수행

㉔ 농기계 이력 및 운행 데이터 기반 농업인 앱/웹 서비스

㉔ 이력 데이터 추가 수집을 위한 농업인 앱/웹 서비스 개발

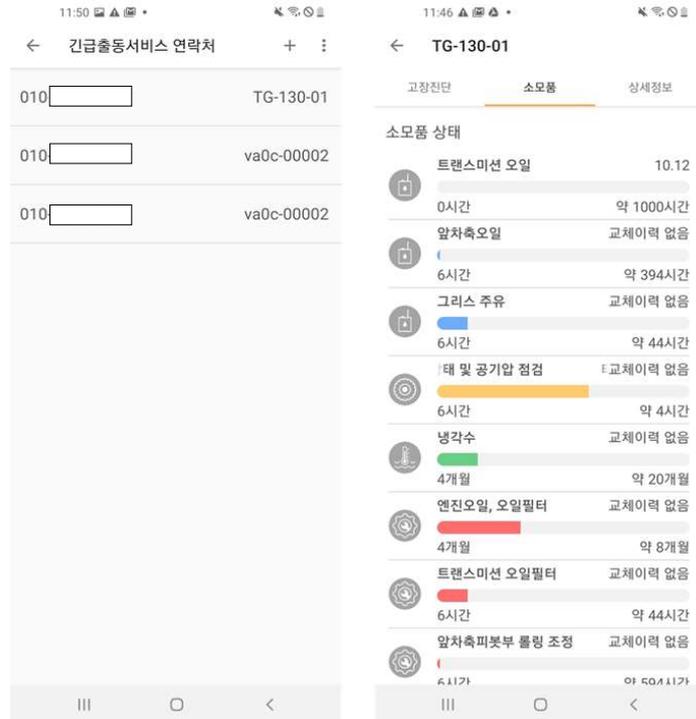
가. 농기계 이력 데이터 신뢰도 향상 및 AI 강화를 위한 작업일지, 고장진단 등 농업인의 생산성 향상 기능 추가 개발



[자동 작성 작업 일지 및 고장진단 앱 기능 개발 화면]

- 농기계의 효율적인 사용 관리와 번거로운 사용 신고 업무에 도움을 줄 수 있는 작업 일지 기능 구현
- 농기계의 상태 유지 및 사고 예방을 위한 고장 진단 기능 구현

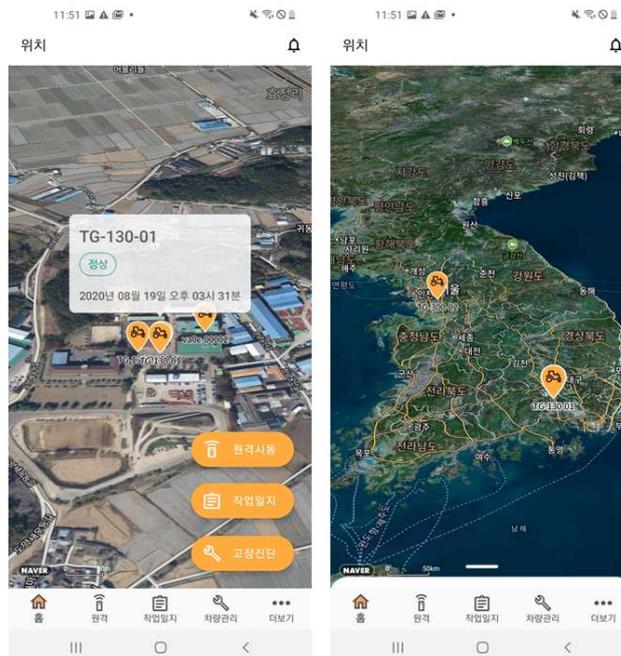
나. AI 강화를 위한 긴급출동 자동연결 서비스, 소모품 교체주기 기능 업데이트 추가



[긴급출동 자동 연결 서비스 및 소모품 교체주기 앱 기능 개발 화면]

- 전복 사고 및 긴급 출동 SOS 버튼 등으로 사전 등록한 지인 및 관공서에 자동 연결
- 농기계의 효율적인 관리를 위한 소모품 교체주기 자동 기록/알람 서비스

다. 편의성과 생산성을 향상을 통해 농업인의 서비스 사용을 장려하여 분산원장 및 AI 알고리즘 개발에 필요한 사용자 데이터의 취득 유도

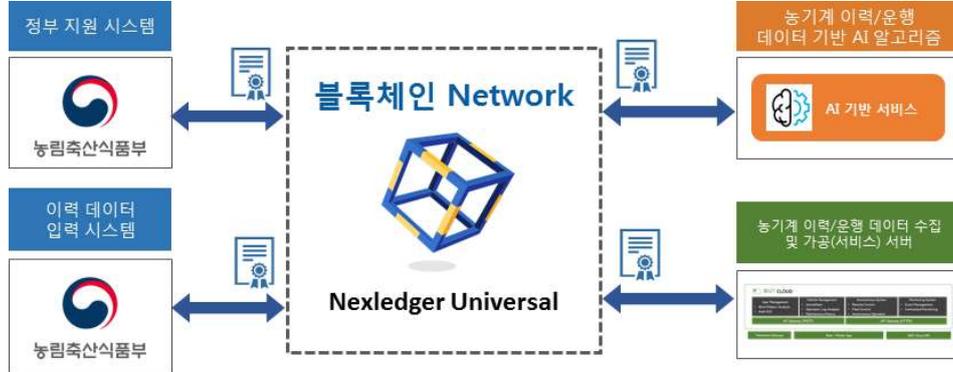


[손쉬운 농기계 관리를 위한 홈메뉴와 지도 기반 서비스]

- 보다 손쉬운 농업인의 농기계 관리를 위한 킷폼 메뉴 기능 개발
- 직관적인 사용이 가능한 지도 기반 서비스 기능 구현

㉔ 농기계 이력 데이터 블록체인 시스템 구현

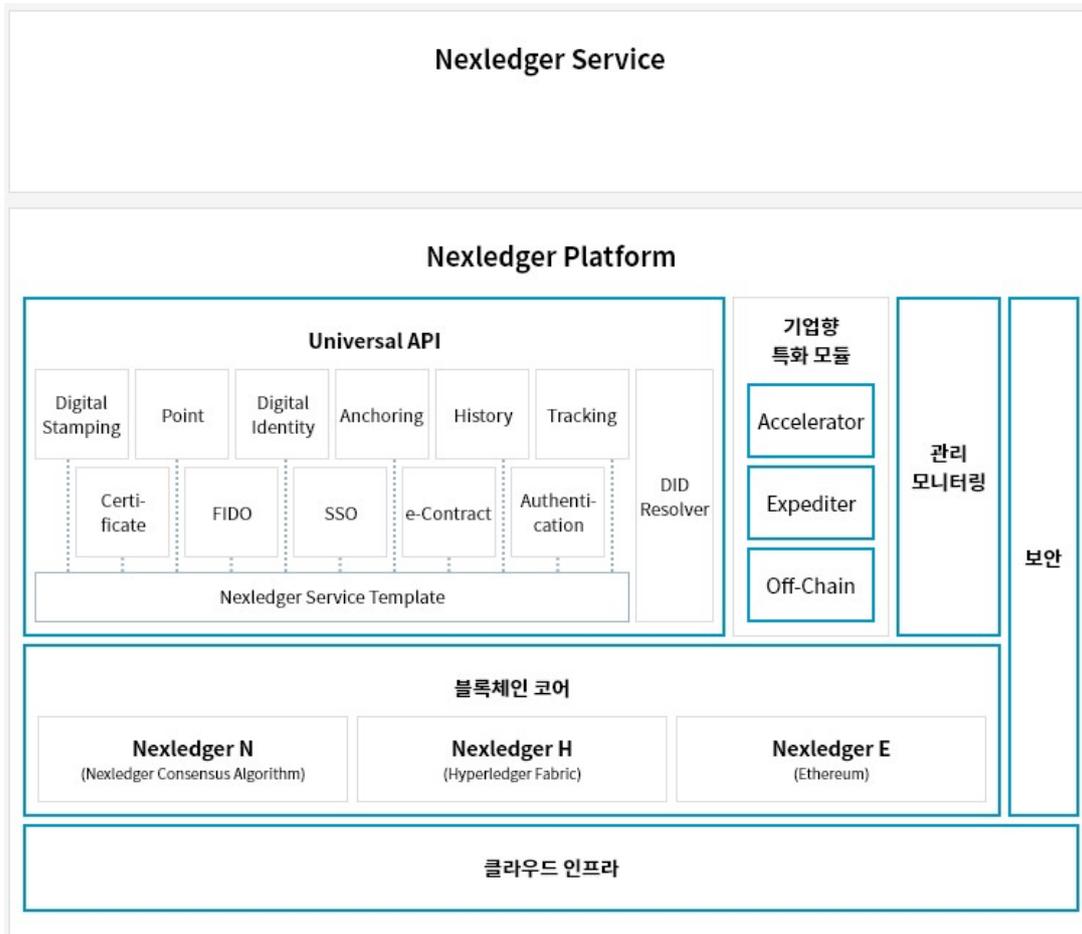
- ㉔ 분산원장을 포함한 블록체인 기반 시스템 아키텍처 설계 및 구현
  - 블록체인 기반 시스템 아키텍처 설계



[ 기관 또는 서비스별 블록체인 네트워크 구성도 ]

각 시스템이 직접 서로 연결되지 않고 블록체인 네트워크를 통해서 서로 정보를 주고 받을 수 있도록 구성하여 상호간의 신뢰성 확보. 블록체인 네트워크는 현재 국내에서 상용 라이브러리로 가장 앞서있는 삼성SDS의 넥스레저를 1차년도에 선정하였고, 2차년도에서는 넥스레저를 이용하여 위의 구성도를 따라 실제적으로 시스템과 넥스레저간의 연결을 구현하였다.

- 블록체인(넥스레저) 아키텍처의 활용



[ 넥스레저 유니버설 구조 ]

선정된 넥스레저의 구조는 크게 넥스레저 서비스 부분과 넥스레저 플랫폼 부분으로 나뉜다. 서비스 부분은 본 과제에서 수행할 내용의 소프트웨어가 차지하고 있다. 다음으로 플랫폼 부분은 다양한 모듈들이 자리잡고 있는데, 본 과제에서는 직접적으로 사용할 부분은 Universal API 이다. 이 중에서 특히 중점적으로 사용한 API는 크게 아래 세 가지 이다.

1. Digital Stamping : 사용자 가입시 사용

Digital Stamping 은 전자문서의 위변조 방지를 위하여 전자문서의 원본확인정보와 시점정보를 Blockchain에 저장한 후, 해당 전자문서가 특정 시점에 존재하였으며 그 이후 변경되지 않았음을 증명해 주는 서비스

2. Digital Identity : 사용자 로그인시 사용

Digital Identity API 는 블록체인을 통해 개인의 신원 정보를 관리할 수 있게 해주는 API. 인증조직은 신뢰할 수 있는 신원 정보를 개인들에게 제공하고, 개인은 인증조직이 발급한 신원정보를 지갑에 저장해 관리한다. 개인은 필요시 열람권을 설정해, 특정 사용자가 특정 기간 동안 지정된 신원정보를 열람할 수 있게 허용할 수 있다. 각 인증기관은 해당 기관이 발급하는 정보들만을 보유하게 되고, 열람자들은 제한된 기간동안 허용된 정보에 대해서만 열람할 수 있게 된다. 넥스레저의 Digital Identity API는 개인정보의 공개범위를 각 개인이 설정할 수 있도록 하여, 개인정보가 본인 동의없이 유출되는 것을 막는다.

3. History : 장비 이력 저장시 사용

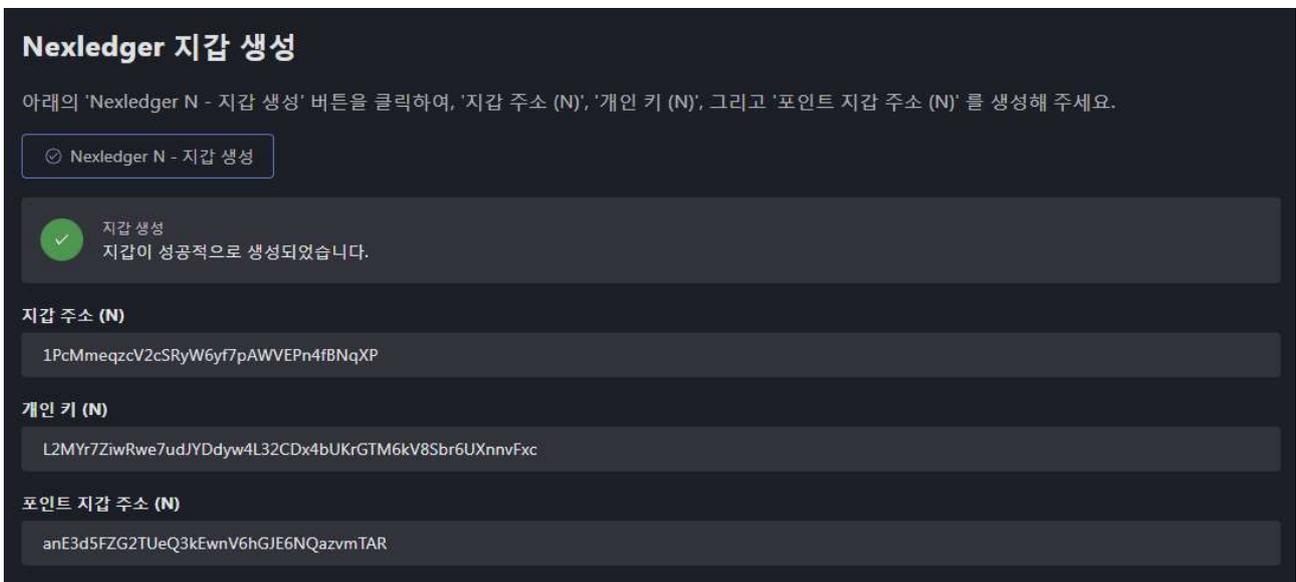
History는 이력 데이터의 위변조 방지를 위하여 발생하는 이력 데이터를 Blockchain에 저장하고, 신뢰가 확보된 이력 내용을 이해 당사자들이 조회할 수 있는 서비스이다.

관리 모니터링과 보안의 경우, 넥스레저의 기본 바탕이므로 구현에 따로 사용하지 않아도 적용이 가능한 부분이다. 기업향 특화 모듈은 현 과제에서는 필요치 않아 배제하였다.

Universal API 아랫단의 블록체인 코어는 본 과제에서 직접 활용하여 사용할 필요는 없으나, API들과 연계 되어 사실상 사용되고 있다고 볼 수 있다.

- 넥스레저를 활용한 구현 과정

관련도와 API들을 선정 한 후에, 넥스레저를 이용하여 아래와 같은 구현 과정을 거치게 되었다.



[ 블록체인 ID를 위한 지갑 생성 - 예시 ]

### 코드 바스켓

'서명된 데이터', '지갑 주소 (N)' 그리고 '사용자 정보' 코드 블록을 코드 바스켓 영역으로 드래그 앤 드롭하고 API를 호출하면 사용자 정보가 등록됩니다.

서명된 데이터

```
IFgnVyh3/IFKAPYTHUeaZ/PGB3431MYBUh2t544UP82IGkdZ/zFoIoAQMYIXHqR3TyCJbFX568Nnc4Eb4/tge6w=
```

지갑 주소 (N)

```
1PcMmeqzcV2cSRyW6yf7pAWVEPn4fBNqXP
```

사용자 정보

```
{
  "name": "홍길동",
  "birth": "1989-10-01",
  "phone": "010-1234-5678",
  "email": "abc@samsungsds.com"
}
```

API 가이드

```

1 1 const request = require("request");
2 2 const nexledgerUniversalURL = "https://nexledger.samsungsds.com";
3 3 const connectors = {
4 4   NCA: "nca",
5 5   Ethereum: "eth",
6 6   Hyperledger: "hlf"
7 7 };
8 8 // Randomly Signed Data. Need to use pair of wallet ID & private key
9 9 const body = {
10 - data: "",
11 - sender: "",
12 - name: "",
13 - birth: "",
14 - phone: "",
15 - email: ""
10+ data: "IFgnVyh3/IFKAPYTHUeaZ/PGB3431MYBUh2t544UP82IGkdZ/zFoIoAQMYIXHqR3TyCJbFX568Nnc4Eb4/tge6w=",
11+ sender: "1PcMmeqzcV2cSRyW6yf7pAWVEPn4fBNqXP",
12+ name: "홍길동",
13+ birth: "1989-10-01",
14+ phone: "010-1234-5678",
15+ email: "abc@samsungsds.com"

```

API 요청

사용자 정보 등록  
사용자 정보가 성공적으로 등록되었습니다.

[ 사용자의 정보를 저장하기 위한 바스켓 생성 - 예시 ]

### 코드 바스켓

[지갑 주소(N)] 코드 블록을 코드 바스켓 영역으로 드래그 앤 드롭하고 API를 호출하면 사용자 정보가 조회됩니다.

지갑 주소 (N)

```
1PcMmeqzcV2cSRyW6yf7pAWVEPn4fBNqXP
```

API 가이드

```

1 1 const request = require("request");
2 2
3 3 const nexledgerUniversalURL = "https://nexledger.samsungsds.com";
4 4 const connectors = {
5 5   NCA: "nca",
6 6   Ethereum: "eth",
7 7   Hyperledger: "hlf"
8 8 };
9 -const walletAddress = "";
9+const walletAddress = "1PcMmeqzcV2cSRyW6yf7pAWVEPn4fBNqXP";
10 10
11 11 const options = {
12 12   method: "GET",
13 13   url: `${nexledgerUniversalURL}/core/v2/identity/${walletAddress}`,
14 14   headers: {
15 15     "Content-Type": "application/json",
16 16     "NL-TENANT-ID": 0,
17 17     "connector": connectors.NCA
18 18   }
19 19 };
20 20

```

API 요청

사용자 정보 조회  
사용자 정보가 성공적으로 조회되었습니다.

[ 저장된 사용자 정보가 다시 조회 되는지 확인 - 예시 ]

위와 같은 사용자에 대한 코드 영역이 끝나면 아래와 같이 웹IDE를 이용하여 개발을 진행하게 되었다.

```

1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.7.0 <0.9.0;
4
5 /**
6  * @title Ballot
7  * @dev Implements voting process along with vote delegation
8  */
9
10 contract Ballot {
11
12     struct Voter {
13         uint weight; // weight is accumulated by delegation
14         bool voted; // if true, that person already voted
15         address delegate; // person delegated to
16         uint votes; // index of the voted proposal
17     }
18
19     struct Proposal {
20         // If you can limit the length to a certain number of bytes,
21         // always use one of bytes1 to bytes32 because they are much cheaper
22         bytes32 name; // short name (up to 32 bytes)
23         uint voteCount; // number of accumulated votes
24     }
25
26     address public chairperson;
27
28     mapping(address => Voter) public voters;
29
30     Proposal[] public proposals;
31
32     /**
33     * @dev Create a new ballot to choose one of 'proposalNames'.
34     * @param proposalNames names of proposals
35     */
36     constructor(bytes32[] memory proposalNames) {
37         chairperson = msg.sender;
38         voters[chairperson].weight = 1;
39
40         for (uint i = 0; i < proposalNames.length; i++) {
41             // "Proposal(...)" creates a temporary
42             // Proposal object and "proposals.push(...)"
43             // appends it to the end of "proposals".
44             proposals.push(Proposal({
45                 name: proposalNames[i],
46                 voteCount: 0
47             }));
48         }
49     }
50
51     /**
52     * @dev Give "voter" the right to vote on this ballot. May only be called by "chairperson".
53     * @param voter address of voter
54     */
55     function giveRightToVote(address voter) public {
56         require(
57             msg.sender == chairperson,
58             "Only chairperson can give right to vote."
59         );
60         require(
61             voters[voter].voted,
62             "The voter already voted."
63         );
64     }
65 }

```

[ Remix를 이용한 개발 예시 - 보안상 흐름 처리 ]

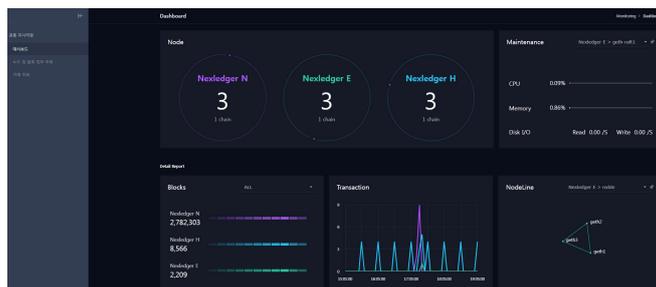
이를 통해서 구성 시스템들과 넥스레저와의 연동 개발을 진행하였다.

향후 비용이 드는 넥스레저를 대체할 오픈소스의 이더리움 계열도 도입에 대해 적극적으로 검토할 필요가 있으나, 아쉽지만 본 과제의 범위에서는 줄이도록 한다.

⑧ 농기계 이력 데이터 블록체인 시스템의 설계 최적화

- 설계 최적화를 위한 성능 모니터링 페이지 개발

상기 구성한 넥스레저의 시스템이 적절하게 가동되는지, 특별한 트래픽은 없는지 등에 대한 모니터링을 위하여 아래와 같은 모니터링 페이지를 구성하였다.



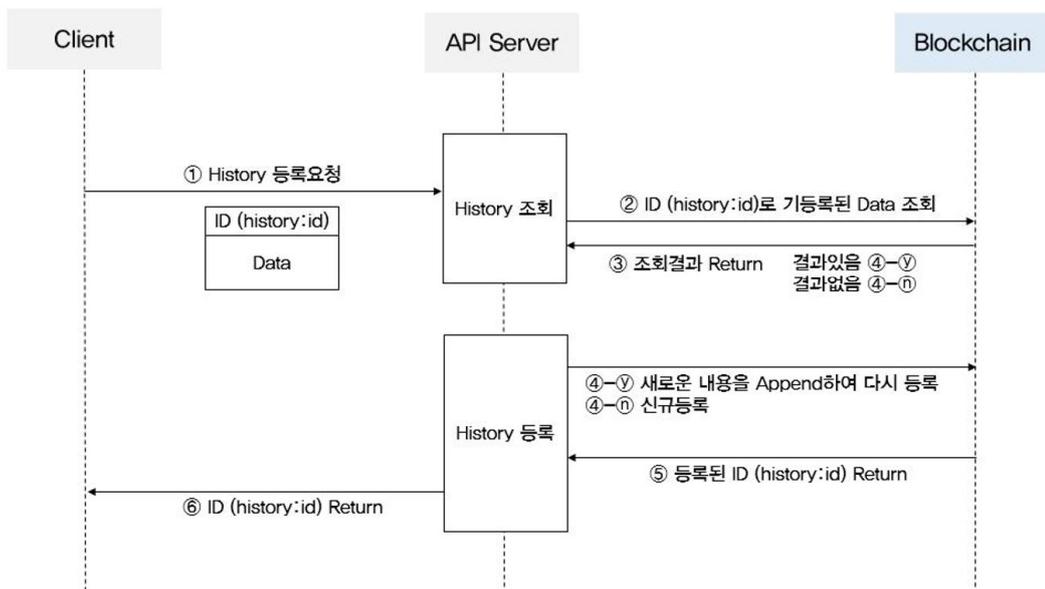
[ 각 코어들의 트래픽 모니터링 ]

[ 네트워크상의 노드들의 정보 조회 ]

[ 노드들간의 트랜잭션 조회 ]

- ④ 농기계 이력 및 운행 데이터 수집 환경과 블록체인 / AI 알고리즘 연동 개발
  - Ⓐ 블록체인 프록시를 통한 이력 및 운용 데이터의 블록체인 네트워크 등록 기능 개발
    - 수집된 농기계 이력 데이터와 블록체인간의 연동
    - 수집된 농기계의 이력을 블록체인 네트워크에 쌓아두기 위해서 아래와 같은 장비이력의 등록, 조회, 일괄 등록 등의 단계를 거치도록 구성하였다. 아래는 주요 단계들이다.

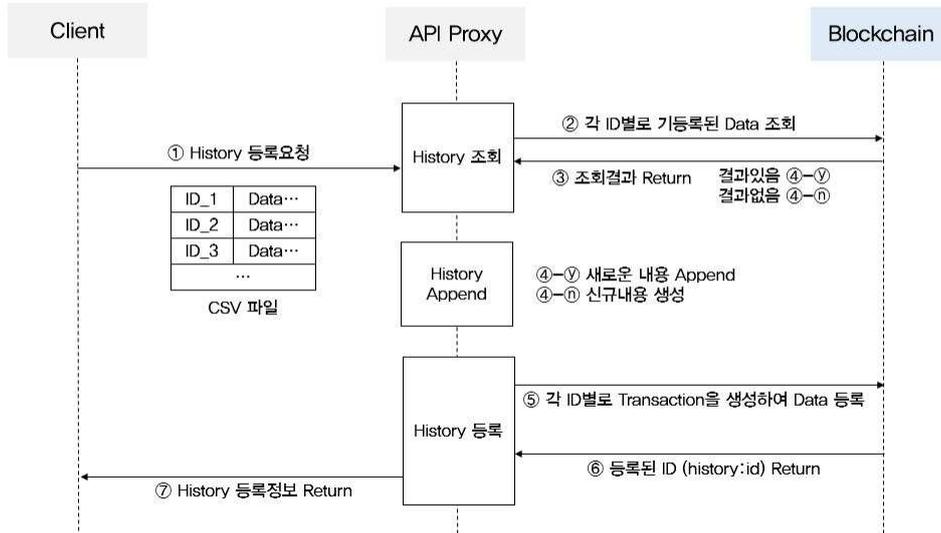
장비이력 등록



[ 장비 이력 등록의 이벤트 다이어그램 ]

장비의 이력이 올라오게 되면 먼저 해당 장비의 이력이 현재 블록체인 네트워크에 등록 되어 있는지 확인한 뒤 조회 결과에 따라 다음 단계로 진행한다. 이미 이력이 있다면, 장비의 기존 이력에 덧붙여 이력을 저장하고, 없다면 새로운 장비를 등록하여 이력을 저장한다. 이력이 저장된 이후에는 등록된 장비 ID를 리턴하여 프로세스가 종료되었음을 리턴한다.

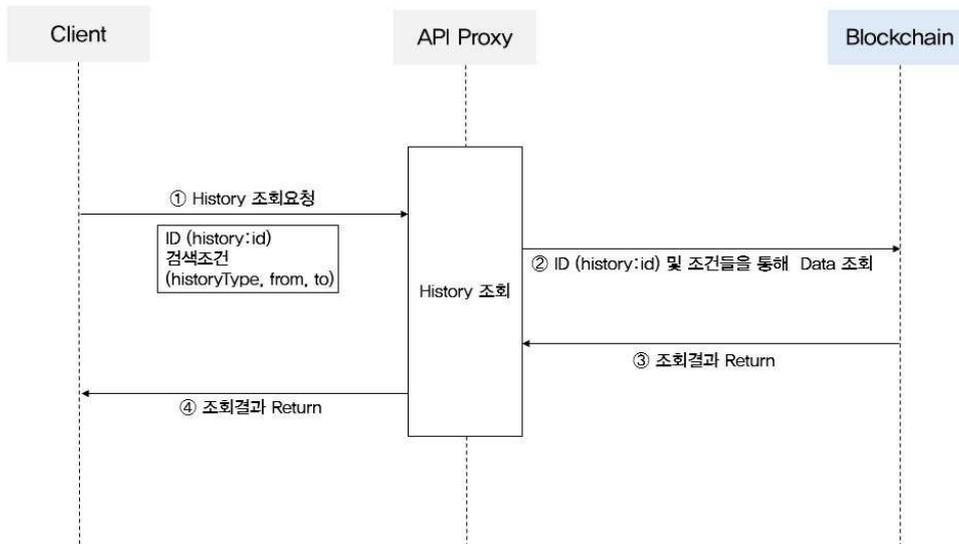
장비이력 일괄 등록



[ 장비 이력 일괄 등록의 이벤트 다이어그램 ]

앞서 하나의 장비만의 이력만 순서대로 업데이트 하는 경우는 실시간 서비스에서 속도적인 면에서 크게 장점을 가질 수는 없다. 이런 점을 보완하고자, 여러 장비의 이력을 한꺼번에 업데이트 하는 프로세스도 신설하였다. 기본적인 플로우 는 앞의 프로세스와 유사하나, 장비 ID 가 다수로 존재하고, 그 ID 별로 모두 트랜잭션을 수행하는 것이 반복적으로 이루어지는 부분이 특징이다.

장비이력 조회



[ 장비 이력 조회 이벤트 다이어그램 ]

마지막으로 이미 넥스레저에 저장된 장비 이력을 조회하는 플로우이다. 특정 장비의 ID가 조회 대상으로

선정되면 API 프록시에게 해당 아이디와 명령이 전달되고, 블록체인 네트워크는 그에 따른 조회 결과를 리턴하여 준다.

- ③ 이력 데이터(분산원장) 외 AI 알고리즘 및 기타 서비스 제공을 위한 API 개발
  - AI알고리즘과의 연동을 위해 아래와 같은 Restful API를 개발하였다. 아래 몇몇 예시들을 나열하였으며, 민감한 부분은 흐림 처리 하였다.

항목명	가능 여부	비고
농기계 위치	○	현재 gps 좌표 조회 가능
연료소비량	○	시작시간, 끝시간을 입력받아 그 사이의 연료소비량 조회 가능
엔진소요동력	○	가장 최근 올라온 엔진 토크 데이터 조회 가능
엔진 가용 시간	○	총 운행시간 (무하별 운행시간 조회 가능) 조회 가능
장비 아이디	○	조회 가능
총 주행 거리	○	총 운행거리 조회 가능
엔진 냉각수 온도	○	가장 최근 올라온 엔진 냉각수 온도 데이터 조회 가능
시간당 연료 소모율	○	(엔진 가용 시간 / 시간당 연료 소모율) 계산 가능 (구현 필요)
누적 연료 소모량	○	총 연료소모량 조회 가능

[ 연동 가능한 데이터 리스트 ]

농기계 위치

- 요청 api : 현재 위치 정보 조회 api
  - 장비 아이디를 이용해 현재 장비의 위치를 얻을 수 있음
- 입력 파라미터
  - equipment\_id : 장비 아이디
- 리턴 데이터 예시

```

{
  "altitude": "127885",
  "longitude": "1279423162",
  "timestamp": "1608579495",
  "equipment_id": "01f76-1bbc-9848-2484-4b69",
  "latitude": "372887856"
}
    
```

- altitude : 고도
- longitude : 경도
- latitude : 위도
- timestamp : 위치가 확인된 시간의 timestamp

[ 장비 위치에 대한 api 매뉴얼 ]

#### 연료 소비량, 누적 연료 소모량

- 요청 api : 장비 별 통계 조회 api
  - 특정 장비에 대해 두 타임스탬프 사이의 통계 데이터를 얻을 수 있음
- 입력 파라미터
  - equipment\_id : 장비 아이디
  - timestamp\_from : 필요한 데이터의 시작 시간 타임스탬프
    - (optional, default = 0)
  - timestamp\_to : 필요한 데이터의 마지막 시간 타임스탬프
    - (optional, default = 현재 시간)
  - statistic\_name : 필요한 통계 데이터 종류 (아래 셋 중 선택)
    - operation\_time : 부하 별 운행시간
    - mileage : 운행거리
    - fuel\_consumption : 연료 소모량
- 리턴 데이터 예시

```
unit:
  value:
    equipment_id: "0000-0000-0000-0000-0000",
    timestamp_from: "2023-01-01 00:00:00",
    timestamp_to: "2023-01-01 00:00:00",
    statistic_name: "fuel_consumption",
    unit: "L",
    value: "1000000.000000000000000000",
    unit: "L",
    value: "10000.0"
```

- unit : 통계 데이터 단위
- value : 통계 데이터 값

#### [ 연료 소비량에 대한 api 매뉴얼 ]

#### 엔진 가용 시간

- 요청 api : 장비 별 통계 조회 api
  - 특정 장비에 대해 두 타임스탬프 사이의 통계 데이터를 얻을 수 있음
- 입력 파라미터
  - equipment\_id : 장비 아이디
  - timestamp\_from : 필요한 데이터의 시작 시간 타임스탬프
    - (optional, default = 0)
  - timestamp\_to : 필요한 데이터의 마지막 시간 타임스탬프
    - (optional, default = 현재 시간)
  - statistic\_name : 필요한 통계 데이터 종류 (아래 셋 중 선택)
    - operation\_time : 부하 별 운행시간
    - mileage : 운행거리
    - fuel\_consumption : 연료 소모량
- 리턴 데이터 예시

```
unit:
  value:
    equipment_id: "0000-0000-0000-0000-0000",
    timestamp_from: "2023-01-01 00:00:00",
    timestamp_to: "2023-01-01 00:00:00",
    statistic_name: "operation_time",
    unit: "h",
    value: "1000000.000000000000000000",
    unit: "h",
    value: "10000.0"
```

- unit : 통계 데이터 단위
- value : 통계 데이터 값
  - load\_low : 타임스탬프 사이 시간 동안 저부하로 운행한 총 운행시간
  - load\_middle : 타임스탬프 사이 시간 동안 중부하로 운행한 총 운행시간
  - load\_high : 타임스탬프 사이 시간 동안 고부하로 운행한 총 운행시간
  - load\_low, load\_middle, load\_high 를 더하여 총 운행시간 계산 가능

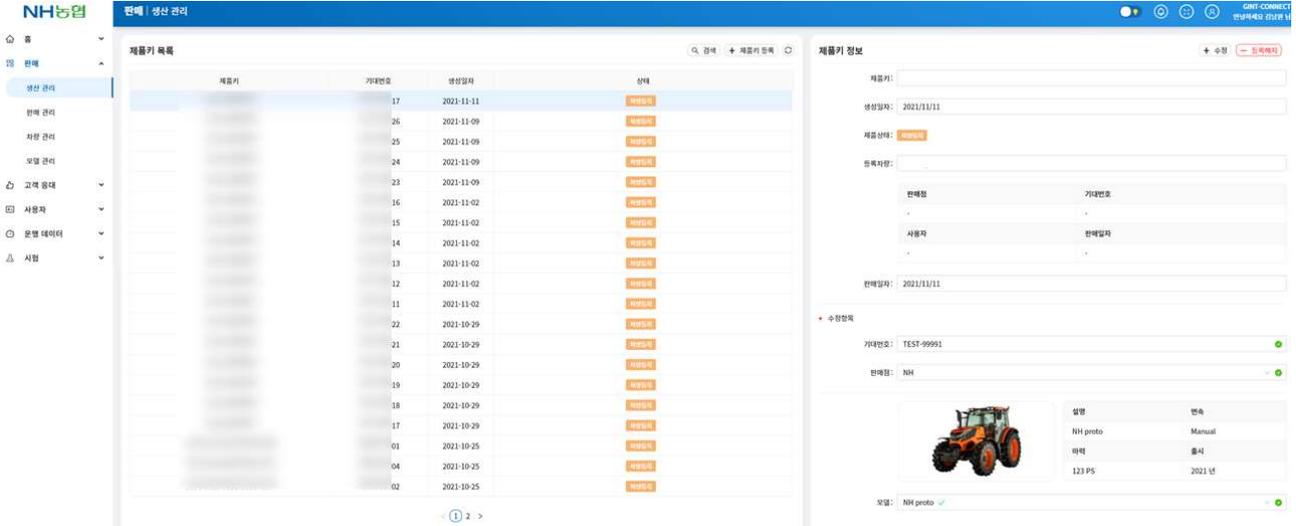
#### [ 엔진가용시간에 대한 api 매뉴얼 ]

이와 같은 매뉴얼을 작성하여 경상대 측에서 장비이력에 대한 분석을 수월히 진행 할 수 있도록 전달하였다.

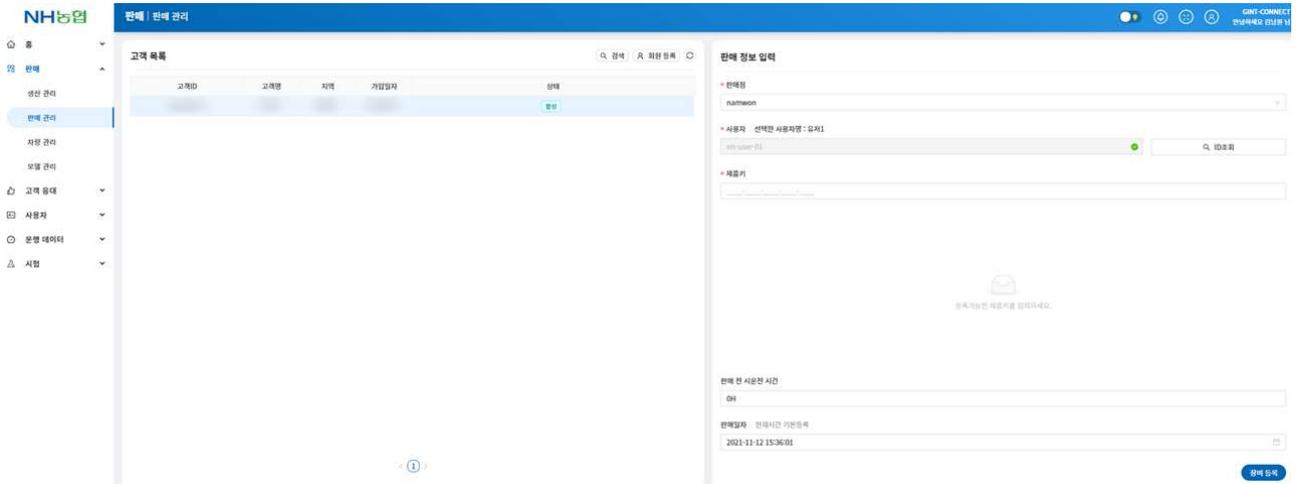
㉔ 농기계 이력 데이터 입력 시스템 개발 및 정부 지원 시스템 개발

㉔ 등록, 소유권 이전, 폐기 등 이력 데이터 관리 웹 서비스 개발

- 농기계의 생애주기를 관리하기 위한 다양한 단계의 웹 서비스를 아래와 같이 개발 하였다.



[ 출하 후 생산 직후 등록한 농기계를 위한 웹서비스 ]



[ 생산 후 판매 된 농기계를 위한 웹서비스 ]

The screenshot shows the '판매 | 차량 관리' (Sales | Vehicle Management) section. On the left is a navigation menu with options like '홈', '판매', '생산 관리', '판매 관리', '차량 관리', '오일 관리', '고객 관리', '사용자', '운영 데이터', and '시뮬'. The main area features a map at the top and a table of tractor records. The table has columns for '기대번호', '사용자ID', '사용자명', '거점지', '판매일자', '상태', and '상세'. The selected tractor (TEST-000021) is highlighted in blue. To the right, a detailed view of the tractor is shown, including a photo of an orange tractor, a table of specifications (Model: NH proto, Serial: Manual, 123 PS, etc.), and a graph titled '스스로 및 드래그 하우 자세를 보기' (View self and drag house posture) showing a line graph of distance over time.

[ 판매 후 농기계가 상용되는 것을 확인 하기 위한 웹서비스 ]

⑥ 면세유 및 농기계 구입자금 지원 관련 웹 서비스 개발

- 정부 및 유관 기관에서 농기계에 대해 적용하고 있는 면세유에 관련된 정책을 결정하기 위해 지원할 수 있는 데이터로는 농기계의 실운영거리가 중요하다고 판단하였다. 이에 따라, GPS 의 위치를 누적하여 실운영거리 데이터로 제공할 수 있도록 하였다.

The screenshot shows the '차량정보' (Vehicle Information) page. It includes a tractor image, a table of specifications, a graph titled '스스로 및 드래그 하우 자세를 보기' (View self and drag house posture) showing a line graph of distance over time, and a section titled '소모품 이력' (Consumables History) listing fuel events. The fuel events table is as follows:

이벤트	일자	운행
그리스 주유	2021-11-08	2일
엔진 오일, 오일 필터	2021-11-08	2일
브레이크 페달 유격	2021-11-08	2일
필 볼트 토오크	2021-11-08	2일

[ 일자별 운행거리를 확인할 수 있는 웹서비스 ]

	A	B	C	D
1	일시	운영시간	운영거리	
2	2021-11-0	0.5분	0.00616 Km	
3	2021-11-0	47.8분	5.38600 Km	
4	2021-11-0	59.5분	3.45963 Km	
5	2021-11-0	28.1분	1.77771 Km	
6	2021-11-0	34.3분	2.48864 Km	
7	2021-11-0	53.0분	3.46057 Km	
8	2021-11-0	24.4분	1.95533 Km	
9	2021-11-0	43.8분	1.68686 Km	
10	2021-11-0	54.3분	1.82120 Km	
11	2021-11-0	25.2분	0.54795 Km	
12	2021-11-0	37.5분	1.94223 Km	
13	2021-11-0	50.3분	4.27877 Km	
14	2021-11-0	45.5분	2.53049 Km	
15	2021-11-0	0.0분	0 Km	
16	2021-11-0	0.0분	0 Km	
17	2021-11-1	0.0분	0 Km	
18	2021-11-1	17.3분	0.12197 Km	
19				

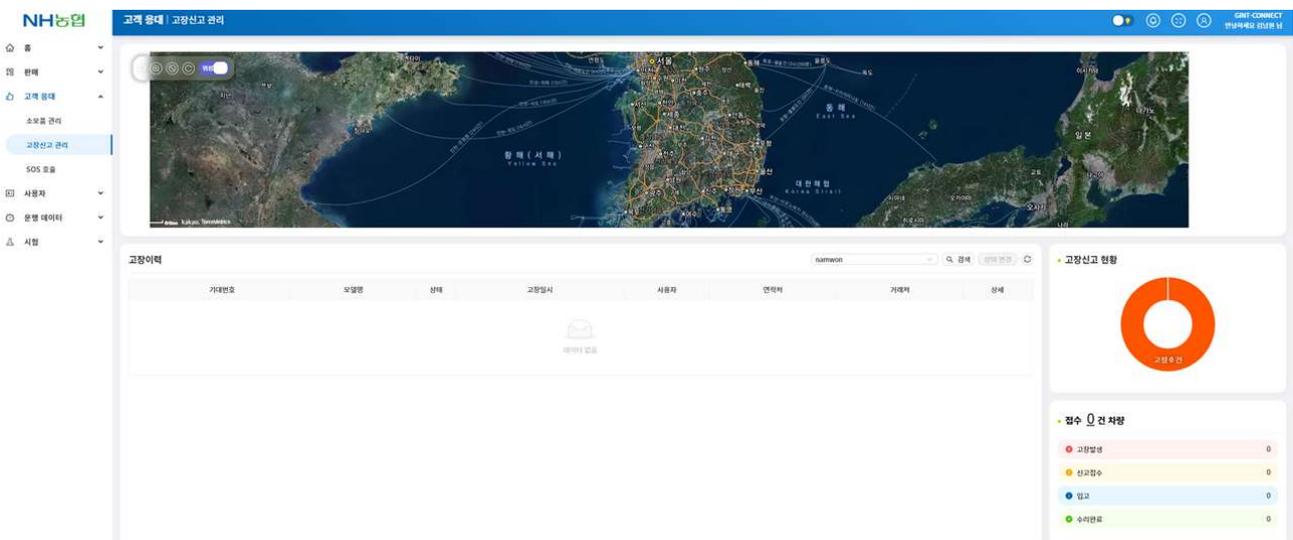
XX4E-00107 운행정보

[ 일자별 운행거리를 엑셀로 다운로드 가능한 기능 예시 ]

또한, 농기계별로 운행거리를 엑셀형식의 파일로 다운로드 받을 수 있게 하여 OA 자료의 전달도 용이하게 하도록 하였다.

© 농기계 수리이력 데이터 관리 웹 서비스 개발

- 농기계에서 CAN통신이 가능한 제품에는 일부 고장코드를 확인 할 수 있다. 이를 텔레매틱스 단말기가 확인한 뒤, 서버로 수신하는 기능을 구현하여 웹에서도 일부 고장에 대해 확인할 수 있도록 하였다.

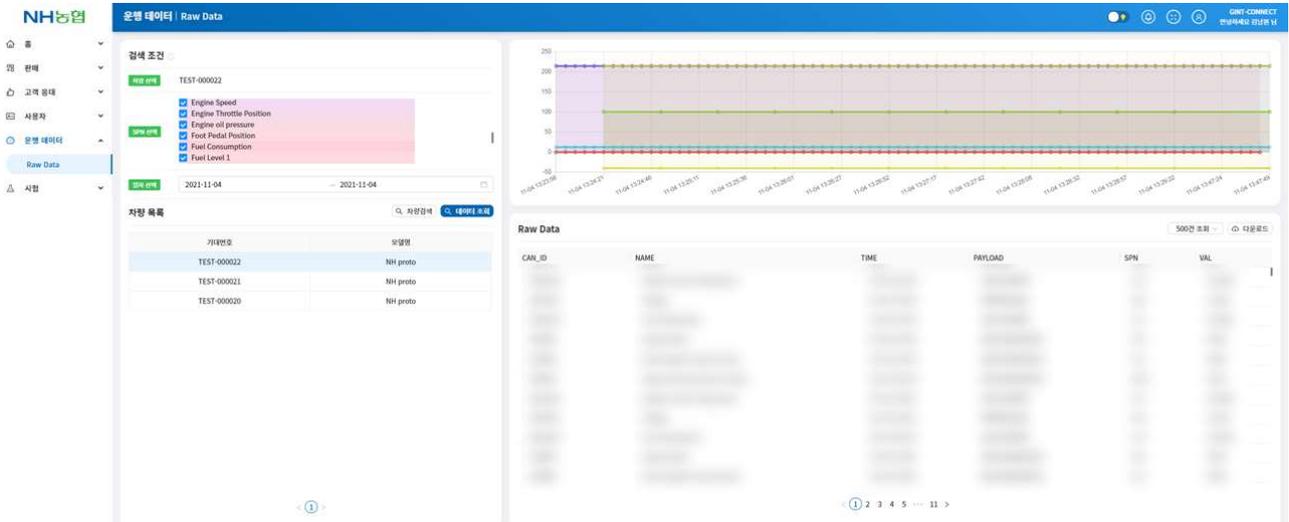


[ 등록된 농기계의 고장코드를 확인할 수 있는 웹페이지 - 현재 0건 ]

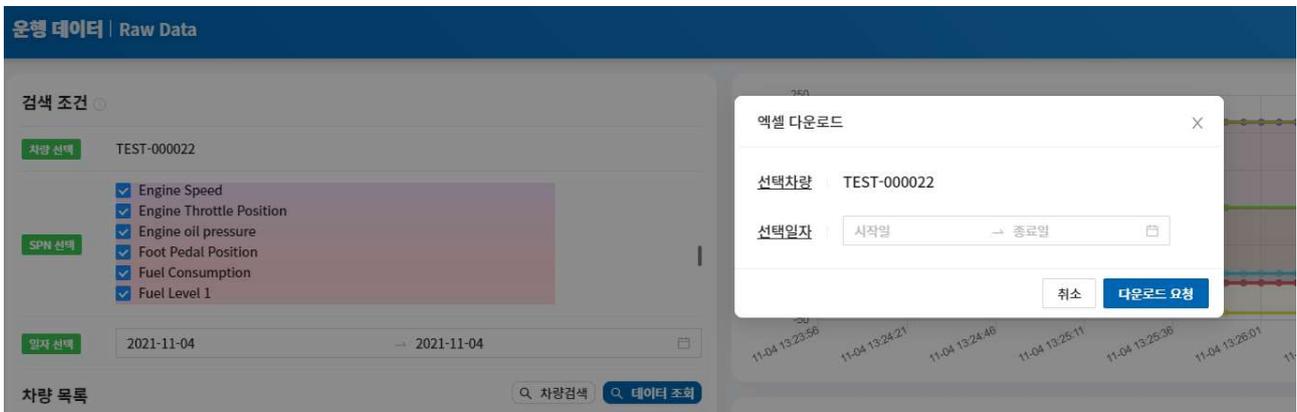
㉔ 농기계 이력 및 운행 데이터 기반 제조사 웹 서비스 개발

㉔ 농기계 운용 데이터 기반 품질 향상을 위한 제조사 웹 서비스 기능 확장

- 농기계에서 나오는 다양한 데이터들을 수집할 수 있는 기능을 단말기에 탑재하고 그 정보들을 업로드하여 그래프로 나타내주어 제조사들이 다양한 정보를 열람할 수 있도록 구성하였다. 또한 이 역시 엑셀 파일 형식으로 다운로드 받아 활용할 수 있도록 하였다.



[ 제조사를 위한 데이터 수집 및 열람 웹페이지 ]



[ 운행데이터를 다운로드 받을 수 있는 서비스 ]

㉔ 농기계 이력 데이터 블록체인 시스템 성능 시험, 모니터링 및 개선

㉔ 농기계 이력 데이터 블록체인 시스템의 성능 최적화 및 개선

- 블록체인 시스템의 트랜잭션 수를 모니터링 하면서 성능 개선
  - 필요없는 프로세스와 SQL 등의 요소 제거
  - 반복적인 테스트를 통해 정량적 목표 달성 확인



[ 블록체인 서비스 모니터링 대시 화면 ]

단 계	시험 사진	비 고																																								
블록체인 모니터링 시스템에서 관련 자료 확인		초당 트랜잭션 수를 확인한다.																																								
초당 TPS 확인	<table border="1" style="width: 100%; text-align: center;"> <tr><td></td><td>1</td><td>4</td><td>0</td><td>2000</td></tr> <tr><td></td><td>1</td><td>2</td><td>0</td><td>1800</td></tr> <tr><td></td><td>0</td><td>3</td><td>1</td><td>664</td></tr> <tr><td></td><td>1</td><td>3</td><td>0</td><td>212</td></tr> <tr><td></td><td>1</td><td>3</td><td>0</td><td>1532</td></tr> <tr><td></td><td>1</td><td>3</td><td>0</td><td>855</td></tr> <tr><td></td><td></td><td>3</td><td>0</td><td>468</td></tr> <tr><td></td><td></td><td></td><td>AVG</td><td>1075.857</td></tr> </table>		1	4	0	2000		1	2	0	1800		0	3	1	664		1	3	0	212		1	3	0	1532		1	3	0	855			3	0	468				AVG	1075.857	TPS 는 약 1075 개로 확인함
	1	4	0	2000																																						
	1	2	0	1800																																						
	0	3	1	664																																						
	1	3	0	212																																						
	1	3	0	1532																																						
	1	3	0	855																																						
		3	0	468																																						
			AVG	1075.857																																						
결과	TPS가 정량적 목표(500개 이상)를 만족함																																									

㉔ 블록체인 플랫폼 모니터링을 통한 검증 및 개선

단 계	시험 사진	비 고
블록체인 모니터링 시스템에서 관련 자료 확인		초당 트랜잭션 수를 확인한다.

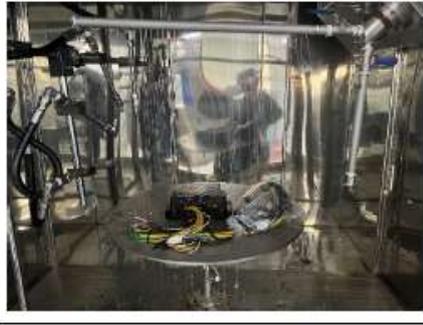


- (2) 시험 시간 : 1h
- (3) 분무 거리 : 400 mm
- (4) 평가 항목 : ① 외관 ② 기능검사
- (5) 시험품 상태 : ① 비포장 상태 ② 비전원 상태
- (6) 시험품 수 : 1 EA

2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
방수시험기	JFM-A-001 / JFM	20191007-01	2022. 11. 08	교정기술원㈜

3) 시험 방법 및 진행 이미지

	
a. 시험 셋팅 이미지 (S2)	b. 시험 셋팅 이미지 (S2)
	
c. 시험 셋팅 이미지 (S2)	-

[ 방수 성능 시험 진행 ]

4) 시험 결과

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

- 이상 전압 시험

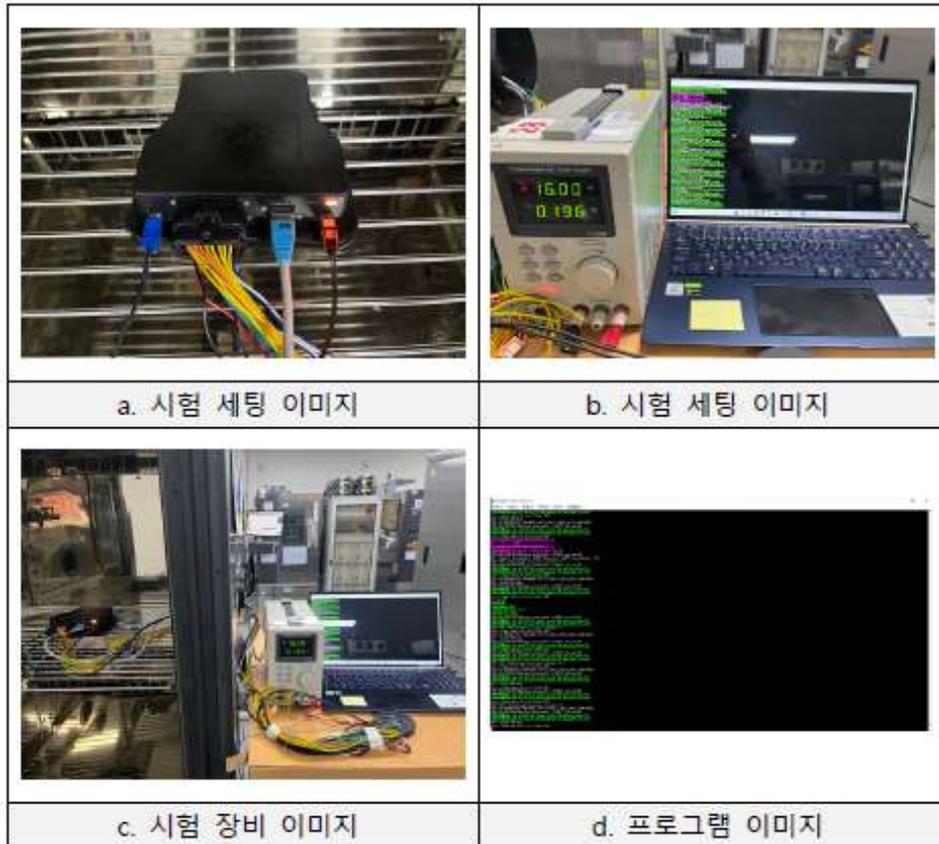
1) 시험 조건

- (1) 시험 전압 : 16 V
- (2) 시험 시간 : 48h
- (3) 시험품 상태 : ① 비포장 ② 작동 상태
- (4) 시험품 평가 : ① 외관 ② 기능검사
- (5) 시험품 수 : 1 EA

2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
항온항습기	EFL-3 / ESPEC	15011353	2022. 11. 12	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

3) 시험 방법 및 진행 이미지



[ 이상전압 시험 셋팅 ]

4) 시험 결과

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

- 내온도성 시험

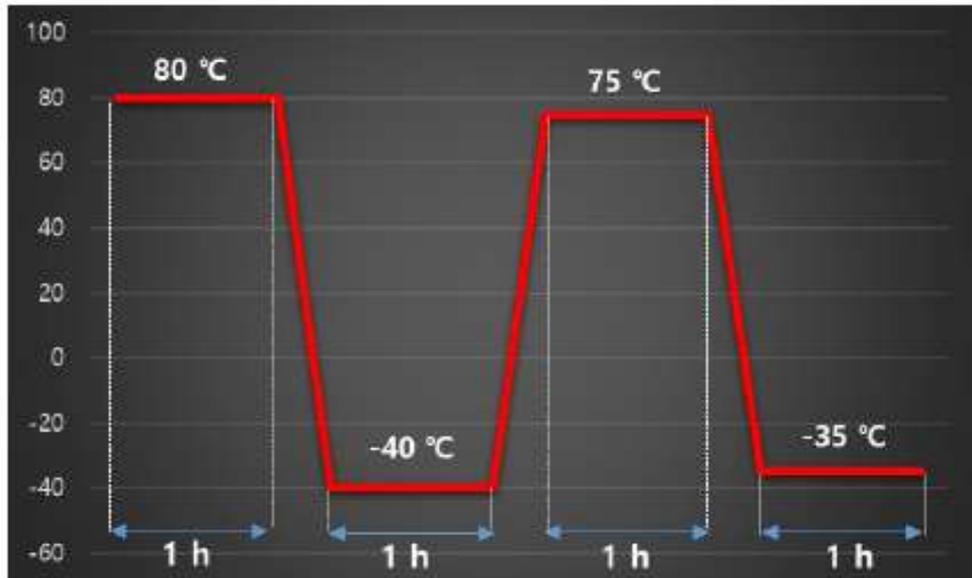
1) 시험 조건

- (1) 시험 온도 : (-40 ~ 75) °C
- (2) 시험 시간 : 4h
- (3) 시험품 상태 : ① 비포장 ② 작동 상태
- (4) 시험품 평가 : ① 외관 ② 기능검사
- (5) 시험품 수 : 1 EA

2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
항온항습기	EFL-3 / ESPEC	15011353	2022. 11. 12	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

3) 시험 방법 및 진행 이미지

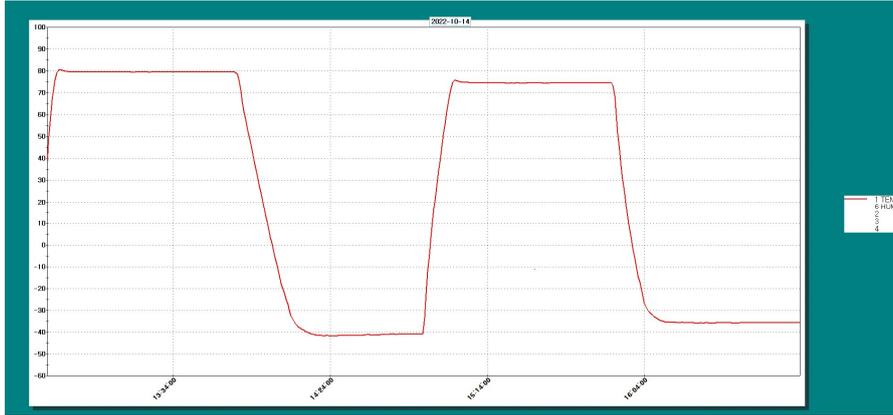


[ 내온도성 시험 방법 패턴 ]



[ 내온도성 시험 셋팅 ]

#### 4) 시험 결과



[ 실제 내온도성 시험 결과 그래프 ]

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

#### - 습도 시험

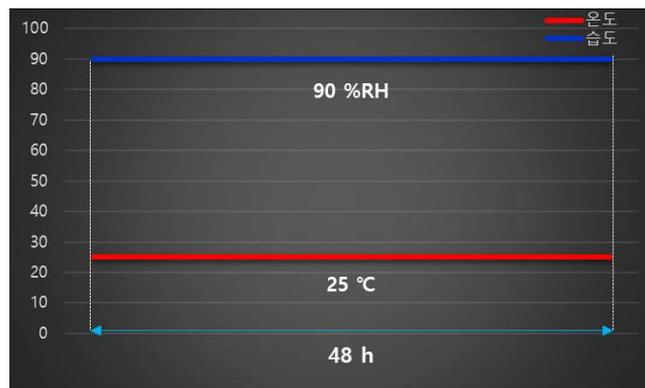
##### 1) 시험 조건

- (1) 시험 온도 : 25 °C
- (2) 시험 습도 : 90 % RH
- (3) 시험 시간 : 48 h
- (4) 시험품 상태 : ① 비포장 ② 작동 상태
- (5) 시험품 평가 : ① 외관 ② 기능검사
- (6) 시험품 수 : 1 EA

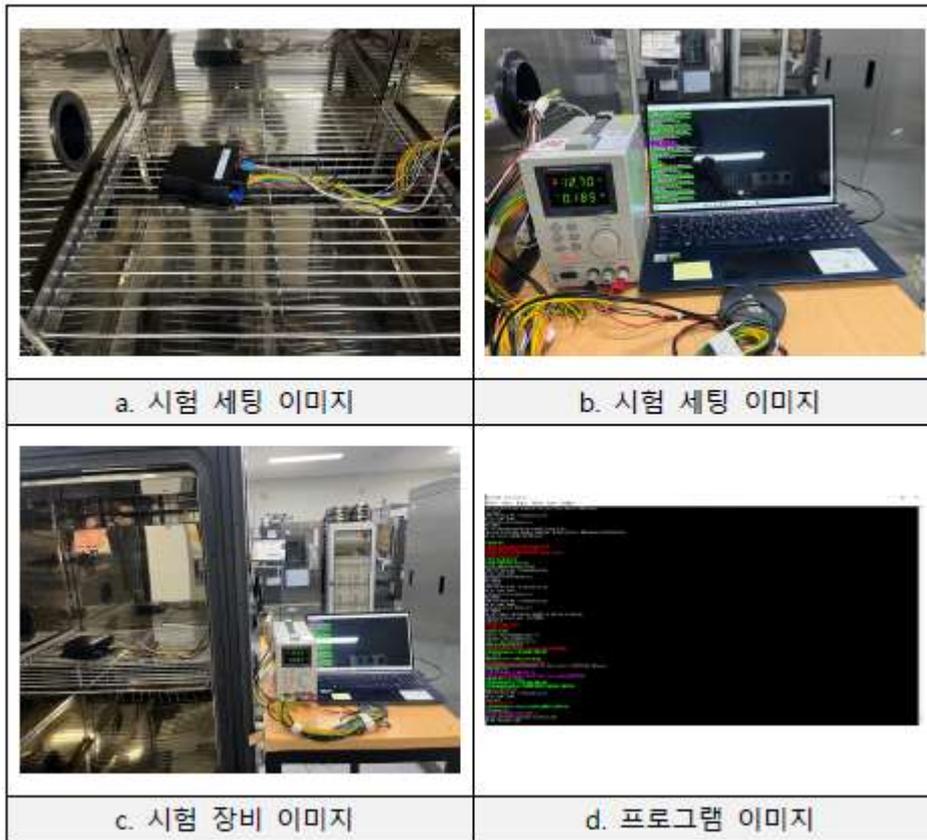
##### 2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
항온항습기	EFL-3 / ESPEC	15011353	2022. 11. 12	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

##### 3) 시험 방법 및 진행 이미지

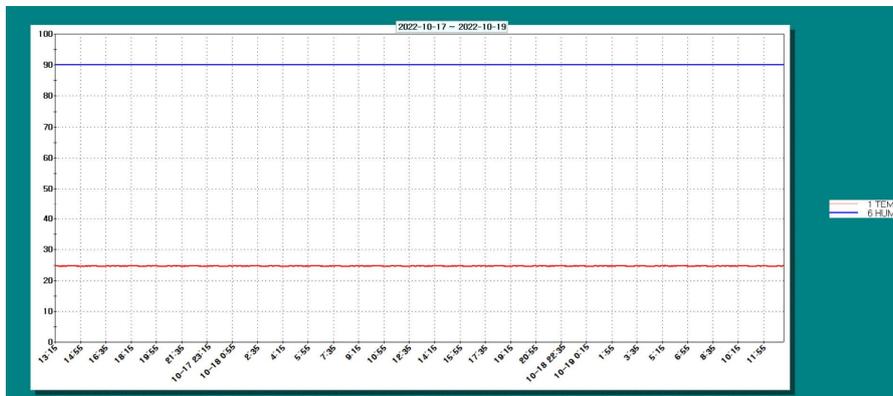


[ 습도 시험 패턴 ]



[ 습도 시험 셋팅 ]

4) 시험 결과



[ 습도 시험 결과 그래프 ]

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

- 저온방치 시험

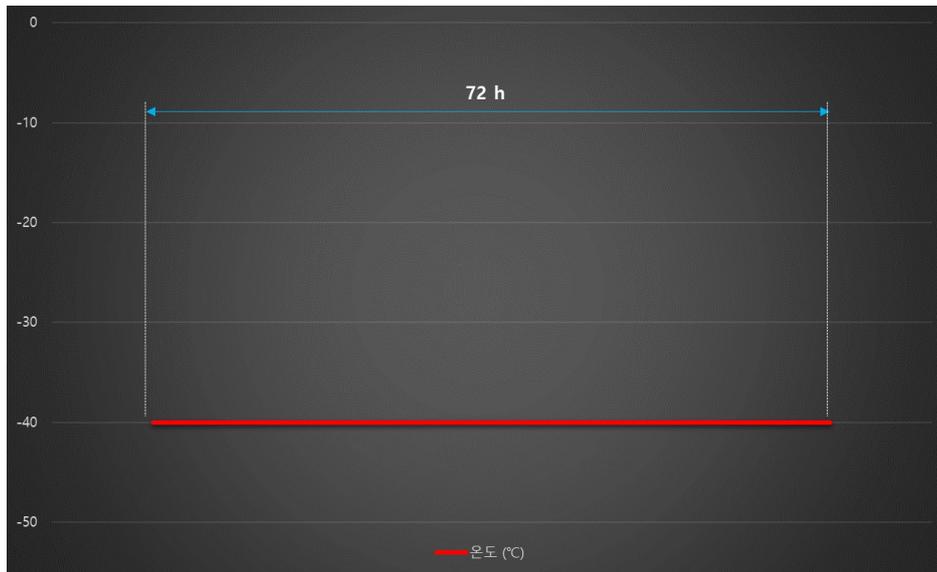
1) 시험 조건

- (1) 시험 온도 : -40 ℃
- (2) 시험 시간 : 72 h
- (3) 시험품 상태 : ① 비포장 ② 작동 상태
- (4) 시험품 평가 : ① 외관 ② 기능검사
- (5) 시험품 수 : 1 EA

2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
항온항습기	EFL-3 / ESPEC	15011353	2022. 11. 12	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

3) 시험 방법 및 진행 이미지

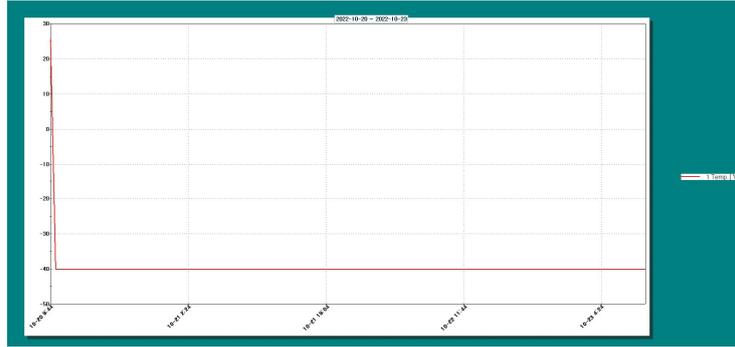


[ 저온 방치 시험 패턴 ]



[ 저온 방치 시험 셋팅 ]

#### 4) 시험 결과



[ 저온 방치 결과 그래프 ]

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

#### - 고온방치 시험

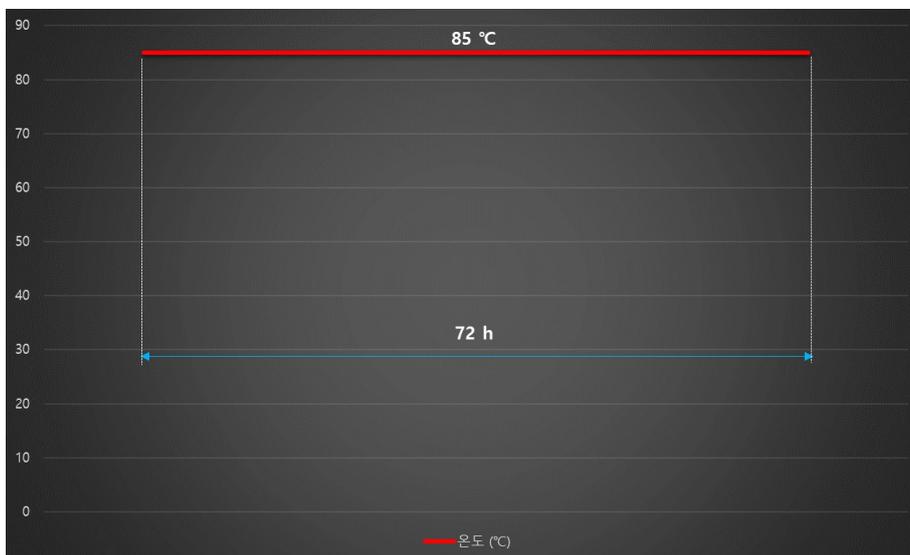
##### 1) 시험 조건

- (1) 시험 온도 : 85 °C
- (2) 시험 시간 : 72 h
- (3) 시험품 상태 : ① 비포장 ② 작동 상태
- (4) 시험품 평가 : ① 외관 ② 기능검사
- (5) 시험품 수 : 1 EA

##### 2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
항온항습기	EFL-3 / ESPEC	15011353	2022. 11. 12	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

##### 3) 시험 방법 및 진행 이미지

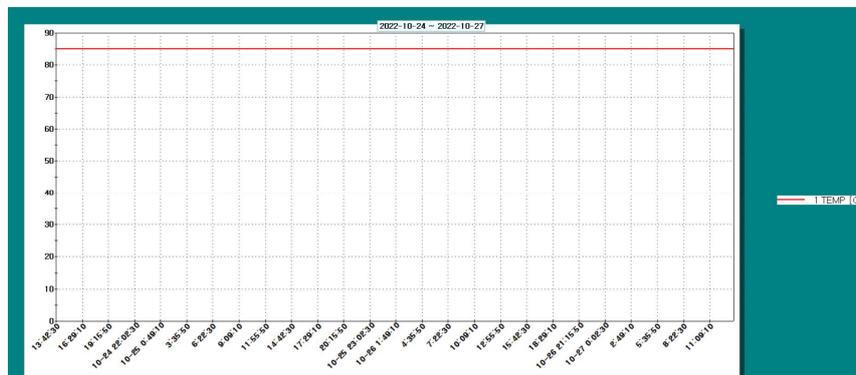


[ 고온 방치 시험 패턴 ]



[ 고온 방치/작동 시험 셋팅 ]

4) 시험 결과



[ 고온 방치 시험 결과 그래프 ]

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

- 진동 시험

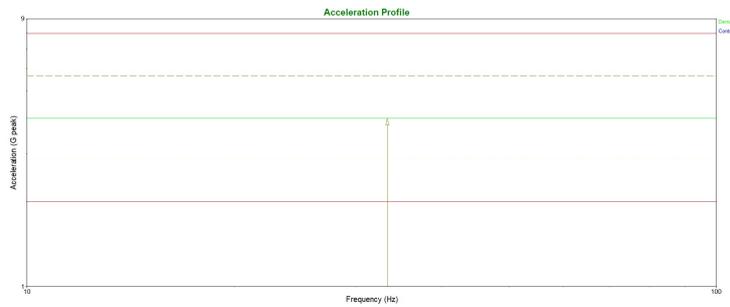
1) 시험 조건

- (1) 진동 주파수 : 33.3 Hz
- (2) 진동 가속도 : 4.0 G
- (3) 진동 방향 및 시간 : 상하(4H), 좌우(2H), 전후(2H)
- (4) 시험품 상태 : ① 비포장 ② 작동 상태
- (5) 시험품 평가 : ① 외관 ② 기능검사
- (6) 시험품 수 : 1 EA

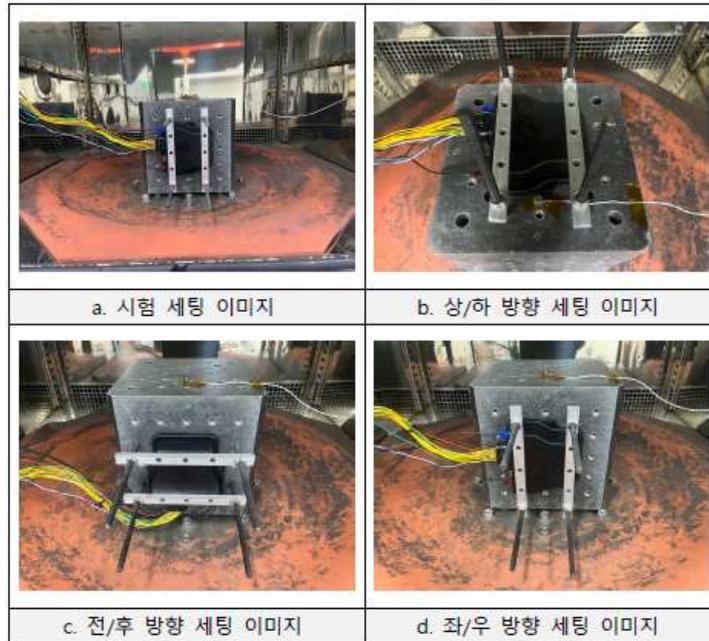
2) 시험 장비

장비명	모델명 / 제조회사	일련번호	차기 교정 예정일	교정기관
진동시험기	EDS-6000LS3-445 / FAMTECH	D1707273	2023. 07. 13	SICT
Power supply	TL350P / TOYOTECH	19100043	2022. 12. 13	SICT

3) 시험 방법 및 진행 이미지

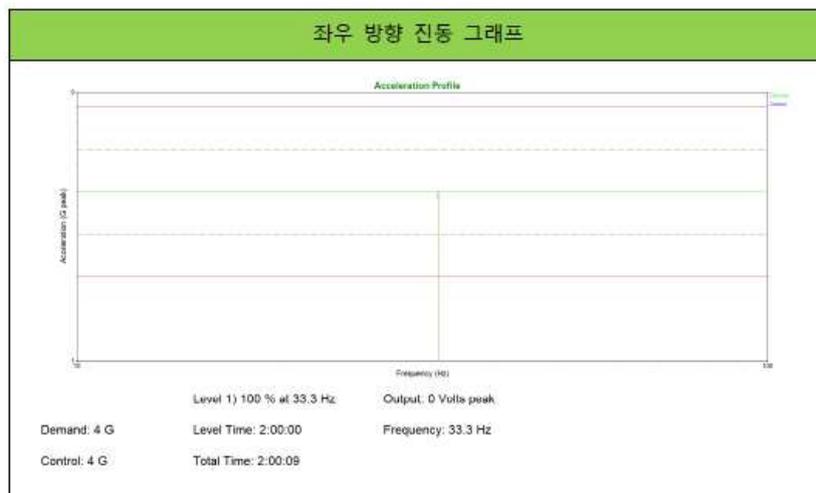


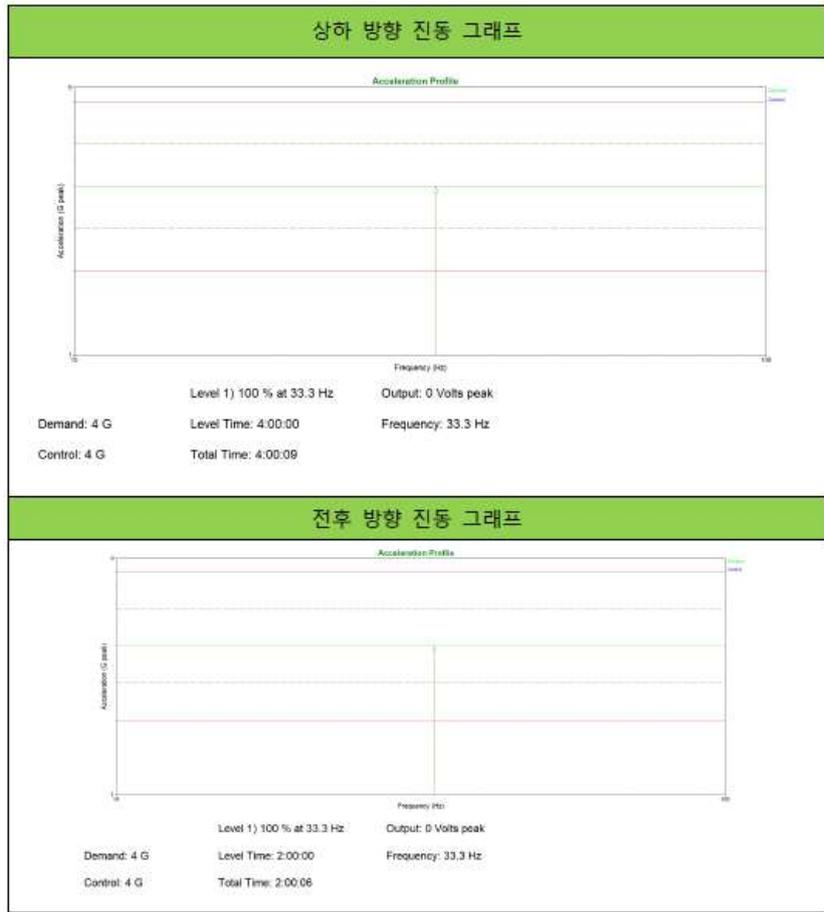
[ 진동 시험 패턴 ]



[ 진동 시험 셋팅 ]

4) 시험 결과





[ 각 방향별 진동 결과 그래프 ]

제품명	구분	점검 항목	시험결과
E1-OC-X (블록체인)	외관 검사	- 시험품 변형 및 Crack 확인	이상 없음 [※부록 1. 참조]
	기능 검사	- 동작 확인	

- 내환경 시험 결과 요약

Leg	No.	시험 항목	시험 조건	평가 기준	시험 결과
-	1	방수 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	2	이상전압 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	3	내온도성 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	4	습도 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	5	저온방치 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	6	고온방치 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음
	7	진동 시험	고객사 요청 규격	신뢰성 시험 후 외관 및 기능검사	이상 없음

- EMC 환경 시험 장비

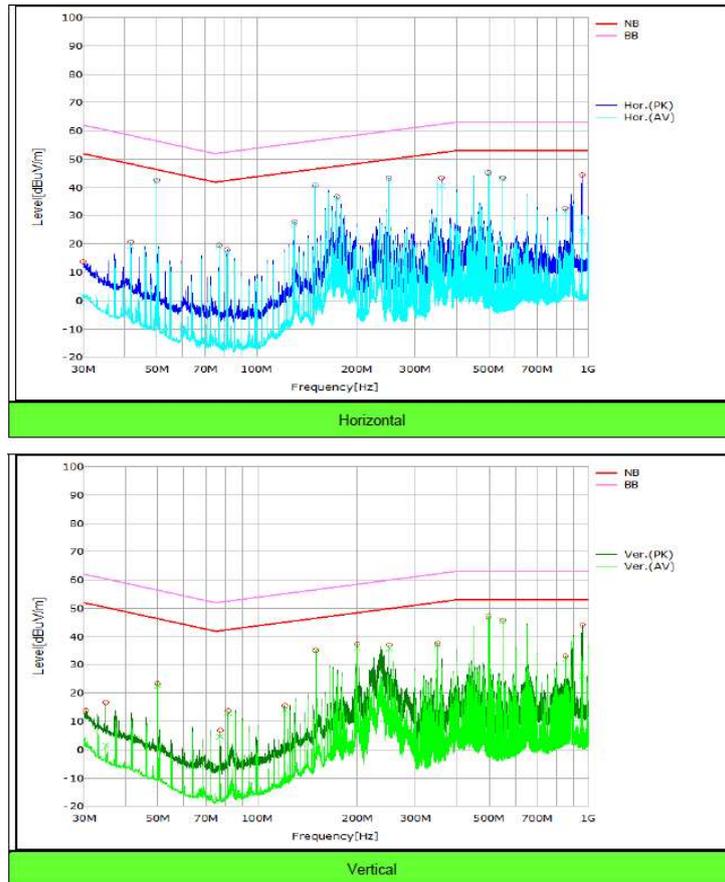
No.	장비명	모델명	시리얼 번호	제조사	교정일자 (yy/mm/dd)	차기 교정일 (yy/mm/dd)
<b>Radiated Emission</b>						
1	EMI Test Receiver	ESCI 3	101175	Rohde & Schwarz	22/11/03	23/11/03
2	Biconical Antenna	VHA 9103	2952	Schwarzbeck	22/07/05	24/07/05
3	Log-Periodic Antenna	VULP 9118A	549	Schwarzbeck	22/07/04	24/07/04
4	LISN	NNBM8124	1020	Schwarzbeck	21/12/23	22/12/23
5	LISN	NNBM8124	1023	Schwarzbeck	21/12/23	22/12/23
6	Pre- Amplifier	TK-PA6S	130012	TESTEK	22/07/18	23/07/18
<b>Conducted Transient Emission</b>						
1	Ultra Compact Simulator	UCS 200 N50	P1844224128	EM TEST	22/07/04	23/07/04
2	Voltage Drop Simulator	VDS200Q	P1910227627	EM TEST	22/07/04	23/07/04
3	Single Line Artificial Network	AN 2050N	0511-02	EM TEST	21/12/23	22/12/23
4	Electronic Switch	BS 200N100	P1646187526	EM TEST	N/A	N/A
5	Shunt resistor box	RS-BOX	P1629182020	EM TEST	N/A	N/A
6	Voltage probe	P5100A	N/A	Tektronix	22/08/24	23/08/24
7	Digital Phosphor Oscilloscope	TDS5104B	B021559	Tektronix	22/09/30	23/09/30
<b>Conducted Immunity on Power Line</b>						
1	Ultra Compact Simulator	UCS 200 N50	P1844224128	EM TEST	22/07/04	23/07/04
2	Voltage Drop Simulator	VDS200Q	P1910227627	EM TEST	22/07/04	23/07/04

- EMC(Radiated Emission) 환경 시험 셋업



[ Radiated Emission 시험 사진 ]

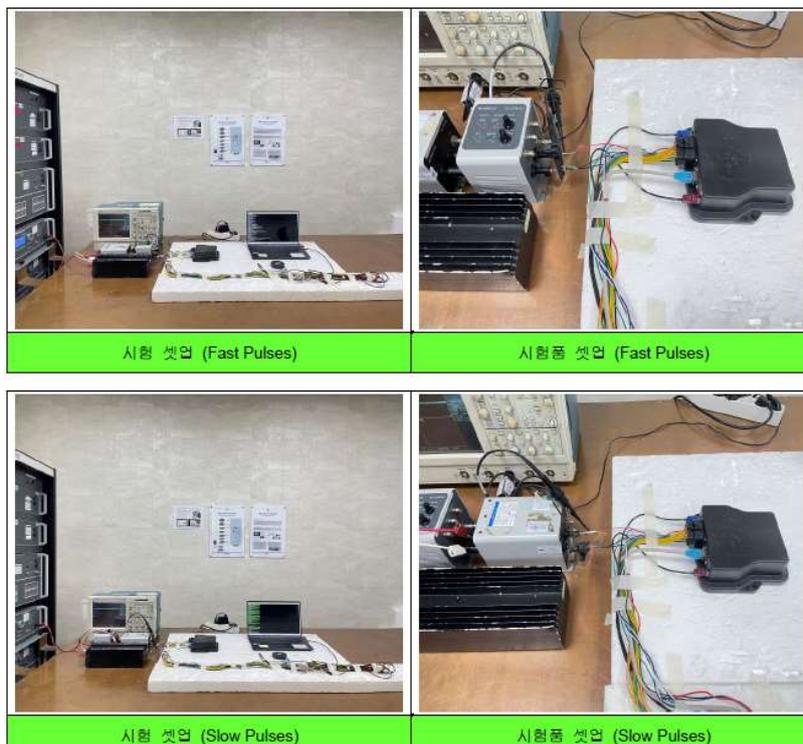
- EMC(Radiated Emission) 환경 시험 결과



[ 시험 결과 그래프 ]

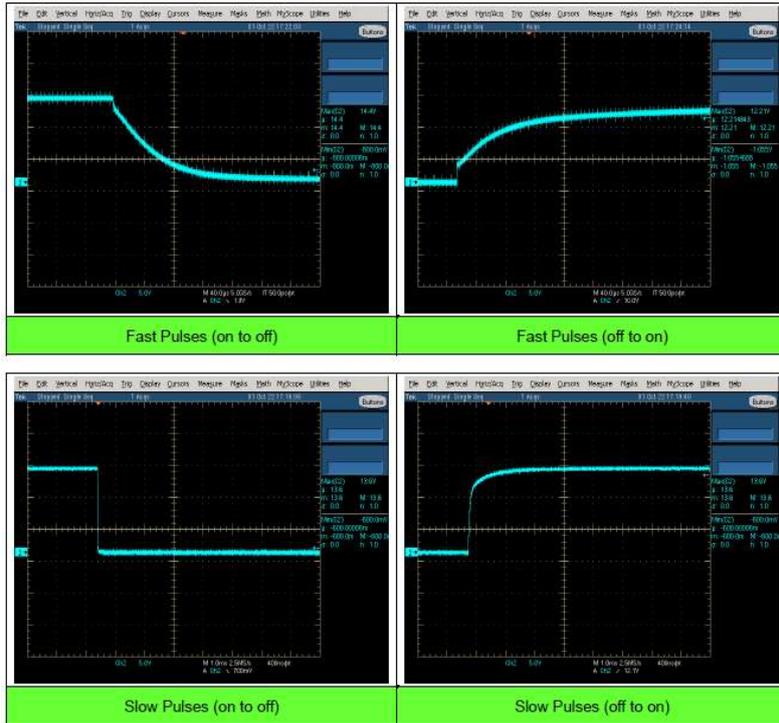
- 측정 수치의 결과가 가이드라인보다 모두 낮음을 확인

- EMC(Conducted Transient Emission) 환경 시험 셋업



[ Conducted Transient Emission 테스트 셋업 ]

- EMC(Conducted Transient Emission) 환경 시험 결과



[ Conducted Transient Emission 측정 결과 ]

- 측정 수치의 결과가 가이드라인보다 모두 낮음을 확인

Test Pulse	Fast Pulses				Slow Pulses			
	On to Off		Off to On		On to Off		On to Off	
Sampling rate	5.00 GS/s		5.00 GS/s		2.50 MS/s		2.50 MS/s	
Trigger level	1.90 V		10.00 V		0.70 V		12.10 V	
Peak amplitude	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Test result	-14.30 V	0.90 V	-14.56 V	-1.29 V	-14.10 V	0.10 V	-14.10 V	0.10 V

- EMC(Conducted Immunity on Power Line) 환경 시험 셋업



[ Conducted Immunity on Power Line 시험 셋팅 ]

Test pulse	Test severity level III U <sub>s</sub> (V)		Minimum number of pulses or test time	Burst cycle/pulse repetition time		Functional status for systems:
	12 V system	24 V system		Min.	Max.	
1	-75	-450	5 000 pulses	0.5 s	5 s	D
2a	+37	+37	5 000 pulses	0.2 s	5 s	D
2b	10	20	10 pulses	0.5 s	5 s	D
3a	-112	-150	1 h	90 ms	100 ms	D
3b	+75	+150	1 h	90 ms	100 ms	D
4	-6	-12	1 pulse	-	-	D

[ 시험 조건 요약 ]

- EMC(Conducted Immunity on Power Line) 환경 시험 결과

인가펄스	인가전압	현상	FPSC
1	-75	시험 중 전원이 Off/On 반복되나, 시험 후 정상 동작함.	Class C
2a	+37	시험 중/후 정상 동작함.	Class A
2b	10	시험 중 전원이 Off/On 반복되나, 시험 후 정상 동작함.	Class C
3a	-112	시험 중/후 정상 동작함.	Class A
3b	+75	시험 중/후 정상 동작함.	Class A
4	-6	시험 중/후 정상 동작함.	Class A

[ Conducted Immunity on Power Line 결과 정리 ]

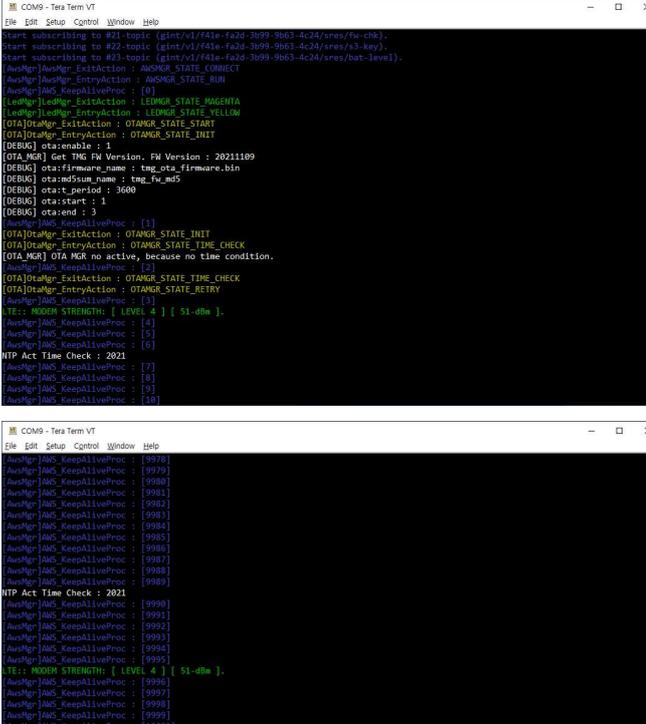
· 모든 시험이 Class D 이상임을 확인

- EMC 환경 시험 결과 요약

시험 항목	시험품 No.	동작 모드	인가레벨 및 조건	FPSC	시험결과 Pass/Fail
Radiated Emission	#1	Mode 1	30 MHz to 1 000 MHz	Limits 를 초과하지 않음	Pass
Conducted Transient Emission	#1	Mode 1	Positive: +75 V Negative: -100 V	Limits 를 초과하지 않음	Pass
Conducted Immunity on Power Line	#1	Mode 1	Level III	Class D	Class C Pass

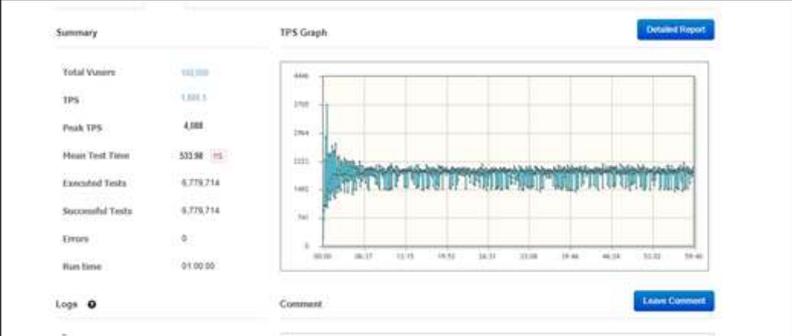
㉔ 농기계 이력 데이터 수집 시스템 서버 성능 시험 및 개선

- 서버성능을 필요 수준 이상으로 사양 조정 및 세부 내역 개발 후 테스트
- 단말기 통신 성공률 테스트 수행

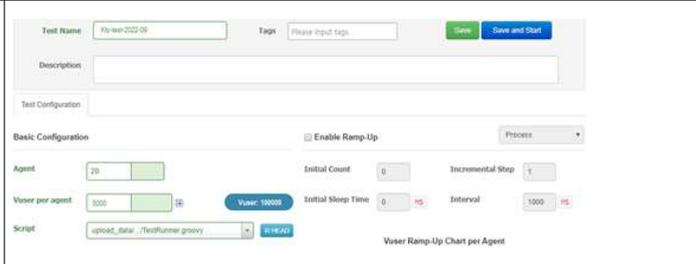
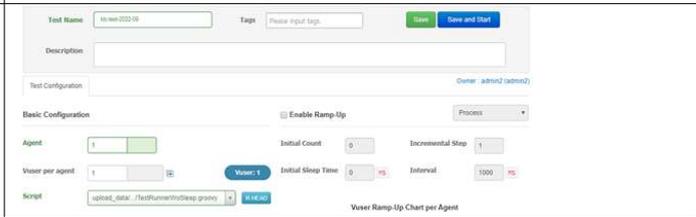
단 계	시험 사진	비 고
<p>단 말기 에 서 KeepAlive 신호 생성</p>		<p>총 1000개의 KeepAlive 신호를 생성한다.</p>
<p>업 로 드된 신호 확인</p>		<p>AWS 콘솔에서 업로드 여부를 확인한다.</p>
<p>결과</p>	<p>1000개 모두 수신 하였으므로 통신성공률 100%로 시험기준(95%이상)을 만족함</p>	

- 데이터 수집/농업인/제조사/정비지원 서버 수용 가입자 테스트 수행

단 계	시험 사진	비 고
<p>테 스트 시작</p>		<p>테스트를 시작하고 약 1시간정도 진행 상황을 확인한다.</p>

테스트 종료		테스트가 완료되면 테스트 결과를 CSV 파일로 export 시킨다.																																																																		
MTT 최대값 확인	<table border="1" data-bbox="470 504 1141 880"> <tr><td>100,000</td><td>4560</td><td>0</td><td>532.123</td><td>82.334</td><td>2152</td></tr> <tr><td>100,000</td><td>5315</td><td>0</td><td>489.655</td><td>86.844</td><td>1899</td></tr> <tr><td>100,000</td><td>4896</td><td>0</td><td>512.284</td><td>88.332</td><td>1344</td></tr> <tr><td>100,000</td><td>2331</td><td>0</td><td>511.223</td><td>84.212</td><td>2111</td></tr> <tr><td>100,000</td><td>8589</td><td>0</td><td>546.215</td><td>89.224</td><td>2354</td></tr> <tr><td>100,000</td><td>5531</td><td>0</td><td>612.234</td><td>85.331</td><td>1897</td></tr> <tr><td>100,000</td><td>2288</td><td>0</td><td>488.367</td><td>81.212</td><td>1356</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td>MAX</td><td>673.65</td><td></td><td></td></tr> <tr><td></td><td></td><td>MID</td><td>543.659</td><td></td><td></td></tr> <tr><td></td><td></td><td>AVG</td><td>533.98</td><td></td><td></td></tr> </table>	100,000	4560	0	532.123	82.334	2152	100,000	5315	0	489.655	86.844	1899	100,000	4896	0	512.284	88.332	1344	100,000	2331	0	511.223	84.212	2111	100,000	8589	0	546.215	89.224	2354	100,000	5531	0	612.234	85.331	1897	100,000	2288	0	488.367	81.212	1356									MAX	673.65					MID	543.659					AVG	533.98			Mean Test Time의 최대값은 약 674 ms, 평균값은 약 534ms로 확인함.
100,000	4560	0	532.123	82.334	2152																																																															
100,000	5315	0	489.655	86.844	1899																																																															
100,000	4896	0	512.284	88.332	1344																																																															
100,000	2331	0	511.223	84.212	2111																																																															
100,000	8589	0	546.215	89.224	2354																																																															
100,000	5531	0	612.234	85.331	1897																																																															
100,000	2288	0	488.367	81.212	1356																																																															
		MAX	673.65																																																																	
		MID	543.659																																																																	
		AVG	533.98																																																																	
결과	Mean Test Time의 최대값은 약 674 ms, 평균값은 약 534ms로 확인되어 1초 이하의 시험기준을 만족함																																																																			

- 데이터 수집/농업인/제조사/정비지원 서버 평균 응답 속도

단 계	시험 사진	비 고
테스트 에이전트 시작		1번 시험항목과 동일하게 조건을 설정하고 100,000개의 User를 실행한다.
추가 테스트 에이전트 생성		100,000개의 User가 실행된 조건에서 추가적인 에이전트를 생성한다.
테스트 스크립트 업로드	<pre data-bbox="438 1630 1134 1854"> 97 ---private String getUniqueId() { 98 ---String id = "test" + ((grinder.getAgentNumber() + 1) * 1000) + ((grinder.getProcessNumber() + 1) * 100); 99 ---return id; 100 ---grinder.getUniqueTag().append(id); 101 ---return id; 102 ---} 103 ---private void prepare() throws Exception { 104 ---//data.equipment_id = getUniqueId(); 105 ---//device_id = getUniqueId(); 106 ---//device_ip = getIpAddress(getDeviceId()); 107 ---//device_hostname = getDeviceName(getDeviceId()); 108 ---//device_username = getDeviceUsername(getDeviceId()); 109 ---//device_password = getDevicePassword(getDeviceId()); 110 ---//device_hostname = getDeviceName(getDeviceId()); 111 ---//device_username = getDeviceUsername(getDeviceId()); 112 ---//device_password = getDevicePassword(getDeviceId()); 113 ---//device_hostname = getDeviceName(getDeviceId()); 114 ---//device_username = getDeviceUsername(getDeviceId()); 115 ---//device_password = getDevicePassword(getDeviceId()); 116 ---} </pre>	테스트를 위한 스크립트를 업로드한다.
테스트 시작		서버에 데이터를 10,000회 업로드하도록 설정하여 테스트를 실행한다.

<p>테스트 종료</p>		<p>테스트가 완료되면 테스트 결과를 CSV파일로 export시킨다.</p>
<p>MTT 최댓값 확인</p>		<p>Mean Test Time의 평균값은 약 840ms로 확인함.</p>
<p>결과</p>	<p>Mean Test Time의 평균값은 약 840ms로 확인되어 3초 이하의 시험기준을 만족함</p>	

• 블록체인 네트워크 초당 트랜잭션 수

<p>블록체인 모니터링 시스템에서 관련 자료 확인</p>	<p>시험 사진</p>	<p>비고</p> <p>초당 트랜잭션 수를 확인한다.</p>
---------------------------------	--------------	-----------------------------------

초당 TPS 확인	1	4	0	2000	TPS 는 약 1075 개로 확 인함
	1	2	0	1800	
	0	3	1	664	
	1	3	0	212	
	1	3	0	1532	
	1	3	0	855	
		3	0	468	
				AVG	
결과	TPS가 정량적 목표(500개 이상)을 만족함				

· 블록체인 네트워크 초당 업로드 속도

단 계	시험 사진	비 고																																	
블록체인 모니터링 시스템에 서 관련 자료 확인	<p>The screenshot shows a blockchain monitoring interface. On the left, there are summary statistics for nodes (노드) and transactions (거래). The main area displays a table of nodes with columns for ID, Name, Status, Config, and Anchor. Below this, there is a detailed view of a specific transaction with columns for ID, Hash, Status, and other metadata.</p>	초당 업 로드 수 를 확인 한다.																																	
초당 업로 드수를 확 인	<table border="1"> <tr> <td>f1afd43cae4ab8892ae0cef69262e6889f1c9689</td> <td>0</td> <td>2111</td> </tr> <tr> <td>c03b6ba96121472b2292393b56afa52049a5c6e</td> <td>0</td> <td>1958</td> </tr> <tr> <td>f1854aadec4bea98738788f02420366a3e566</td> <td>1</td> <td>1580</td> </tr> <tr> <td>f75fdf0522018e01ecb540e3858d451dc60a055f</td> <td>1</td> <td>1423</td> </tr> <tr> <td>362b1195db59167adf5d5fa5fa0e059b250e52</td> <td>1</td> <td>2500</td> </tr> <tr> <td>f3c56ffb93cf196b7e4b9db7286daf1490766d</td> <td>0</td> <td>1333</td> </tr> <tr> <td>8aaf1868fc683d14932aec0f9198a1ebd205c</td> <td>1</td> <td>3232</td> </tr> <tr> <td>630e9f9a78c97209def6cd8daba2d133d920c9</td> <td>1</td> <td>2231</td> </tr> <tr> <td>ae7d3decdd641e918a3280c5928aae13db37a7</td> <td>1</td> <td>2266</td> </tr> <tr> <td>i81077d28342c60ea7497ca4963ca0a768b7bc2</td> <td>0</td> <td>2131</td> </tr> <tr> <td></td> <td>AVG</td> <td>2076.5</td> </tr> </table>	f1afd43cae4ab8892ae0cef69262e6889f1c9689	0	2111	c03b6ba96121472b2292393b56afa52049a5c6e	0	1958	f1854aadec4bea98738788f02420366a3e566	1	1580	f75fdf0522018e01ecb540e3858d451dc60a055f	1	1423	362b1195db59167adf5d5fa5fa0e059b250e52	1	2500	f3c56ffb93cf196b7e4b9db7286daf1490766d	0	1333	8aaf1868fc683d14932aec0f9198a1ebd205c	1	3232	630e9f9a78c97209def6cd8daba2d133d920c9	1	2231	ae7d3decdd641e918a3280c5928aae13db37a7	1	2266	i81077d28342c60ea7497ca4963ca0a768b7bc2	0	2131		AVG	2076.5	초 당 업 로드 수 는 약 2076개 로 확인 함
f1afd43cae4ab8892ae0cef69262e6889f1c9689	0	2111																																	
c03b6ba96121472b2292393b56afa52049a5c6e	0	1958																																	
f1854aadec4bea98738788f02420366a3e566	1	1580																																	
f75fdf0522018e01ecb540e3858d451dc60a055f	1	1423																																	
362b1195db59167adf5d5fa5fa0e059b250e52	1	2500																																	
f3c56ffb93cf196b7e4b9db7286daf1490766d	0	1333																																	
8aaf1868fc683d14932aec0f9198a1ebd205c	1	3232																																	
630e9f9a78c97209def6cd8daba2d133d920c9	1	2231																																	
ae7d3decdd641e918a3280c5928aae13db37a7	1	2266																																	
i81077d28342c60ea7497ca4963ca0a768b7bc2	0	2131																																	
	AVG	2076.5																																	
결과	업로드 수가 정량적 목표(2000개 이상)을 만족함																																		

㉔ 농기계 이력 데이터 입력 시스템 개발 및 정부 지원 시스템 상용화 준비

Ⓐ 농기계 이력 데이터 관리를 위한 정부 지원 시스템 실증 및 개선

- 실증 참여 기관인 농기계 임대사업소를 대상으로 한 시스템 개발

실증에 참여한 임대사업소에 달려있는 단말기를 통해 각 트랙터의 구분을 하고, 이를 통해 대  
여 및 반납에 대한 추적이 가능하도록 구성함

**NH농협** 로그아웃

블록체인 데이터

- 테스트차량 정보
- 차량이동 정보
- 작업지역 이력 정보
- 상/하반기 작업정보
- 블록체인 송/수신

### 테스트차량 정보

Show 10 entries Search:

ID	VIN	중운행일수	운행시작일	마지막운행일
vonxcsp000017	TEST-000017	108일	2021-11-01	2022-09-03
vonxcsp000018	TEST-000018	105일	2021-11-01	2023-01-09
vonxcsp000023	TEST-000023	42일	2021-11-09	2022-12-22
vonxcsp000025	TEST-000025	171일	2021-11-09	2023-01-25
vonxcsp000026	TEST-000026	48일	2021-11-09	2022-09-20

Showing 1 to 5 of 5 entries Previous 1 Next

[ 임대사업소 트랙터 리스트 ]

- 임대 대상 트랙터들의 이동 위치를 월별로 확인 가능한 시스템 개발
- 임대사업소 운영의 가장 필요한 정보인 트랙터의 위치에 대해 GPS기반 정보를 지도에 표기하여 위치 파악 및 지역이탈에 대해 확인 가능하도록 개발함

**NH농협** 로그아웃

블록체인 데이터

- 테스트차량 정보
- 차량이동 정보
- 작업지역 이력 정보
- 상/하반기 작업정보
- 블록체인 송/수신

### 차량이동 정보

차량선택 (\*) 연도  
 장호원 23 (vonxcsp000023) 2022

월  
4월

검색 열기/닫기

차량이동 정보 RAW Data

Logged in as: 김호관리자 (gint-)

**NH농협** 로그아웃

블록체인 데이터

- 테스트차량 정보
- 차량이동 정보
- 작업지역 이력 정보
- 상/하반기 작업정보
- 블록체인 송/수신

### 차량이동 정보

차량선택 (\*) 연도  
 정읍 26 (vonxcsp000026) 2022

월  
5월

검색 열기/닫기

차량이동 정보 RAW Data

Logged in as: 김호관리자 (gint-admin)

[ 임대 대상 트랙터들의 위치를 지도에 표시하여 트래킹 ]

- ⑥ 정부 및 시군구 사용자 피드백을 바탕으로 한 서비스 사용성 개선
  - 허용지역 외 사용에 대한 일목요연한 내용을 보여주기 위한 서비스 개발  
임대 사용에 대해 가장 대표적인 불법이 지역 이탈이며, 이를 간편히 모니터링하기 위한 시스템 개발함

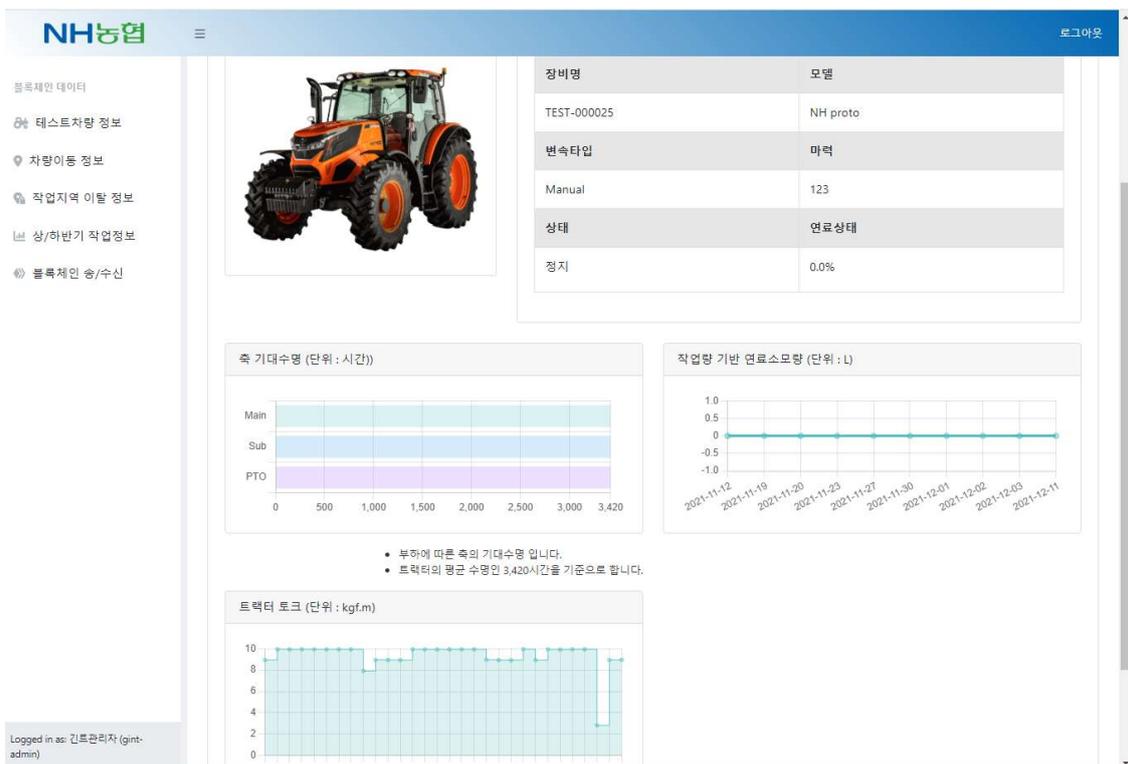
**작업지역 이탈 정보**  
총 레코드 수 : 11건  
Show 10 entries Search:

기대번호	기대명	이탈일자	주소	이탈시작시간	이탈종료시간
vonxsp000018	상주 18	2022-02-18	경상북도 예천군 풍양면 청운리	2022-02-18 02:24:01.0	2022-02-18 06:21:36.0
vonxsp000023	장호원 23	2022-05-11	충청북도 음성군 갑곡면 주천리	2022-05-11 00:28:57.0	2022-05-11 01:00:54.0
vonxsp000023	장호원 23	2022-05-11	충청북도 음성군 갑곡면 원당리	2022-05-11 00:57:33.0	2022-05-11 00:58:39.0
vonxsp000023	장호원 23	2022-05-03	충청북도 음성군 갑곡면 주천리	2022-05-03 22:52:26.0	2022-05-03 23:05:12.0
vonxsp000023	장호원 23	2022-05-03	충청북도 음성군 갑곡면 원당리	2022-05-03 22:54:31.0	2022-05-03 22:55:21.0
vonxsp000023	장호원 23	2022-04-10	충청북도 음성군 갑곡면 주천리	2022-04-10 22:07:58.0	2022-04-10 22:49:08.0
vonxsp000023	장호원 23	2022-04-10	충청북도 음성군 갑곡면 원당리	2022-04-10 22:10:24.0	2022-04-10 22:10:44.0
vonxsp000025	정을 25	2022-11-11	전라북도 고창군 아산면 반암리	2022-11-11 11:15:31.0	2022-11-11 16:55:24.0
vonxsp000025	정을 25	2022-11-10	전라북도 고창군 아산면 반암리	2022-11-10 07:19:29.0	2022-11-10 16:20:38.0
vonxsp000025	정을 25	2022-11-09	전라북도 고창군 아산면 반암리	2022-11-09 13:44:58.0	2022-11-09 16:26:35.0

Showing 1 to 10 of 11 entries Previous 1 2 Next

[ 임대사업소 영외 지역 이탈 정보 리스트업 화면 ]

- ⑦ 정부 지원 시스템을 통한 AI 알고리즘 결과 연동
  - 트랙터 수명 예측 AI 알고리즘을 적용하여, 일부 트랙터에 대한 수명 예측 총남대와 협업한 알고리즘을 바탕으로 축 기대수명의 잔여를 표기



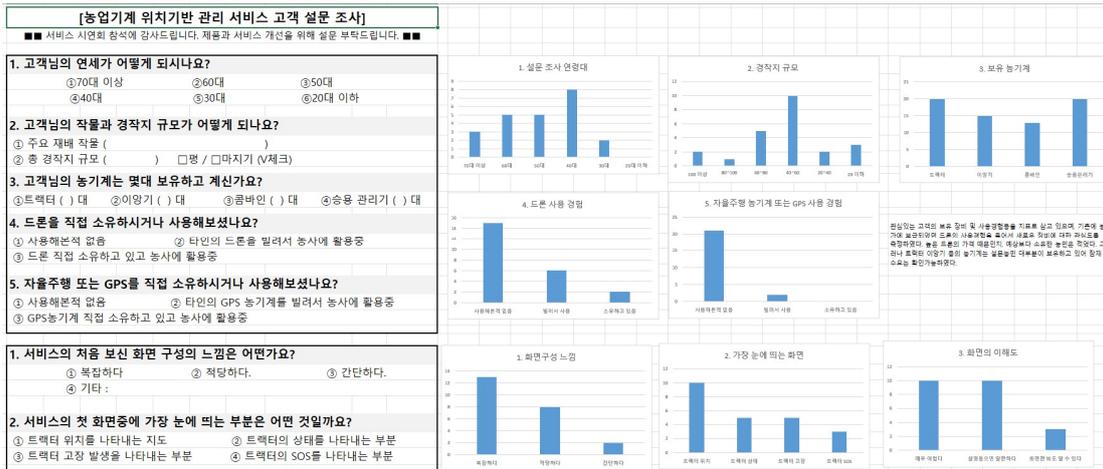
[ 정부지원 시스템에서 AI 알고리즘 결과 연동 화면 ]

- ㉔ 농기계 이력 및 운행 데이터 기반 농업인/제조사 서비스 업데이트 기능 상용화 준비
- ㉕ 농업인/제조사 서비스 실증 및 개선
  - 시연회를 통한 홍보 및 서비스 소개



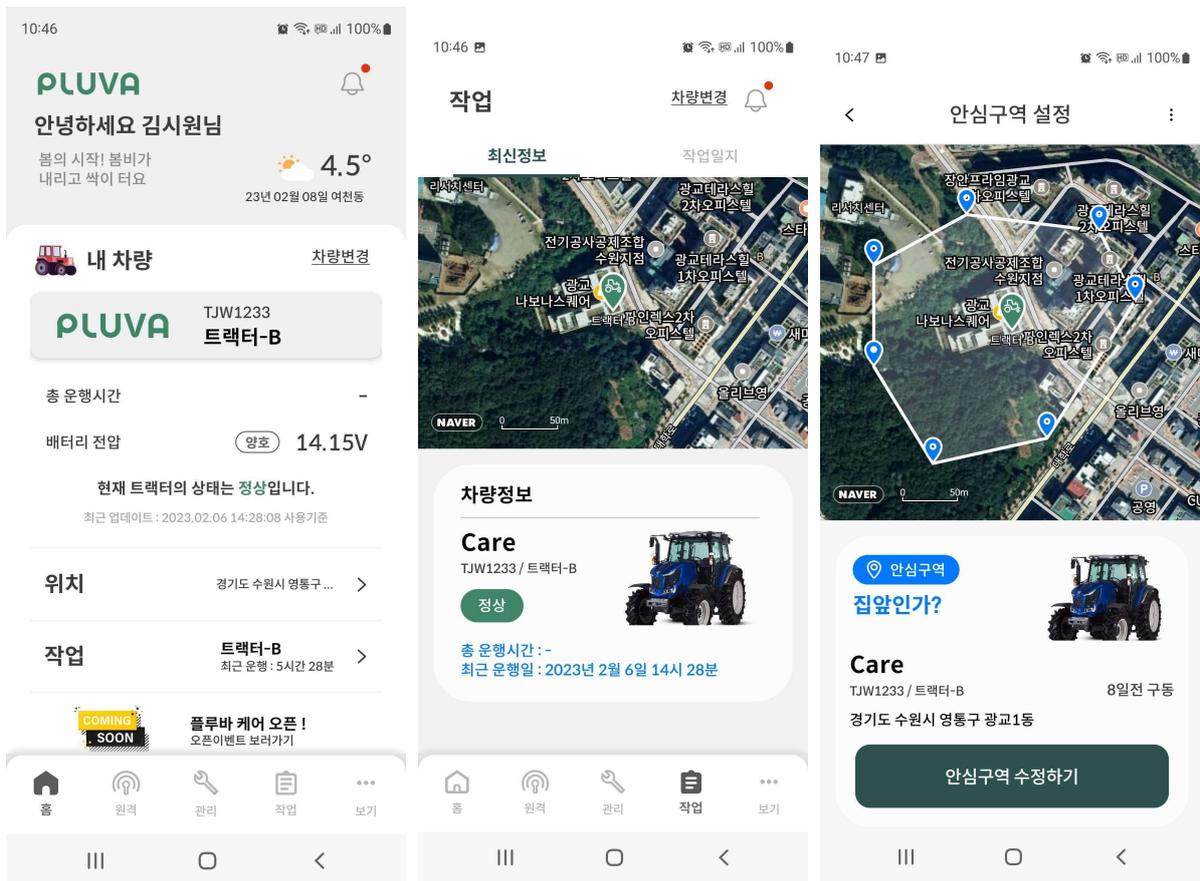
[ 시연회를 통한 홍보 및 서비스 설명 ]

㊸ 실 사용자 피드백을 바탕으로 한 서비스 사용성 개선  
 - 서비스 사용자 및 설명 청취자들을 대상으로 한 설문 조사 실행



[ 설문 조사 결과 일부 발췌 ]

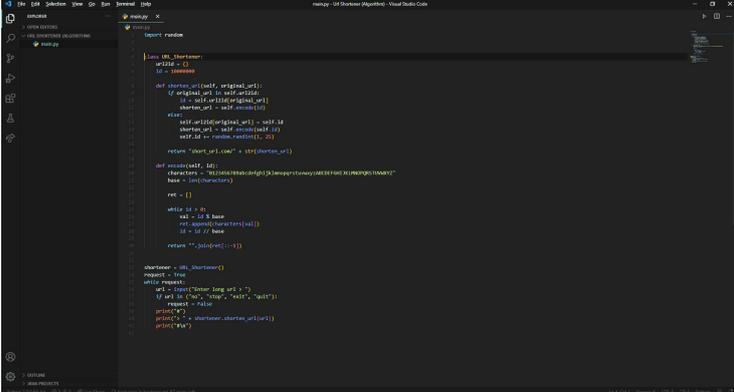
- 조사결과에 따른 UI 변경 -> 글자수 확대, 요소 배치 변경



[ 변경된 UI 일부 예시 ]

㉔ 농업인/제조사 서비스를 통한 SI 알고리즘 결과 연동

- 경상대에서 제공한 Restful API를 금년도에 추가 하여 연동되는지 확인 완료

단 계	시험 사진	비 고																																																																														
연동 서비스를 위한 스펙확인	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>API 요청시 필요한 정보 URL : /operation url http://192.168.1.100:30000/predict_fuel_torque</p> <p>Method : Get Params : key에 해당하는 이름, value에 해당하는 값</p> <p>1. 트랙터 가동시 값이 : tractor_warning (트랙터의 가동기록을 실시간으로 수집하여 해당 트랙터의 최근 위험 기록을 반환)</p> <p>Params - eqidparam(엔진 아이디) - lg130-prdno(00000001) - lg300-prdno(00000001) - lg500-prdno(00000002)</p> <p>Result - type(operation 종류) - data(현재 날짜) - eqidparam(엔진 아이디) - date_warning(경고 일자) - warning(경고 타입)</p> <p>2. 트랙터 가동시 값이 : tractor_machining (트랙터의 작업현황을 통해 트랙터의 가동률 값)</p> <p>Params - eqidparam(엔진 아이디) - lg130-prdno(00000001) - lg300-prdno(00000001) - lg500-prdno(00000002)</p> <p>Result - type(operation 종류) - data(현재 날짜) - eqidparam(엔진 아이디) - status(현재 상태)</p> <p>3. 트랙터 토크 관련 연료 소모량 예측 : predict_fuel_torque (트랙터의 토크 기록을 이용하여 연료 소모량 예측) ● 입력받은 데이터들을 최종결과에 사용함</p> </div> <div style="width: 45%;"> <p>Params - input토크 기록 데이터) 1. Format - [토 100, 100, 20, 30, 50, 100]을 기준으로 데이터를 입력할 수있음. 또한 데이터에 "" 입력시 오류 발생</p> <p>Result - type(operation 종류) - data(현재 날짜) - predict_fuel(예측된 연료소모량) 1. Format - %.1f (소수점 첫 번째 자리까지 출력)</p> <p>4. 트랙터 작업량 관련 연료 소모량 예측 : predict_fuel_operation (트랙터의 정해진 시간동안 작업량 기록을 이용하여 연료 소모량 예측)</p> <p>Params - input(작업량 단위) : Second - mid(작업량 단위 단위 작업량) : Second</p> <p>Result - type(operation 종류) - data(현재 날짜) - predict_fuel(예측된 연료소모량) 2. Format - %.1f (소수점 첫 번째 자리까지 출력)</p> </div> </div>	Restful API 형식의 연동 서비스 스펙을 문서로 확인																																																																														
연동 서비스를 사용		스크립트를 실행하여 데이터를 서버와 연동 확인.																																																																														
데이터 수집 서버에서 다운로드된 데이터 확인	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Name</th> <th>CAN ID</th> <th>SPN</th> <th>Payload</th> <th>Value</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td>Engine Coolant Temperature</td> <td>18feee00</td> <td>110</td> <td>423eb025f</td> <td>26</td> <td>Celsius</td> </tr> <tr> <td>Voltage</td> <td>18fef700</td> <td>168</td> <td>ffffff1801</td> <td>14</td> <td>V</td> </tr> <tr> <td>Fuel Temperature</td> <td>18feee00</td> <td>174</td> <td>423eb025f</td> <td>22</td> <td>Celsius</td> </tr> <tr> <td>Engine Throttle Position</td> <td>18fef200</td> <td>51</td> <td>730000000</td> <td>31.6</td> <td>%</td> </tr> <tr> <td>Wheel Based Speed</td> <td>18fef113</td> <td>84</td> <td>8E+13</td> <td>1.5</td> <td>km/h</td> </tr> <tr> <td>Engine Coolant Temperature</td> <td>18feee00</td> <td>110</td> <td>433ec825f</td> <td>27</td> <td>Celsius</td> </tr> <tr> <td>Voltage</td> <td>18fef700</td> <td>168</td> <td>ffffff1901</td> <td>14.05</td> <td>V</td> </tr> <tr> <td>Fuel Temperature</td> <td>18feee00</td> <td>174</td> <td>433ec825f</td> <td>22</td> <td>Celsius</td> </tr> <tr> <td>Fuel Rate</td> <td>18fef200</td> <td>183</td> <td>730000000</td> <td>5.75</td> <td>L/h</td> </tr> <tr> <td>Instantaneous Fuel Economy</td> <td>18fef200</td> <td>184</td> <td>730000000</td> <td>0</td> <td>km/L</td> </tr> <tr> <td>Engine Speed</td> <td>cf00400</td> <td>190</td> <td>f1959f7029</td> <td>1326</td> <td>rpm</td> </tr> <tr> <td>Actual Engine Torque Percent</td> <td>cf00400</td> <td>513</td> <td>f1959f7029</td> <td>34</td> <td>%</td> </tr> </tbody> </table>	Name	CAN ID	SPN	Payload	Value	Unit	Engine Coolant Temperature	18feee00	110	423eb025f	26	Celsius	Voltage	18fef700	168	ffffff1801	14	V	Fuel Temperature	18feee00	174	423eb025f	22	Celsius	Engine Throttle Position	18fef200	51	730000000	31.6	%	Wheel Based Speed	18fef113	84	8E+13	1.5	km/h	Engine Coolant Temperature	18feee00	110	433ec825f	27	Celsius	Voltage	18fef700	168	ffffff1901	14.05	V	Fuel Temperature	18feee00	174	433ec825f	22	Celsius	Fuel Rate	18fef200	183	730000000	5.75	L/h	Instantaneous Fuel Economy	18fef200	184	730000000	0	km/L	Engine Speed	cf00400	190	f1959f7029	1326	rpm	Actual Engine Torque Percent	cf00400	513	f1959f7029	34	%	데이터 수집 서버에 접근하여 데이터가 올바르게 다운로드 되었는지 확인 한다.
Name	CAN ID	SPN	Payload	Value	Unit																																																																											
Engine Coolant Temperature	18feee00	110	423eb025f	26	Celsius																																																																											
Voltage	18fef700	168	ffffff1801	14	V																																																																											
Fuel Temperature	18feee00	174	423eb025f	22	Celsius																																																																											
Engine Throttle Position	18fef200	51	730000000	31.6	%																																																																											
Wheel Based Speed	18fef113	84	8E+13	1.5	km/h																																																																											
Engine Coolant Temperature	18feee00	110	433ec825f	27	Celsius																																																																											
Voltage	18fef700	168	ffffff1901	14.05	V																																																																											
Fuel Temperature	18feee00	174	433ec825f	22	Celsius																																																																											
Fuel Rate	18fef200	183	730000000	5.75	L/h																																																																											
Instantaneous Fuel Economy	18fef200	184	730000000	0	km/L																																																																											
Engine Speed	cf00400	190	f1959f7029	1326	rpm																																																																											
Actual Engine Torque Percent	cf00400	513	f1959f7029	34	%																																																																											
결과	연동 서비스의 개수가 시험 기준(3개 이상)을 만족함																																																																															

- 공동연구기관(경상국립대학교) :

㉠. AI를 적용한 농기계 이력/운행 데이터 분석을 위한 데이터셋 항목 연구

㉠. 농기계 가동률, 유휴 농기계 추적 등을 위한 데이터 항목 도출

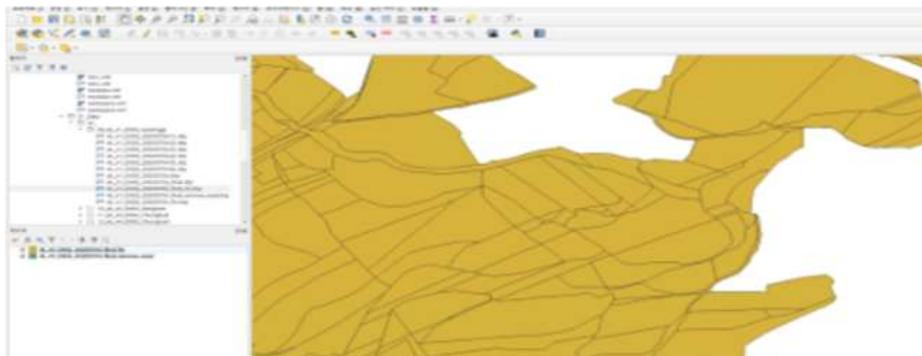
- 1차년도 연구에서는 블록체인 기술을 통해 농기계의 등록 관리 및 이에 따른 전주기 사용 이력 관리 시스템을 구축하는 기초 자료 조사를 하고자 하였음
- 본 연구에서는 농기계가 작업 또는 주행 중인지 판별할 수 있는 총 23개의 후보 인자를 선정함

구분	데이터 셋 항목명	구분	데이터 셋 항목명
1	농기계 위치 (개인위치정보사업자에게만 허가)	13	현재 변속 단수
2	연료소비량	14	미션 오일 온도
3	엔진소요동력	15	PTO출력
4	엔진 가용 시간	16	키오프 이후 배터리 잔량
5	장비 아이디	17	PTO 총 사용시간
6	총 주행 거리	18	엔진 회전수(RPM)
7	트랙터 움직임 감지	19	고장 진단 정보 일체
8	엔진 냉각수 온도	20	소음
9	CO2 배출량	21	수평도
10	시간당 연료 소모율	22	무게
11	누적연료소모량(고분해능)	23	CO2배출량
12	축매탱크 잔여레벨(요소수)		

[데이터 셋 항목]

가. 농기계 위치

- GPS를 이용하여 실시간으로 현재 농기계의 위치와 상태를 받을 수 있음.
- GPS 데이터는 오픈소스인 국가 정보 포털 - 오픈 API를 이용할 수 있음.  
(<http://openapi.nsy.a.go.kr/nsya>), (<https://qgis.org/ko/site/>)
- 지도 데이터를 구축 할 때 농경지로부터 도로에 해당하는 부분을 제거 해야 함.
- 정확한 농경지 오픈데이터 추가조사가 필요함.
- 파이썬 연동을 통해 GPS 데이터가 입력 됐을 경우, 어떤 지번주소(python)에 포함되 어 있는지 찾아내는 알고리즘 구현이 필요함.



[데이터 추출 결과]



[파이선 연동]

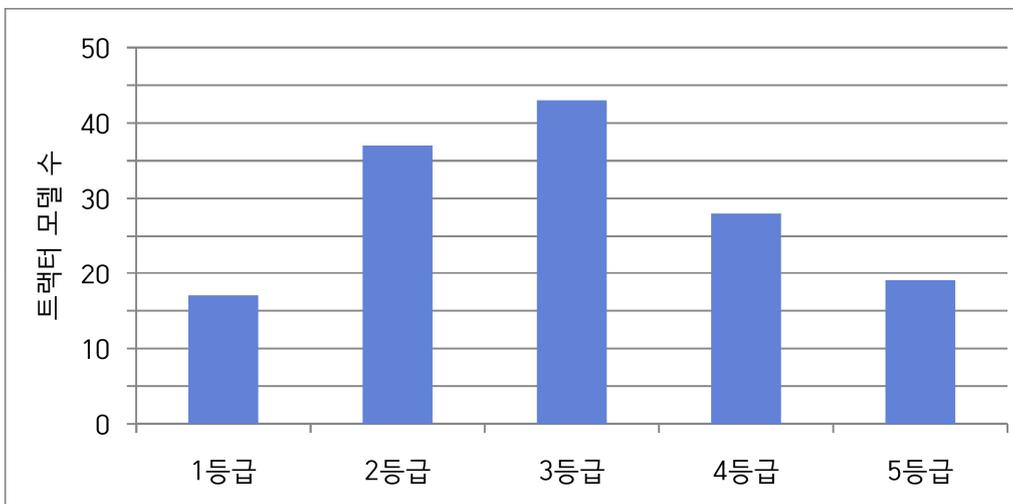
나. 연료 소비량

- 연료소비량은 스로틀 엔진회전속도, 기어단수, 작업종류에 따라 영향을 크게 받음.
- 토양의 종류에 따라서도 영향을 받음.
- 연료소비율은 스로틀 엔진회전속도에 비례, 기어단수에 반비례함.
- 실제 도로 주행일때의 연료소비량 데이터가 필요하며, 시험을 통한 데이터가 필요함.

구분	쟁기 작업	마른로타리 작업	무논로타리 작업	무논정지 작업
연료소비	0.33~0.36 L/kW · h	0.30~0.45 L/kW · h	0.19~0.34 L/kW · h	0.28~0.39 L/kW · h

\* 농용 트랙터 연료소모량에 미치는 요인분석, 박석호 (2010)

[작업에 따른 연료 소비]



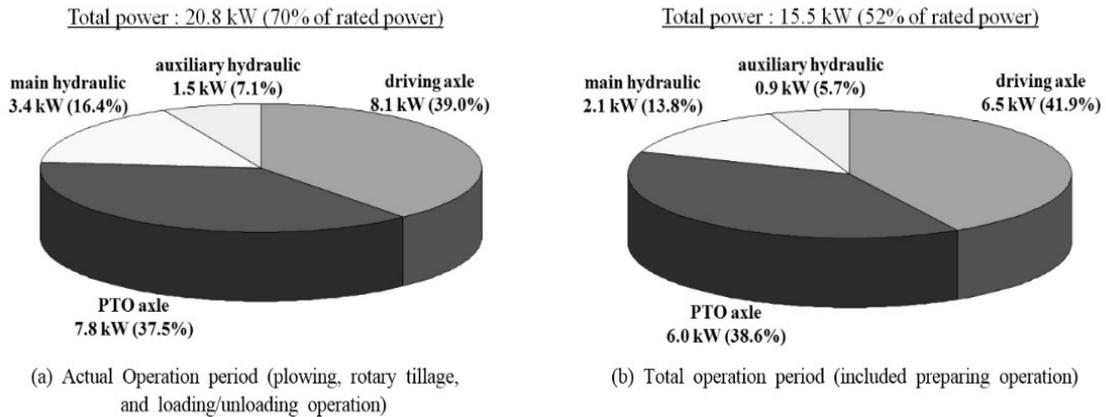
[연료 소비 등급별 트랙터 모델 수]

다. 엔진 소요 동력

- 트랙터 주요 구성요소의 소요동력은 실 작업 구간의 경우 전체 소요동력이 20.8 kW로 나타남.
- 전체 경로 구간의 경우 트랙터의 소요동력은 15.5 kW로 정격출력의 약 52%만 사용
- 실제 작업시 단순 주행보다 더 많은 동력이 요구됨.
- 도로 주행에 대한 표본이 부족함으로 시험을 통해 단순 주행에 대한 표본이 요구됨.

구분	플라우 경운 작업	로타리 경운 작업	로더 작업
실제작업 구간(평균)	23.1kW	24.6kW	14.9kW
전체경로 (평균)	17.5kW	19.1kW	8.9kW
작업여부	○	○	○

\* 농업용 트랙터의 주요 농작업 소요동력 분석 , 김용주 (2011)  
[작업에 따른 소요 동력]



[트랙터 주요 부품의 전력 요구량 분석]

\* 농업용 트랙터의 주요 농작업 소요동력 분석 , 김용주 (2011)

#### 라. 엔진 가용 시간

- 국내 트랙터의 연간 이용 시간은 엔진 출력에 따라 142~237 시간의 범위에 있었으며 37~58 kW, 22~36 kW, 59 kW 이상, 21 kW 이하의 순서로 트랙터의 사용빈도가 높은 것으로 나타났음.
- 작업별 트랙터 평균 이용 시간 비율은 로타리 경운 작업 45%, 플라우 경운 작업 29%, 로더 작업 19%, 운반 4%, 기타 작업 3%의 순서로 나타났음.

#### 마. 트랙터 움직임 감지

- 트랙터의 움직임을 통해 트랙터가 쉬는 중인지 작동 중인지 알 수 있음.
- 트랙터의 이력관리를 위해 트랙터의 미사용 기간은 매우 중요함.
- 트랙터의 움직임 확인을 위한 툴(tool)은 그림과 같다.

(<https://emanual.robotis.com/docs/kr/parts/sensor/pir-10/>)



[동작 감지 센서 PIR-10]

바. 엔진 냉각수 온도

- 냉각수의 정상 온도는 80~90도 사이임.
- 엔진의 온도를 낮출 때 필요한 이론상 냉각수의 적정온도는 70~95도로 나타났음.
- 70~95도 사이의 농기계는 정상적인 운행 상태이며 작업 또는 주행중임.
- 하지만 70도 이하의 경우 외부와의 열 교환으로 인해 냉각수 온도가 내려간 것이므로 농기계는 작동 중이라 볼 수 없음.
- 95도 이상의 경우 과도한 일을 하고 있는 것이며 레드존으로 들어가 정상 상태 아님.
- 냉각수 온도는 냉각수 온도센서를 통해 측정 가능함.

70~95도	X<70 OR X>95도
· 작업 OR 주행 중	·이 경우 냉각수의 온도가 레드존으로 들어가므로 정상인 아닌 상태 ·이것으로 작업중인지 주행중인지 알 수 없음

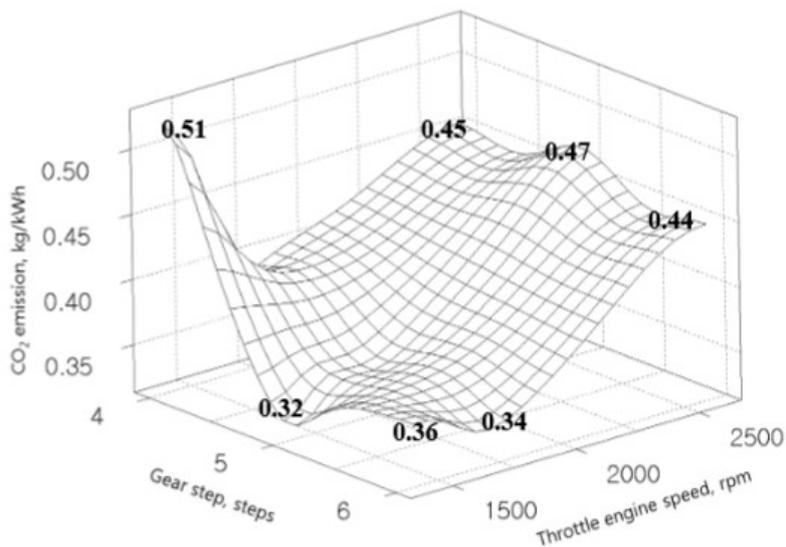
[냉각수 온도에 따른 상태]

사. CO2 배출량

- 트랙터 CO2 배출량은 엔진회전 속도에 비례하여 증가함.
- 이는 연료소비율과 유사한 경향으로 나타남.
- 작업별 CO2 배출량은 차이가 났으며, 표로 나타내면 다음과 같음.
- CO2 배출량을 얻는 TOOL은 (GreenLine, MK2, Eurotron, Italy)이 있음
- 이 TOOL은 CO2 배출량이 퍼센트(%)로 출력되기 때문에 시간당 CO2 배출량(kg/h)으로 환산해야 함.

쟁기작업	마른로타리 작업	무논로타리 작업	무논정지 작업
0.33~0.36 L/kW·h	0.30~0.45 L/kW·h	0.19~0.34 L/kW·h	0.28~0.39 L/kW·h

\* 농용 트랙터 연료소모량에 미치는 요인분석, 박석호 (2010)  
[작업에 따른 CO2 배출량]



[기어 스텝에 의한 트랙터 CO2 배출 특성 및 쟁기 작업 시 스로틀 엔진 속도]

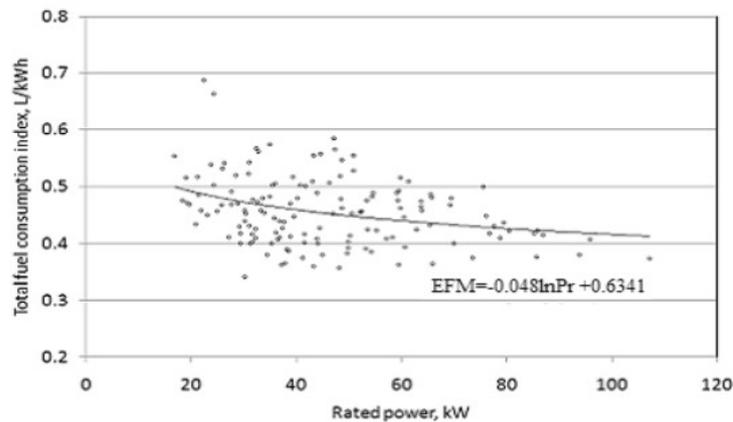
\* 농용 트랙터 연료소모량에 미치는 요인분석, 박석호 (2010)

아. 시간당 연료 소모율

- Nahmgung(2001)의 2001년 연구결과에 의하면 쟁기 작업의 출력 수준은 전형적인 작업 속도인 5~10 km/h일 때 대부분 정격 출력의 50~90% 정도이었으며, 쟁기 작업의 21.1%, 15.8%, 26.3%, 21.1%에서는 출력 수준이 각각 정격 출력의 50~60%, 60~70%, 70~80%, 80~90%인 것으로 나타났음.
- 기본 연료 소비율 지표의 범위와 평균은 트랙터의 정격 출력이 클수록 좁아지고 감소함.
- 정격 출력이 클수록 연료 소비율이 낮아 에너지 효율은 높았음.
- 단순 주행 연료 소비율 지표범위는 추가적인 시험을 통해 요구됨.

작업 연료 소비율 지표범위	주행 연료 소비율 지표범위
0.29~0.52 L/kW·h (평균은 0.385 L/kW·h 표준 편차는 0.0616L/kW·h)	표본X
작업	주행

\* 농업용 트랙터의 연료 소비 효율 등급화, 김수철 (2010)  
[연료소비율 범위]



[총연료소비지수 분포]

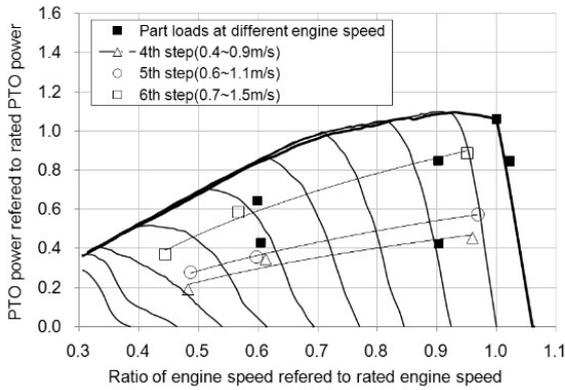
\* 농업용 트랙터의 연료 소비 효율 등급화, 김수철 (2010)

자. PTO 출력

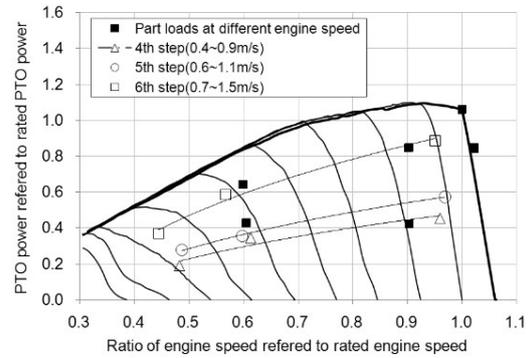
- PTO 토크 측정은 유니버설조인트에 스트레인게이지를 부착하고 라디오 텔레메트리시스템을 이용하여 할 수 있음.
- PTO출력은 엔진회전수에 비례하여 증가함.
- 전반적으로 농작업에 소요되는 PTO 출력은 4.4 kW(정격출력의 17%)에서 26.0kW(정격 출력의 100%)까지 다양한 분포로 나타났음.
- 쟁기작업 시 트랙터 출력은 1154~1265 rpm에서 4.8~9.4 kW, 1475~1593 rpm에서 8.7~15 kW, 2417~2517 rpm에서 11.6~22.7 kW로 엔진회전수에 따라 차이를 보였음.
- 쟁기작업의 트랙터 출력은 1475~1593 rpm에서 8.7~15 kW로 나타났음.
- 마른 로타리, 무논 로타리, 무논 정지 작업도 쟁기작업과 유사한 경향으로 나타났음.
- 정격 엔진회전수 2340 rpm(정격엔진회전수의 90%)에 맞추어 경운정지 작업을 수행한다면 엔진회전수 1560 rpm(정격 엔진회전수의 60%)에서 나타난 결과와 비슷한 경향을 보일 것으로 판단됨.
- 트랙터의 단순 주행 PTO출력의 표본이 부족하여 시험이 필요함.

마른 로타리 작업	무논 로타리 작업	무논 정지 작업	주행
10.6~23.3kW,	15.8~25.7kW	9.2~15.0kW	표본X
작업	작업	작업	작업

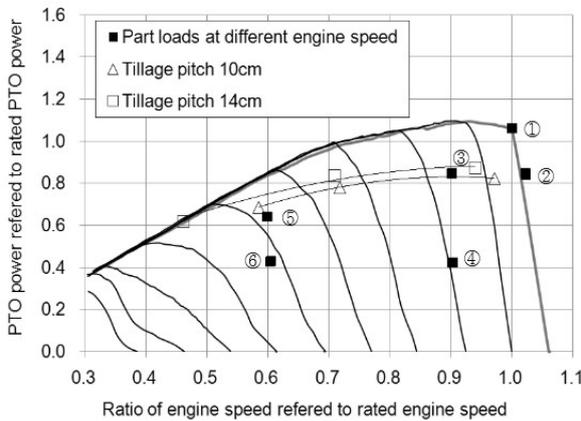
[작업에 따른 PTO 출력]



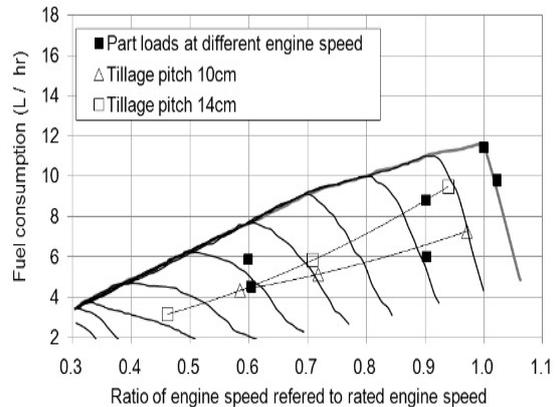
[쟁기 작업 시 트랙터 PTO 전원]



[쟁기 작업 시 연료 소비율]



[회전 운전 시 트랙터 PTO 전원]



[로타리 작동 시 연료 소비율]

\* 트랙터 PTO 출력과 농작업 부하 특성, 박석호 (2010)

차. 엔진 회전수 (RPM)

- 엔진 회전수 변경 부분 부하시험 PTO 출력은 그림 8~11에 해당됨.
- 각 지점에 대한 시험조건 설정은 OECD 기준에서 제시하고 있는데, 이 기준에 따라 엔진 회전수와 출력을 조정하여 각 지점에 대한 연료소모량을 측정하여 트랙터 성능을 비교하는 기준으로 사용되고 있음.
- 이 기준은 실제 트랙터 농작 시 엔진 회전수와 부하변동을 고려해 설정한 기준으로 밭작업이 대부분인 유럽의 기준에 맞춰진 것이기 때 문에 논 작업이 대부분의 우리나라의 경우와 다를 수 있음.
- 시험조건으로 설정한 엔진 회전수는 부하에 따라 변화하며, 설정한 엔진회전수가 낮을 수록 엔진회전수가 많이 저하되는 것으로 나타났음.
- 전반적으로 농작업에 소요되는 PTO 출력은 4.4 kW(정격출력의 17%)에서 26.0kW(정격출력의 100%)까지 다양한 분포로 나타났음(Choi and NahmGung, 2000). 마른 로타리, 무논 로타리, 무논 정지 작업의 출력은 각각 10.6~23.3 kW, 15.8~25.7 kW, 9.2~15.0 kW로 작업 속도와 엔진 회전수에 비례하여 증가함.

쟁기작업	로타리 경운작업 10CM 피치	로타리 경운작업 14CM 피치	주행
-154~1265 rpm: 4.8~9.4 kW -1475~1593 rpm: 8.7~15 kW -2417~2517 rpm: 11.6~22.7 kW	17.6 kW(1447 rpm), 20 kW(1792 rpm), 21.1 kW(2450 rpm)	15.8 kW(1125 rpm), 21.4 kW(1768 rpm), 22.4 kW(2368 rpm),	표본X
작업	작업	작업	주행

\*트랙터 PTO 출력과 농작업 부하 특성, 박석호 (2010)  
[작업에 따른 엔진 회전수]

카. 트랙터 소음

- 해외 농기시장에서는 트랙터의 운전실 내부 소음수준을 80 dB 이하로 관리하고 있음.
- 국내에서 사용되고 있는 트랙터는 대부분 85 dB를 초과하고 있는 실정임.
- 트랙터 운전실의 소음 저감에 관한 연구(김원진)에 의하면 엔진 회전수에 따른 트랙터의 좌, 우측 총합 소음레벨의 최대값은 좌측 2237 rpm에서 85.4 dB, 우측 1968 rpm에서 87.1 dB로 평가됨.



[소음측정기]

\* 출처 : [http://www.hysc.kr/mall/m\\_mall\\_list.php?ps\\_ctid=02070100](http://www.hysc.kr/mall/m_mall_list.php?ps_ctid=02070100)

타. 수평도

- 농작업시 트랙터의 기울기는 변동이 클 것으로 예상됨.
- 도로주행시 트랙터의 기울기 변동과 작업시 트랙터 기울기의 변동의 차이는 클 것으로 예상되나 표본으로 나타내기에는 자료가 부족하여 추가 시험이 필요함.
- 시중에 출시되어 있는 기울기 센서로 측정 가능함.

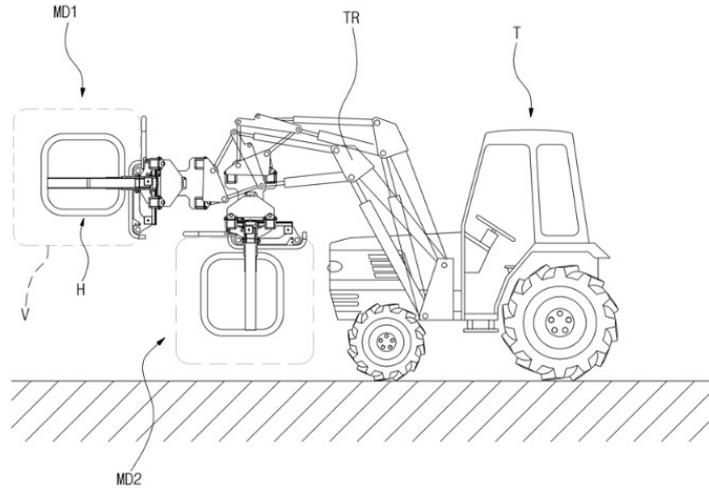


[수평 측정기]

\* 출처 : <http://itempage3.auction.co.kr/DetailView.aspx?itemno=B253805774>

파. 무게

- 트랙터는 작업시 보통 로더를 장착하여 곡물과 같은 농산물이나 퇴비, 목초, 베일 등을 적재할수 있도록 함과 동시에 이를 중거리 운반하는데도 사용 되고 있음.
- 작업시에 로더를 추가하거나 농산물등을 적재 시키면 무게는 크게 변동할 것이며 이를 토대로 단순 주행과 작업을 구별할 수 있음.
- 로더와 적재된 무게까지 함께 측정할 수 있는 제품이 시중에 나와 있기 때문에 어렵지 않게 측정 가능할 것임.



[트랙터 무게 측정 장치]

\* 출처 : <https://patents.google.com/patent/KR101813651B1/ko>

㉔. 데이터 셋 기타 항목

가. 장비 아이디

- 각 농기계마다 장비 아이디를 등록하여 관리함.
- 농기계의 장비 아이디를 등록할 경우 고장 및 이력 관리에 용이함
- 장비 아이디가 등록된 농기계들의 경우 각 농기계 별로 실제 주행거리에 대한 데이터를 모을수 있음.

나. 총 주행거리

- 트랙터의 총 주행거리는 GPS를 통한 방법과 누적 연료소모량을 통하여 알 수 있음.
- 트랙터의 주행에는 단순 도로 주행과 작업 주행이 있음.
- 총 주행거리를 통해 평균적으로 작업거리를 알 수 있음.
- 농기계의 출발지점으로부터 작업지까지의 거리를 빼면 평균적으로의 작업거리의 계산이 가능함.

다. 누적연료소모량(고분해능)

- 트랙터 수명동안의 사용에 따른 총 연료사용량은 다음과 같음.
- $F = YFC \times D$   
 $YFC = \text{연간 사용량 } L/yr$   
 $Dt = \text{트랙터 수명}(10년), yr$
- 트랙터의 누적 연료소모량을 통해 평균 작업량을 얻을 수 있음.

라. 변속 단수

- 요즘 트랙터는 전 후진을 클러치 없이 파워 셔틀로 가능함
- 보통 작업은 전 후진을 많이 하며 파워 셔틀을 많이 쓸 것이고, 예전 트랙터는 클러치를 많이 사용함
- 트랙터는 크게 주 변속이 1~4단으로 나뉨.
- 1단은 초저속 주행으로 깊이 갈이를 할 때 사용함.
- 2단과 3단은 로터리 작업시에 주로 사용됨.
- 4단은 보통 주행 할 경우 쓰여짐

#### 마. 미션 오일 온도

- 적정 미션 오일 온도는 80도 내외 이다.
- 미션 오일 온도는 작업중일 때 높게 나타남.

#### 바. 키오프 이후 배터리잔량

- 키오프 이후 배터리잔량은 모든 작업을 마치고 난 트랙터의 남은 배터리 양임.
- 키오프 이후 배터리의 남은 양을 통해 트랙터의 평균 작업량을 알 수 있음.

#### 사. PTO 총사용시간

- OECD 시험기준에 따라 트랙터의 출력을 측정할결과로 정격엔진회전수에 대한 엔진회전수의 비, 정격 PTO 출력에 대한 PTO 출력의 비로 나타냄.
- 시험조건으로 설정한 엔진회전수는 부하에 따라 변화하며, 설정한 엔진회전수가 낮을수록 엔진회전수가 많이 저하되는 것으로 나타났음.
- 농작업에 소요되는 PTO 출력은 4.4 kW(정격출력의 17%)에서 26.0 kW(정격출력의 100%)까지 다양한 분포로 나타났다
- 즉 작업에 따라 PTO 출력의 범위는 다양하게 나타났고 PTO 출력량당 사용 시간을 보면 어떤 작업을 주로 했는지 알 수 있음.

### ㉔. 농기계 이력/운행데이터 시스템과의 데이터 인터페이스 연구

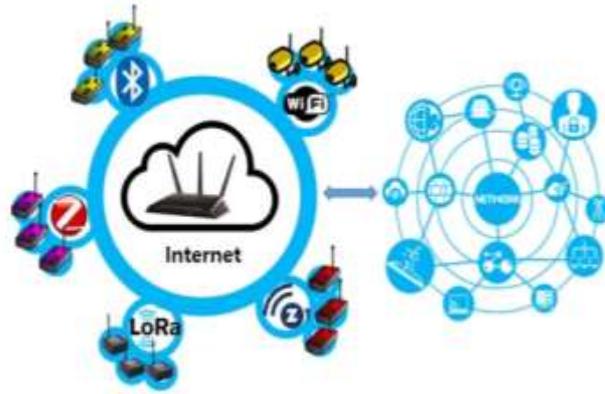
#### ㉔A 블록체인, 이력/운행 데이터 시스템으로부터의 데이터 수집 방법론 도출

##### 가. OpenWrt 기반 IoT 게이트웨이

- OpenWrt는 무선랜 라우터 및 공유기(게이트웨이)를 위한 비실시간 리눅스 기반의 오픈소스 운영체제임.
- 게이트웨이 기능을 지원하는 임베디드 보드들은 일반적으로 제한된 자원(CPU,메모리 등)을 갖기 때문에 범용 PC에서와 같이 리눅스의 모든 기능을 구현하는 것 대신 게이트웨이로서 필요한 기능만을 선택적으로 사용함.
- opkg라는 패키지 관리 시스템을 이용하여 사용자 또는 개발자가 자유롭게 필요 패키지 또는 소프트웨어를 설치하여 OpenWrt 기반 게이트웨이를 개발하여 다양한 방식으로 사용할 수 있음.

##### 나. IP 기반 이기종 네트워크 인터페이스 연동

- 현재 대부분의 IoT 시스템은 영역별 독립적으로 데이터를 생산해 내고 있으나, IoT 시스템의 활용 극대화를 위해서는 독립적으로 생성되는 대량 또는 다양한 정보가 공유되고 융합될 수 있는 환경 구축이 필수적임.
- 영역별 독립적으로 사용되는 이기종 IoT네트워크 인터페이스를 IP 기반으로 연동할 수 있는 IoT게이트웨이를 제안함.

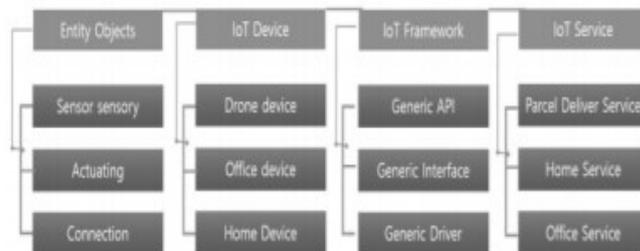


[이기종 장치의 인터넷 연결 및 IoT 게이트웨이를 통한 네트워크]

\* IOT 빅데이터 수집을 위한 IP기반 이기종 네트워크 인터페이스 연동 게이트웨이, 강지현(2019)

다. A Generic Interface Architecture

- IoT Platform Architecture는 Service Layer, Configuration Layer, Device Layer 등으로 구성됨.
- 범용 API 드라이버는 메타 모델, 장치 관리자, 환경구성 관리자 및 보안 관리자와 같은 IoT 플랫폼과 연결된 범용 인터페이스임.
- IoT 기반의 서비스가 자동으로 구성됨.
- 이 서비스는 다른 서비스 환경을 구성하는 데 사용될 수 있음.
- 서비스 자동발견으로 서비스를 효과적으로 등록하고 발견하여 메타데이터를 통해서 데이터 서비스를 등록함.
- 서비스 재사용성으로 코드의 재사용을 촉진하기 위해 다양한 서비스로 구성됨.
- 이기종 인터페이스를 위한 IoT 범용 인터페이스는 서로 다른 방식으로 서비스를 호출하며 범용 인터페이스를 호출하는 Get 방식을 통해 상위 레이어를 호출함.
- 이 알고리즘은 이기종 IoT 서비스를 위한 내부에이전트를 통해서 매개변수를 전달하여 호출되는 방식임.

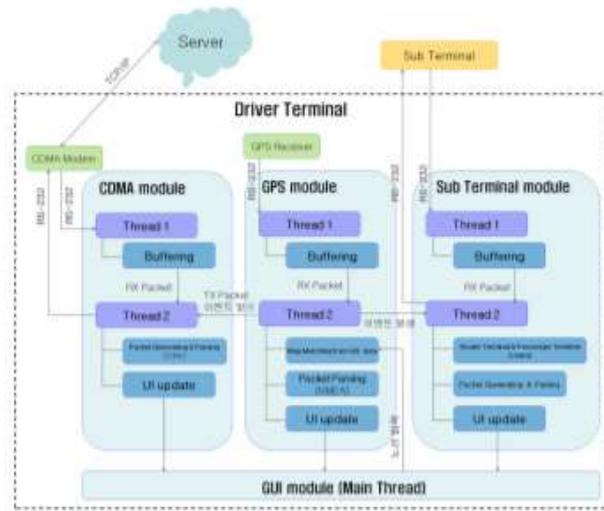


[IoT 프레임 워크의 메타 모델]

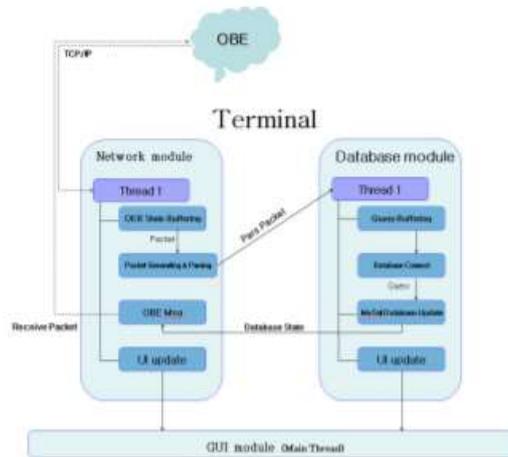
\*출처 : IoT 플랫폼을 위한 범용 인터페이스, 김미(2018)

라. GPS와 CDMA 모뎀이 장착된 차량 단말을 이용하여 위치 추적 및 운행 정보 추적

- 기존 사례에서는 GPS와 CDMA 모뎀이 장착된 차량 단말을 이용하여 작업 및 이동 중인 청소 차량의 위치를 서버 PC에 전송하고 서버에서는 차량의 위치를 데이터 베이스에 저장하고 범용 지도 접속 방식을 지원하는 홈페이지를 통해 인터넷에 접속된 모든 사람에게 위치 정보를 지도와 함께 표출해 주는 사설 LBS 시스템 플랫폼을 개발하여 공공 청소차 운행 관리 시스템에 적용하여 그 동작을 확인함.
- 차량 단말은 GPS 위치 정보를 정확히 추출해 내고 주기적으로 이동통신망을 통해 서버 PC에 저장할 수 있으며 서버는 차량 위치 정보를 Google Map 지도에 실시간으로 표시할 수 있었음.
- 또한 서버 데이터 베이스에 저장된 로그 정보를 이용하여 차량의 이동 경로를 표시할 수 있으며 이동 거리, 소요 시간 등의 운행 정보를 추출할 수 있었음.



[단말기 소프트웨어 구성]

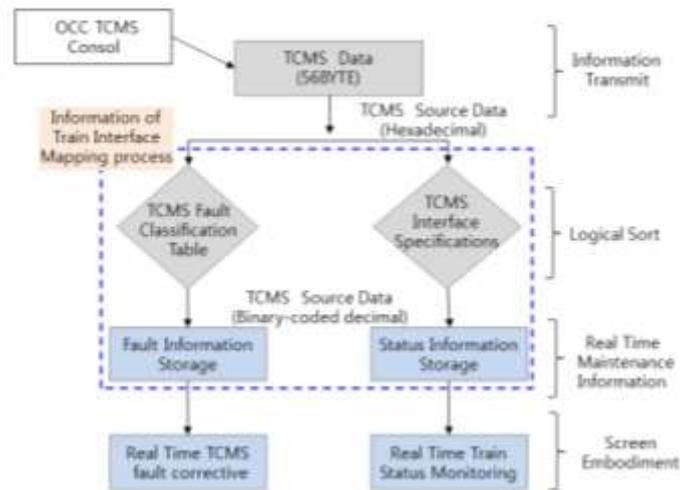


[서버 소프트웨어 구성]

\* 공공 청소차 운행 관리를 위한 추적 시스템 개발, 백승원 (2011)

마. 장비 상태 및 고장정보 맵핑 인터페이스 (유사 사례 - 전동차)

- 각종 상태정보는 무선통신망과 전동차 TCMS 인터페이스를 통하여 유지보수정보시스템으로 실시간 전송됨.
- TCMS는 고장 발생시 고장의 종류, 위치 및 시간 등의 정보를 보여주면서 알람을 출력함.
- 기존 사례에서는 중대한 고장인 경우에는 고장 조치 방법 뿐만 아니라 수동운전 시 차량운행 거리, 속도, 차량번호 및 행선지 등 운전이 필요한 정보를 현시함.
- 대량의 전동차 TCMS의 각종 정보는 데이터 처리 알고리즘을 이용하여 56byte의 작은 데이터 테이블로 전송함.
- 이러한 대량정보로부터 차량기지의 유지보수 정보시스템에 효율적인 전달을 위해서는 고장정보를 일정한 규칙에 의거하여 코드화하는 것이 중요함.

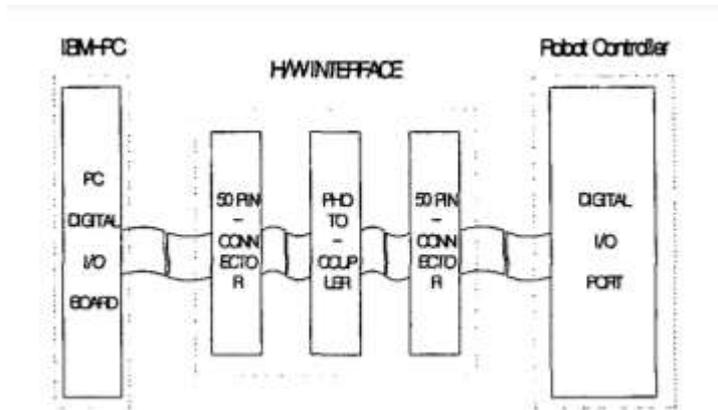


[TCMS 정보의 데이터 처리 알고리즘]

\* 무인전동차의 실시간 상태 진단을 위한 유지보수 정보시스템 인터페이스에 대한 개념 설계 ,  
한준의(2017)

바. 병렬 I/O 인터페이스 소프트웨어 PCP(Process Control Programs)

- 이 논문에서는 산업용 로봇을 교시하기 위한 병렬 데이터 통신 interface에 대한 병렬 I/O Data 전송 실험에 대해서 살펴봄.
- 센서 데이터를 실사근으로 로봇에게 전달해야만 하는 목적에 의해서 병렬 데이터 전송을 위한 하드웨어 Interface 보드에 대해 살펴봄
- 제작된 하드웨어 Interface를 통해서 I/O packet을 전송하고 전송받기 위한 프로그램에 대해서도 살펴봄.
- 이렇게 제작된 I/O Date 전송 Interface 시스템의 성능을 분석하여 실시간으로 Data를 전송할 수 있는지를 살펴봄
- 실시간으로 병렬 데이터를 제작된 Interface를 통해서 전송할 수 있다.



[병렬 I/O 하드웨어 Interface 구성도]

\* 산업용 6관절 로봇의 원격제어를 위한 실시간 병렬데이터 통신 인터페이스 , 최명환(2001)

㉔ 전달된 농기계 이력/운행 데이터셋을 활용한 기본 AI 알고리즘 설계

㉔ AI 알고리즘

가. 전문가 시스템

- 가장 직관적으로 인간이 지닌 지적 능력을 알고리즘으로 구현한 방법이 전문가 시스템임.
- 의사 등 특정 분야 전문가들의 지식이나 추론 방법을 컴퓨터로 옮겨 전문가 이외의 사람들도 이와 같은 지혜를 이용할 수 있도록 한 것임.
- 지식 베이스라고 명명한 데이터베이스에 전문지식을 저장하여 추론 기능을 사용, 최대한 전문가에

가까운 판단을 내리도록 함.

- 가장 간단하게 인간의 지성을 알고리즘으로 구현할 수 있는 장점이 있으나 사람이 직접 많은 수의 규칙을 집어넣어야 하는 것을 전제로 하기에 지식이 바뀌거나 추가된다면 일일이 수정/추가해야 함.
- 의학이나 생물 분야에서 널리 사용되고 있는 AI 알고리즘이다.

#### 나. 머신러닝

- 머신러닝은 컴퓨터를 인간처럼 학습시켜 스스로 규칙을 형성할 수 있도록 한 방법임.
- 주로 통계적인 접근을 사용하며 데이터를 기반으로 통계 모형을 만드는, 모델링 방법을 주로 이용함.
- 학습방법에 따라 지도 학습(Supervised Learning), 비지도 학습(Unsupervised Learning)으로 나눌 수 있음.
- 지도 학습은 기존에 발생한 입력(x)과 이에 관한 결과(y)로 설계한 알고리즘을 학습하는 방식임.
- 알고리즘은 각 입력에 관한 결과를 알고 있기에 새로운 입력에 관한 결과를 추론할 수 있음.
- 이를 세분화하면 회귀(Regression)와 분류(Classification)가 있음.
- 회귀는 정해지지 않은 범위에서 특정한 값을 찾는 것이며 분류는 정해진 선택지 중 가장 가능성 큰 선택지를 선택하는 것임.
- 비지도 학습은 결과(y)가 따로 없는 데이터에 대해서 학습하는 것임. 결과(y)가 함께 있는 데이터가 아니라 입력(x) 데이터만으로 어떠한 결과를 찾는 방식임.
- 데이터의 새로운 특성이나 분포를 알고 싶을 때 사용하는 머신러닝 기법임. 비지도 학습으로 설계된 알고리즘을 세분화하면 군집화(Clustering)와 차원축소(Dimensionality Reduction) 등이 있음.

#### 다. 퍼셉트론

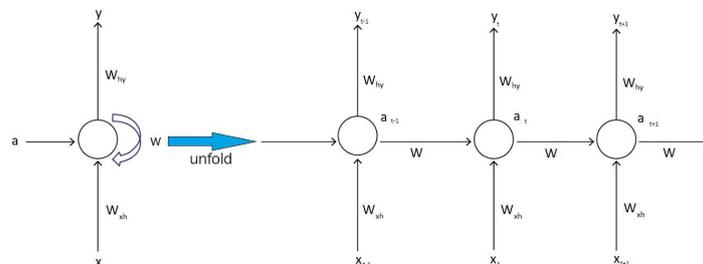
- 퍼셉트론의 주요 아이디어는 동물 신경계의 신호전달 원리를 본 따 알고리즘을 설계한 것임.
- 동물의 신경계는 뉴런을 통해 신호를 전달함.
- 뉴런은 여러 입력 신호를 받아 신호를 활성화할지 비활성화할지 결정하는데, 활성화와 비활성은 신호의 총합이 정해진 임계값을 넘었는지에 따라 결정됨.
- 또한, 각 입력 신호에는 고유한 가중치(weight)가 부여되며 가중치가 클수록 해당 신호가 중요하다고 볼 수 있음.
- 이러한 뉴런을 알고리즘으로 옮기면(3가지 입력 신호가 들어온다고 가정) 입력 신호를  $x_1, x_2, x_3$ 로 두고 각 신호의 가중치를  $w_1, w_2, w_3$ 로 둬.
- 이후  $x_1w_1 + x_2w_2 + x_3w_3$ 가 임계값을 넘었는지 판단하여 이를 활성화할지 비활성화 할지 결정함.
- 퍼셉트론의 가중치를 개선하는 방식은 위에서 언급한 지도 학습과 유사한 방식으로 이루어짐.
- 특정 입력 신호와( $x_1, x_2, x_3$ ) 이값이 임계값을 넘었는지의 유무(1, 0)를 통해 가중치를 변화시킴.
- 데이터들이 일정한 선으로 분류가 가능한 선형으로 분포가 되어있다면 퍼셉트론 알고리즘을 적용할 수 있으나, 데이터들이 특정 직선으로 구분할 수 없는, 비선형으로 분포된다면 퍼셉트론을 적용할 수 없는 단점이 있음.

#### 라. 딥러닝

- 딥러닝은 퍼셉트론과 개념은 유사하지만 변형되고 확장된 개념임.
- 퍼셉트론은 이전에 인간의 지닌 지적 능력을 그대로 묘사하기 위해 동물의 신경계를 알고리즘으로 옮김.
- 하지만 결국 0이나 1로 활성화하기에 비선형으로 분포된 데이터들에 퍼셉트론을 적용할 수가 없었음.
- 딥러닝은 이러한 단점을 극복하기 위해 발전된 방법이다. 개념적으로 딥러닝은 여러 퍼셉트론을 연동한 것임.
- 하나의 직선을 긋는다면 비선형 데이터를 나눌 수 없겠으나 여러 직선을 동시에 긋는다면 비선형

- 데이터를 나눌 수 있음.
- 하지만 이러한 아이디어는 Vanishing gradient 문제를 해결할 수 없음.
- Vanishing gradient는 최초 입력값이 네트워크의 마지막까지 이어지지 않는 것.
- 단순한 비선형 데이터 분포가 아닌, 복잡한 비선형 데이터에서 발생하는 분류 문제를 해결하기 위해 섀넬 수 없는 많은 직선을 그어야 함.
- 따라서 더 많은 단의 네트워크가 필요하고 최초 입력이 마지막까지 이어져야 할 것임
- 하지만 기존 퍼셉트론은 활성화 함수에서 1 또는 0을 출력하면서 최초 입력이 마지막까지 이어지지 않았음.
- 따라서 현재의 딥러닝은 활성화 함수를 동물의 신경계처럼 사용하지 않고 변형하여 사용하는 방법이 많이 사용됨.
- 활성화 함수에서 1이나 0을 출력하는 것이 아니라 0이 넘으면 입력을 그대로 다음 단으로 전달하고 0보다 작으면 0을 넘기는 것임.
- 또한, 데이터의 과적합을 막기 위해 학습 시 의도적으로 네트워크 일부분을 끊고 학습하거나(drop out) 잃어버린 입력을 다시 전달해주는(residual network) 등 다양한 방법을 활용하며 딥러닝 기술이 발전하고 있음.
- 이는 동물의 신경계를 그대로 묘사하고자 했던 퍼셉트론과 차이가 있음.
- 딥러닝은 머신러닝과도 차이를 보이는데 가장 큰 차이는 은닉계층의 유무임.
- 머신러닝은 모델에 입력과(x) 출력을(y) 넣어줄 때 가장 출력과 연관 있는 x를 넣어줘야 됨. 이를 특성(feature)이라고 함.
- 이 특성은 출력과 연관성 있어야 하므로 모델을 설계하는 사람이 이를 찾아주어야 됨.
- 반면 딥러닝은 은닉계층에 의해 모델 내에서 특성을 자동으로 찾아주기에 비교적 간편함.
- 하지만 그에 따라 많은 데이터가 필요하다는 단점이 있음.

#### 마. RNN 구조



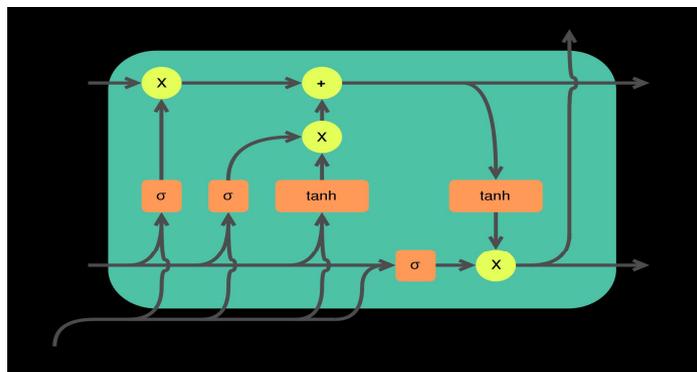
[RNN 구조]

- RNN(Recurrent Neural Network)은 딥러닝 기반 네트워크의 한 종류로 이전 계층이 다음 계층에 영향을 주고 이러한 것들이 연결되면서 연속적인 데이터의 흐름을 모델에 반영할 수 있는 특징이 있음.
- 딥러닝 모델(Fully connected Network) 구조를 보면 모든 인풋이 모든 은닉계층에 영향을 주는 것을 알 수 있음.
- 입력값이 순서에 상관없이 독립적일 경우 위 구성이 적합할 수 있으나 언어, 주가, 전력사용량 등 연속적인 데이터(Sequential Data)의 경우 데이터의 연속적인 흐름의 입력일 경우 그림 2의 구조가 적합하지 않음.
- 이때 사용하는 것이 RNN으로 연속적인 데이터의 흐름을 모델에 반영하면서 결과를 도출할 수 있음.
- RNN(Recurrent Neural Network) 구조는 입력에서 은닉계층으로 값이 전달되고 은닉계층에서 연속적으로 서로 영향을 주는 것을 알 수 있음.
- 이러한 모델 구조를 통해 연속적인 데이터 흐름을 모델에 반영할 수 있음.
- 하지만 이러한 RNN 구조에 단점이 있는데, 출력과 먼 입력이 출력에 영향을 주기 힘들다는 것임.
- RNN 구조상 입력이 은닉계층을 거치면서 한쪽으로 나아감.

- 이때 원하는 출력에 가까운 입력은 많은 계층을 거치지 않기에 출력에 큰 영향을 주지만 원하는 출력과 먼 입력은 도중 많은 계층을 거치기에 최초 입력과 많이 바뀌어 출력에 많은 영향을 줄 수 없다는 한계가 있다.

바. LSTM(Long-Short Term Memory) 모델

- LSTM(Long-Short Term Memory) 모델은 위에서 언급한 RNN의 단점을 극복한 모델 구조임.
- RNN의 경우 출력과 먼 입력은 출력에 많은 영향을 줄 수 없다는 한계가 있음. LSTM은 이를 셀이라고 하는 계층으로 극복함.
- 은닉계층에 셀 계층을 추가하여 모든 은닉계층의 값을 출력으로 보내는 것이 아니라, 선택적으로 은닉계층의 값을 출력 부분으로 보낼 수 있음.
- 이에 따라 입력이 모든 계층을 거치면서 원래 값을 잃어버리는 단점을 막을 수 있음.
- 구현이 어렵다는 단점이 있으나 많은 딥러닝 라이브러리들이 모듈화된 모델을 제공함으로써, 쉽게 설계 중인 딥러닝 모델에 적용할 수 있음.

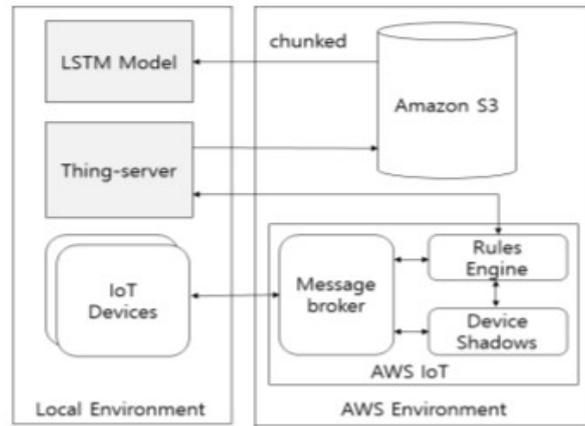


[LSTM 아키텍처]

\* 출처 : ([https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory))  
 \* 고효율 에너지 저장시스템을 위한 AI 알고리즘 연구, 장규희 (2019)

사. IoT 스트리밍 센서 데이터에 기반한 실시간 PM10 농도 예측 LSTM 모델 (기존 사례)

- 본 논문에서는 LSTM 예측 모델의 입력으로 사용하기 위한 데이터를 AWS IoT 환경에서 IoT 센서로부터 PM10 스트리밍 데이터를 수집함.
- 실시간으로 축적되는 데이터 처리를 위해 IoT 기기에서 측정된 PM10 스트리밍 데이터를 AWS IoT와 자체 제작한 Thing-server(소켓 서버)를 통해 Amazon S3에 저장하는 아키텍처를 보여준다.
- IoT 기기는 실시간 측정값을 Thing-server로 전송하고, Thing-server는 각 클라이언트에서 전송한 1초마다 측정된 값을 20초 단위의 평균값으로 계산하고 매일 24시간 분량의 파일로 만들어서 매일 자정(00시 00분 00초)에 CSV 형태로 Amazon S3 버킷에 저장함.
- S3에 저장된 데이터는 시계열 데이터이므로 로컬 환경에 구성된 LSTM 모델의 입력으로 사용될 수 있도록 시퀀스(sequence) 데이터로 변환함.
- 변환된 시퀀스 데이터는 슬라이딩 윈도우 프로세스를 통해 다양한 훈련 데이터셋을 만드는데 사용될 수 있음.
- 스트리밍 데이터의 실시간 예측 서비스에 대한 플로차트를 보여줌.
- 서비스 아키텍처는 기기로부터 스트리밍 데이터를 받아 들이는 AWS IoT, 훈련 데이터 저장을 위한 Amazon S3 저장장치, Local 컴퓨팅 환경에서의 학습을 위한 LSTM 모델, 모델을 배치하기 위한 Docker, 실시간 예측 서비스를 제공하기 위해 Docker상에 설치된 TensorFlow Serving, 애플리케이션과 TensorFlow Serving 사이의 통신을 담당하는 Thing-server(소켓 서버)로 구성하였음.
- 기존 사례에서는 한 다양한 알고리즘 중에서 스트리밍 시계열 데이터에 알맞은 LSTM 모델을 이용하여 IoT 센서 데이터를 학습하여 입력 값들을 분석하고, 애플리케이션에서 FAN을 ON시키는 것과 같은 환경요소에 따른 필요한 정보를 제공할 수 있는 예측 서비스를 구축하였음.



[AWS IoT에서 데이터 수집]

\* IoT 스트리밍 센서 데이터에 기반한 실시간 PM10 농도 예측 LSTM 모델 , 김삼근 (2020)

1) 농기계 이력/운영 데이터 시스템과 연동 및 트레이닝 데이터 특징 분석

가) 농기계 이력/운영 시스템과의 API 연동 및 데이터 수집

(1) 데이터 제공

- 주관기관인 긴트와 협의를 통해 농기계 사용이력에 따른 데이터의 항목을 토의하고 REST API를 통해 시스템과 연동
- 본 연구에서는 실질적으로 데이터 구현이 가능한 항목을 도출하였으며, 총 9개의 후보 인자의 데이터 수집

표 1. API연동 데이터 셋 항목

구분	데이터 셋 항목명	가능 여부	비고
1	장비 아이디	○	조회 가능
2	농기계 위치	○	현재 gps 좌표 조회 가능
3	연료소비량	○	시작시간, 끝시간을 입력받아 그 사이의 연료소비량 조회 가능
4	엔진소요동력	○	가장 최근 올라온 엔진 토크 데이터 조회 가능
5	엔진 가용 시간	○	총 운행시간 (부하별 운행시간 조회 가능) 조회 가능
6	총 주행 거리	○	총 운행거리 조회 가능
7	엔진 냉각수 온도	○	가장 최근 올라온 엔진 냉각수 온도 데이터 조회 가능
8	CO2 배출량	X	
9	시간당 연료 소모율	○	(엔진 가용 시간 / 시간당 연료 소모율) 계산 가능 (구현 필요)
10	누적 연료 소모량	○	총 연료소모량 조회 가능

(2) 제공 가능 데이터 명세

- (구분 1) 장비 아이디 : 장비에 대한 고유한 아이디

□ (구분 2) 농기계 위치

- 요청 api : 현재 위치 정보 조회 api

- 장비 아이디를 이용해 현재 장비의 위치를 얻을 수 있음
- 입력 파라미터
- equipment\_id : 장비 아이디
- 리턴 데이터 예시

```

"altitude": "127335",
"longitude": "1270423162",
"timestamp": "1628579495",
"equipment_id": "0f76-1bbc-9848-2484-4b69",
"latitude": "372997856"

```

- altitude : 고도
- longitude : 경도
- latitude : 위도
- timestamp : 위치가 확인된 시간의 timestamp

□ (구분 3) 연료 소비량, 누적 연료 소모량

- 요청 api : 장비 별 통계 조회 api
- 특정 장비에 대해 두 타임스탬프 사이의 통계 데이터를 얻을 수 있음
- 입력 파라미터
- equipment\_id : 장비 아이디
- timestamp\_from : 필요한 데이터의 시작 시간 타임스탬프
- (optional, default = 0)
- timestamp\_to : 필요한 데이터의 마지막 시간 타임스탬프
- (optional, default = 현재 시간)
- statistic\_name : 필요한 통계 데이터 종류 (아래 셋 중 선택)
- operation\_time : 부하 별 운행시간
- mileage : 운행거리
- fuel\_consumption : 연료 소모량

- 리턴 데이터 예시

```

"equipment_id": "0f76-1bbc-9848-2484-4b69",
"statistic_name": "fuel_consumption",
"timestamp_from": "0",
"timestamp_to": "1633956143.638123035430908203125",
"unit": "L",
"value": "1459.5"

```

- unit : 통계 데이터 단위
- value : 통계 데이터 값

□ (구분 4) 엔진 소요 동력, 엔진 냉각수 온도

- 요청 api : 차량 현재 상태 조회 api
- 가장 최근에 업데이트 된 상태 데이터 조회 가능
- 상태 데이터에는 엔진 토크 데이터, 냉각수 온도, 전압, 엔진 스피드 등의 데이터가 있음
- 입력 파라미터
- equipment\_id : 장비 아이디
- 리턴 데이터 예시
- 엔진 토크 데이터

```
{  
  "name": "engine_torque",  
  "timestamp": "1628164302",  
  "value": "-125",  
  "unit": "%"  
},
```

○ 엔진 냉각수 온도

```
{  
  "name": "engine_coolant_temperature",  
  "timestamp": "1620363389",  
  "value": "78",  
  "unit": "Celsius"  
},
```

- unit : 통계 데이터 단위
- value : 통계 데이터 값

□ (구분 5) 엔진 가용 시간

- 요청 api : 장비 별 통계 조회 api
- 특정 장비에 대해 두 타임스탬프 사이의 통계 데이터를 얻을 수 있음
- 입력 파라미터
- equipment\_id : 장비 아이디
- timestamp\_from : 필요한 데이터의 시작 시간 타임스탬프
- (optional, default = 0)
- timestamp\_to : 필요한 데이터의 마지막 시간 타임스탬프
- (optional, default = 현재 시간)
- statistic\_name : 필요한 통계 데이터 종류 (아래 셋 중 선택)
- operation\_time : 부하 별 운행시간
- mileage : 운행거리
- fuel\_consumption : 연료 소모량
- 리턴 데이터 예시

```

"equipment_id": "0f76-1bbc-9848-2484-4b69",
"statistic_name": "operation_time",
"timestamp_from": "0",
"timestamp_to": "1633959296.805715084075927734375",
"value": {
  "load_low": "10532.00000",
  "load_middle": "868.00000",
  "load_high": "26.00000"
},
"unit": "sec"

```

- unit : 통계 데이터 단위
- value : 통계 데이터 값
- load\_low : 타임스탬프 사이 시간 동안 저부하로 운행한 총 운행시간
- load\_middle : 타임스탬프 사이 시간 동안 중부하로 운행한 총 운행시간
- load\_high : 타임스탬프 사이 시간 동안 고부하로 운행한 총 운행시간
- load\_low, load\_middle, load\_high 를 더하여 총 운행시간 계산 가능

□ (구분 6) 총 주행 거리

- 요청 api : 장비 별 통계 조회 api
- 특정 장비에 대해 두 타임스탬프 사이의 통계 데이터를 얻을 수 있음
- 입력 파라미터
- equipment\_id : 장비 아이디
- timestamp\_from : 필요한 데이터의 시작 시간 타임스탬프
- (optional, default = 0)
- timestamp\_to : 필요한 데이터의 마지막 시간 타임스탬프
- (optional, default = 현재 시간)
- statistic\_name : 필요한 통계 데이터 종류 (아래 셋 중 선택)
- operation\_time : 부하 별 운행시간
- mileage : 운행거리
- fuel\_consumption : 연료 소모량
- 리턴 데이터 예시

```

"equipment_id": "0f76-1bbc-9848-2484-4b69",
"statistic_name": "mileage",
"timestamp_from": "0",
"timestamp_to": "1633959686.6081039905548095703125",
"value": "2.34663",
"unit": "km"

```

- unit : 통계 데이터 단위
- value : 통계 데이터 값

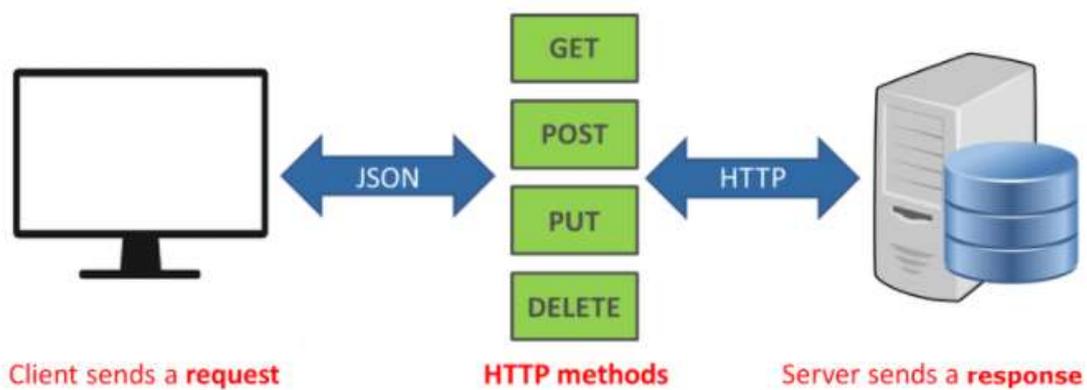
□ (구분 7) 시간당 연료 소모율

- (누적 연료 소모량 / 엔진 가용 시간) 로 계산 가능
- 요청 api : 구현되어있지 않음

※ API 란 데이터와 기능의 집합을 제공하여 컴퓨터 프로그램간 상호작용을 촉진하며, 서로 정보를 교환가능 하도록 하는 것

※ REST API : REST 기반으로 서비스 API를 구현한 것. 최근 OpenAPI(누구나 사용할 수 있도록 공개된 API: 구글 맵, 공공 데이터 등), 마이크로 서비스(하나의 큰 애플리케이션을 여러 개의 작은 애플리케이션으로 쪼개어 변경과 조합이 가능하도록 만든 아키텍처) 등을 제공하는 업체 대부분은 REST API를 제공한다.

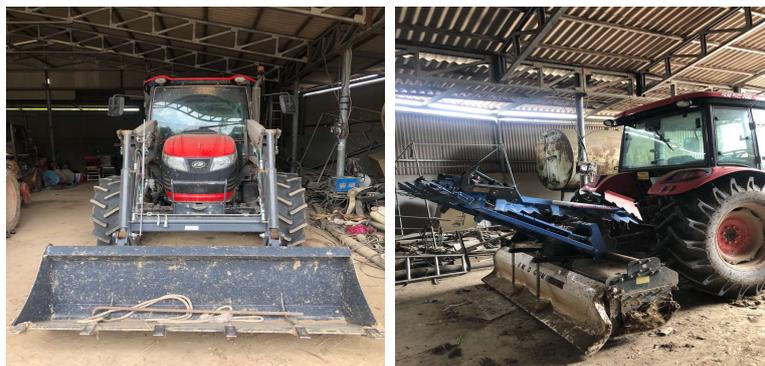
※ REST API의 특징 : 사내 시스템들도 REST 기반으로 시스템을 분산해 확장성과 재사용성을 높여 유지보수 및 운용을 편리하게 할 수 있다. REST는 HTTP 표준을 기반으로 구현하므로, HTTP를 지원하는 프로그램 언어로 클라이언트, 서버를 구현할 수 있다. 즉, REST API를 제작하면 델파이 클라이언트 뿐 아니라, 자바, C#, 웹 등을 이용해 클라이언트를 제작할 수 있다.



[ REST API 구조 ]

나) 농기계 이력/운행 트레이닝 데이터 특징 분석

(1) 실데이터 수집 및 특징 분석



[AL600P 트랙터]

(가) 농업기계 선정

- 농작업 기계중에서 작업빈도가 가장 많은 트랙터를 선정
- 트랙터 작업에 관련된 도메인 지식을 얻고자 실제 농가를 방문
- 전문가의 도메인 지식을 토대로 설문지 작성

**농기계(트랙터) 연중 작업 현황조사**

☑ 개요 : 농기계에 대한 이력 관리가 미흡한 실정기기에 농기계 운용(이력, 운행, 사용 연한)에 관한 데이터수집 및 응용 가공하여 정보 및 민간 수요에 알맞은 농산업 서비스를 제공하고자 아래와 같은 설문을 진행하고자 합니다. 참여해주셔서 감사합니다.

1. 주소 : 경상남도 \_\_\_\_\_ 시군 \_\_\_\_\_ 동면

2. 성명 : \_\_\_\_\_ (연령 : \_\_\_\_\_ )

3. 트랙터 제조사 : \_\_\_\_\_, 제조년도 : \_\_\_\_\_, 규격 : \_\_\_\_\_ 마력, 기계잔존수명 : \_\_\_\_\_ 년

4. 작업 형태 : (벼농사) 경작면적 : \_\_\_\_\_

작업내용	작업시기(월) (초·중·말)	작업일 수(일)	일 평균 작업시간(H)	연료 소비량(말통)
쟁기 작업				
1차 로터리 작업				
2차 로터리 작업				
비료 살포 작업				
씨래 작업				
모내기 (묘 운반)				
기타 :				

5. 작업 형태 : (보리농사) 경작면적 : \_\_\_\_\_

작업내용	작업시기(월) (초·중·말)	작업일 수(일)	일 평균 작업시간(H)	연료 소비량(말통)
보리씨앗 살포				
1차 로터리				
베토 작업				
기타 :				

6. 작업 형태 : (밭농사) 경작면적 : \_\_\_\_\_

작업내용	작업시기(월) (초·중·말)	작업일 수(일)	일 평균 작업시간(H)	연료 소비량(말통)
로터리 작업				
기타 :				



[트랙터 연중 작업 현황 조사 설문지]



[진주·산청대리점 : YANMAR]

(나) YANMAR 대리점 방문 및 도메인 지식습득

- 농기계를 판매하는 업체이자 실질적인 수리 및 이력 내용을 관리하는 것이 주 업무
- 트랙터의 작업별 이용원리, 특징 파악
- 트랙터 기종별 하단에 위치한 규격 및 제조번호 유무 확인
- 온도, 습도, 가열된 정도, 주유 현황, 주행 거리 등 트랙터의 다양한 정보 확인



[트랙터 하단부, 트랙터 컨트롤박스]

주소	성명	연령	트랙터 제조사	제조년도	규격(마력)	가게사용년수	경작면적	장기적인 작업시간(월)(주,중,일)	장기적인 작업일 수(일)	장기적인 일 평균 작업시간(시)	장기적인 연료 소비량(말)	1차 로타리 작업 작업시간(시)(주,중,일)	1차 로타리 작업 작업일 수(일)
경상남도 사천시 사남면	조인태	60	국제	2015	70	18	10ha	11	20	4	35	4	10
경상남도 사천시 용현면	차봉주	62	LS	2021	62	20	4ha	11	10	8	50	5	10
경상남도 사천시 서포면	백경선	50	LS	2018	68	20	7ha	11	10	8	35	5	10
경상남도 사천시 사남면	규광훈	65	대우	2013	70	16	5ha					3	3
경상남도 사천시 서포면	김유민	48	안마	2016	53	15	3ha	11	7	8	42	4	14
경상남도 사천시 사천읍	서광복	-	LS	2017	80	20	10ha	4	7	5	9	5	5
경상남도 사천시 북동면	김성우	67	LS	2014	85	20	20ha	11	30	9	140	4	30
경상남도 사천시 서포면	김광수	37	LS	2011	115	20	20ha	3	30	7	150	4	30
경상남도 사천시 이봉동	노영철	45	국제	2019	110	18	7ha	10	30	8	30	4	10
경상남도 사천시 풍동면	정근영	58	대우	2020	127	16	13ha	4	25	7	12	5	25
경상남도 진주시 이반성면	김광수	65	LS	2015	52	10	3	11	15	4	140	5	6
경상남도 진주시 이반성면	심필로	70	대우	2010	70	10	7	10	7	4	3.5	5	5
경상남도 진주시 이반성면	김광훈	81	대우	2009	64	10	4	5.10	7	4	3.5	2.3	5
경상남도 진주시 이반성면	이광복	67	대우	2004	73	5	5.5	11	3-4	6	3	1.12	2-3
경상남도 진주시 이반성면	정호복	69	대우	2018	60	15	3					3.4	8
경상남도 진주시 이반성면	최정호	69	대우	2018	63	10	3					3.4	8
경상남도 진주시 이반성면	최인달	73	구부다	2020	70	20	4					4	5
경상남도 진주시 이반성면	이용익	57	대우	2020	58	15	1.5					11	2
경상남도 진주시 이반성면	손홍복	77	대우	2011	43	5	1.6					5	3
경상남도 진주시 이반성면	허복룡	73	대우	2019	73	17	3	2.3	7	2	2	5	10
경상남도 진주시 사봉면	남현민	70	국제	2011	45	10	6000평					4	5
경상남도 진주시 사봉면	남승현	68	구부다	2012	55	10	20,000평					4	7
경상남도 진주시 사봉면	정호복	59	LS	2017	55	10	7.5	11	4	6	12	4	6
경상남도 진주시 사봉면	김태우	60	LS	2006	72	5	5					30	5
경상남도 진주시 사봉면	정환진		국제		75	4	80마지	11.12	8		13	3	12

[엑셀 데이터 수집]

주소	성명	연령	트랙터제조사	제조년도	규격(마력)	기계사용년수	경작면적(평수)	평기작업 일수	평기작업 일수/일	평기작업 일평 균작업시간(분)	평기작업 총시간	평기작업 연료 소비량(리터)
경상남도 사천시 사남면	조인래	60	국제	2015	70	18	32250	11	20.0	4.0	80.0	35.0
경상남도 사천시 용현면	차봉주	62	LS	2021	62	20	12100	11	10.0	8.0	80.0	50.0
경상남도 사천시 서포면	박영석	50	LS	2018	68	20	21175	11	10.0	8.0	80.0	35.0
경상남도 사천시 사남면	구정훈	65	대동	2013	70	16	15125				0	
경상남도 사천시 서포면	김동민	48	안파	2016	53	15	9075	11	7.0	8.0	56.0	42.0
경상남도 사천시 사천읍	서보경		LS	2017	80	20	32250	3	7.0	5.0	35.0	9.0
경상남도 사천시 옥동면	한성수	67	LS	2014	85	20	60500	11	30.0	8.0	240.0	140.0
경상남도 사천시 서포면	김영우	37	LS	2011	115	20	60500	3	30.0	7.0	210.0	150.0
경상남도 사천시 미동읍	노영철	45	국제	2019	110	18	21175	10	30.0	8.0	240.0	30.0
경상남도 사천시 절동면	정연철	58	대동	2020	127	16	39325	3	25.0	7.0	175.0	12.0
경상남도 진주시 미반성면	강광호	65	LS	2015	62	10	9075	10	15.0	4.0	60.0	140.0
경상남도 진주시 미반성면	심활보	70	동양	2010	70	10	21175	10	7.0	4.0	28.0	3.5
경상남도 진주시 미반성면	김광중	81	대동	2009	64	10	12100	10	7.0	4.0	28.0	3.5
경상남도 진주시 미반성면	이영록	67	동양	2004	73	5	16637	11	3.5	6.0	21.0	3.0

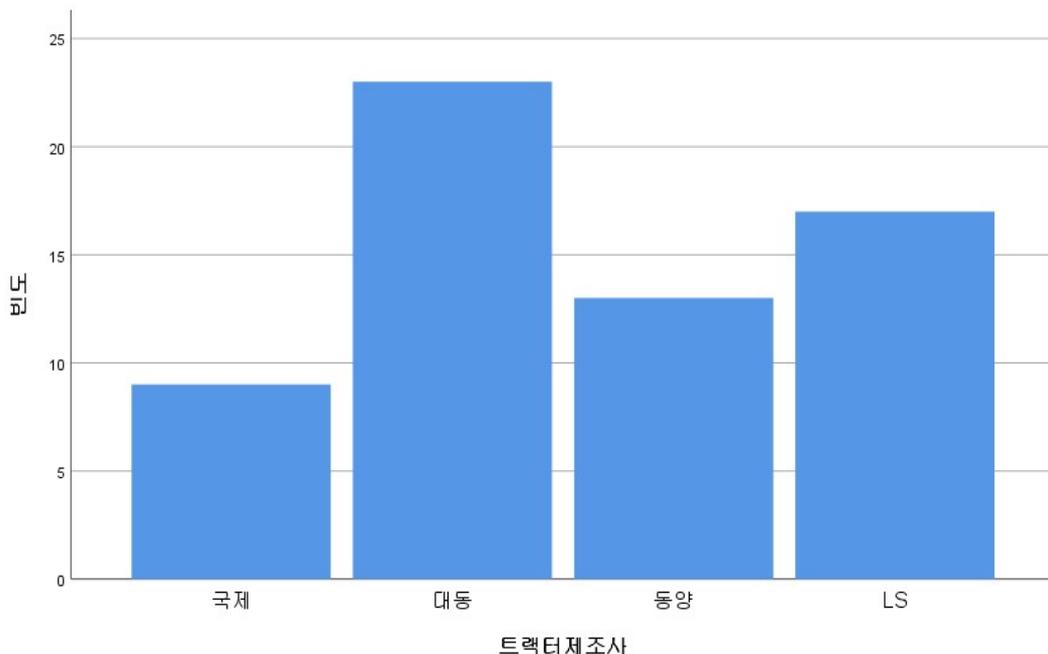
[SPSS 데이터 수집]

(다) 데이터 항목수집 및 전처리

- 사천시 : 10농가, 진주시 : 62농가
- 공통변수 : 주소, 성명, 연령, 트랙터제조사, 제조년도, 규격(마력), 기계사용년 수, 경작면적 (평수), 작업 시기, 작업일 수, 작업시간, 연료소비량
- 작업내용
  - 벼농사 : 쟁기작업, 1차로터리 작업, 2차 로터리 작업, 비료 살포작업, 씨래 작업, 모내기 작업(모 운반), 기타 작업
  - 보리농사 : 보리씨앗 살포작업, 1차 로터리 작업, 베토 작업, 기타작업
  - 밭농사 : 로터리 작업, 기타 작업
  - 기타\_콩 : 로터리 작업, 콩 파종 작업
  - 기타\_밀, 라이그라스 : 보리씨앗 살포작업, 1차 로터리 작업, 베토 작업, 기타 작업
  - 기타\_밀, 귀리 : 보리씨앗 살포작업, 곤포 작업, 1차 로터리 작업, 베토 작업

(라) 트레이닝 데이터 항목 특징 분석

- 트랙터 제조사
  - 빈도 5이하인 제조사들을 제외한 막대그래프
  - 대동이 23개로 제일 빈도가 높고 LS가 17개, 동양 13개, 국제 9개 순으로 나타남.



[트랙터 제조사 빈도 막대 그래프]

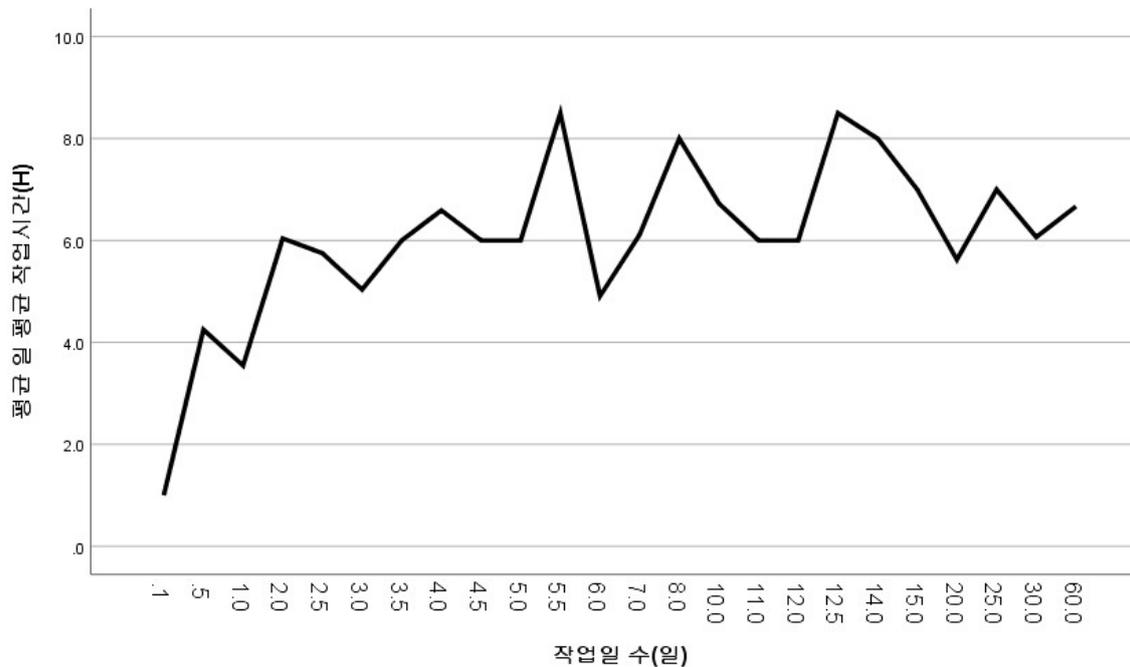
□ 기계사용년 수

- 1~ 20년 이상인 기계들의 년수를 범주화함
- 1~5, 6~10, 11~15, 16~20으로 5년의 간격을 두고 빈도표를 통해 분포를 확인
- 1~5년의 사용 트랙터가 41.7%로 가장 많았고, 20년 이상의 트랙터들도 7.5%인 것을 확인



□ 작업일 수와 작업시간

- 작업일 수와 작업시간의 상관관계를 알아보고자 함
- 작업일 수와 작업시간을 곱하여 작업 총시간의 변수를 생성



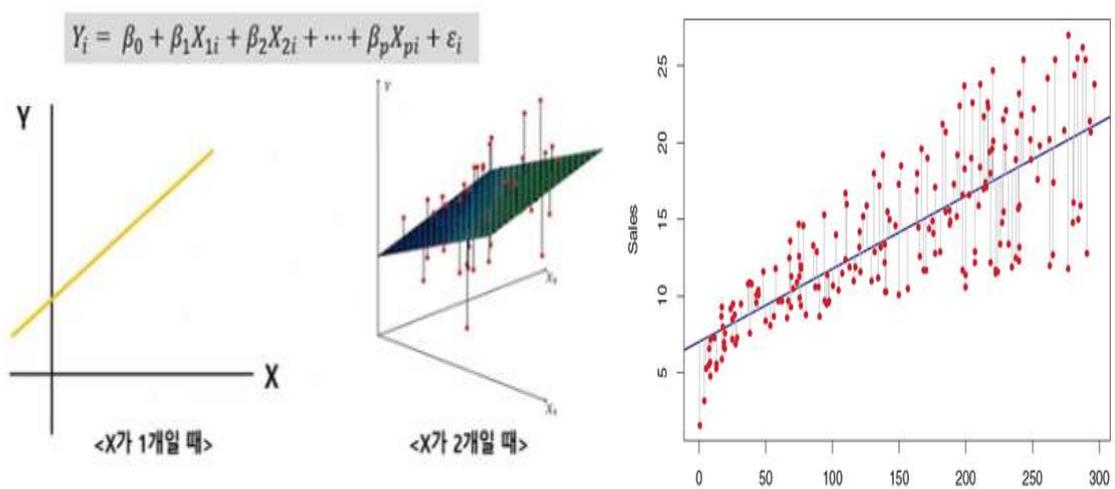
2) 농기계 이력/운행 데이터 기반 AI 알고리즘 개발

가) 트랙터 유류소모량 추정 기본 AI 알고리즘 개발

(1) 유류소모량 추정에 사용된 AI 머신러닝 및 딥러닝 알고리즘

(가) 다중 선형회귀 (Linear Regression)

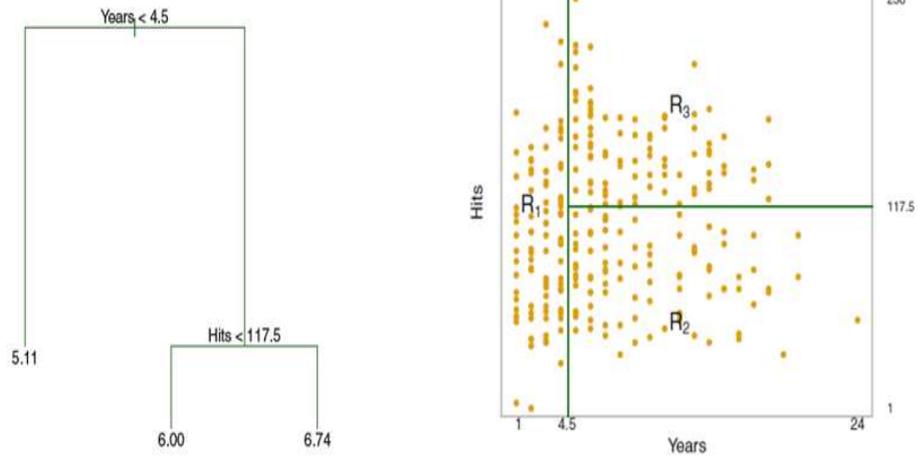
- 두 개 이상의 설명변수 X에 기초하여 양적 반응변수 Y를 예측
- X와 Y사이의 선형적 상관관계가 있다고 가정
- $X_i$  : i번째 설명변수,  $\beta_i$  : i번째 설명변수와 반응변수의 연관성 수량화
- $X_i$ 의 한 유닛 증가가 Y에 미치는 영향(다른 설명변수 변동X)
- 계수 추정할 때 사용하는 방법인 최소제곱법을 사용하여 RSS를 최소화하는 계수추정값 B0와 B1을 선택



[다중 선형회귀식과 최소제곱법을 이용한 계수추정]

(나) 회귀결정트리 (DecisionTree Regressor)

- 계층화나 분할을 한 뒤 예측하는 형태
- 훈련관측치가 속하는 영역의 평균 또는 최빈값을 이용
- 설명변수 공간 즉,  $X_1, X_2, \dots, X_p$ 에 대한 가능한 값들의 집합을 J개의 서로 겹치지 않는 영역  $R_1, R_2, \dots, R_p$ 로 분할
- 영역  $R_j$ 에 속하는 모든 관측치들에 동일한 예측을 하며, 예측값은  $R_j$ 의 훈련 관측치들에 대한 반응변수 값들의 평균이다.



[의사결정나무와 RSS를 최소화하는 예측 Box]

\* 출처 :

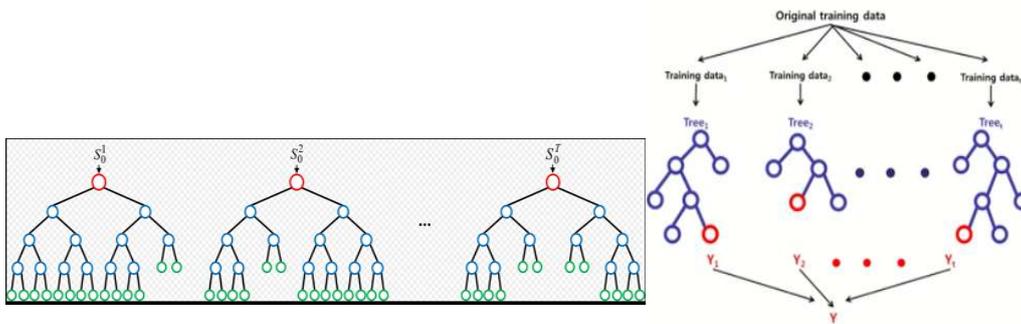
<https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-4-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%ACDecision-Tree>

코드 실행	<pre> from sklearn.linear_model import LinearRegression lr = LinearRegression().fit(X_train, y_train) print("훈련 데이터 점수:", lr.score(X_train, y_train)) print("테스트 데이터 점수:", lr.score(X_test, y_test))  from sklearn.tree import DecisionTreeRegressor tree = DecisionTreeRegressor().fit(X_train, y_train) print("훈련 데이터 점수:", tree.score(X_train, y_train)) print("테스트 데이터 점수:", tree.score(X_test, y_test))  from sklearn.svm import SVR svr = SVR() svr.fit(X, y) print("훈련 데이터 점수:", svr.score(X_train, y_train)) print("테스트 데이터 점수:", svr.score(X_test, y_test)) </pre>
분석 결과	<p> 훈련 데이터 점수: 0.4082240039970364  테스트 데이터 점수: 0.5580374205015124  훈련 데이터 점수: 0.9906742440729459  테스트 데이터 점수: 0.665231702751091  훈련 데이터 점수: 0.17225217037081297  테스트 데이터 점수: 0.16627240809510402 </p>

[선형회귀와 회귀결정트리 알고리즘 accuracy\_score 결과]

(다) 랜덤포레스트 (RandomForest Regression)

- 앙상블 모델 중 하나로 다수의 의사결정나무에 의한 예측을 종합하는 방법
- 계층 구조로 이루어진 노드와 에지들의 집합
- 훈련단계에서 종단노드에 대한 매개변수, 내부 노드와 관련된 노드 분할 함수의 매개변수를 최적화 하는 작업 진행
- 테스트 단계에서 루트 노드에서 시작, 각 노드에 분할 함수를 입력 데이터 v에 적용
- 입력 데이터가 단말 노드에 도달할 때까지 반복되는 원리



[랜덤포레스트와 최적화하는 작업]

\* 출처 : <https://dinonotes.com/archives/2338>

(라) 부스팅 (Boosting)

- 편향을 줄이고 지도학습의 차이를 줄이기위한 앙상블 메타 알고리즘
- 모델의 Residual를 가지고 weak learner를 강화
- residual을 예측하는 형태의 모델
- Adaboost는 앙상블 부스팅에서 가장 대표적인 방법이며 단순하면서 강력한 알고리즘
- Adaboost는 이전 분류기가 틀린 부분을 adaptive하게 바꾸어가며 잘못 분류되는 데이터에 집중하도록 하는 것
- AdaBoost는 의사결정나무의 성능을 향상시키는데 가장 많이 사용
- gradient boosting은 경사하강법으로 가중치를 부여하여 손실(loss)값이 작아지는 방향으로 파라미터를 움직임



[adaboost와 gradient boosting의 진행과정]

\* 출처 : <https://lsjsj92.tistory.com/544>

```

from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor(n_estimators=200, random_state=0).fit(X_train,y_train)
print("훈련 데이터 점수:", forest.score(X_train, y_train))
print("테스트 데이터 점수:", forest.score(X_test, y_test))

from sklearn.ensemble import GradientBoostingRegressor
gb = GradientBoostingRegressor(n_estimators=10, random_state=0).fit(X_train,y_train)
print("훈련 데이터 점수:", gb.score(X_train, y_train))
print("테스트 데이터 점수:", gb.score(X_test, y_test))

from sklearn.ensemble import AdaBoostRegressor
ada = GradientBoostingRegressor().fit(X_train,y_train)
print("훈련 데이터 점수:", ada.score(X_train, y_train))
print("테스트 데이터 점수:", ada.score(X_test, y_test))

```

훈련 데이터 점수: 0.9522619730339619  
 테스트 데이터 점수: 0.6314661576540138  
 훈련 데이터 점수: 0.6909135968661058  
 테스트 데이터 점수: 0.5429172689422483  
 훈련 데이터 점수: 0.9592220446955078  
 테스트 데이터 점수: 0.7026499906965471

[랜덤포레스트와 부스팅 알고리즘 accuracy\_score 결과]

(마) 릿지 (Ridge)

- 통계학에서 능형 회귀라고 부르고 선형회귀에 규제가 추가된 것
- 하이퍼파라미터인  $\alpha$ 은 모델을 얼마나 많이 규제할지 조절
- $\alpha=0$ 이면 릿지 회귀는 선형회귀와 같아지고,  $\alpha$ 가 커질수록 모든 가중치가 0에 가까워져 결국 데이터의 평균을 지나는 수평선이 됨.

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^n |\theta_i|$$

[Lidge 모델 식]

(바) 라쏘 (Lasso)

- 릿지 회귀처럼 비용함수에 규제 항을 더함
- L2 노름의 제곱을 2로 나눈 것 대신 가중치 벡터의 L1 노름을 사용
- 덜 중요한 변수의 가중치를 완전히 제거함으로 다항 회귀의 차수가 0으로 수렴
- 라쏘의 비용함수가  $\theta=0$ 에서 미분가능 하지 않기 때문에 서브그래디언트 벡터g를 사용

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^n \theta_i^2$$

[Lasso 모델 식]

(사) 엘라스틱 넷 (Elastic Net)

- 릿지 회귀와 라쏘 회귀를 절충한 모델
- 규제 항은 릿지와 회귀의 규제항을 단순히 더해 사용
- 두 규제 항의 혼합 정도를 혼합비율 r을 사용해 조절
- 변수의 수가 훈련 샘플의 수보다도 많고, 변수 몇 개가 강하게 연관되어 있을 때 엘라스틱 넷을 선호

$$J(\theta) = \text{MSE}(\theta) + r\alpha \sum_{i=1}^n |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^n \theta_i^2$$

[Elastic Net 모델 식]

\* 출처 : [https://yganalyst.github.io/ml/ML\\_chap3-4/](https://yganalyst.github.io/ml/ML_chap3-4/)

## # 라소, 엘라스틱넷, 리지

```
from sklearn.linear_model import Lasso
lasso = Lasso().fit(X_train, y_train)
print("훈련 데이터 점수:", lasso.score(X_train, y_train))
print("테스트 데이터 점수:", lasso.score(X_test, y_test))

from sklearn.linear_model import ElasticNet
elasticnet = ElasticNet(alpha=0.2, l1_ratio=0.2).fit(X_train, y_train)
print("훈련 데이터 점수:", elasticnet.score(X_train, y_train))
print("테스트 데이터 점수:", elasticnet.score(X_test, y_test))

from sklearn.linear_model import Ridge
ridge = Ridge().fit(X_train, y_train)
print("훈련 데이터 점수:", ridge.score(X_train, y_train))
print("테스트 데이터 점수:", ridge.score(X_test, y_test))
```

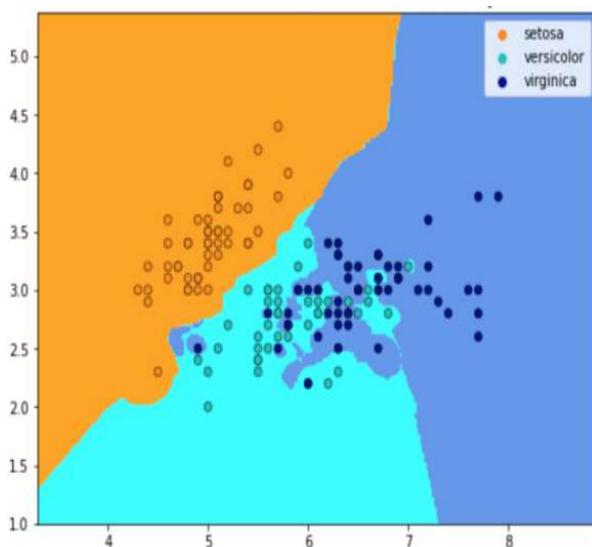
```
훈련 데이터 점수: 0.4511804082192905
테스트 데이터 점수: 0.49466600215528933
훈련 데이터 점수: 0.45122473333428714
테스트 데이터 점수: 0.49352874261322843
훈련 데이터 점수: 0.4512248472263263
테스트 데이터 점수: 0.4934631054238693
```

[라소, 리지, 엘라스틱넷 알고리즘 accuracy\_score 결과]

(아) K-최근접 이웃 (KNeighbors Regressor)

- 데이터로부터 가까운 'k'개의 다른 데이터의 레이블을 참조하여 분류하는 알고리즘
- 유사한 레코드들의 평균을 찾아 새로운 레코드에 대한 예측값으로 사용
- 지도학습 알고리즘. 예측변수에 따른 정답데이터를 기반으로 새로운 변수의 정답을 찾아가는 방법
- 모델을 훈련 및 학습, 피팅하는 과정이 있는데 K-최근접 이웃 알고리즘은 이 과정이 없는 Lazy model
- 특징들이 어떤 척도에 존재하는지, 가까운 정도에 어떻게 측정할 것인지, k를 어떻게 설정할 것인지에 따라 예측결과가 달라짐.

\* 출처 : <https://lovelydiary.tistory.com/372>



## # k-최근접 이웃 == KNN

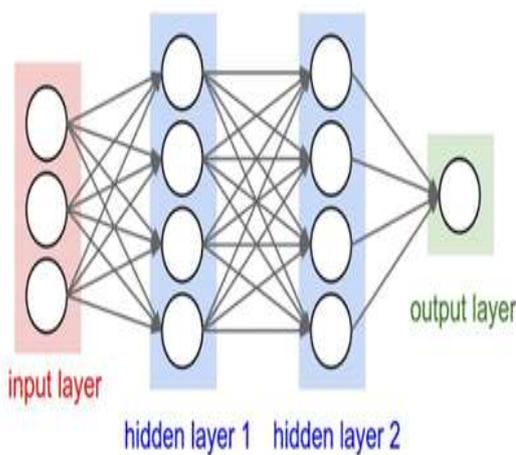
```
from sklearn.neighbors import KNeighborsRegressor
reg=KNeighborsRegressor(n_neighbors=3)
reg.fit(X_train, y_train)
print(reg.score(X_train, y_train))
print(reg.score(X_test, y_test))
```

```
0.8444836861974648
0.7130032745153163
```

[K-최근접 이웃 알고리즘 accuracy\_score 결과]

(자) 회귀 다층퍼셉트론 (Multi-Layer Perceptron)

- 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘
- x는 입력값, w는 가중치 y는 출력값
- XOR게이트는 기존의 AND, NAND, OR 게이트를 조합한 형태
- 입력층과 출력층 사이에 존재하는 은닉층을 더 추가한 딥러닝 알고리즘
- 학습 단계에서 손실 함수와 옵티마이저를 사용하고 인공 신경망이 심층 신경망인 경우



```
# 회귀 다층퍼셉트론
from sklearn.neural_network import MLPRegressor
mlp = MLPRegressor(max_iter=20000, hidden_layer_sizes=(200,), random_state=100)
mlp.fit(X_train, y_train)

print("훈련 데이터 점수:", mlp.score(X_train, y_train))
print("테스트 데이터 점수:", mlp.score(X_test, y_test))
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
mlp.fit(X_train_scaled, y_train)

print("훈련 데이터 점수:", mlp.score(X_train_scaled, y_train))
print("테스트 데이터 점수:", mlp.score(X_test_scaled, y_test))

훈련 데이터 점수: -112.17003758227902
테스트 데이터 점수: -107.17667308250746
훈련 데이터 점수: 0.8143030128232552
테스트 데이터 점수: 0.5966466809902882
```

[회귀 다층퍼셉트론 accuracy\_score 결과]

\* 출처 : <https://wikidocs.net/24958>

(2) 농기계 사용패턴 및 특징 분석

(가) 기계사용 년 수별 규격(마력)의 차이

- 일원배치 분산분석 실시
- 유의확률이 0.005보다 작으므로 기계사용 년 수 별로 규격(마력)의 차이가 있음
- 분산의 동질성 검정결과 모든 기준이 유의수준 0.05보다 높으므로 분산이 같다고 해석
- 집단비교결과 Duncan의 경우 20년 이상 사용한 기계의 규격이 20년 미만 사용한 기계의 규격보다 높고 5년 이하로 사용한 기계의 규격이 6년 이상 사용한 기계의 규격보다 낮은 것을 확인
- Scheffe 방법으로는 5년 이하로 사용한 기계의 규격 정도의 차이만 있을 뿐 Duncan 방법의 경우와 같음

ANOVA					
규격(마력)					
구분	제곱합	자유도	평균제곱	F	유의확률
집단-간	8646.633	4	2161.658	7.137	.000
집단-내	18779.158	62	302.890		
전체	27425.791	66			

분산의 동질성 검정					
		Levene 통계량	자유도1	자유도2	유의확률
규격(마력)	평균을 기준으로 합니다.	1.410	4	62	.241
	중위수를 기준으로 합니다.	.957	4	62	.438
	자유도를 수정한 상태에서 중위수를 기준으로 합니다.	.957	4	48.488	.440
	절삭평균을 기준으로 합니다.	1.246	4	62	.301

		규격(마력)			
		N	유의수준 = 0.05에 대한 부분집합		
기계사용년수범주화			1	2	3
Duncan <sup>a,b</sup>	20년이상	5	40.00		
	16~20	8	46.88	46.88	
	11~15	12	53.25	53.25	
	6~10	12		62.33	62.33
	0~5	30			72.77
	유의확률			.121	.071
Scheffe <sup>a,b</sup>	20년이상	5	40.00		
	16~20	8	46.88		
	11~15	12	53.25	53.25	
	6~10	12	62.33	62.33	
	0~5	30			72.77
	유의확률			.112	.214

동질적 부분집합에 있는 집단에 대한 평균이 표시됩니다.

a. 조화평균 표본크기 9.524을(를) 사용합니다.

b. 집단 크기가 동일하지 않습니다. 집단 크기의 조화평균이 사용됩니다. I 유형 오차 수준은 보장되지 않습니다.

[기계사용 년 수별 규격 일원배치 분산분석]

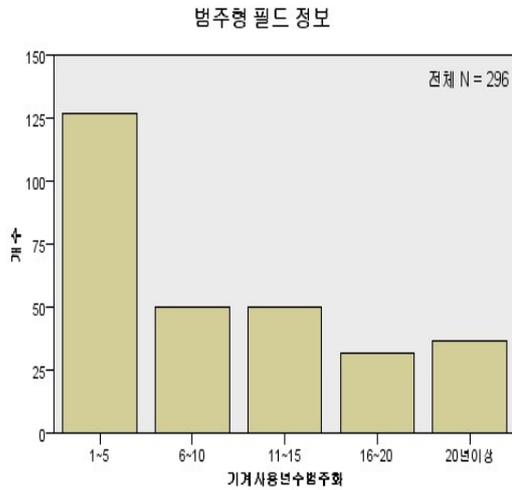
(나) 기계사용 년 수에 따른 연료 소비량 차이

- 정규성 검정 결과 기계사용 년 수 범주 모두 유의확률이 유의수준 0.05보다 낮으므로 정규성 가정이 만족 되지 않음
- 따라서 비모수 검정방법인 크루스칼 왈리스(Kruskal Wallis test) 검정 실시
- 사후 검정결과 “기계사용 년 수” 범주변수에 대한 p-value값이 0.05 미만이므로 귀무가설 기각 즉, ‘기계사용 년 수 범주에 따라 연료 소비량은 차이가 있다’라는 결론을 내릴 수 있음
- 대응별 비교탭 확인결과 ‘16~20년’의 기계사용범주가 ‘20년 이상’ 범주와 차이가 없지만 다른 범주들보다 낮은 것으로 확인
- ‘1~5년’ 범주와 ‘6~10년’ 범주가 다른 기계사용범주들보다 비교적 높게 차이가 있는 것으로 확인
- 20년 이상 사용한 기계들의 경우 6~15년 사용한 기계들과 유의한 차이가 없음을 확인

		정규성 검정					
기계사용년수범주화		Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
		통계량	자유도	유의확률	통계량	자유도	유의확률
연료 소비량(L)	1~5	.240	127	.000	.667	127	.000
	6~10	.316	50	.000	.635	50	.000
	11~15	.243	50	.000	.749	50	.000
	16~20	.329	32	.000	.680	32	.000
	20년이상	.266	37	.000	.653	37	.000

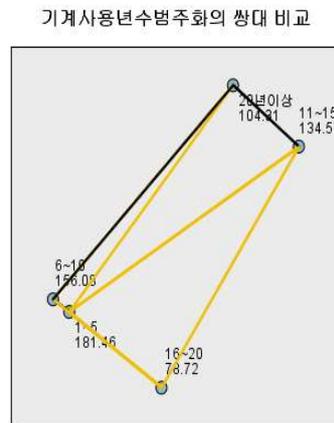
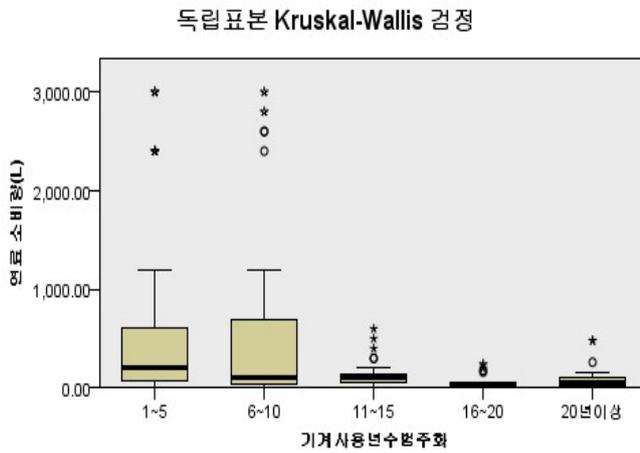
a. Lilliefors 유의확률 수정

[정규성 검정 결과표]



전체 N	296
검정 통계량	51.838
자유도	4
근사 유의수준 (양쪽검정)	.000

[범주형 자료 빈도 막대 그래프와 검정 표]



[Kruskal-Wallis 검정 box-cox 그래프와 상대 비교표]

각 노드는 기계사용년수범주화의 표본 평균 순위를 보여줍니다

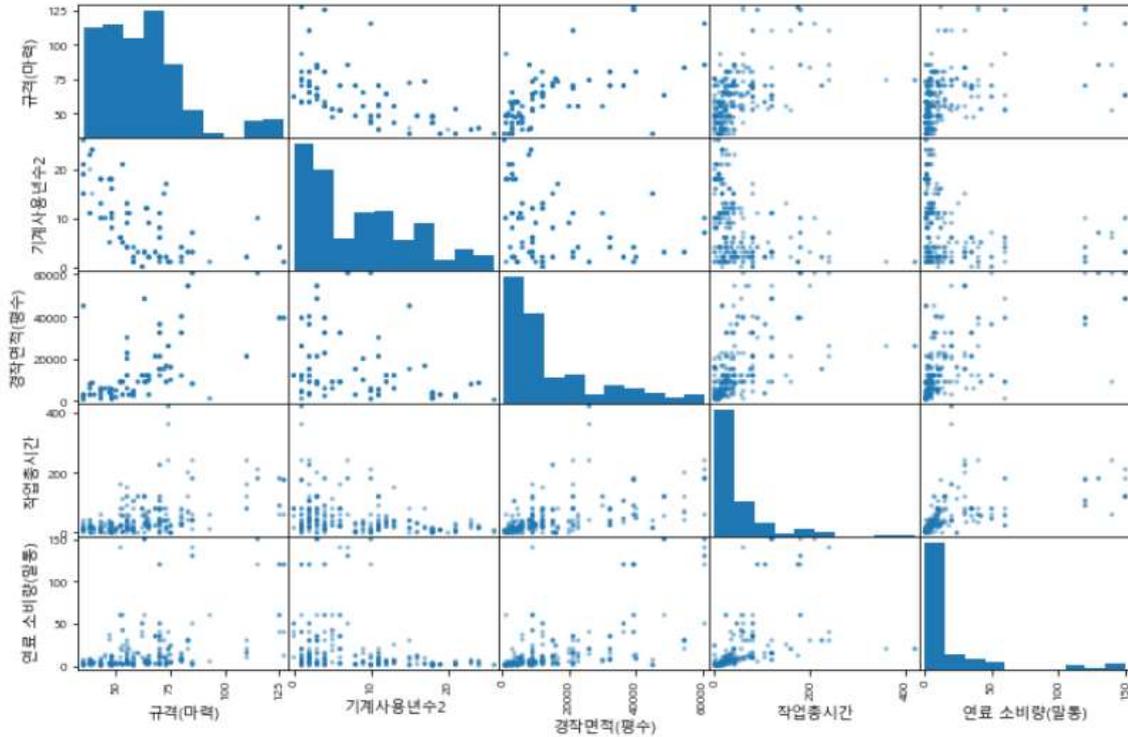
표본1-표본2	검정 통계량	표준 오차	표준 검정 통계량	Sig.	조정된 유의수준
16~20-20년이상	-25.592	20.630	-1.241	.215	1.000
16~20-11~15	55.851	19.347	2.887	.004	.039
16~20-6~10	77.361	19.347	3.999	.000	.001
16~20-1~5	102.738	16.904	6.078	.000	.000
20년이상-11~15	30.259	18.532	1.633	.103	1.000
20년이상-6~10	51.769	18.532	2.793	.005	.052
20년이상-1~5	77.146	15.965	4.832	.000	.000
11~15-6~10	21.510	17.092	1.258	.208	1.000
11~15-1~5	46.887	14.268	3.286	.001	.010
6~10-1~5	25.377	14.268	1.779	.075	.753

각 행은 표본 1 및 표본 2 분포가 동일한 귀무가설을 검정합니다. 근사 유의수준(양쪽 검정)이 표시됩니다. 유의 수준은 .05입니다. 유의성 값이 다중 검정에 대해 Bonferroni 수정으로 조정되었습니다.

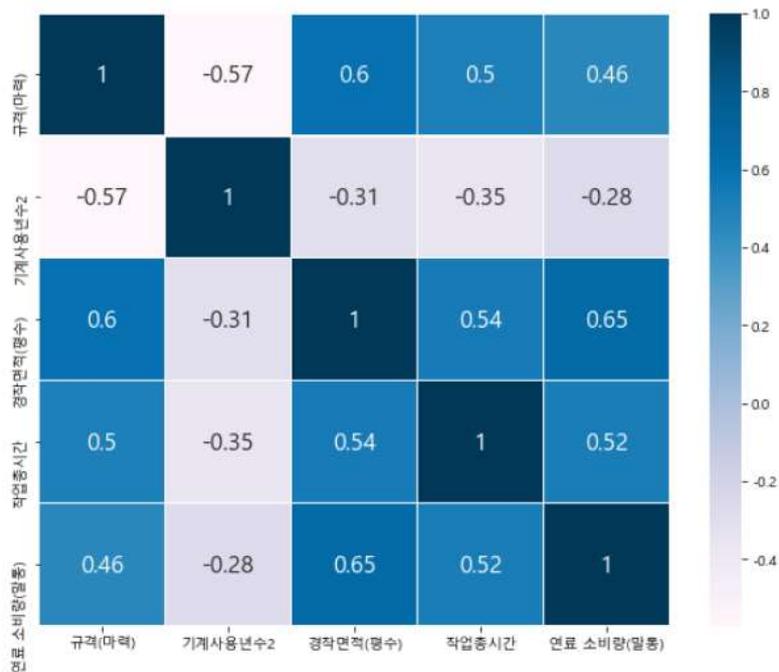
[대응별 비교 탭]

(나) 독립변수와 연료소비량의 상관관계

- 연료소비량을 예측하기 위한 변수와 상관성을 파악하기 위해 산점도와 상관관계 히트맵을 작성
- 경작면적(평수) 변수와 0.65로 어느 정도 높은 양의 상관관계가 있는 것을 확인
- 작업 총시간과 규격(마력) 변수와도 0.52, 0.46으로 양의 상관관계가 있는 것을 확인
- 기계사용년수는 -0.28로 적은 음의 상관성을 보이는 것으로 확인
- 통계 모델이나 시모델을 형성할 때 중요한 지표가 될 것으로 보임



[독립변수들의 산점도 분포표]



[Correlation coefficient Heat map]

3) 다양한 기법을 이용한 AI 알고리즘 고도화

가) 통계기반 영향 인자 도출 후 고도화 반영

(1) 연료 소비량 예측 다중 회귀선형 모델 고도화

- 독립변수 : 트랙터제조사, 규격(마력), 기계사용 년 수, 작업 총 시간,
- 종속변수 : 연료 소비량
- ANOVA 테이블 결과 R<sup>2</sup>값이 0.459로 독립변수들이 종속변수를 충분하게 잘 설명하는 것으로 보임
- 자기 상관계수 값이 2.233로 2에 가까운 값으로 자기 상관은 존재하지 않는 것으로 판단
- 회귀모형의 진단을 나타내는 ANOVA 테이블에 대한 분석표를 보면, F값은 10.377이고, 유의확률은 유의수준 0.05를 넘기지 않으므로 모형이 적합
- 각각의 독립변수들이 종속변수에 영향을 미치지지에 대한 분석표인 계수를 보면 경작면적(평수), 작업 총시간 변수가 유의수준 0.05를 넘지 않아 종속변수에 유의미한 독립변수인 것을 확인 가능
- 규격(마력), 기계사용 년 수 변수는 유의수준 0.05를 넘어 유의미한 변수라고 볼 수는 없음
- 회귀모형 공선성의 가정을 확인하기 위해 공선성의 통계량인 VIF계수를 보면 0.1보다 작거나 10보다 크지 않으므로 공선성이 존재하지 않는다고 할 수 있음

모형 요약<sup>b</sup>

모형	R	R 제곱	수정된 R 제곱	추정값의 표준오차	Durbin-Watson
1	.677 <sup>a</sup>	.459	.414	25.9280	2.233

a. 예측자: (상수), 2021-제조년도, 경작면적(평수), 쟁기작업 총 시간, 규격(마력)  
 b. 종속변수: 쟁기작업\_연료 소비량(말통)

ANOVA<sup>a</sup>

모형	회귀	제곱항	자유도	평균제곱	F	유의확률
1	회귀	27903.320	4	6975.830	10.377	.000 <sup>b</sup>
	잔차	32940.815	49	672.262		
	전체	60844.135	53			

- a. 종속변수: 쟁기작업\_연료 소비량(말통)  
 b. 예측자: (상수), 2021-제조년도, 경작면적(평수), 쟁기작업 총 시간, 규격(마력)

계수<sup>a</sup>

모형	비표준화 계수	표준화 계수	t	유의확률	공선성 공차	통계량 VIF
1 (상수)	6.579	17.332	.380	.706		
규격(마력)	-.209	.247	-.846	.401	.455	2.198
경작면적(평수)	.001	.000	2.842	.007	.610	1.640
쟁기작업 총 시간	.248	.070	3.525	.001	.605	1.653
2021-제조년도	-.034	.630	-.053	.958	.681	1.468

- a. 종속변수: 쟁기작업\_연료 소비량(말통)

[연료 소비량 예측 다중 선형회귀 모델]

## 나) 머신러닝 기반 특징 도출 후 고도화

### (1) AI 머신러닝 모델 파라미터 조정 및 고도화 작업

#### (가) Gradient Boosting Regressor Grid Search 과정

- 그리드 서치를 통해 옵션을 직접 부여하여 최적의 옵션을 찾는 과정 실시
  - 가장 중요한 매개변수인 learning\_rate를 통해 이전 트리의 오차를 얼마나 강하게 보정할 것인지 제어
  - 앙상블에 트리를 결정하는 n\_estimators 값을 조정함으로써 모델의 복잡도를 설정
  - max\_depth 매개변수를 통해 트리의 복잡도를 설정
  - alpha 값을 조정함으로 가중치에 대한 정도를 부여
  - 테스트 데이터 점수 결과 정확도가 0.64 -> 0.75 증가
- \* 출처 : <https://tensorflow.blog/>

```
# 그리드서치 옵션부여
from sklearn.model_selection import GridSearchCV
param_grid = {'learning_rate' : [0.075, 0.08, 0.085, 0.09, 0.95],
              'n_estimators' : [260, 265, 270, 275, 280, 285],
              'max_depth' : [1, 2],
              'alpha' : [0.0003, 0.0004, 0.0005, 0.0007, 0.001]}

# 그리드서치 최적옵션 찾기 수행
grid_search = GridSearchCV(gb, param_grid, cv = 5)
grid_search.fit(X, y)
grid_search.best_params_
```

```
{'alpha': 0.0003, 'learning_rate': 0.95, 'max_depth': 1, 'n_estimators': 260}
```

[GridSearch option control]

## 랜덤포레스트, 그래디언트부스팅, 아다부스팅 ¶

```
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor(n_estimators=110, random_state=0, max_leaf_nodes=40).fit(X_train, y_train)
print("훈련 데이터 점수:", forest.score(X_train, y_train))
print("테스트 데이터 점수:", forest.score(X_test, y_test))

from sklearn.ensemble import GradientBoostingRegressor
gb = GradientBoostingRegressor(random_state=0).fit(X_train, y_train)
print("훈련 데이터 점수:", gb.score(X_train, y_train))
print("테스트 데이터 점수:", gb.score(X_test, y_test))

from sklearn.ensemble import AdaBoostRegressor
ada = AdaBoostRegressor(random_state=0).fit(X_train, y_train)
print("훈련 데이터 점수:", ada.score(X_train, y_train))
print("테스트 데이터 점수:", ada.score(X_test, y_test))
```

```
훈련 데이터 점수: 0.9472386564924671
테스트 데이터 점수: 0.7400215369519418
훈련 데이터 점수: 0.9694935928492624
테스트 데이터 점수: 0.7543491686385229
훈련 데이터 점수: 0.8693718289879924
테스트 데이터 점수: 0.6466667349883782
```

[Parameter 조정 후 test score 점수]

(나) Random Forest Regressor Grid Search 과정

- 중요 매개변수에는 n\_estimators, max\_features, max\_depth가 있음
  - n\_estimators는 크면 클수록 많은 트리를 평균화하기 때문에 과대 적합을 줄여 안정적인 모델을 만드는 역할을 함. 긴 훈련시간이 단점
  - max\_features는 각 트리가 얼마나 무작위가 될지를 결정
  - max\_depth는 가지치기 옵션
  - bootstrap을 통해 데이터양을 증가시켜 오류를 최소화하는 역할을 함
- \* 출처 : <https://tensorflow.blog/>

```
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'n_estimators': [20, 30, 50, 70, 100, 110, 125]},
    {'bootstrap': [True], 'n_estimators': [30, 50, 60, 90, 100], 'max_features': [5, 8, 9, 10, 15, 20]},
]

forest_reg = RandomForestRegressor(n_jobs=-1)
grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True, n_jobs=-1)

grid_search.fit(X, y)
grid_search.best_params_
```

{'n\_estimators': 110}

[GridSearch option control]

(다) KNN Grid Search 과정

- KNN 모델에서 데이터간의 거리(Distance)는 중요한 지표이자 변수
  - 거리를 구하는 방법으로서 일반적으로 많이 사용하는 두 가지는 (1)유클리디안 거리, (2)맨하탄 거리
  - K 즉, n\_neighbors의 값을 어떻게 설정하느냐에 따라 결과가 달라짐
  - 모델의 적합성 관점에서 Overfitting 모델인지 Underfitting 모델인지를 결정
- \* 출처 : <https://leonard92.tistory.com/12>

## # k-최근접 이웃 (KNN)

```
from sklearn.neighbors import KNeighborsRegressor
reg=KNeighborsRegressor(n_neighbors=10, metric = 'manhattan', weights= 'distance' )
reg.fit(X_train, y_train)
print(reg.score(X_train, y_train))
print(reg.score(X_test, y_test))
```

0.9906742440729459  
0.8099277083293004

[Parameter 조정 후 test score 점수]

```

# 그리드서치 옵션부여
from sklearn.model_selection import GridSearchCV
param_grid = {'n_neighbors' : list(range(1,20)),
              'weights' : ["uniform", "distance"],
              'metric' : ['euclidean', 'manhattan', 'minkowski']
              }

# 그리드서치 최적옵션 찾기 수행
grid_search = GridSearchCV(reg, param_grid, cv = 5)
grid_search.fit(X, y)
grid_search.best_params_

{'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'distance'}

```

[GridSearch option control]

(라) 연료 소비량 예측 다중 회귀선형 모델 성능 향상 방안

- 트랙터 제조사, 규격(마력), 기계사용 년 수, 작업 총 시간, 연료 소비량을 변수로 사용한 다중 회귀선형 모델의 예측 결과를 실제 연료 소비량으로 사용하기에는 낮은 예측 수준으로 나타남
- 상관관계가 높은 변수들을 추가함으로 성능을 향상시킬 수 있을 것으로 판단됨
- 자동계측기능을 통하여 수집된 데이터가 아닌 설문조사를 통해 수집된 데이터로써, 센서를 활용한 신뢰성 높은 데이터를 수집함으로 고도화할 수 있을 것으로 보임

다) 연료 소비량 예측 다중 회귀선형 모델 최적의 변수 선정을 통한 고도화

(1) 최적의 변수 선정을 위한 상관관계 분석

- 트랙터의 실사용 데이터를 수집하여 추가적으로 데이터 분석
- 사용된 데이터 수 : row \* column (52921 \* 31)
- 독립변수 : 30개의 트랙터 실사용 수집 데이터

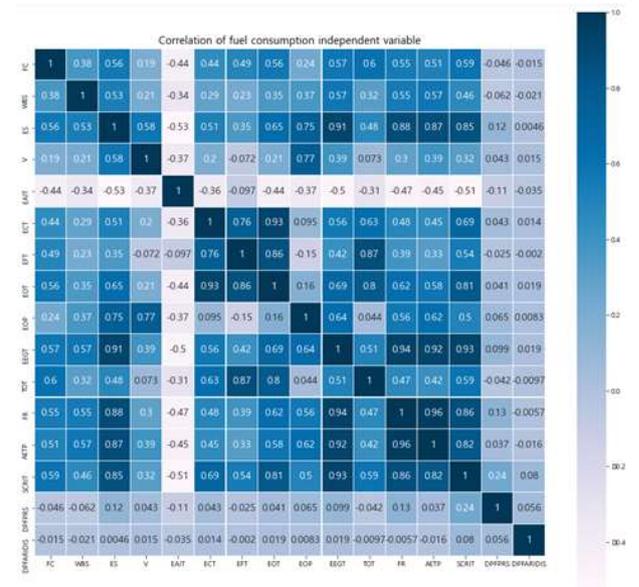
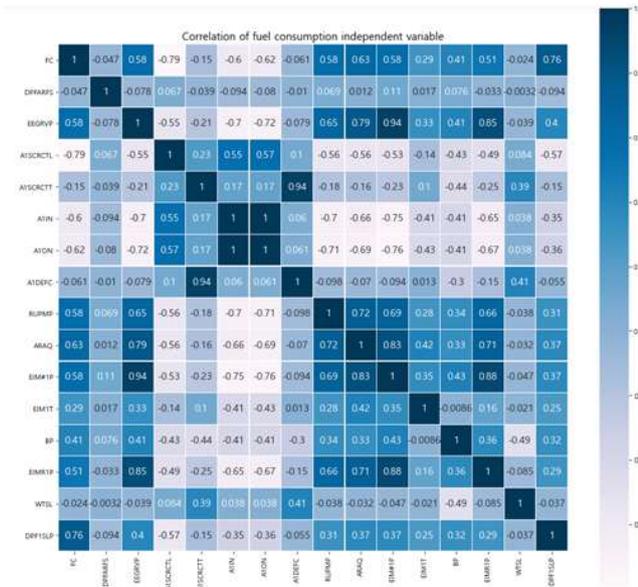
Actual Engine Torque Percent
Actual reducing agent quantity
Aftertreatment 1 DIESEL EXHAUST FLUID CONCENTRATION
Aftertreatment 1 Intake Nox
Aftertreatment 1 Outlet Nox
Aftertreatment 1 SCR Catalyst Tank Level
Aftertreatment 1 SCR Catalyst Tank Temperature
Barometric Pressure
Diesel Particulate Filter 1 Soot Load Percent
Diesel Particulate Filter Active Regeneration Forced Status
Diesel Particulate Filter Active Regeneration Inhibited Due to Inhibit Switch
Diesel Particulate Filter Lamp Command
Diesel Particulate Filter Passive Regeneration Status
Engine Air Inlet Temperature
Engine Coolant Temperature
Engine Exhaust Gas Recirculation Valve Position
Engine Exhaust Gas Temperature
Engine Fuel Temperature
Engine Injector Metering Rail 1 Pressure
Engine Intake Manifold #1 Pressure
Engine Intake Manifold 1 Temperature
Engine Oil Pressure
Engine Oil Temperature
Engine Speed
Fuel Rate
Relative urea pump module pressure
SCR inlet temperature
Transmission Oil Temperature
Voltage
Wheel Based Speed

□ 종속변수 : Fuel Consumption

□ 피어슨 상관 계수(  $-1 < r < -0.5$ ,  $+0.5 < r < 1$ )를 이용하여 연료 소비량과 상관관계가 있는 16개의 변수를 선정

$$r = \frac{\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \div \sqrt{\frac{\sum_i^n (X_i - \bar{X})^2}{n - 1}} \sqrt{\frac{\sum_i^n (Y_i - \bar{Y})^2}{n - 1}}$$

[피어슨 상관 계수 식]



[Correlation of fuel consumption independent variables]

□ 선정된 16개의 독립변수 :

Actual Engine Torque Percent
Actual reducing agent quantity
Aftertreatment 1 Intake Nox
Aftertreatment 1 Outlet Nox
Aftertreatment 1 SCR Catalyst Tank Level
Diesel Particulate Filter 1 Soot Load Percent
Engine Exhaust Gas Recirculation Valve Position
Engine Exhaust Gas Temperature
Engine Injector Metering Rail 1 Pressure
Engine Intake Manifold #1 Pressure
Engine Oil Temperature
Engine Speed
Fuel Rate
Relative urea pump module pressure
SCR inlet temperature
Transmission Oil Temperature

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Fuel Consu	Engine Spt	Engine Oil	Engine Exh	Transmissi	Fuel Rate	Actual Eng	SCR inlet	Engine Exh	Aftertreatr	Aftertreatr	Aftertreatr	Relative ur	Actual red	Engine Int	Engine Inj	Diesel Part
52902	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52903	165	953.75	11312	465.125	68.875	1.95	9	414.7813	0.0775	47.6	3276.75	3276.75	0	0	12	62.89844	1
52904	165	927.25	10516	463.875	44.26563	2.2	10	433.3281	0	47.6	2093.175	2112.675	2500	0	12	73.39844	1
52905	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52906	165	2186.125	11664	467.75	50.3125	15.9	32	494.9375	21.5725	35.2	444.6	208.25	5000	171.9	14	58.5	6
52907	165	952.375	11204	444.875	68.875	1.7	4	470.2813	0	39.4	555.4	197.9	5000	25.05	10	58.19922	2
52908	165	950	10340	443.75	20.34375	2.85	8	387.5313	0	47.6	3276.75	3276.75	-80	0	0	51.89844	0
52909	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52910	165	1032.375	11680	704.25	36.875	3.55	8.5	434.8906	10.99125	47.6	516.35	273.1	2500	10.425	108	119.8984	1.5
52911	165	0	10420	473	68.875	0	0	393.2344	0	47.6	3276.75	3276.75	-80	0	5	31.04883	0
52912	165	0	10532	455.625	26.625	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52913	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52914	165	0	10420	455.625	22.29688	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52915	165	0	10532	455.625	26.625	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52916	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52917	165	0	10532	455.625	26.625	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52918	165	0	10532	455.625	26.625	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52919	165	0	10420	455.625	22.29688	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0
52920	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52921	165	1374.24	11163.76	558.996	43.57462	6.798686	17.97979	480.3806	9.965653	41.06946	1932.693	1766.393	2840.909	58.67259	54.66791	83.97271	2.395748
52922	165	0	10420	455.625	26.625	0	0	393.2343	0	47.6	3276.75	3276.75	-120	0	5	30.94922	0

[선정된 17개 변수의 데이터셋]

(2) 피어슨 상관 계수(  $-1 < r < -0.5$ ,  $+0.5 < r < 1$ )를 통하여 선정된 변수의 다중회귀분석 모델

□ 사용된 데이터 수 : row \* column (52921 \* 17)

□ Training : Test = 80 : 20

□ 도출된 다중회귀분석 식 :

$$FC = -0.0512 - 1.0490(a) + 0.0624(b) + 0.4693(c) + 0.0504(d) + 0.1062(e) + 0.2409(f) + 0.0257(g) + 0.0586(h) - 0.1021(i) - 0.1393(j) - 0.0429(k) - 0.0594(l) + 0.0490(m) + 0.5538(n) - 0.1402(o)$$

where:

FC : Fuel Consumption

a : Aftertreatment 1 SCR Catalyst Tank Level

b : Engine Speed

c : Diesel Particulate Filter 1 Soot Load Percent

d : Engine Injector Metering Rail 1 Pressure

e : Engine Intake Manifold #1 Pressure

f : Actual reducing agent quantity

g : Relative urea pump module pressure

h : Aftertreatment 1 Intake Nox

i : Aftertreatment 1 Outlet Nox

j : Engine Exhaust Gas Recirculation Valve Position

k : SCR inlet temperature

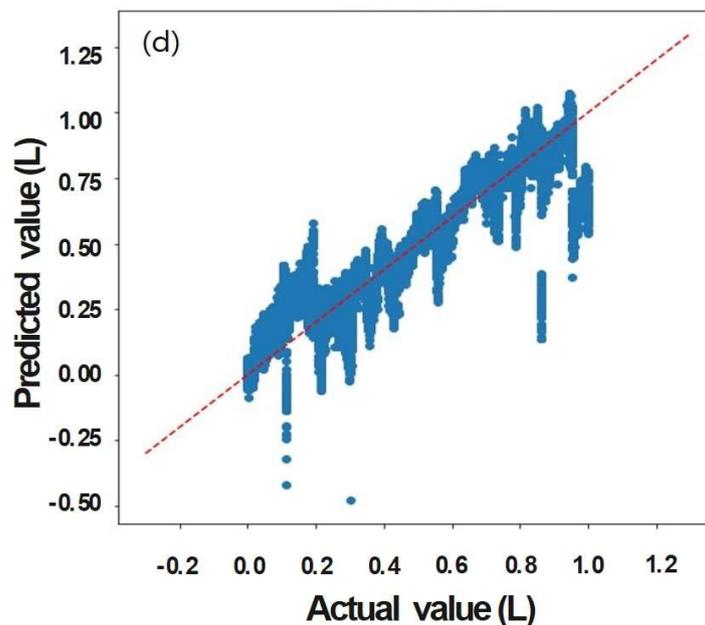
l : Actual Engine Torque Percent

m : Fuel Rate

n : Transmission Oil Temperature

o : Engine Exhaust Gas Temperature

p : Engine Oil Temperature



[Scatter plot of predicted value and actual value]

□ R<sup>2</sup> 결과가 0.836 정도로 도출됨

□ 위를 통해 상관 관계가 높은 변수를 추가함으로써 다중 회귀선형 모델의 예측 수준을 향상시킬 수 있음을 확인함

□ 이후 상관 관계가 높은 변수의 추가와 최적의 변수 선정, 센서 및 자동계측 기능을 활용한 높은 신뢰성을 가진 좋은 데이터를 수집하고, 다양한 AI 기법을 활용하여 예측 모델의 고도화를 진행

한다면 높은 예측 수준을 가지는 결과를 도출할 수 있을 것으로 판단됨

- © AI를 적용한 농기계 이력/운행 데이터 알고리즘 고도화 및 서비스를 위한 API 연동
- AI를 적용한 농기계 이력/운행 데이터 알고리즘 고도화

1. 데이터 정규화

□ 정규화 (Normalization)

- 데이터의 범위를 0 ~ 1 사이의 값으로 변환함.
- 머신러닝에서 scale이 큰 feature의 영향이 비대해지는 것을 방지함.
- 딥러닝에서 Local Minima에 빠질 위험이 감소함.
- 딥러닝의 학습률 증가와 과적합 문제를 해결할 수 있음.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

[정규화 (Normalization) 수식]

```
for month in fuels.keys():
    for day in fuels[month].keys():
        y = np.array([fuels[month][day])).reshape(-1, 1)
        y = torch.Tensor(y).cuda()
        x = data['torque'][month][day]
        if not len(data['torque'][month][day]) > width**2:
            for i in range(width**2 - len(data['torque'][month][day])):
                x.append(0)
        else:
            x = data['torque'][month][day][:width**2]
        x = np.array(x).reshape(-1, 1)
        scaler.fit(x)
        dump.append((x, y))

for x, y in dump:
    x = scaler.transform(x)
    x = torch.Tensor(x).cuda()
    trainData.append((x,y))

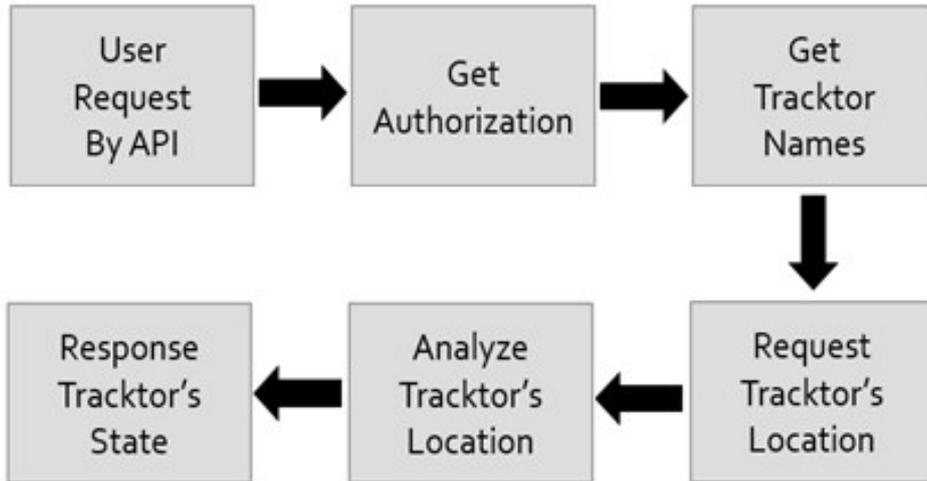
joblib.dump(scaler, 'scaler_torque.save')
```

[정규화 구현 코드 (Preprocessor.py)]

## 2. 트랙터 고도 기반 위험도 검출 및 상태 기반 경고 서비스 구현

### □ 프로세싱 흐름도 (Flowchart)

- API를 통해 사용자의 요청을 감지함.
- 사용자의 요청을 분석함.
- 연동된 데이터 API로부터 트랙터의 현재 상태 데이터를 수집함
- 수집한 트랙터의 데이터를 분석 후 결과를 반환함.



### □ 데이터베이스 (Database) 구축

- 데이터베이스 구현을 위하여 SQLite3를 이용하였음.
- 실시간 트랙터의 상태 데이터를 저장할 수 있음.
- 실시간 감지된 위험 기록을 저장할 수 있음.
- 데이터를 유지보수와 이용을 용이하게 함.

```
if not os.path.isfile('./database.db'): # 데이터 베이스가 없을 경우 생성
    con = sqlite3.connect('./database.db') # 데이터 베이스 생성
    cur = con.cursor() # 데이터 베이스 연결
    cur.execute("CREATE TABLE MODEL(name text);") # 트랙터 정보를 저장하는 테이블 생성
    cur.execute("CREATE TABLE LOG(type text, name text, date text);") # 로그를 저장하는 테이블 생성
    con.commit() # 데이터 베이스 저장
    cur.close()
    con.close()
self.loadModel() # 트랙터 정보 로드
```

[데이터베이스 구현 코드]

### □ 트랙터 데이터 권한 획득 알고리즘

- 해당 알고리즘은 트랙터 데이터 분석을 위해 시스템이 API 접근키를 획득하는 알고리즘임.
- 접근키 요청을 위해 HTTP 프로토콜을 이용하였음.
- 접근키 획득을 위해 JSON을 이용하였음.

```
def getAuth(self): # 권한 획득
    self.authToken = requests.get(self.authUrl) # 권한 요청
    self.authToken = self.authToken.json()['auth_token'] # 권한 획득
```

[API 권한 요청 구현 코드]

□ 트랙터 모델명 로드 알고리즘

- 해당 알고리즘은 시스템이 트랙터 데이터를 이용하기 위해 모델명을 불러오는 알고리즘임.
- 데이터베이스에 저장되어 있는 트랙터 정보를 불러옴.
- 트랙터 정보를 불러오기 위해 SELECT 명령어를 이용하였음.
- 각 트랙터에 실시간 상태 데이터를 저장하기 위해 QList를 이용하였음.

```
def loadModel(self): # 트랙터 정보 로드
    con = sqlite3.connect('./database.db')
    cur = con.cursor() # 데이터 베이스 연결
    cur.execute('SELECT * FROM MODEL') # 트랙터 정보 로드
    for row in cur:
        self.modelNames.add(row)
        self.memory.setdefault(row, self.qList(10)) # 트랙터당 QList 생성
    cur.close()
    con.close()
```

[트랙터 데이터 로드 구현 코드]

```
class qList(): # 갯수가 정해진 리스트
    def __init__(self, limitCount):
        self.limitCount = limitCount
        self.list = list()
    def add(self, value): # 정해진 갯수를 초과할 경우 맨 앞 데이터 삭제
        if len(self.list) >= self.limitCount:
            del self.list[0]
            self.list.append(value)
        else:
            self.list.append(value)
    def toNumpy(self): # Numpy 자료형으로 변환
        return np.array(self.list)
    def gradient(self): # 기울기 계산
        return numpy.gradient(self.toNumpy())
```

[QList 구현 코드]

□ 트랙터 등록 알고리즘

- 해당 알고리즘은 시스템이 사용할 수 있도록 데이터베이스에 트랙터 정보를 등록함.
- 데이터베이스에 업로드하기 위해 INSERT 명령어를 이용하였음.

```
def addModel(self, name): # 트랙터 등록
    con = sqlite3.connect('./database.db')
    cur = con.cursor() # 데이터 베이스 연결
    cur.execute('INSERT INTO MODEL VALUES(?,);', (name,)) # 트랙터 정보 추가
    cur.close()
    con.commit() # 데이터 베이스 저장
    cur.close()
    con.close()
    self.loadModel() # 트랙터 정보 갱신
```

[트랙터 등록 구현 코드]

□ 트랙터 위치기반 위험 알고리즘

- 해당 알고리즘은 트랙터의 위치 데이터 중 고도의 급격한 변화 시 위험을 감지하는 알고리즘임.
- 현재 트랙터의 위치 데이터를 얻기 위해 연동 데이터 API를 이용하였음.
- API의 current-location 서비스를 이용하였으며 데이터는 각 트랙터의 QList에 저장됨.
- 수집된 각 트랙터 데이터의 기울기를 계산하여 급격한 고도변화 감지시 위험 기록을 생성함.
- 생성된 위험 기록은 연동된 데이터베이스에 실시간으로 기록됨.
- 해당 알고리즘은 10초를 기준으로 반복 수행됨.

```
def monitoringLocation(self): # 트랙터 위치 확인
    timeStart = datetime.now()
    while True: # 무한 반복
        operation = "current-locations" # 트랙터 위치 확인 명령어
        timeNow = datetime.now()
        if (timeNow - timeStart).seconds >= 3600: # 3600초 경과
            self.getAuth()
            timeStart = datetime.now()
        url = self.url + operation # 명령어 추가
        headers = {
            'Authorization': self.authToken, # 권한 등록
        }
        try:
            for model in self.modelNames:
                params = {
                    'equipment_id': model # 트랙터 별 수형
                }
                response = requests.get(url, headers = headers, params = params) # 트랙터 위치 요청
                response = response.json()
                self.memory[model].add(int(response['altitude'])) # 트랙터 고도 추가
                try:
                    gradients = self.memory[model].gradient() # 트랙터 고도 기울기 계산
                except:
                    gradients = []
                if not gradients == []:
                    for gradient in gradients:
                        if gradient >= self.threshold: # 기울기 별 임계 값 비교
                            con = sqlite3.connect('./database.db')
                            cur = con.cursor() # 데이터 베이스 연결
                            cur.execute("INSERT INTO LOG VALUES(?, ?, ?);", ('Location-danger', model, str(datetime.now()))) # 기울기가 임계 값을 넘을 경우 경고 로그 추가
                            con.commit() # 데이터 베이스 저장
                            con.close()
                            cur.close()
                            con.close()
            print("Got Tractor Data From API Server ", str(datetime.now()))
        except:
            print(traceback.format_exc())
            print("Failed Tractor Data From API Server ", str(datetime.now()))
        pass
    time.sleep(10) # 해당 주기로 반복
```

[트랙터 위치기반 위험 알고리즘 구현 코드]

	type	name	date
	필터	필터	필터
1	test_warning	tg130-proto-00000001	2022-09-08
2	test_warning	tg300-proto-00000001	2022-11-04
3	test_warning	tg300-proto-00000002	2022-10-18

[위험 기록 데이터베이스 테이블]

□ 트랙터 상태기반 경고 알고리즘

- 해당 알고리즘은 트랙터가 시스템에 등록된 가동 유무 상태와 차이를 감지하는 알고리즘임.
- 현재 트랙터의 위치 데이터를 얻기 위해 연동 데이터 API를 이용하였음.
- API의 current-location 서비스를 이용하였으며 데이터는 각 트랙터의 QList에 저장됨.
- 수집된 각 트랙터 데이터를 분석하여 현재 시스템에 등록된 가동상태 여부와 다를 경우 경고 기록을 생성함.
- 생성된 경고 기록은 연동된 데이터베이스에 실시간으로 기록됨.

○ 해당 알고리즘은 10초를 기준으로 반복 수행됨.

```
def monitoring(self, modelName): # 트랙터 상태 확인
    if modelName == '': return 'error' # 비어있을 경우
    operation = "latest-state" # 트랙터 상태 확인 명령어
    url = self.url + operation # 명령어 추가
    headers = {
        'Authorization': self.authToken, # 권한 등록
    }
    params = {
        'equipment_id': modelName
    }
    response = requests.get(url, headers=headers, params=params) # 트랙터 정보 찾기
    response = response.json() # 'wheel_based_speed', 'engine_speed', 'engine_torque'
    wheel = int(response[0]['value'])
    speed = int(response[1]['value'])
    torque = int(response[3]['value'])
    if wheel == 0 and speed == 0 and torque == 0: # 트랙터 상태 모니터링
        return 'stop' # 트랙터 정지 메시지
    else:
        return 'moving' # 트랙터 가동중 메시지
```

[트랙터 상태기반 경고 알고리즘 구현 코드]

□ 위험 기록 확인 알고리즘

- 해당 알고리즘은 데이터베이스에 저장되어 있는 위험 기록을 반환하는 알고리즘임.
- 사용자는 제공되는 API를 통해 해당 필요로 하는 트랙터 모델명을 시스템에 요청함.
- 시스템은 사용자가 요청한 모델명과 SELECT 명령어를 이용하여 최근 해당 트랙터의 위험 기록을 사용자에게 반환함.
- 해당 트랙터의 최근 위험 기록이 존재하지 않는다면 Null을 반환함.

```
def checkWarning(self, modelName):
    if modelName == '': return ('empty') # 비어있을 경우
    con = sqlite3.connect('./database.db')
    cur = con.cursor() # 데이터 베이스 연결
    sql = 'SELECT type,name,date FROM LOG WHERE name=?'
    cur.execute(sql, (modelName,))
    rows = cur.fetchall()
    rows = rows[-1]
    cur.close()
    con.close()
    if not rows:
        return ('empty')
    else:
        return rows
```

[위험 기록 확인 알고리즘 구현 코드]

### 3. 트랙터 토크 기반 연료 예측 서비스 구현

#### □ 랜덤 시드 설정

- 시스템의 난수들은 랜덤 시드를 기반으로 생성됨.
- 랜덤 시드를 특정 값으로 고정하여 난수 생성을 통제함.
- 고정된 난수를 바탕으로 딥러닝 모델의 성능을 균일하게 판단할 수 있음.

```
def seed_everything(seed=7):  
    random.seed(seed)  
    os.environ['PYTHONHASHSEED'] = str(seed)  
    np.random.seed(seed)  
    torch.manual_seed(seed)  
    torch.cuda.manual_seed(seed)  
    torch.backends.cudnn.deterministic = True
```

[랜덤 시드 설정 구현 코드]

#### □ 트랙터 토크 기반 학습데이터 생성 알고리즘

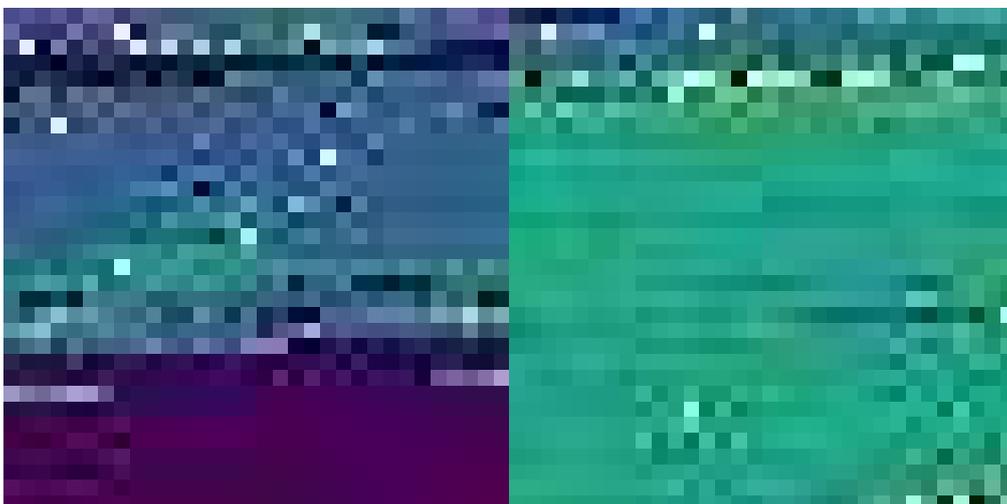
- 해당 알고리즘은 시계열 토크 데이터를 기반으로 이미지 형태의 데이터셋을 생성하는 알고리즘임.
- 시계열 토크 데이터는 이상치와 결측치를 포함할 수 있으며 해당 데이터들은 딥러닝 모델 학습에 큰 영향을 끼칠 수 있음.
- 기존의 RNN 방식의 시계열 분석 딥러닝 모델은 한번에 하나의 데이터만 처리할 수 있어 딥러닝 모델이 특정 값으로 수렴하는데 많은 시간이 소요됨.
- 시계열 토크 데이터를 하나의 이미지 형태의 데이터로 변환 및 생성한 뒤, 해당 데이터를 CNN 기반 딥러닝 모델의 합성곱 연산을 통해 멀티 연산을 할 수 있음.
- 따라서 트랙터의 토크 데이터의 값의 크기에 따라 연료소비량을 예측할 수 있음.

```

def batchBytorque(xData, yData, width):
    engineTorque = pandas.read_csv(xData)
    scaler = Normalizer()
    fuel = pandas.read_csv(yData)
    data = {
        'speed': {},
        'torque': {},
        'fuel': {}
    }
    for index in range(len(engineTorque)):
        row = engineTorque.iloc[index]
        if not row['Month'] in data['torque']:
            data['torque'][row['Month']] = {}
        if not row['Day'] in data['torque'][row['Month']]:
            data['torque'][row['Month']][row['Day']] = list()
        data['torque'][row['Month']][row['Day']].append(row['Value']*0.01)
    keys=[]
    for index in range(len(fuel)):
        row = fuel.iloc[index]
        if not row['Month'] in data['fuel']:
            data['fuel'][row['Month']] = {}
        if not row['Day'] in data['fuel'][row['Month']]:
            data['fuel'][row['Month']][row['Day']] = list()
            keys.append(row['Day'])
        if not index == 0:
            prvRow = fuel.iloc[index - 1]
            prvM = prvRow['Month']
            prvD = prvRow['Day']
            data['fuel'][prvM][prvD] = max(data['fuel'][prvM][prvD])
            prvD = row['Day']
        data['fuel'][row['Month']][row['Day']].append(row['Value'])
    finalMonth = _list(data['fuel'].keys())[-1]
    finalDay = list(data['fuel'][finalMonth].keys())[-1]
    data['fuel'][finalMonth][finalDay] = max(data['fuel'][finalMonth][finalDay])

```

[트랙터 토크 기반 학습데이터 생성 알고리즘 구현 코드 일부]



[이미지화된 토크 데이터]

- 트랙터 토크 기반 딥러닝 모델 초기화
  - 학습에 사용되는 하이퍼 파라미터를 정의함.
  - 학습률 (Learning rate)를 1e-4로 정의함.
  - 모델의 학습에 사용되는 토크 데이터를 불러옴.
  - 모델의 테스트에 사용되는 토크 데이터를 불러옴.

```

super(CNN, self).__init__()
self.inputH = 128
self.inputW = 128
self.padding = 0
self.channel = 1
self.stride = 2
self.fH = 10
self.fW = 10
self.drop = nn.Dropout(0.5)
self.conv1 = nn.Conv2d(in_channels=self.channel, out_channels=self.channel, kernel_size=self.fW,
                       stride=self.stride).cuda()
self.conv2 = nn.Conv2d(in_channels=self.channel, out_channels=self.channel, kernel_size=self.fW,
                       stride=self.stride).cuda()
self.maxpooling1 = torch.nn.MaxPool2d(6, stride=self.stride, padding=0, dilation=1,
                                       return_indices=False, ceil_mode=False).cuda()
self.maxpooling2 = torch.nn.MaxPool2d(3, stride=1, padding=0, dilation=1,
                                       return_indices=False, ceil_mode=False).cuda()
self.out = nn.Linear(64, 1).cuda()

```

[딥러닝 알고리즘 모델 초기화 코드]

- 트랙터 토크 기반 딥러닝 모델 구현
  - 딥러닝 모델을 추상화된 모듈을 이용하여 구현함.
  - 합성곱 연산을 통해 데이터를 처리함.
  - 맥스풀링 과정을 통해 가중치의 비대를 방지함.
  - 드랍아웃을 이용하여 모델의 과적합을 방지함.

```

def forward(self, x):
    x = x.view(1, self.channel, self.inputW, self.inputH)
    x = self.conv1(x)
    x = self.drop(F.relu(x))
    x = self.maxpooling1(x)
    x = self.conv2(x)
    x = self.drop(F.relu(x))
    x = self.maxpooling2(x)
    x = self.drop(F.relu(x))
    x = self.out((x.view(-1, 64)))
    return x

```

[토크 기반 딥러닝 모델 구현 코드]

□ 트랙터 토크 기반 딥러닝 모델 학습

- 트랙터 토크 기반 딥러닝 모델을 학습 데이터셋을 이용하여 학습한 후 테스트 데이터 셋을 이용하여 모델의 성능을 평가를 반복하며 모델의 가중치를 갱신함.
- 딥러닝 모델의 최적화 함수는 Adam 최적화 함수를 사용함.
- 모델의 손실함수는 RMSE 오차함수를 이용함.
- 모델의 최저 오차를 매번 갱신하여 모델이 최선의 결과를 도출할 수 있도록함.

```
def train(self):
    self.writer = SummaryWriter()
    self.cnn = CNN()
    self.cnn.cuda()
    criterion = RMSELoss().cuda()
    optimizer = optim.Adam(self.cnn.parameters(), lr=self.learningRate)
    # scheduler = optim.lr_scheduler.CosineAnnealingLR(optimizer=optimizer, T_max=30, eta_min=0)

    try: # Load Saved Model
        checkpoint = torch.load('Prdfuel.model')
        self.cnn.load_state_dict(checkpoint['model_state_dict'])
        optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
        self.trainEpoch = checkpoint['epoch']
        # loss = checkpoint['loss']
        # self.model.eval() for test
        print('** Model save data is detected')
        print('* Train Epoch:', self.trainEpoch)
    except:
        pass

    self.cnn.train()
    bestLoss = -1

    for epoch in range(self.epochs):
        idx = 1
        lossTotal = 0
        for x, y in self.trainData:
            if keyboard.is_pressed('F12'):
                torch.save({'epoch': self.trainEpoch,
                            'model_state_dict': self.cnn.state_dict(),
                            'optimizer_state_dict': optimizer.state_dict(),
                            'loss': loss},
                            'Prdfuel.model')
                print('** Model is saved')
                self.writer.flush()
                self.writer.close()
                exit()

            output = self.cnn(x) # log is except
            loss = criterion(output, y)
            lossTotal += loss.item() * self.batchSize
            print("Epoch: %d Step: %d/%d Loss: %.2f" % (self.trainEpoch, idx, len(self.trainData), loss.item()))
            optimizer.zero_grad()
```

[모델의 학습 구현 코드 일부]

□ 트랙터 토크 기반 딥러닝 모델 테스트

- 학습된 딥러닝 모델의 실성능을 평가함.
- 모델의 최고의 학습 상태를 기반으로 테스트를 진행함.
- 실사용 환경을 가정하여 구현되었기 때문에 테스트 및 실서비스 제공에 이용됨.

```

def test(x): # Must x is list or array
    if len(x) == 0: return "0.0"
    cnn = CNN()
    width = 128 # Const
    cnn.cuda()
    scaler = joblib.load('scaler_v2.save')
    optimizer = optim.Adam(cnn.parameters(), lr=0.001)
    try: # Load Saved Model
        checkpoint = torch.load('Prdfuel.model')
        cnn.load_state_dict(checkpoint['model_state_dict'])
        optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
        cnn.eval()
    except:
        return 0
    if not len(x) > width ** 2:
        for i in range(width ** 2 - len(x)):
            x.append(0)
    else:
        x = x[:width ** 2]
    x = np.array(x).reshape(-1, 1)
    x = scaler.transform(x)
    x = x.reshape(1, width, width) # Make image
    x = torch.tensor(x, dtype=torch.float32).cuda()
    with torch.no_grad():
        output = cnn(x) # log is except
        result = output.item()
        if result < 0 : result = 0
        return "%.1f"%(result)

```

[모델의 테스트 구현 코드 일부]

#### 4. 트랙터 작업량 기반 연료 예측 서비스 구현

- 트랙터 작업량 기반 학습데이터 생성 알고리즘
  - 해당 알고리즘은 시계열 작업량 데이터를 기반으로 학습 데이터셋을 생성하는 알고리즘임.
  - 모델의 하이퍼 파라미터인 Batch size를 이용하여 학습에 사용되는 Batch의 형태의 학습 데이터 세트를 생성함.

```

def set(self):
    import random
    data = pandas.read_excel('data.xlsx')
    try:
        self.scaler = joblib.load('scaler_v2.save')
    except:
        pass
    dumpTrain = []
    trainset = []
    testset = []
    for i in range(1, len(data)):
        col = data.loc[i]
        low = np.array(float(col['low'])).reshape(-1,1)
        middle = np.array(float(col['middle'])).reshape(-1,1)
        high = np.array(float(col['high'])).reshape(-1,1)
        fuel = np.array(float(col['fuel'])).reshape(-1,1)
        dumpTrain.append([low, middle, high, fuel])

    random.shuffle(dumpTrain)
    dumpTest = dumpTrain[:3] # Test data 추출
    del dumpTrain[:3]

    for low, middle, high, fuel in dumpTrain:
        self.scaler.fit(low)
        self.scaler.fit(middle)
        self.scaler.fit(high)
        low = self.scaler.transform(low)
        middle = self.scaler.transform(middle)
        high = self.scaler.transform(high)
        trainset.append([low, middle, high, fuel])

    for low, middle, high, fuel in dumpTest:
        low = self.scaler.transform(low)
        middle = self.scaler.transform(middle)
        high = self.scaler.transform(high)
        testset.append([low, middle, high, fuel])

    joblib.dump(self.scaler, 'scaler_v2.save')

```

[학습 데이터셋 생성 알고리즘 코드 일부]

#### □ 랜덤 시드 설정

- 시스템의 난수들은 랜덤 시드를 기반으로 생성됨.
- 랜덤 시드를 특정 값으로 고정하여 난수 생성을 통제함.
- 고정된 난수를 바탕으로 딥러닝 모델의 성능을 균일하게 판단할 수 있음.

```

def seed_everything(seed=7):
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True

```

[랜덤 시드 설정 구현 코드]

- 트랙터 작업량 기반 연료 예측 모델 초기화
  - 학습에 사용되는 하이퍼 파라미터를 정의함.
  - 학습률 (Learning rate)를 1e-4로 정의함.
  - 모델의 학습에 사용되는 토크 데이터를 불러옴.
  - 모델의 테스트에 사용되는 토크 데이터를 불러옴.
  - 모델에 사용될 입력 레이어를 정의함.

```
class ANN(nn.Module):
    def __init__(self):
        super(ANN, self).__init__()
        self.layerLow = nn.Linear(1, 48).cuda()
        self.layerMiddle = nn.Linear(1, 48).cuda()
        self.layerHigh = nn.Linear(1, 48).cuda()
        self.layer1 = nn.Linear(144, 36).cuda()
        self.layer2 = nn.Linear(36, 1).cuda()
        self.drop = nn.Dropout(0.8).cuda()
```

[모델 초기화 코드 일부]

- 트랙터 작업량 기반 연료 예측 모델 구현
  - 딥러닝 모델을 추상화된 모듈을 이용하여 구현함.
  - Fully Connected 레이어의 연산을 통해 데이터를 처리함.
  - 오차 역전파를 위해 Relu 활성화 함수를 이용함.
  - 드랍아웃을 이용하여 모델의 과적합을 방지함.

```
def forward(self, x):
    xLow, xMiddle, xHigh = x.split(1, dim=1)
    aLow = self.layerLow(xLow)
    aMiddle = self.layerMiddle(xMiddle)
    aHigh = self.layerHigh(xHigh)
    x = torch.cat([aLow, aMiddle, aHigh], dim=2)
    x = F.relu(x)
    x = self.layer1(x)
    x = self.drop(x)
    x = self.layer2(x)
    x = torch.reshape(x, [-1, 1])
    return x
```

[모델 구현 코드]

- 트랙터 작업량 기반 연료 예측 모델 학습
  - 딥러닝 모델을 학습 데이터셋을 이용하여 학습한 후 테스트 데이터 세트를 이용하여 모델의 성능을 평가를 반복하며 모델의 가중치를 갱신함.
  - 딥러닝 모델의 최적화 함수는 Adam 최적화 함수를 사용함.
  - 모델의 손실함수는 RMSE 오차함수를 이용함.
  - 모델의 최저 오차를 매번 갱신하여 모델이 최선의 결과를 도출할 수 있도록함.

```

def train():
    epochs = 1000
    learningRate = 0.001
    trainEpoch = 0
    writer = SummaryWriter()
    loader = Loader()
    criterion = RMSELoss().cuda()
    trainBatch, trainTarget, testBatch, testTarget = loader.set()
    ann = ANN().cuda()
    optimizer = optim.Adam(ann.parameters(), lr=learningRate)

    try: # Load Saved Model
        checkpoint = torch.load('Prdfuel_v2.model')
        ann.load_state_dict(checkpoint['model_state_dict'])
        optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
        trainEpoch = checkpoint['epoch']
        print '** Model save data is detected'
        print '* Train Epoch:', trainEpoch
    except:
        pass

    ann.train()
    bestLoss = -1

    for epoch in range(epochs):
        lossTotal = 0
        for idx, x in enumerate(trainBatch):
            output = ann(x) # log is except
            loss = criterion(output, trainTarget[idx])
            lossTotal += loss.item()
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

        lossTotal /= trainBatch.shape[0]
        print("Epoch: %d Loss: %.2f" % (trainEpoch, lossTotal))
        writer.add_scalar("Loss/Train", lossTotal, trainEpoch)
        trainEpoch += 1

        lossTotal = 0
        ann.eval()

```

[모델 학습 코드 일부]

□ 트랙터 작업량 기반 연료 예측 모델 테스트

- 학습된 모델의 실성능을 평가함.
- 모델의 최고의 학습 상태를 기반으로 테스트를 진행함.
- 실사용 환경을 가정하여 구현되었기 때문에 테스트 및 실서비스 제공에 이용됨.

```

def test(low, middle, high=0):
    scaler = joblib.load('scaler_v2.save')
    vlow = np.array([float(low)]).reshape(-1, 1)
    vmiddle = np.array([float(middle)]).reshape(-1, 1)
    vhigh = np.array([float(high)]).reshape(-1, 1)

    vlow = scaler.transform(vlow)
    vmiddle = scaler.transform(vmiddle)
    vhigh = scaler.transform(vhigh)

    batch = torch.tensor([[vlow, vmiddle, vhigh]], dtype=torch.float32)
    batch = batch.view(-1, 3, 1).cuda()

    ann = ANN().cuda()
    optimizer = optim.Adam(ann.parameters(), lr=0.0001)

    checkpoint = torch.load('Prdfuel_v2_best.model')
    ann.load_state_dict(checkpoint['model_state_dict'])
    optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
    ann.eval()
}
with torch.no_grad():
    output = ann(batch) # log is except
    result = output.item()
    if result < 0: result = 0
}
return "%.1f" % (result)

```

[모델 테스트 코드 일부]

## 5. API 서비스 제공을 위한 서버 구축

- 트랙터 고도 기반 위험도 검출 서비스
  - API 서비스 제공을 위한 알고리즘을 구현함.

```

@app.route('/tractor_warning', methods=['GET']) # 트랙터 고도 기반 위험도 검출 서비스 (최근 위험 정보 반환)
def location():
    modelName = request.args.get('equipment', default='', type=str)
    monitor = tractorMonitor.Monitor()
    result = monitor.checkWarning(modelName)
}
if len(result) == 1:
}
    response = {
        'type': 'tractor_location',
        'date': datetime.datetime.now(),
        'equipment': modelName,
        'date_waring': '',
        'waring': ''
    }
}
else:
}
    response = {
        'type': 'tractor_location',
        'date': datetime.datetime.now(),
        'equipment': modelName,
        'date_waring': result[2],
        'waring': result[0]
    }
}
return jsonify(response), 200

```

[트랙터 고도 기반 위험도 검출 서비스 API]

- 트랙터 상태 기반 경고 서비스
  - API 서비스 제공을 위한 알고리즘을 구현함.

```
@app.route('/tractor_monitoring', methods=['GET']) # 트랙터 현재 작동여부 모니터링 서비스
def monitoring():
    modelName = request.args.get('equipment', default='', type=str)
    monitor = tractorMonitor.Monitor()
    result = monitor.monitoring(modelName)
    response = {
        'type': 'tractor_monitoring',
        'date': datetime.datetime.now(),
        'equipment': modelName,
        'state': result
    }
    return jsonify(response), 200
```

[트랙터 상태 기반 경고 서비스 API]

- 트랙터 토크 기반 연료 예측 서비스
  - API 서비스 제공을 위한 알고리즘을 구현함.

```
@app.route('/predict_fuel_torque', methods=['GET'])
def predcitBytorque(): # 토크 기반 연료 예측 서비스
    input = request.args.get('input', default='', type=str)
    input = input.split(',')
    result = prdFuel_v1.test(input)
    response = {
        'type': 'predict_fuel_torque',
        'date': datetime.datetime.now(),
        'predicted_fuel': result
    }
    return jsonify(response), 200
```

[트랙터 토크 기반 연료 예측 서비스 API]

- 트랙터 작업량 기반 연료 예측 서비스
  - API 서비스 제공을 위한 알고리즘을 구현함.

```
@app.route('/predict_fuel_operation', methods=['GET']) # 작업량 기반 연료 예측 서비스
def predcitByoperation():
    low = request.args.get('low', default=0, type=float)
    middle = request.args.get('middle', default=0, type=float)
    result = prdFuel_v2.test(low, middle)
    response = {
        'type': 'predict_fuel_operation',
        'date': datetime.datetime.now(),
        'predicted_fuel': result
    }
    return jsonify(response), 200
```

[트랙터 작업량 기반 연료 예측 서비스 API]

- API 서버 구현
  - API 서버를 구현함.

```

if __name__ == "__main__":
    monitor = tractorMonitor.Monitor()
    receiver = threading.Thread(target=monitor.monitoringLocation, args=())
    receiver.setDaemon(True)
    receiver.start()
    app.run(
        host="",
        port=8000)

```

[ API 서버 구현 코드 ]

6. 인공지능 관련 활용 정량적 데이터

- Engine Speed
  - Training : Tractor c의 5월 12일 ~ 11월 5일 92,424 개의 데이터
  - Test : Tractor b의 5월 12일 ~ 11월 5일 21,902 개의 데이터
  - Training : Test = 80.8 : 19.2
  
- Engine Demand Percent Torque
  - Training : Tractor c의 6월 4일 ~ 11월 5일 92,238 개의 데이터
  - Test : Tractor b의 5월 12일 ~ 11월 5일 21,902 개의 데이터
  - Training : Test = 80.8 : 19.2
  
- Fuel Consumption
  - Training : Tractor c의 6월 4일 ~ 11월 5일 90,122 개의 데이터
  - Test : Tractor b의 6월 3일 ~ 11월 5일 20,046 개의 데이터
  - Training : Test = 81.8 : 18.2

	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
92417	11	5	c	Engine Speed	16:02:21	2349.75
92418	11	5	c	Engine Speed	16:02:26	2294.25
92419	11	5	c	Engine Speed	16:02:31	1350.75
92420	11	5	c	Engine Speed	16:02:36	1378.75
92421	11	5	c	Engine Speed	16:02:41	2111
92422	11	5	c	Engine Speed	16:02:46	1895.75
92423	11	5	c	Engine Speed	16:02:51	977
92424	11	5	c	Engine Speed	16:02:56	1872.5
92425	11	5	c	Engine Speed	16:03:01	1799.5

[Engine Speed\_Training Data]

	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
21895	11	5	b	Engine Speed	14:02:57	1104.5
21896	11	5	b	Engine Speed	14:03:02	1015.25
21897	11	5	b	Engine Speed	14:03:07	1012.25
21898	11	5	b	Engine Speed	14:03:12	1289
21899	11	5	b	Engine Speed	14:03:17	1013.75
21900	11	5	b	Engine Speed	14:03:21	0
21901	11	5	b	Engine Speed	14:03:27	0
21902	11	5	b	Engine Speed	14:03:32	0
21903	11	5	b	Engine Speed	14:03:35	0

[Engine Speed\_Test Data]

	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
90115	11	5	c	Fuel Consumption	16:02:14	1238
90116	11	5	c	Fuel Consumption	16:02:20	1238.5
90117	11	5	c	Fuel Consumption	16:02:24	1238.5
90118	11	5	c	Fuel Consumption	16:02:30	1238.5
90119	11	5	c	Fuel Consumption	16:02:40	1238.5
90120	11	5	c	Fuel Consumption	16:02:44	1238.5
90121	11	5	c	Fuel Consumption	16:02:50	1238.5
90122	11	5	c	Fuel Consumption	16:02:54	1238.5
90123	11	5	c	Fuel Consumption	16:03:00	1238.5

[Fuel Consumption\_Training Data]

	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
20039	11	5	b	Fuel Consumption	14:02:41	163
20040	11	5	b	Fuel Consumption	14:02:47	163
20041	11	5	b	Fuel Consumption	14:02:51	163
20042	11	5	b	Fuel Consumption	14:02:57	163
20043	11	5	b	Fuel Consumption	14:03:01	163
20044	11	5	b	Fuel Consumption	14:03:05	163
20045	11	5	b	Fuel Consumption	14:03:12	163
20046	11	5	b	Fuel Consumption	14:03:16	163
20047	11	5	b	Fuel Consumption	14:03:20	163

[Fuel Consumption\_Test Data]

	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
92231	11	5	c	Engine Demand Percent Torque	16:02:21	95
92232	11	5	c	Engine Demand Percent Torque	16:02:26	0
92233	11	5	c	Engine Demand Percent Torque	16:02:31	0
92234	11	5	c	Engine Demand Percent Torque	16:02:36	40
92235	11	5	c	Engine Demand Percent Torque	16:02:41	8
92236	11	5	c	Engine Demand Percent Torque	16:02:46	96
92237	11	5	c	Engine Demand Percent Torque	16:02:51	97
92238	11	5	c	Engine Demand Percent Torque	16:02:56	96
92239	11	5	c	Engine Demand Percent Torque	16:03:01	97

[Engine Demand Percent Torque\_Training Data]

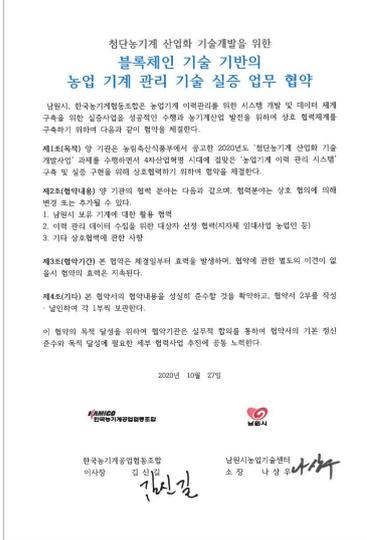
	A	B	C	D	E	F
1	Month	Day	TractorID	Variable	Time	Value
21895	11	5	b	Engine Demand Percent Torque	14:02:57	74
21896	11	5	b	Engine Demand Percent Torque	14:03:02	42
21897	11	5	b	Engine Demand Percent Torque	14:03:07	41
21898	11	5	b	Engine Demand Percent Torque	14:03:12	65
21899	11	5	b	Engine Demand Percent Torque	14:03:17	39
21900	11	5	b	Engine Demand Percent Torque	14:03:21	0
21901	11	5	b	Engine Demand Percent Torque	14:03:27	0
21902	11	5	b	Engine Demand Percent Torque	14:03:32	0
21903	11	5	b	Engine Demand Percent Torque	14:03:35	0

[Engine Demand Percent Torque\_Test Data]

- 공동연구기관(한국농기계공업협동조합) :

㉔ 농기계 이력 데이터 블록체인 시스템 실증 준비

㉔ 실증시험 시군선정



[남원시 실증 지역 협약]

- 농기계 보급률, 실증 사업화 협력 의지를 바탕으로 한 선행 검토
- 농기계 사용실태를 기초로 실증시험 시군을 선정하여 MOU 체결
- 실증 대상 지역 선정 1차 완료 (농협중앙회, 상주시, 남원시 등)

㉔ 실증대상 농기계 선정

- 실증대상 농기계 선정을 위한 농기계 사용실태 파악
- 실태 파악을 통한 확장성과 범용성을 만족하는 대상 모델 선정
- 자주식 트랙터 1차 실증 적용 잠정 확정

기관 / 시군구	모델 선정(자주식 우선)
농협중앙회 20.12.15. 업무 협약 예정	
상주시 20.12.10. 업무 협약 예정	
남원시 20.10.27. 업무 협약 완료	

[실증 지역 협약 상태와 대상 농기계 선정]

- 실증대상 농기계 선정 : 농업용트랙터, 농업용굴삭기
- 실증시험 시군선정 : 상주시(농업기술센터), 남원시(농업기술센터)



[ 실증을 위한 농기계 기술 센터 면담 ]



[ 실증을 위한 단말기 설치 예시 ]



[ 실증 전 시연 운전 ]

- ④ 시군구 대상 실증 사업화를 통한 결과 도출 및 정부 지원과 연계한 정책 제언
  - 실증사업화 대상 시군구 및 농업기계 선정
  - 선정 농업기계와 관련한 정부(지방자치단체 포함)의 지원현황 파악
  - 농업기계 관리기술 실증사업 3건 이상 추진

농협경제지주(주) 업무협약	상주시 농업기술센터 업무협약	남원시 농업기술센터 업무협약
<p style="text-align: center;"><b>원단농기계 산업화 기술개발을 위한 불특체인 기술 기반의 농업 기계 관리 기술 실증 업무 협약</b></p> <p>농협경제지주, 한국농기계협동조합은 농업기계 관리관리를 위한 시스템 개발 및 데이터 체계 구축을 위한 실증사업을 성공적인 수행의 농기계산업 발전을 위하여 상호 협력체계를 구축하기 위하여 다음과 같이 협약을 체결한다.</p> <p>제1조(목적) 양 기관은 농업혁신시용부처에 공모한 2020년도 '원단농기계 산업화 기술 개발사업' 과제를 수행하면서 4차산업혁명 시대에 걸맞은 '농업기계 이력 관리 시스템' 구축 및 실증 구현을 위해 상호협력하기 위하여 협약을 체결한다.</p> <p>제2조(협약내용) 양 기관의 협력 분야는 다음과 같으며, 협력분야는 상호 협력에 의해 변경 또는 추가될 수 있다. 1. 농업 이력 관리에 대한 활용 협력 2. 이력 관리 데이터 수집을 위한 대상자 선정 방법(농협담당자 및 농민 등) 3. 기타 상호협력에 관한 사항</p> <p>제3조(협력기간) 본 협약은 체결일부서 효력을 발생하며, 협정에 관한 별도의 서적이 없을 시 협정이 효력은 지속된다.</p> <p>제4조(기타) 본 협약에서 협약내용을 성실하게 준수할 것을 확인하고, 협정 2주를 작성·발행하여 각 1부씩 보관한다.</p> <p>이 협약의 목적 달성을 위하여 협약기관은 실무적 협력을 통하여 협약서의 기존 명칭 준수와 목적 달성에 필요한 세부 협약사항 추진에 공조 노력한다.</p> <p style="text-align: right;">2020년 12월 15일</p> <p style="text-align: center;">  </p>	<p style="text-align: center;"><b>원단농기계 산업화 기술개발을 위한 불특체인 기술 기반의 농업기계 관리기술 실증업무 협약</b></p> <p>상주시, 한국농기계협동조합은 농업기계 관리관리를 위한 시스템 개발 및 데이터 체계 구축을 위한 실증사업을 성공적인 수행의 농기계산업 발전을 위하여 상호 협력체계를 구축하기 위하여 다음과 같이 협약을 체결한다.</p> <p>제1조(목적) 양 기관은 농업혁신시용부처에 공모한 2020년도 '원단농기계 산업화 기술 개발사업' 과제를 수행하면서 4차산업혁명 시대에 걸맞은 '농업기계 이력 관리 시스템' 구축 및 실증 구현을 위해 상호협력하기 위하여 협약을 체결한다.</p> <p>제2조(협약내용) 양 기관의 협력 분야는 다음과 같으며, 협력분야는 상호 협력에 의해 변경 또는 추가될 수 있다. 1. 농업 이력 관리에 대한 활용 협력 2. 이력 관리 데이터 수집을 위한 대상자 선정 방법(지자체 담당자 및 농민 등) 3. 기타 상호협력에 관한 사항</p> <p>제3조(협력기간) 본 협약은 체결일부서 효력을 발생하며, 협정에 관한 별도의 서적이 없을 시 협정이 효력은 지속된다.</p> <p>제4조(기타) 본 협약에서 협약내용을 성실하게 준수할 것을 확인하고, 협정 2주를 작성·발행하여 각 1부씩 보관한다.</p> <p>이 협약의 목적 달성을 위하여 협약기관은 실무적 협력을 통하여 협약서의 기존 명칭 준수와 목적 달성에 필요한 세부 협약사항 추진에 공조 노력한다.</p> <p style="text-align: right;">2020년 12월 15일</p> <p style="text-align: center;">  </p>	<p style="text-align: center;"><b>원단농기계 산업화 기술개발을 위한 불특체인 기술 기반의 농업 기계 관리 기술 실증 업무 협약</b></p> <p>남원시, 한국농기계협동조합은 농업기계 관리관리를 위한 시스템 개발 및 데이터 체계 구축을 위한 실증사업을 성공적인 수행의 농기계산업 발전을 위하여 상호 협력체계를 구축하기 위하여 다음과 같이 협약을 체결한다.</p> <p>제1조(목적) 양 기관은 농업혁신시용부처에 공모한 2020년도 '원단농기계 산업화 기술 개발사업' 과제를 수행하면서 4차산업혁명 시대에 걸맞은 '농업기계 이력 관리 시스템' 구축 및 실증 구현을 위해 상호협력하기 위하여 협약을 체결한다.</p> <p>제2조(협약내용) 양 기관의 협력 분야는 다음과 같으며, 협력분야는 상호 협력에 의해 변경 또는 추가될 수 있다. 1. 농업 이력 관리에 대한 활용 협력 2. 이력 관리 데이터 수집을 위한 대상자 선정 방법(지자체 담당자 및 농민 등) 3. 기타 상호협력에 관한 사항</p> <p>제3조(협력기간) 본 협약은 체결일부서 효력을 발생하며, 협정에 관한 별도의 서적이 없을 시 협정이 효력은 지속된다.</p> <p>제4조(기타) 본 협약에서 협약내용을 성실하게 준수할 것을 확인하고, 협정 2주를 작성·발행하여 각 1부씩 보관한다.</p> <p>이 협약의 목적 달성을 위하여 협약기관은 실무적 협력을 통하여 협약서의 기존 명칭 준수와 목적 달성에 필요한 세부 협약사항 추진에 공조 노력한다.</p> <p style="text-align: right;">2020년 12월 15일</p> <p style="text-align: center;">  </p>

- 농기계 이력데이터 활용을 위한 정책건의안 도출

: 텔레매틱스 장비를 통해 취합된 농기계 운행데이터를 활용하여 농작업 면적을 산출하고, 수기로 작성하는 농작업 일지를 온라인상으로 대체할 수 있을 것임

농작업 대행 실적 등을 투명하게 관리하여 농협중앙회에서 지역농협에 지원하는 운영자금의 효과를 극대화 할 수 있음

면세유 관리 측면에서도 농작업 이력관리를 통해 투명하게 면세유 사용 이력 등을 관리할 수 있을 것임

㉔ 선정 시군구 시스템 실증 시험 결과 보고서 작성 및 정책 건의, 세미나 개최 및 정책 건의

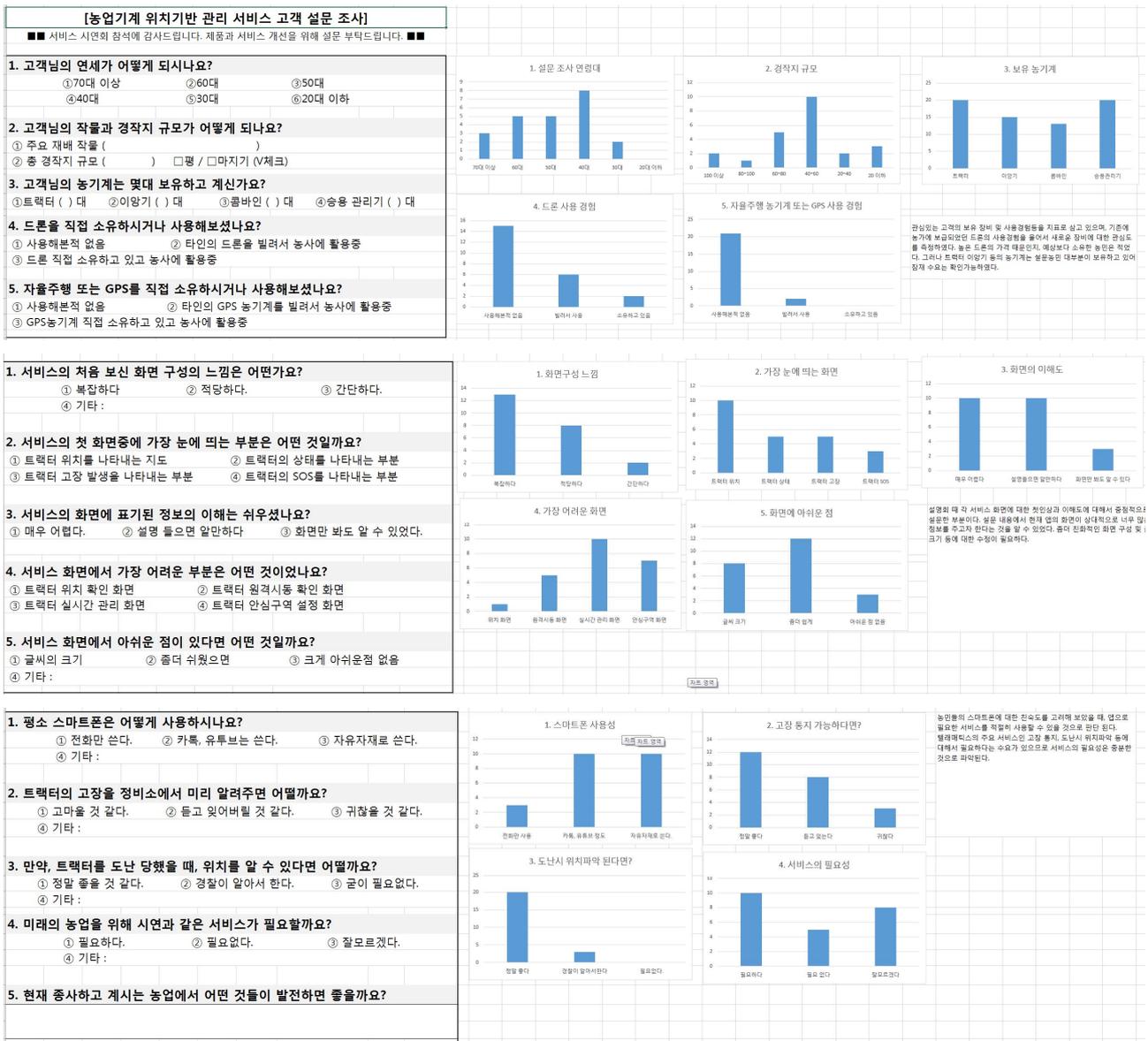
㉔ 실증 사업화 결과 보고서 작성

- 실증사업화 및 시연회 결과 취합

: 상주시, 남원시 농업기술센터 내 실증사업 결과 분석

- 컨소시엄 및 시군구 내용 검토

: 실증사업 결과 분석에 따른 향후 보완사항 등 모색



[시군구 사용자 설문조사 바탕 보고서 작성 발취]

㉔ 농업기계 관리기술 정책제언을 위한 외부 의견 수렴(세미나 개최 등)

- 목적 : 농업기계 관리기술의 정책적 활용을 위한 방안 검토 및 정책제안

- 주요내용 :

- 블록체인 단말기를 통한 데이터의 종류와 활용방법 등에 대한 의견수렴 및 정책제언 방향설정 (농기계공급업체, 농기계임대사업소, 농업인, 기타)
- 단말기 공급업체, 단말기 부착 농기계 사용자 등 예상되는 문제점 및 개선방향 (단말기 소프트웨어의 자동업데이트 기능, 단말기 부착 농기계 연료소모량 등 측정기능, 현재 위치 기능 및 도난방지기능, 긴급구조요청 기능, 농기계정책 보험 연계 등)
- 블록체인 관리기술의 농업기계 기종 검토 및 정책제언 도출  
(1단계) '23.7 개정 농업기계화촉진법에 따른 적용대상 농기계 : 농업기계 출고부터 폐차까지의 생애이력 주기 관리(트랙터·콤바인·이앙기)  
☞ 향후 적용대상 기종검토 : 스마트농업 빅데이터 축적을 위한 전동자주형 농업기계(드론, 스피드스프레이어, 농업용굴착기 및 농업용로우더 등)

전문가 자문회의 4회



①08.18(금), 24명(농기업체, 유관기관 전문가 등)



②08.30(화), 26명(농기업체, 유관기관 전문가 등)



③10.21(금), 9명(농기업체, 유관기관 전문가 등)



④11.24(목), 17명(농기업체, 유관기관 전문가 등)

평가보고 세미나 2회



①10.06(목), 137명(업체 117사 120명, 기관 9, 조합 6, 기타2)



②11.29(화), 76(업체 57사 66명, 기관 5, 조합 6, 기타2)

- ◎ 농기계 이력 관리 시스템 실증 사업화 결과, 외부의견 수렴을 바탕으로 한 정책 건의
  - 실증 사업 자료를 기반으로 한 정책 제안
    - : 면세유 지원사업 관련 사용내역 투명성 확보
    - 농기계 이력관리를 통한 체계적인 농기계 보험요율 산정 및 관리
    - 농업기계화촉진법에서 정한 농기계 사후검정 등으로의 이력관리 연계 적용
    - 지능형 농기계(자율주행, 로봇 등)의 활용범위 확대 제고
    - 농기계에 부착된 단말기(센서)를 통해 장비 운용상태 및 고장유무를 실시간으로 확인하여 효율적인 영농작업 추진
    - 장비별 운용현황 분석을 통한 효율적인 영농방법 제안으로 농가 소득 증대 기여



'60년의 역사, 100년의 미래'  
**KAMICO**  
한국농기계공업협동조합

수신 농림축산식품부장관  
경유 농기자재정책팀장

**제 목 「블록체인 기술기반의 농업기계 관리기술」에 대한 정책활용 건의**

- 귀 부의 무궁한 발전을 기원합니다.
  - 우리조합은 농림축산식품 연구개발사업 과제(과제명 : 블록체인 기술기반의 농업기계 관리기술 개발사업)를 통해 개발된 기술을 정부에서 추진중인 농업 기계화사업의 효율적 정책개발 및 활용을 위하여 불임과 같이 건의드립니다.
- 붙임 : 정책활용 제안서 1부, 끝.

**한국농기계공업협동조합 이사장 김신길**

사람	사람	대리(기안자)	부장	팀장	전문이사	전권
조세영(125)	김형민(124)	김대은(123)	안상희(122)	장길수		2022.12.21 이기중
협조자						
시행 정책지원팀-651(2022.12.21.)						
우 31041 충청남도 천안시 서북구 쌍거울 정자1길180 / www.kamico.or.kr						
전화번호 041-411-2122-5 팩스번호 041-555-4491 Email : k2@kamico.or.kr						

[붙임]

**블록체인 기술기반의 농업기계 관리기술 정책활용 제안**

- 현황 및 문제점
  - 농업기계는 자동차나 건설기계와 달리 등록하는 절차가 없어 관련 장비에 대한 주요 이력(제조 - 출고 - 수리 - 폐기 등) 관리가 어려움
  - 지방자치단체나 농업 등의 농기계임대사업 경우 농업기계의 관리를 해당 구입처별로 별도 관리를 하고 있어 통합관리 어려움
  - 지방자치단체 농기계임대시 타 행정구역으로의 이동이나 실제 작업 시간, 연료소모량 등을 파악하기 어려움
- 필요성
  - 농업기계화촉진법(2023.7월 시행예정) 개정내용에 따르면 농업기계 제조업자 및 수입업자(판매위탁자 포함)는 차대각인 농업기계(트랙터, 콤팩트, 이앙기)는 농식품부가 구축하는 전산정보처리시스템을 통해 농림축산식품부장관에게 신고하여야 함
  - 농업기계의 실제 동작시 현재 위치, 작업시간 등 실시간 파악하여 임대사업소 등에서는 농기계의 효율적 관리를 위해 필요
- 주요 개발 내용
  - 농업기계의 생애주기 이력(출고-중고-폐기) 관리를 위한 정부 전산정보처리시스템의 효율적 구축과 관리기능 지원
  - 농업기계의 현재 장비의 위치, 실제 작업시간 등을 실시간 파악 가능
- 기대효과
  - 농업기계의 관리를 위한 농업기계화촉진법, 지자체 및 농협 등의 농기계임대사업 등 관련장비의 효율적 관리 지원
  - 농업기계 개인 소유주에 있어서도 관련 농기계의 현지 위치정보를 파악할 수 있어 장비 도난 방지 예방에도 기여
- 정책활용건의
  - 관련 블록체인 기술적용 단말기 부착을 통한 농업기계 생애주기 및 전산정보처리시스템 등의 관리기술 활용
  - 농기계 정비 이력관리(수리 및 정비 등) 관리를 통해 농기계정책보험 등에 가입시 농업인 복지 등을 확대

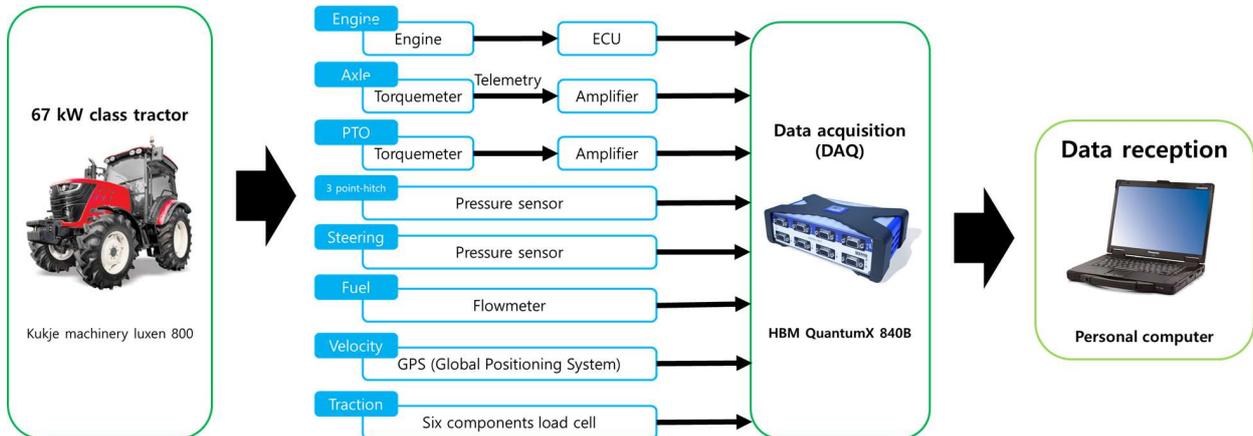
[정책제안 일부 발췌]

- 위탁기관1 (충남대학교 산학협력단) :

가) 트랙터 주요 동력 소모원 선정 및 부하계측 시스템 설계·개발

① 트랙터 주요 동력 소모원 선정

- 부하 계측 시스템은 아래 그림과 같이 엔진, 차축, PTO(Power Take Off), 3점히치, 조향장치 등에 센서를 부착하여 데이터 측정이 가능하도록 구성하였음.



[ 부하 계측 시스템 구성 ]

- 엔진은 CAN (Controller Area Network) 통신을 이용하여 토크, 회전속도를 측정하여 작업 간 소요되는 전체 동력을 측정할 수 있게 구성하였음.

② 트랙터 계측 시스템 개발

- 농작업간 차축에서의 토크를 측정하기 위해서 차축에 토크미터를 설치하였으며, 무선 텔레메트리 통신을 이용하여 앰플리파이어로 데이터가 전송됨. 이후 앰플리파이어에서 신호가 증폭되어 계측기로 데이터를 수집할 수 있게 구성되어 있음. 토크미터는 전륜은 최대 15 kNm까지 측정이 가능한 MW 15 kNm(Manner, Germany), 후륜은 최대 30 kNm까지 측정이 가능한 MW 30 kNm (Manner, Germany)로 선정하였음. 설계상 제원을 고려하여 후륜에 측정 범위가 큰 토크

미터를 선정하였음.

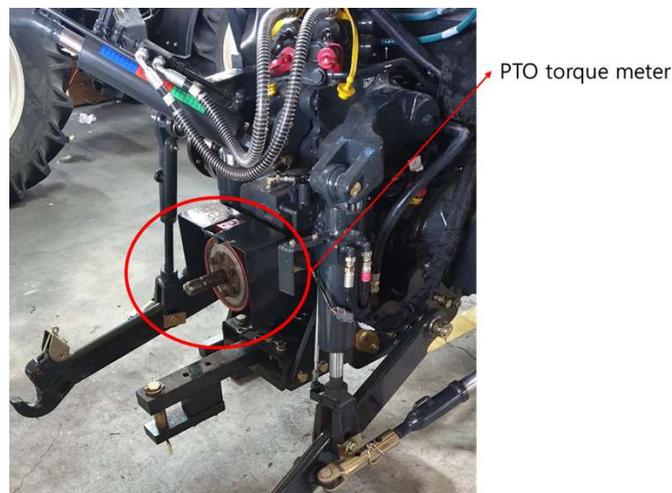


[ 트랙터에 설치된 차축 토크미터 ]

표 65. 차축 토크미터 제원

Term	Front axle	Rear axle
Nominal load	15 kNm	30 kNm
Maximum load	400%	400%
Sensor	Strain gauge (strain gauge resistor $\geq 350 \Omega$ )	Strain gauge (strain gauge resistor $\geq 350 \Omega$ )
Bridge supply voltage	10.0 V	10.0 V
Sample rate	4 kS/s	4 kS/s
Bandwidth	0 to 1 kHz	0 to 1 kHz
Temperature	-25 to +125°C	-25 to +125°C

- 로타리 작업간 PTO의 소요동력을 측정하기 위해서 PTO 토크미터를 설치하였음. PTO는 앰플리 파이어와 유선으로 연결되어 회전속도 및 토크를 측정할 수 있게 구성되었음. PTO 토크미터는 최대 5 kNm 까지 측정이 가능한 MW B 5 kNm(Manner, Germany)로 선정하였음.

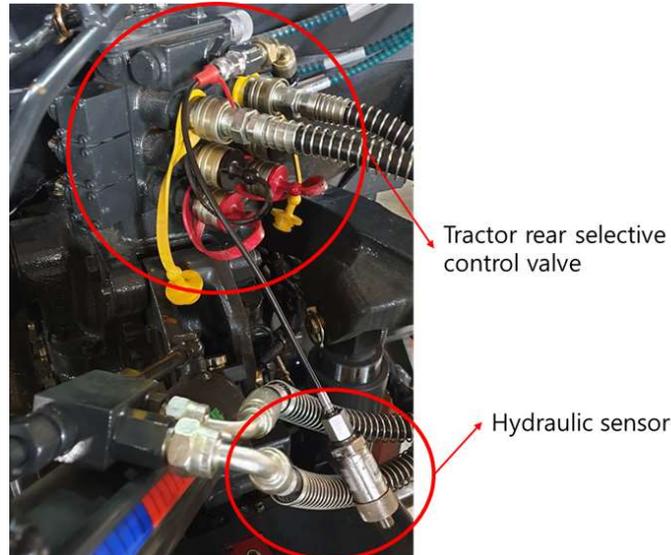


[ 트랙터에 설치된 PTO 토크미터 ]

표 66. PTO 토크미터 제원

Term	Value
Nominal load	5 kNm
Maximum load	400%
Pulse rate RPM detection	90 pulses
Sensor	Strain gauge (strain gauge resistor $\geq 350 \Omega$ )
Sample rate	6.62 kHz
Resolution	16 Bit with 16 Bit CRC
Maximum speed	2,000 rpm
Temperature	-10 to +85°C

- 트랙터의 유압장치는 크게 후방의 3점하치와 조향장치로 구분됨. 3점하치의 부하를 측정하기 위해서 차량 후방에 위치한 주유압포트에 유압 센서를 설치하였으며, 조향 장치의 부하를 측정하기 위해서 차량 엔진룸안에 유압센서를 설치하였음. 유압센서는 HDA 4748-H-0250 (HYDAC, Germany)으로 선정하였으며, 트랙터 릴리프 밸브의 제원을 고려하여 최대 250 bar 까지 측정가능한 센서를 선정하였음.



[ 트랙터 후방에 설치된 유압 센서 ]

표 67. 유압센서 제원

Term	Value
Mechanical Connection Type	G1/4 A ISO 1179-2, Male
Input Overload Pressure	500 bar
Input Measuring Pressure Range	0 to 250 bar
Output Typical Accuracy According to DIN 16086, Terminal Based (Full Scale)	$\leq \pm 0.25 \%$

- GPS는 작업 간 차량 속도를 측정하기 위해서 설치되었으며, RTK-GPS 성능을 보이고 있는 Ellipse2-D (SBG systems, France)로 선정하였음.



[ Ellipse2-D GPS ]

표 68. GPS 센서 제원

Term	Value
Weight	180g
Input voltage	9 - 36V
Velocity accuracy	0.03 m/s RMS
Signal tracking	GPS: L1, L2, L2C GLONASS:L1,L2 BeidouB1,B2, SBAS.QZSS

- 각 센서들은 DAQ로 유선으로 연결되게 구성되어 있으며, DAQ는 LAN 통신을 통해 PC로 데이터를 전송함. 데이터들은 계측기 전용 프로그램인 CatmanEasyAP (3.3, HBM, Germany)를 이용하여 PC에 저장되었음.



[ 계측용 트랙터 시스템 구성 ]

나) 필드 실험을 통한 주요 농작업 부하 데이터 수집 및 분석

① 부하 계측 수행을 위한 트랙터 및 작업기 선정

- 부하 계측을 수행하기 위해서 트랙터 및 작업기 선정을 수행하였음. 트랙터는 67 kW (@2,200 rpm)트랙터인 LUXEN 800으로 선정하였으며 중량 및 크기는 각각 4,000 kg 그리고, 4,020 × 2,270 × 2,790 mm (length × width × height) 임.



[ 국제기계 LUXEN800 ]

표 69. LUXEN 800의 세부 제원

Item		Specification
Length × Width × Height (mm)		4,020 × 2,270 × 2,790
Weight (kg)		4,000
Engine power (kW)		67
Transmission	주변속	기계식(동기물림식)
		4단(1,2,3,4)
	부변속	기계식(상시물림식)
		4단(초저속,A,B,C)

- 이랑쟁기 및 로타리 작업 수행을 위해 작업기 선정을 수행하였음. 이랑쟁기(WJSP-8, WOONGJIN MACHINERY Co., LTD., Korea)의 중량 및 크기는 각각 495 kg 그리고, 2,180 × 2,800 × 1,285 mm (length × width × height) 임. 로터베이터(N225, CELLI, Italy)의 중량 및 크기는 각각 718 kg 그리고 750 × 2,420 × 1,060 mm mm (length × width × height) 임.



[ 계측 시험에 사용된 이랑쟁기(좌) 및 로터베이터(우) ]

표 70. 이랑쟁기 작업기의 세부제원

Item	Plow
Company	Woongin Co. Ltd.
Model	WJSP-8
Length × Width × Height (mm)	2,180 × 2,800 × 1,285
Weight (kg)	495

표 71. 로타베이터 작업기의 세부제원

Item	Rotavator
Company	CELLI
Model	N225
Length × Width × Height (mm)	750 × 2,420 × 1,060
Weight (kg)	718

② 작업단수 및 계측시험 필드지

- 트랙터 작업 단수는 사용자의 설문 및 문헌연구를 참고하여 선정하였음. 쟁기 작업시에는 B2 (6.0 km/h) 및 B3 (7.6 km/h)로단수를 사용하였으며, 로타리 작업시에는 A3 (2.7 km/h) 및 A4 (3.7 km/h) 단수를 사용하였음.



[ 계측 시스템을 이용한 쟁기 및 로타리 작업 ]

- 필드 시험은 조건별로 2회 이상 반복 수행하였으며, 사용자 선호도가 가장 높은 C-type의 작업 패턴을 선정하였음.

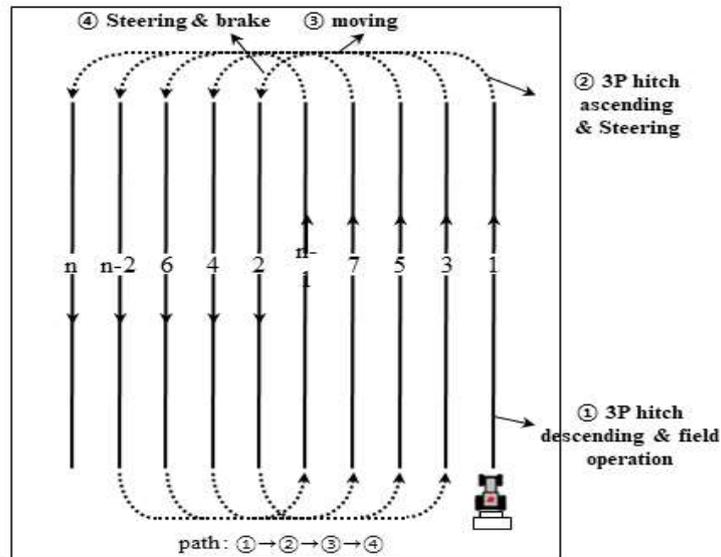


그림 310. 시 사용된 C-type의 작업 패턴

- 계측시험은 충남 당진(36°55'49.8" N, 126°37' 57.3"E)에 위치한 필드에서 수행하였다. 필드시험지는 약 1,500평 정도의 규모로 측정되었음.

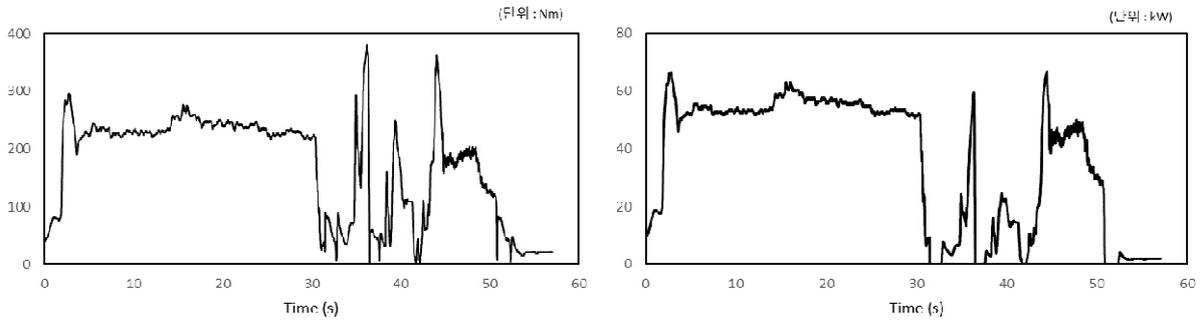


[ 쟁기 및 로타리 작업을 수행한 필드 시험지 ]

③ B2 단수에서의 부하 계측 시험 결과 - 1

- 엔진 토크 및 출력

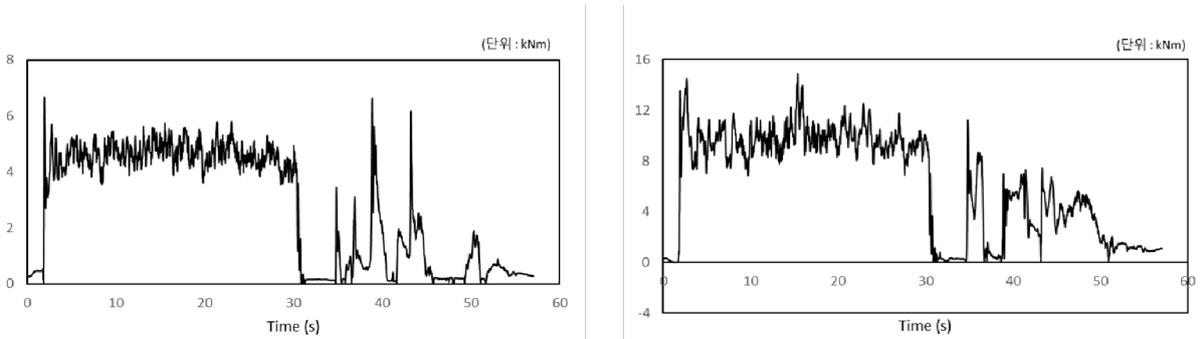
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 5 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 190~276 Nm, 약 45~62 kW의 범위를 보였음. 이후 약 30sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ B2 조건에서의 쟁기 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

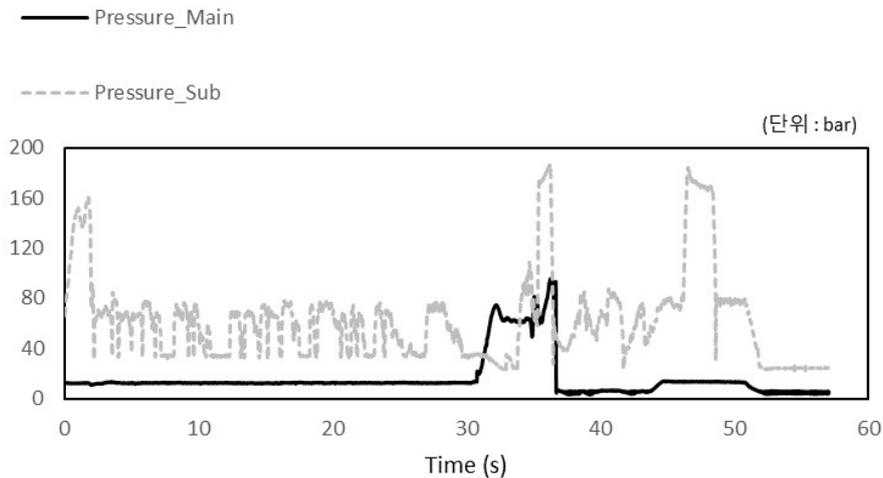
트랙터 차축 토크 및 출력은 아래 그림과 같이 쟁기 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 3~5 kNm 범위를 보이고 있으며, 후륜 토크는 약 6~14 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2~3배 더 높게 나타나고 있음.



[ B2 조건에서의 쟁기 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

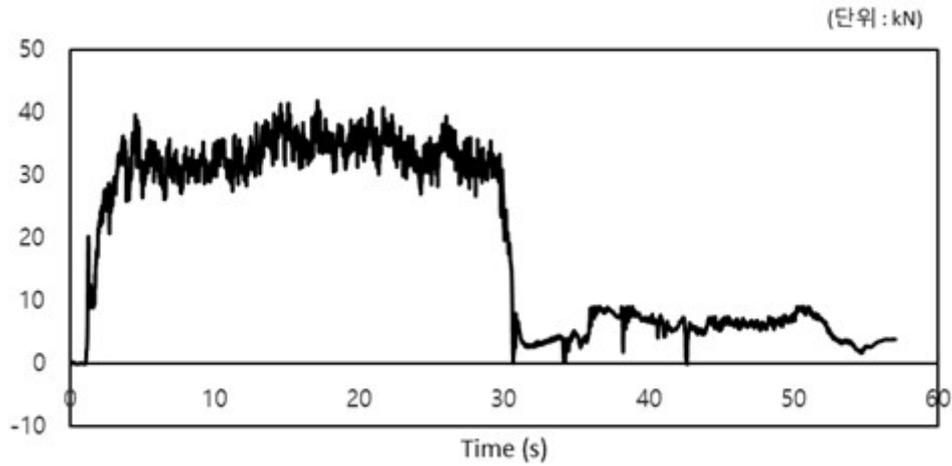
메인 펌프는 쟁기 작업 시 쟁기의 수평유지를 위해 유압회로를 작동시켜 대부분 12bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났음. 보조 펌프는 쟁기 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 32~84 bar 범위로 나타났음.



[ B2 조건에서의 쟁기 작업 시 메인 및 보조 유압펌프의 압력 ]

- 견인력

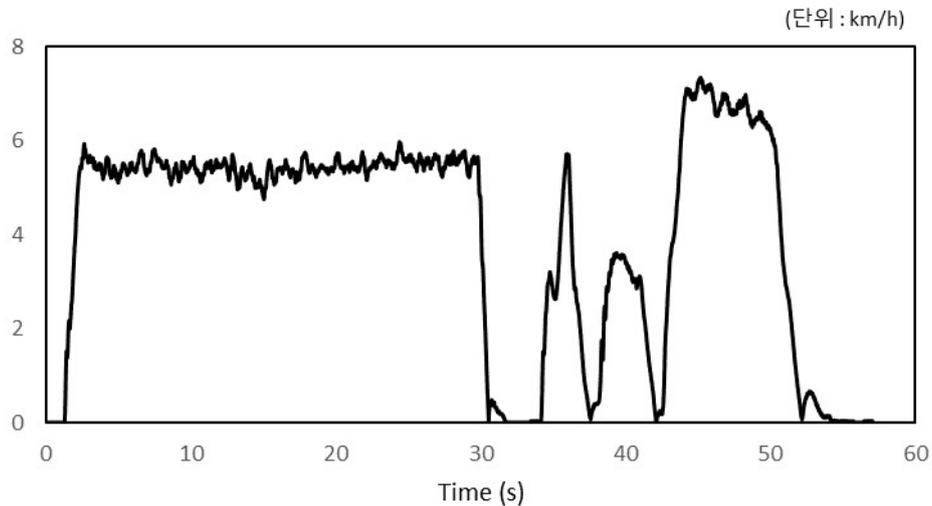
이랑쟁기 작업간 견인력의 최소, 평균, 최대, 표준편차는 각각 20, 33, 41, 2.8 kN 으로 나타났음. 아래 그림과 같이 작업 시작함에 따라 5sec에서 견인력이 증가하였다가, 작업이 종료됨에 따라 30sec에서 견인력이 감소하는 것을 확인할 수 있음.



[ B2 조건에서의 쟁기 작업 시 견인력 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 5km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 4.5, 5.4, 5.9, 0.2 km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ B2 조건에서의 쟁기 작업 시 작업 속도 ]

- B2 조건에서 쟁기 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

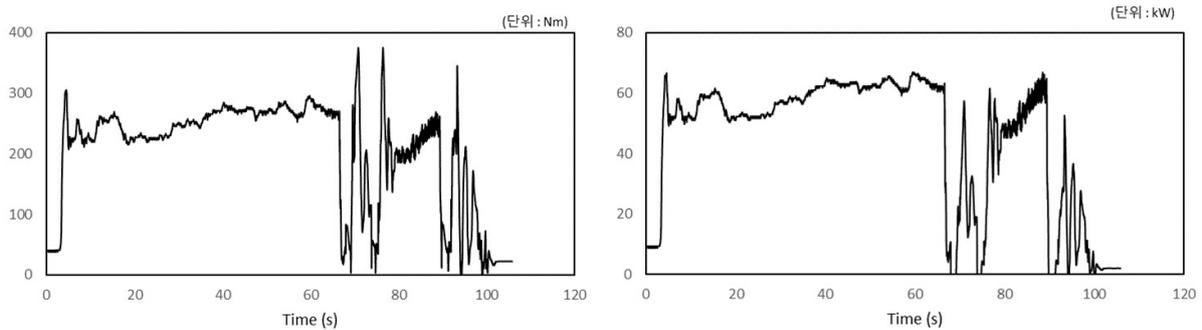
표 72. B2 조건에서 쟁기 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	190	235	276	12
	Rotational speed (rpm)	2,167	2,219	2,300	18
	Power (kW)	45	54	62	2.6
Torque_front (Nm)		3	4	5	0.4
Torque_rear (Nm)		6	9	14	1
Main pressure (bar)		11	12	13	0.2
Sub pressure (bar)		32	55	84	15
Traction (kN)		29	42	50	2
Velocity (km/h)		4.5	5.4	5.9	0.2

④ B2 단수에서의 부하 계측 시험 결과 - 2

- 엔진 토크 및 출력

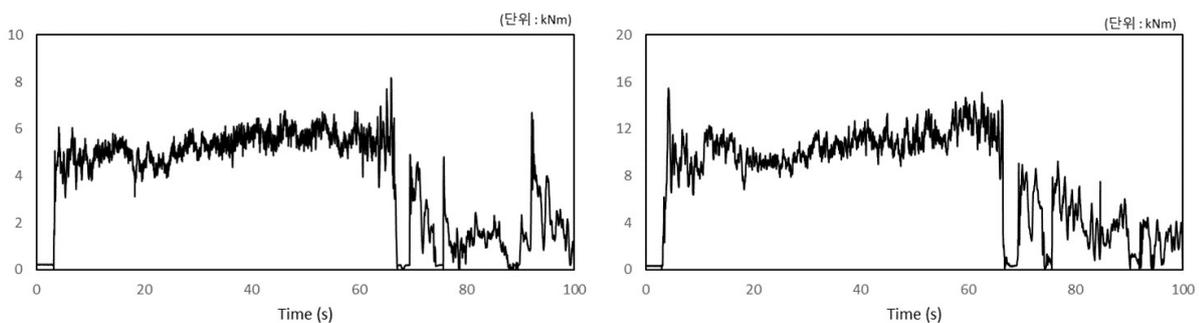
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 5 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 207~296 Nm, 약 49~66 kW의 범위를 보였음. 이후 약 65sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ B2 조건에서의 쟁기 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

트랙터 차축 토크 및 출력은 아래 그림과 같이 쟁기 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 3~7 kNm 범위를 보이고 있으며, 후륜 토크는 약 6~15 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2배 더 높게 나타나고 있음.

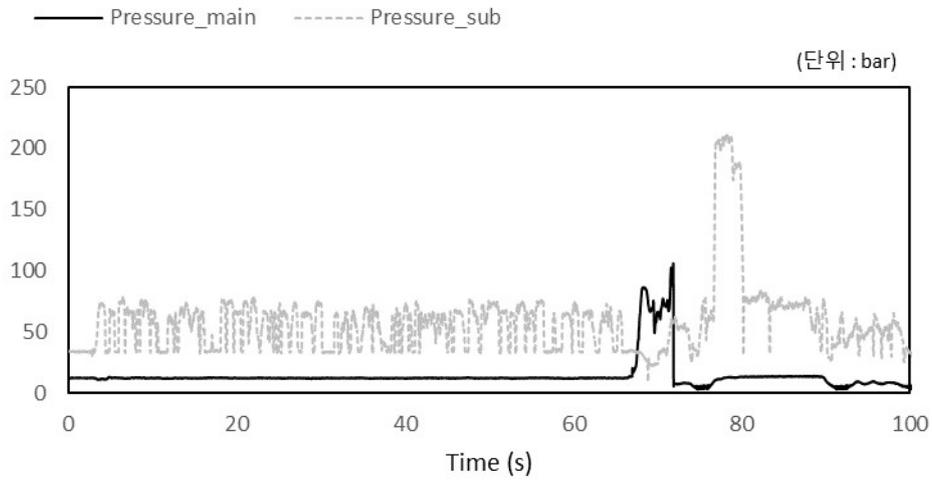


[ B2. 조건에서의 쟁기 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

메인 펌프는 쟁기 작업 시 쟁기의 수평유지를 위해 유압회로를 작동시켜 대부분 12bar 수준을 유

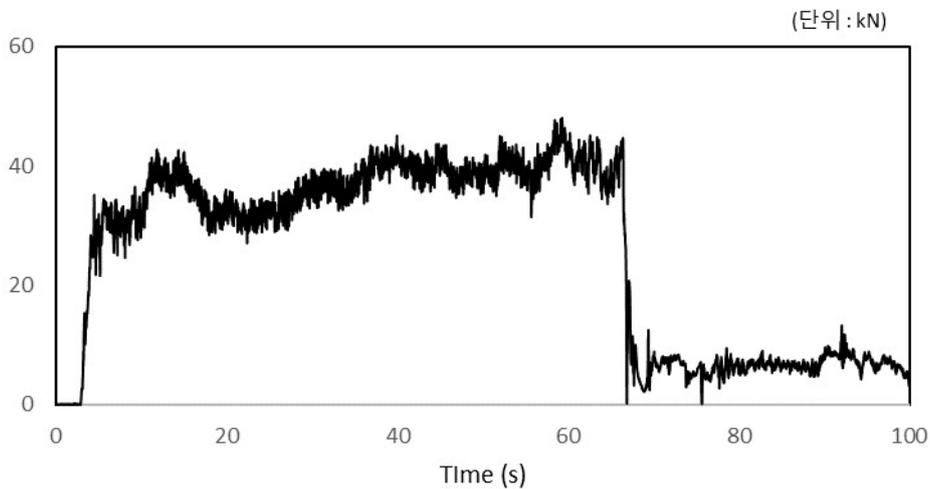
지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났음. 보조 펌프는 쟁기 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 32~78 bar 범위로 나타났음.



[ B2. 조건에서의 쟁기 작업 시 메인 및 보조 유압펌프의 압력]

- 견인력

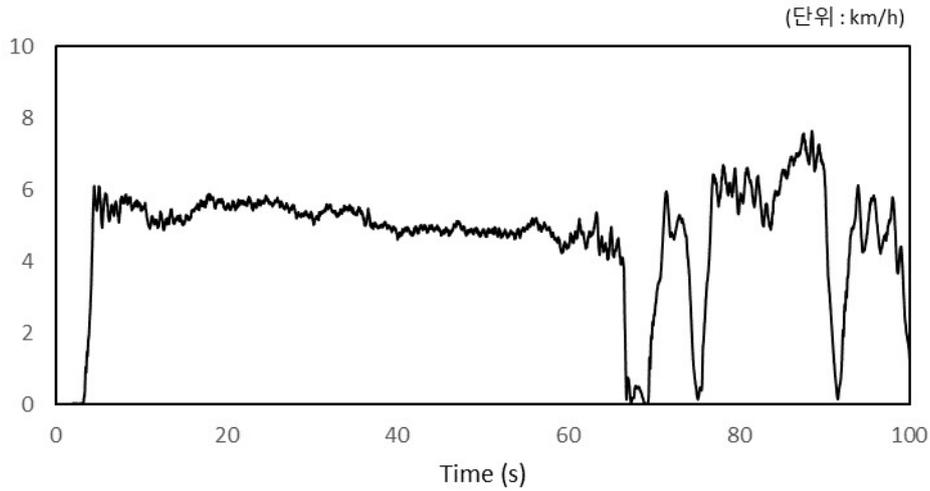
이랑쟁기 작업간 견인력의 최소, 평균, 최대, 표준편차는 각각 21, 36, 48, 4 kN 으로 나타났음. 아래 그림과 같이 작업 시작함에 따라 5sec에서 견인력이 증가하였다가, 작업이 종료됨에 따라 65sec에서 견인력이 감소하는 것을 확인할 수 있음.



[ B2 조건에서의 쟁기 작업 시 견인력 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 5km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 4, 5, 6, 0.3km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ B2 조건에서의 쟁기 작업 시 작업 속도 ]

- B2 조건에서 쟁기 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

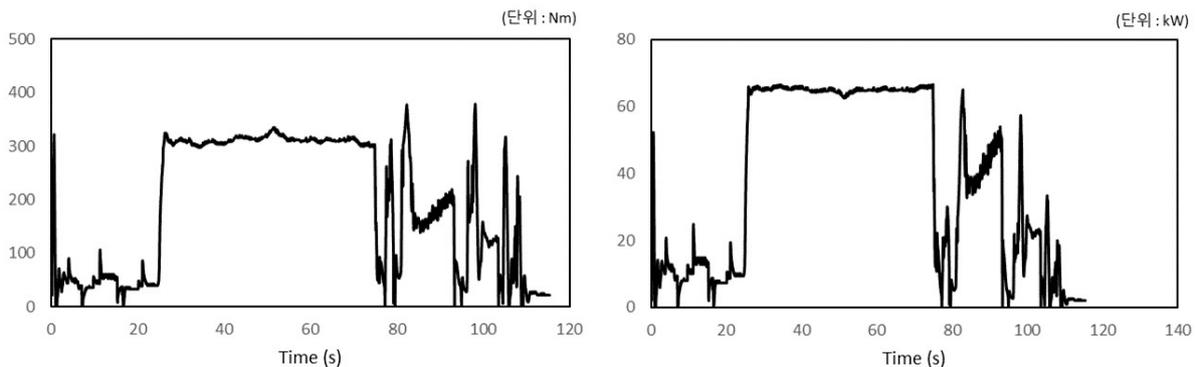
표 73. B2 조건에서 쟁기 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	207	254	296	20
	Rotational speed (rpm)	2,153	2,204	2,269	17
	Power (kW)	49	58	66	4.6
Torque_front (Nm)		3	5	7	0.5
Torque_rear (Nm)		6	10	15	1.4
Main pressure (bar)		11	12	13	0.2
Sub pressure (bar)		32	55	78	14
Traction (kN)		21	37	48	4
Velocity (km/h)		4	5	6	0.3

⑤ B3 단수에서의 부하 계측 시험 결과 - 1

- 엔진 토크 및 출력

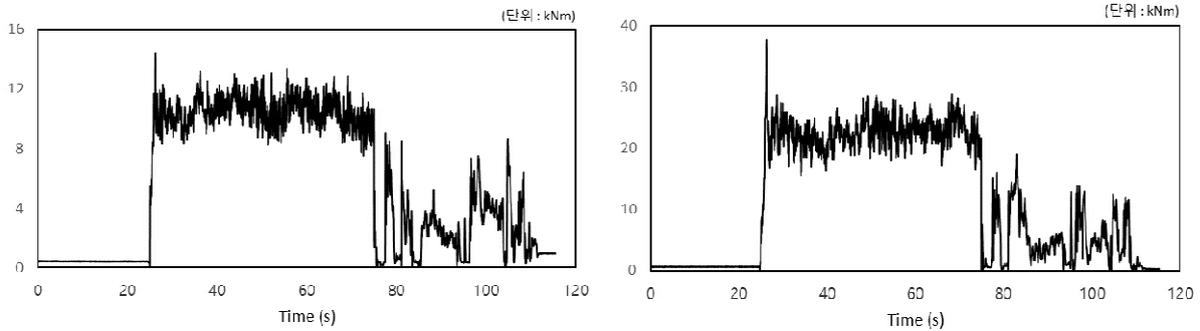
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 25 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 70~335 Nm, 약 16~66 kW의 범위를 보였음. 이후 약 75sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ B3 조건에서의 쟁기 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

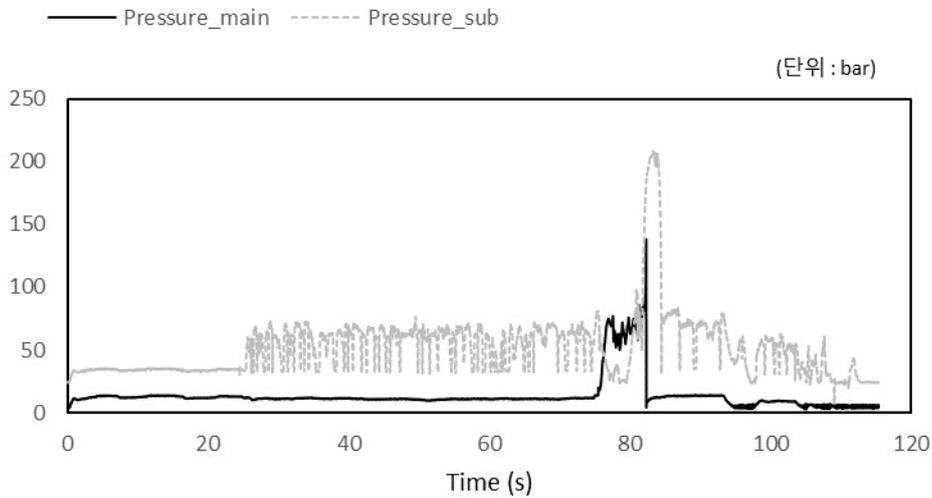
트랙터 차축 토크 및 출력은 아래 그림과 같이 쟁기 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 9~12 kNm 범위를 보이고 있으며, 후륜 토크는 약 18~25 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2~3배 더 높게 나타나고 있음.



[ B3 조건에서의 쟁기 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

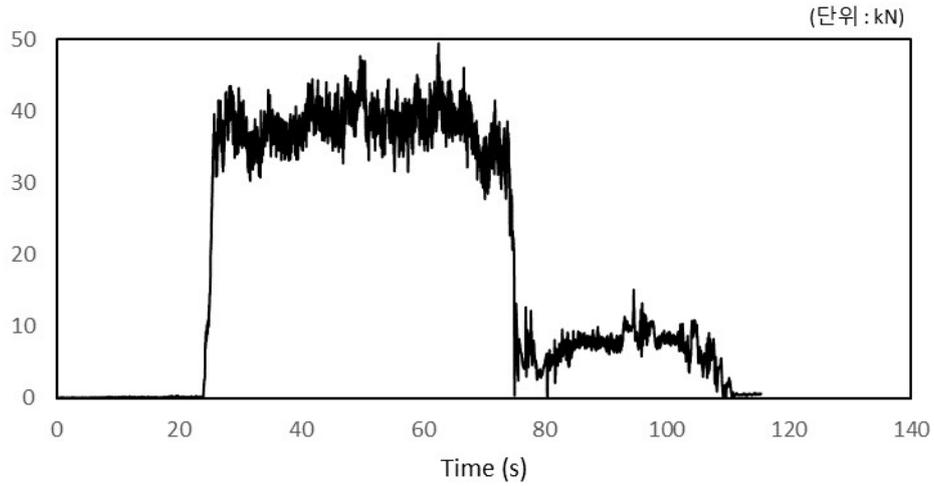
메인 펌프는 쟁기 작업 시 쟁기의 수평유지를 위해 유압회로를 작동시켜 대부분 11bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났음. 보조 펌프는 쟁기 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 30~76 bar 범위로 나타났음.



[ B3 조건에서의 쟁기 작업 시 메인 및 보조 유압펌프의 압력 ]

- 견인력

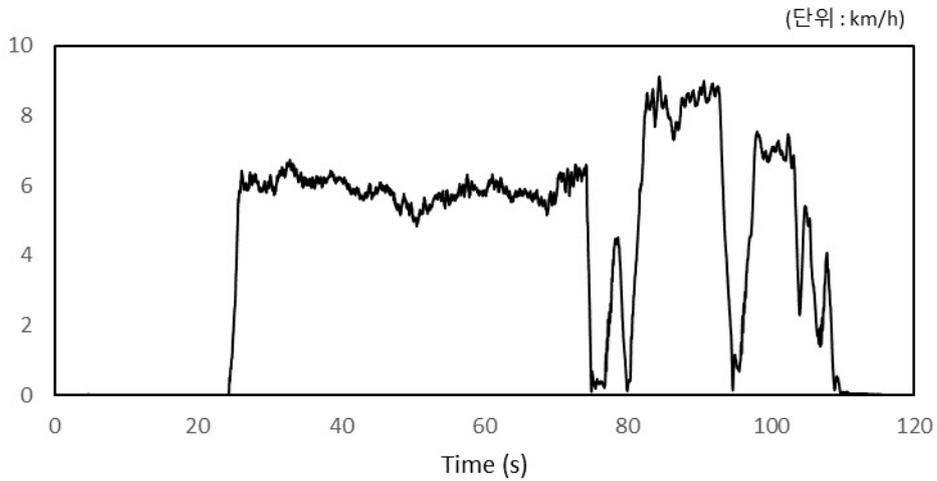
이랑쟁기 작업간 견인력의 최소, 평균, 최대, 표준편차는 각각 1.4, 37, 49, 4 kN 으로 나타났음. 아래 그림과 같이 작업 시작함에 따라 25sec에서 견인력이 증가하였다가, 작업이 종료됨에 따라 75sec에서 견인력이 감소하는 것을 확인할 수 있음.



[ B3 조건에서의 쟁기 작업 시 견인력 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 5.8km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 0.1, 5.8, 6.7, 0.6km/h로 나타났으며, 이는 작업단수를 고려하여 적정한 수준으로 나타났음.



[ B3 조건에서의 쟁기 작업 시 작업 속도 ]

- B3 조건에서 쟁기 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

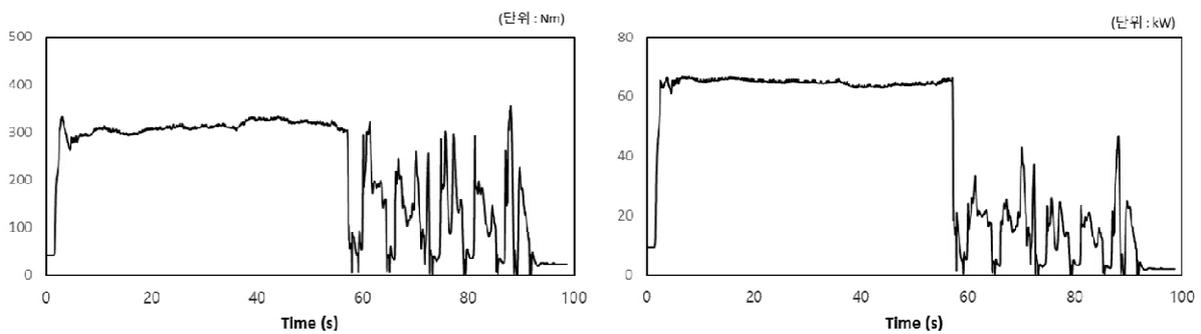
표 74. B3 조건에서 쟁기 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	70	308	335	17
	Rotational speed (rpm)	1,794	1,999	2,270	61
	Power (kW)	16	64	66	3
Torque_front (Nm)		0.6	5.2	7.2	0.5
Torque_rear (Nm)		0.07	11	18	1.4
Main pressure (bar)		9	11	13	0.4
Sub pressure (bar)		30	57	76	12
Traction (kN)		1.4	37	49	3.9
Velocity (km/h)		0.1	5.8	6.7	0.6

⑥ B3 단수에서의 부하 계측 시험 결과 - 2

- 엔진 토크 및 출력

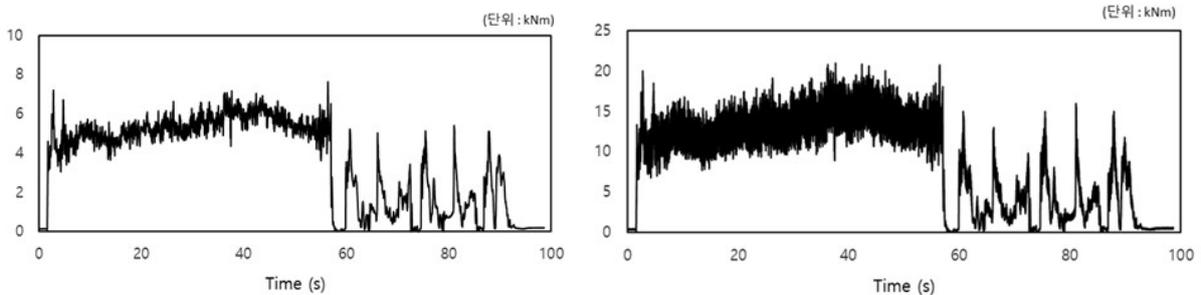
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 5 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 261~334 Nm, 약 61~66 kW의 범위를 보였음. 이후 약 55sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ B3 조건에서의 쟁기 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

트랙터 차축 토크 및 출력은 아래 그림과 같이 쟁기 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 3~7 kNm 범위를 보이고 있으며, 후륜 토크는 약 7~18 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2~3배 더 높게 나타나고 있음.

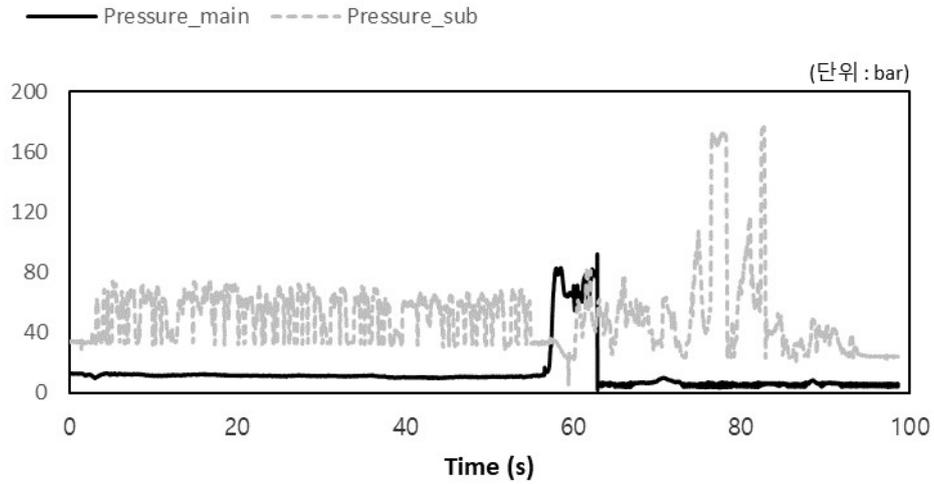


[ B3 조건에서의 쟁기 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

메인 펌프는 쟁기 작업 시 쟁기의 수평유지를 위해 유압회로를 작동시켜 대부분 11bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났

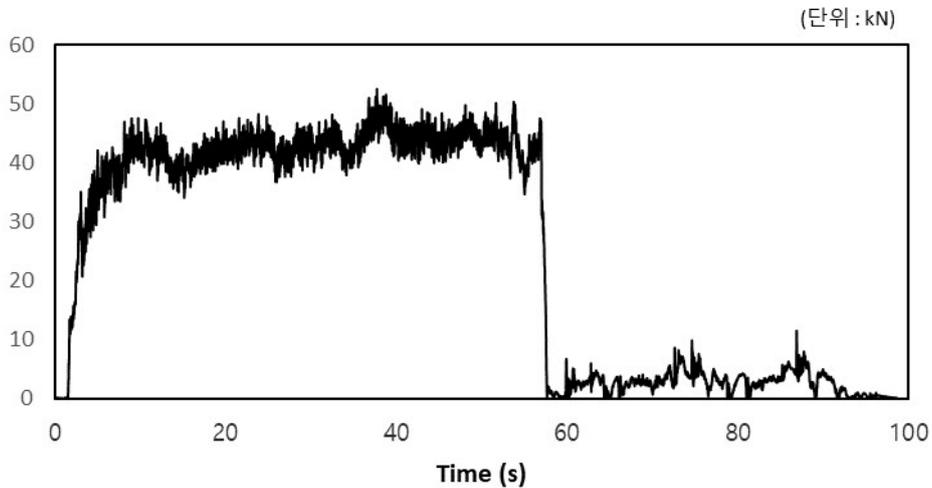
음. 보조 펌프는 쟁기 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 28~73 bar 범위로 나타났음.



[ B3 조건에서의 쟁기 작업 시 메인 및 보조 유압펌프의 압력 ]

- 견인력

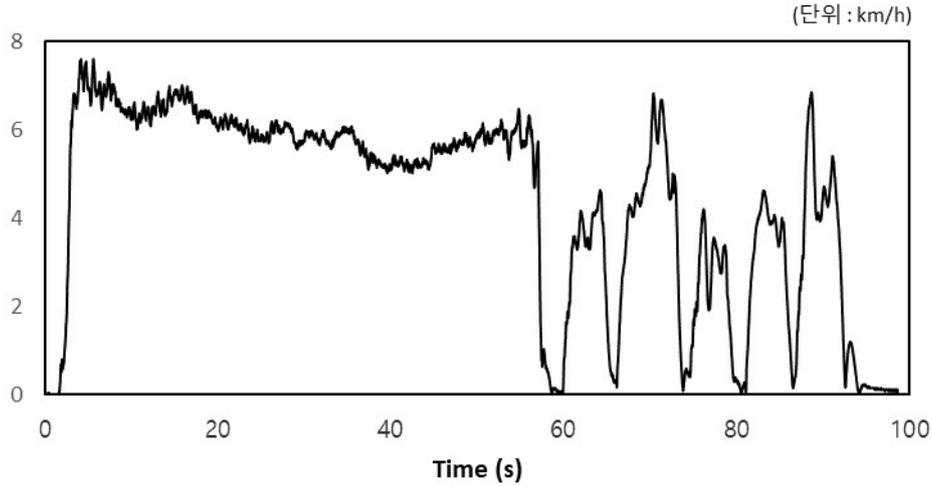
이랑쟁기 작업간 견인력의 최소, 평균, 최대, 표준편차는 각각 20, 42, 52, 3.7 kN 으로 나타났음. 아래 그림과 같이 작업 시작함에 따라 1.5sec에서 견인력이 증가하였다가, 작업이 종료됨에 따라 55sec에서 견인력이 감소하는 것을 확인할 수 있음.



[ B3 조건에서의 쟁기 작업 시 견인력 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 6km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 4.3, 6.0, 7.6, 0.5km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ B3 조건에서의 쟁기 작업 시 작업 속도 ]

- B3 조건에서 쟁기 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

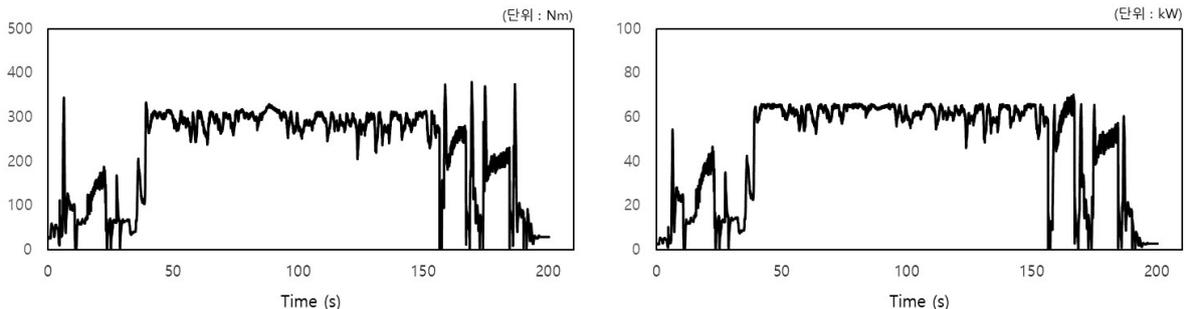
표 75. B3 조건에서 쟁기 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	261	310	334	11
	Rotational speed (rpm)	1,810	1,993	2,230	97
	Power (kW)	61	64	66	0.9
Torque_front (Nm)		3	5	7	0.6
Torque_rear (Nm)		66	66	66	0
Main pressure (bar)		9	11	13	0.7
Sub pressure (bar)		28	53	73	12
Traction (kN)		20	42	52	3.7
Velocity (km/h)		4.3	6	7	0.5

⑦ A3 단수에서의 부하 계측 시험 결과 - 1

- 엔진 토크 및 출력

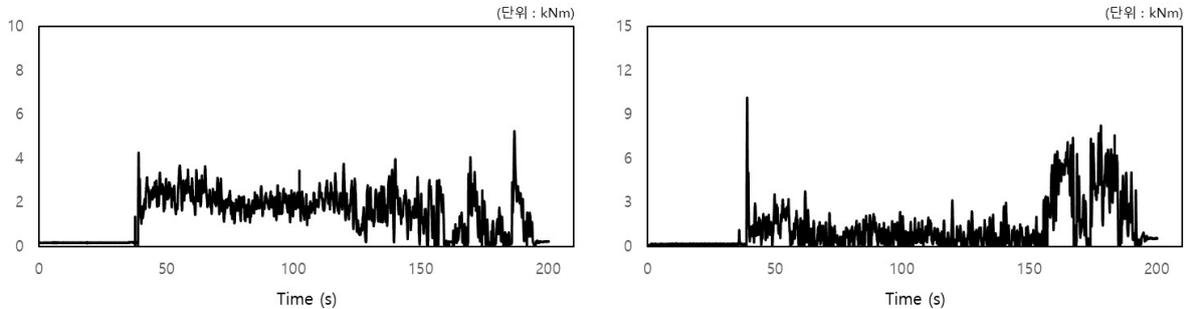
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 39 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 100~330 Nm, 약 22~66 kW의 범위를 보였음. 이후 약 154sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ A3 조건에서의 로터리 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

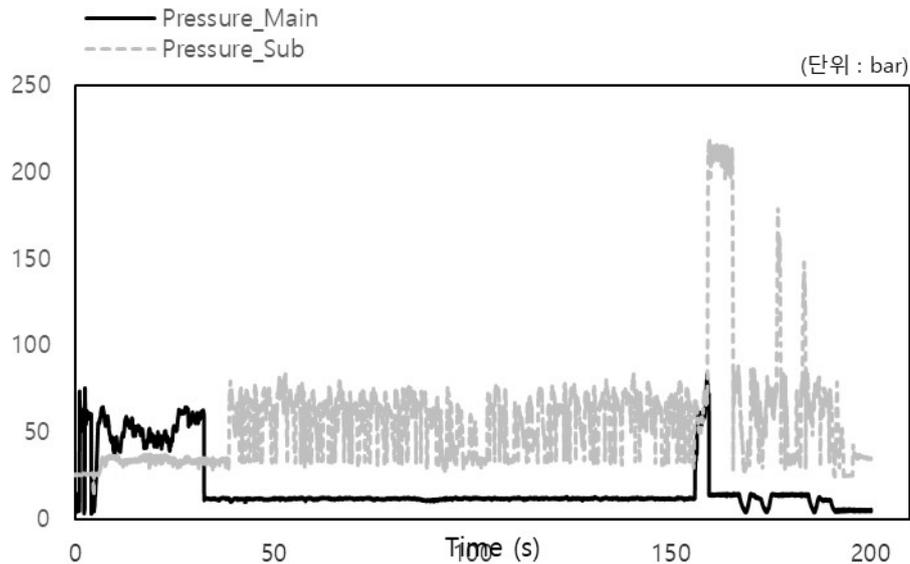
트랙터 차축 토크 및 출력은 아래 그림과 같이 로터리 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 0~4.3 kNm 범위를 보이고 있으며, 후륜 토크는 약 0~10.2 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2.3 배 더 높게 나타나고 있음.



[ A3 조건에서의 로터리 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

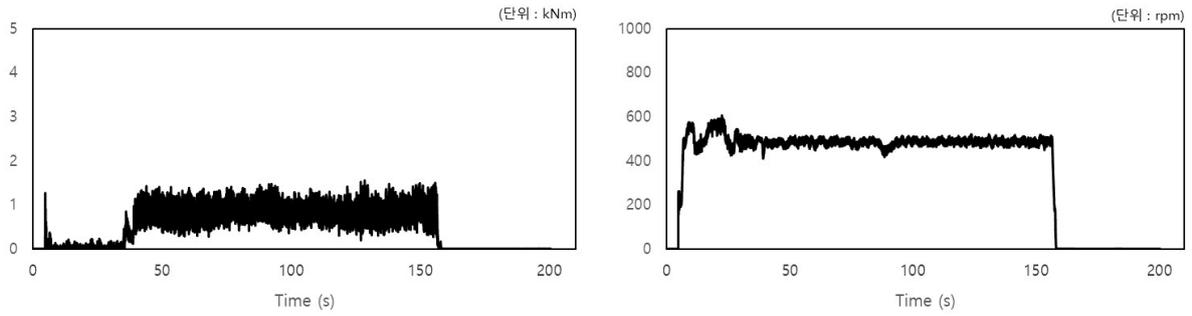
메인 펌프는 로터리 작업 시 로터리의 수평유지를 위해 유압회로를 작동시켜 대부분 12bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar 이상 증가하는 구간도 나타났음. 보조 펌프는 로터리 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 30~80 bar 범위로 나타났음.



[ A3. 조건에서의 로터리 작업 시 메인 및 보조 유압펌프의 압력 ]

- PTO 부하분석

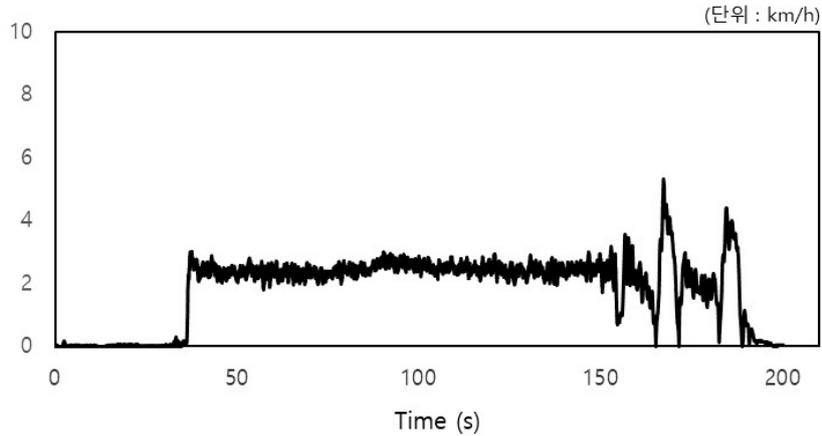
PTO 토크의 최소, 평균, 최대, 표준편차는 각각 0.1, 0.8, 1.6, 0.2 kNm로 나타났음. 회전속도의 경우 최소, 평균, 최대, 표준편차는 각각 410, 485, 521, 13 rpm으로 나타났음. 출력의 최소, 평균, 최대, 표준편차는 각각 10, 40, 57, 10 kW로 나타났음.



[ A3 조건에서의 로터리 작업 시 PTO 토크 및 회전속도 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 2.4 km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 1.7, 2.4, 3.2, 0.2 km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ A3 조건에서의 로터리 작업 시 작업 속도 ]

- A3 조건에서 로터리 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

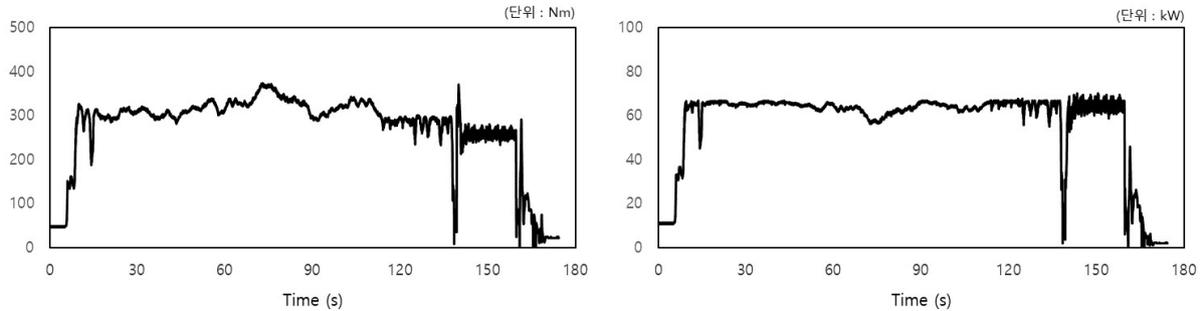
표 76. A3 조건에서 로터리 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	103.19	288.09	333.49	30.09
	Rotational speed (rpm)	1,806	2,042	2,152	44
	Power (kW)	22.53	61.54	66.1	5.92
Torque_front (Nm)		1.31	1,973.33	4,274.86	644.92
Torque_rear (Nm)		0.03	885.72	10,165.42	804.5
Main pressure (bar)		9.86	11.86	13.08	0.36
Sub pressure (bar)		27.66	57.6	83.8	14.4
PTO	Torque (Nm)	115.09	848.66	1,555.71	227.11
	Rotational speed (rpm)	410.29	484.73	521.02	12.56
	Power (kW)	5	43	57	11
Velocity (km/h)		1.71	2.44	3.18	0.18

⑧ A3 단수에서의 부하 계측 시험 결과 - 2

- 엔진 토크 및 출력

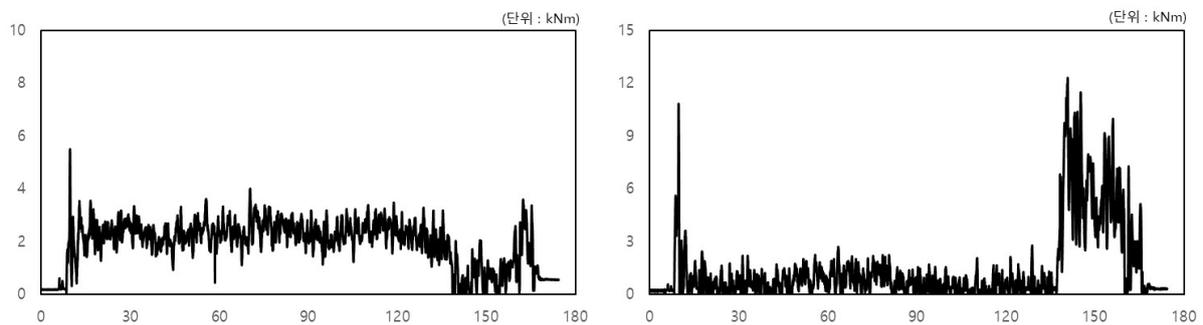
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 7 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 130~370 Nm, 약 32~68 kW의 범위를 보였음. 이후 약 136sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ A3 조건에서의 로터리 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

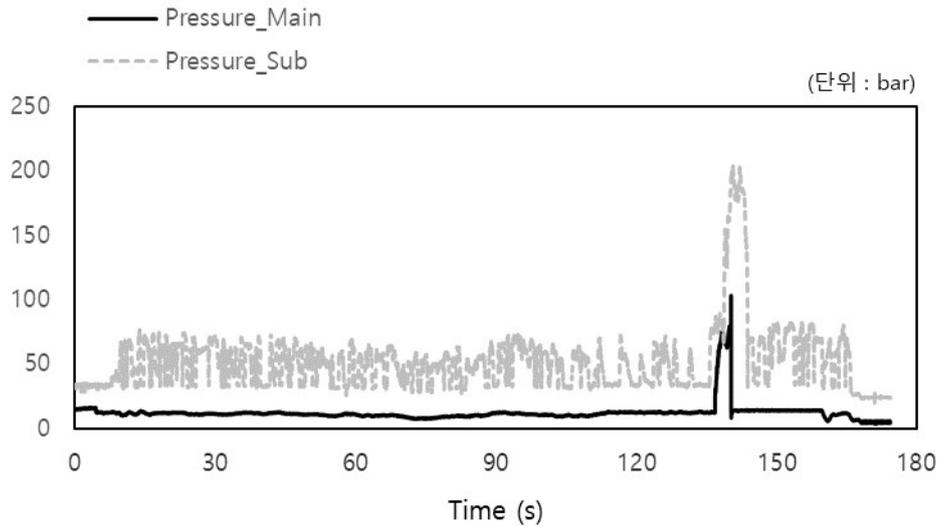
트랙터 차축 토크 및 출력은 아래 그림과 같이 로터리 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 0~5.5 kNm 범위를 보이고 있으며, 후륜 토크는 약 0~10.8 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2배 더 높게 나타나고 있음.



[ A3 조건에서의 로터리 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

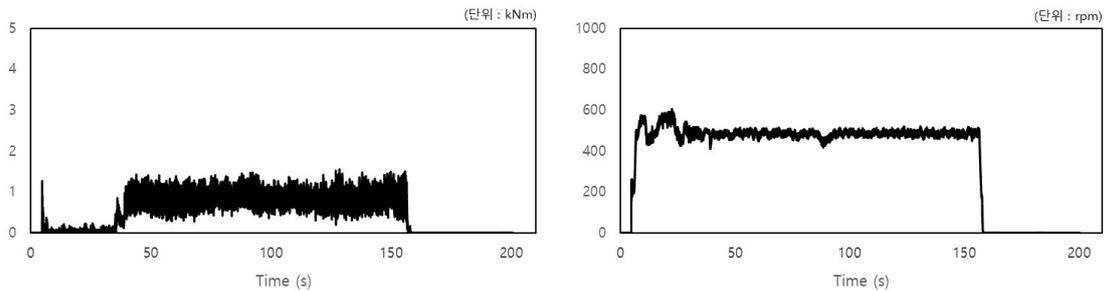
메인 펌프는 로터리 작업 시 로터리의 수평유지를 위해 유압회로를 작동시켜 대부분 11bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났음. 보조 펌프는 로터리 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 25~78 bar 범위로 나타났음.



[ A3 조건에서의 로터리 작업 시 메인 및 보조 유압펌프의 압력 ]

- PTO 부하분석

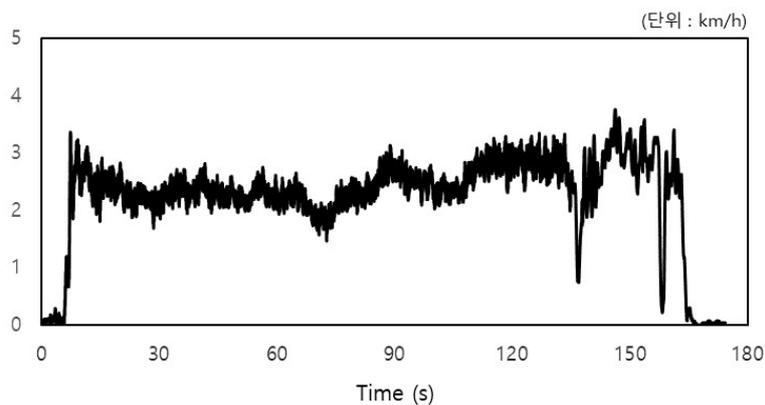
PTO 토크의 최소, 평균, 최대, 표준편차는 각각 0.04, 0.96, 1.86, 0.26 kNm로 나타났음. 회전 속도의 경우 최소, 평균, 최대, 표준편차는 각각 328, 469, 555, 45 rpm으로 나타났음. 출력의 최소, 평균, 최대, 표준편차는 각각 2, 46, 65, 12 kW로 나타났음.



[ A3 조건에서의 로터리 작업 시 PTO 토크 및 회전속도 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 2.4 km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 1.5, 2.4, 3.4, 0.3 km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ A3 조건에서의 로터리 작업 시 작업 속도 ]

- A3 조건에서 로터리 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

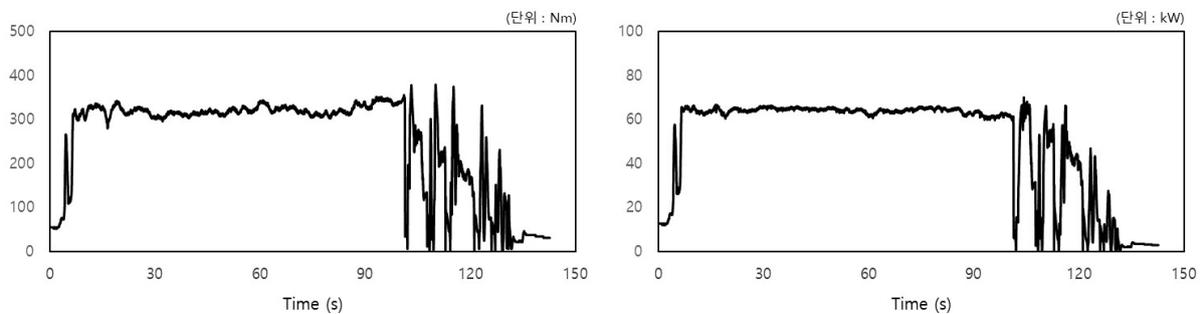
표 77. A3 조건에서 로터리 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	134.51	309.43	374.06	29.25
	Rotational speed (rpm)	1,451	1,979	2,297	185
	Power (kW)	31.56	63.67	67.81	4.05
Torque_front (Nm)		10.26	2,297.07	5,498.15	522.62
Torque_rear (Nm)		0.09	742.64	10,823.84	789.01
Main pressure (bar)		7.03	10.96	13.63	1.28
Sub pressure (bar)		25.53	49.08	77.75	12.88
PTO	Torque (Nm)	45.09	961.78	1,865.88	268.09
	Rotational speed (rpm)	328.56	469.78	555.98	44.88
	Power (kW)	2	47	65	12
Velocity (km/h)		1.47	2.44	3.37	0.3

⑨ A4 단수에서의 부하 계측 시험 결과 - 1

- 엔진 토크 및 출력

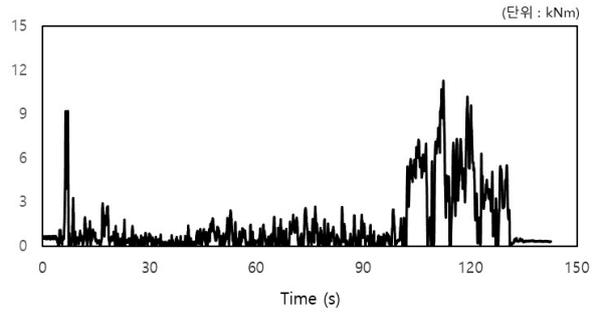
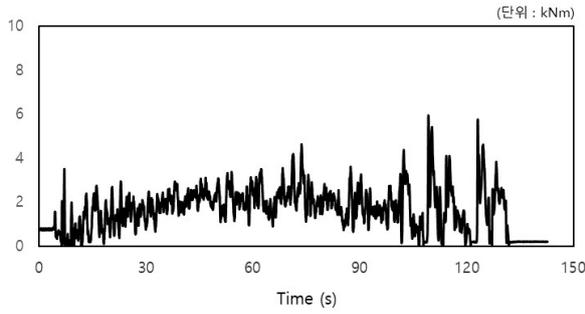
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 6 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 110~350 Nm, 약 26~66 kW의 범위를 보였음. 이후 약 96sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ A4 조건에서의 로터리 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

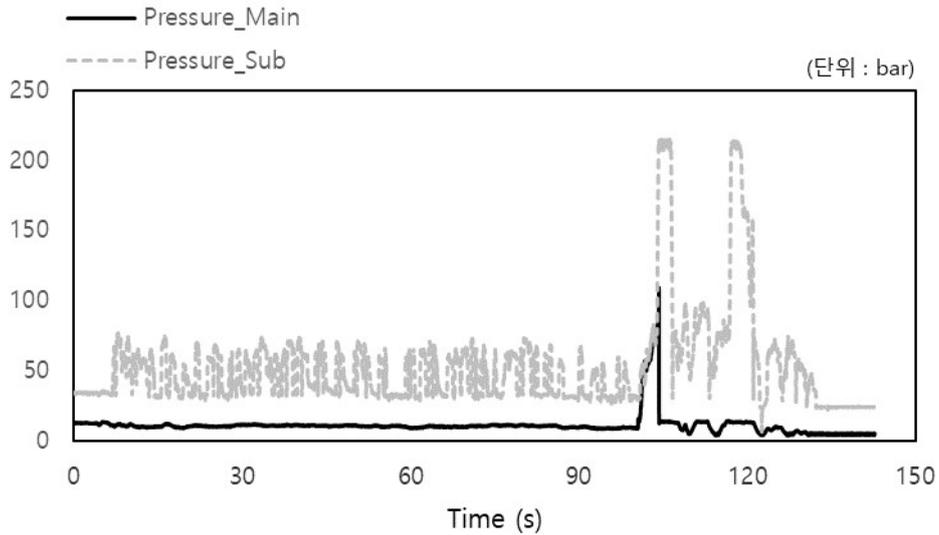
트랙터 차축 토크 및 출력은 아래 그림과 같이 로터리 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 0~4.6 kNm 범위를 보이고 있으며, 후륜 토크는 약 0~9.2 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2배 더 높게 나타나고 있음.



[ A4 조건에서의 로터리 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

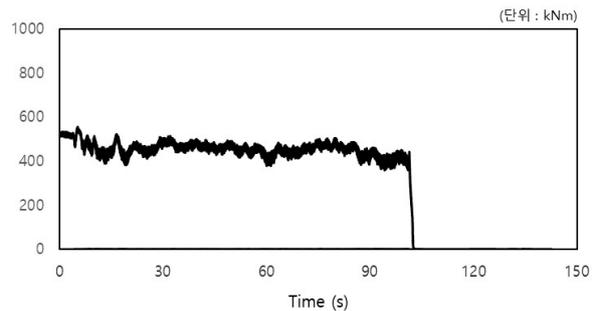
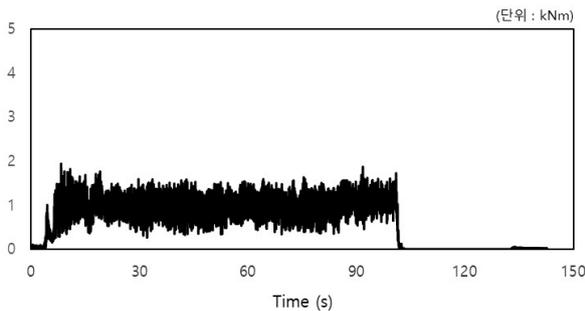
메인 펌프는 로터리 작업 시 로터리의 수평유지를 위해 유압회로를 작동시켜 대부분 11bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar 이상 증가하는 구간도 나타났음. 보조 펌프는 로터리 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 26~78 bar 범위로 나타났음.



[ A4 조건에서의 로터리 작업 시 메인 및 보조 유압펌프의 압력 ]

- PTO 부하분석

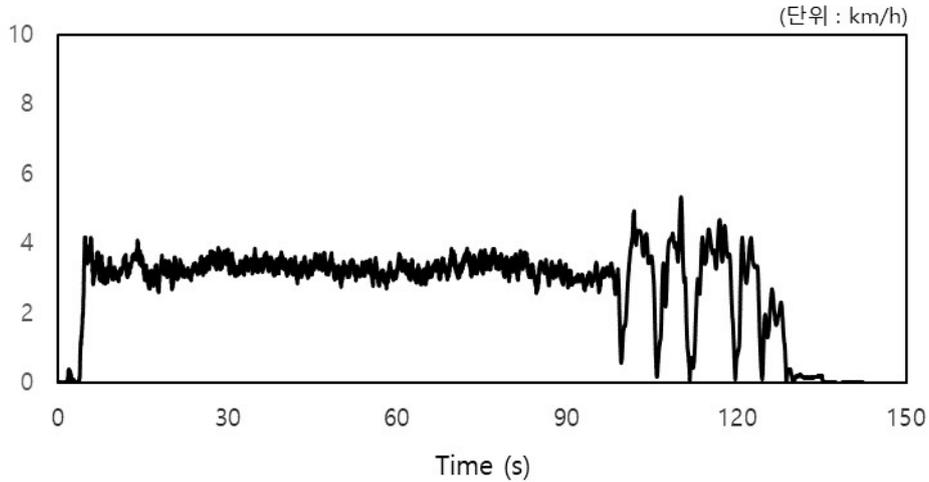
PTO 토크의 최소, 평균, 최대, 표준편차는 각각 0.15, 0.98, 1.94, 0.26 kNm로 나타났음. 회전 속도의 경우 최소, 평균, 최대, 표준편차는 각각 359, 455, 539, 25 rpm으로 나타났음. 출력의 최소, 평균, 최대, 표준편차는 각각 8, 47, 65, 13 kW로 나타났음.



[ A4 조건에서의 로터리 작업 시 PTO 토크 및 회전속도 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 3.3 km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 2.6, 3.3, 4.2, 0.2 km/h로 나타났으며, 이는 작업단수를 고려하여 적절한 수준으로 나타났음.



[ A4 조건에서의 로터리 작업 시 작업 속도 ]

- A4 조건에서 로터리 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

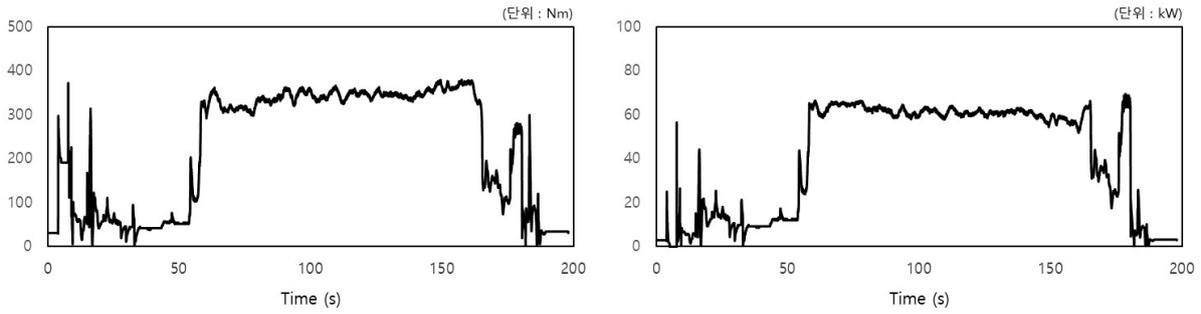
표 78. A4 조건에서 쟁기 경운 시 부하 분석 결과

		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	111.14	318.05	351.88	20.11
	Rotational speed (rpm)	1,652	1,918	2,255	98.3
	Power (kW)	26.22	63.72	66.73	3.28
Torque_front (Nm)		0.47	1,844.92	4,634.83	733.84
Torque_rear (Nm)		0.01	650.63	9,222.89	799.41
Main pressure (bar)		8.58	10.73	13.55	0.71
Sub pressure (bar)		26.22	47.44	78.24	14.09
PTO	Torque (Nm)	155.73	982.16	1,949.54	264.84
	Rotational speed (rpm)	359.28	455.11	538.88	25.35
	Power (kW)	8	47	65	13
Velocity (km/h)		2.58	3.29	4.18	0.23

⑩ A4 단수에서의 부하 계측 시험 결과 - 2

- 엔진 토크 및 출력

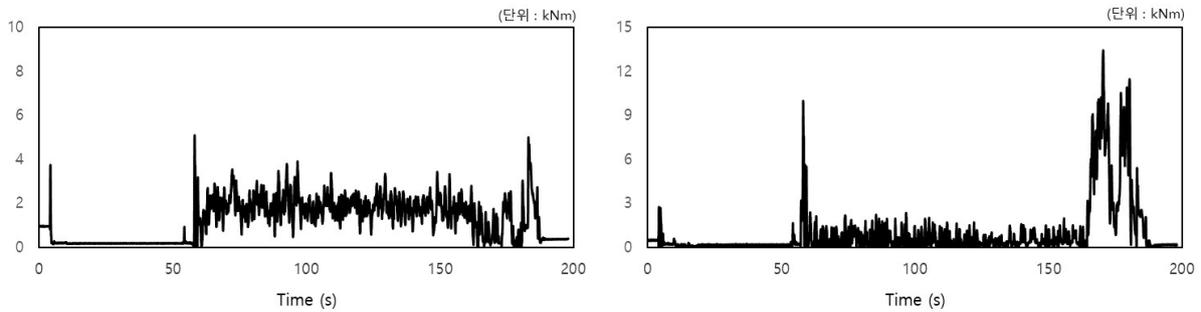
트랙터 엔진 토크 및 출력은 아래 그림과 같이 약 56 sec에 작업을 시작함에 따라 급격하게 증가하는 개형으로 나타내며, 작업구간에서 토크, 출력 각각 약 102~380 Nm, 약 24~66 kW의 범위를 보였음. 이후 약 163sec에 작업이 종료됨에 따라 토크 및 출력이 감소하는 개형으로 나타났음.



[ A4 조건에서의 로터리 작업 시 엔진 토크 및 출력 ]

- 차축 토크 및 출력

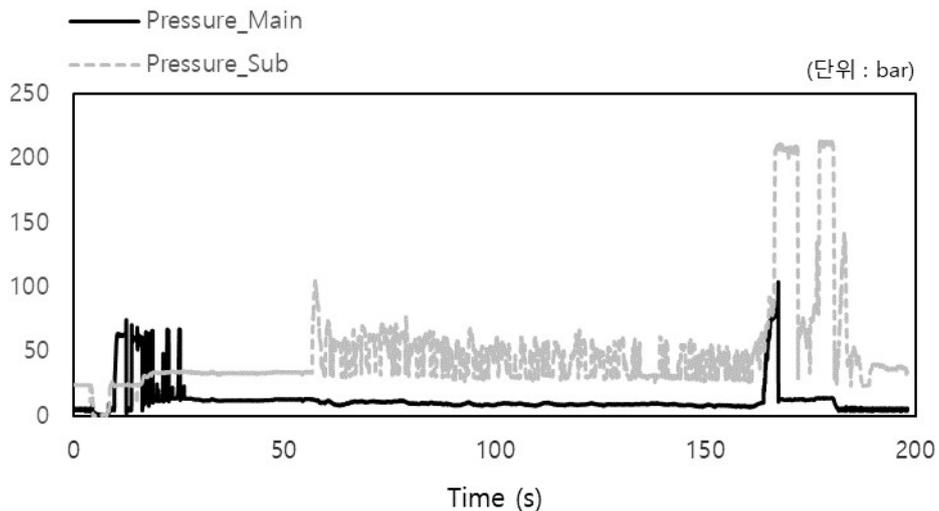
트랙터 차축 토크 및 출력은 아래 그림과 같이 로터리 경운 시작 후 부착 작업기 견인을 위하여 급격하게 증가하는 개형으로 나타내고 있음. 트랙터 전륜 토크는 약 0~5.1 kNm 범위를 보이고 있으며, 후륜 토크는 약 0~9.9 kNm의 범위를 보이고 있음. 후륜의 토크가 전륜 토크의 약 2배 더 높게 나타나고 있음.



[ A4 조건에서의 로터리 작업 시 전륜(좌) 및 후륜(우) 차축 토크 ]

- 메인 및 보조 펌프 압력 및 출력

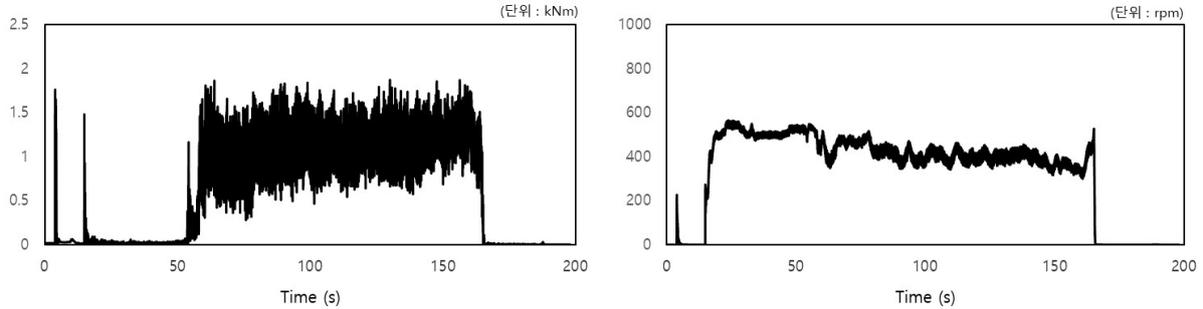
메인 펌프는 로터리 작업 시 로터리의 수평유지를 위해 유압회로를 작동시켜 대부분 9 bar 수준을 유지하였으며 높은 부하 발생 시 히치를 제어하여 순간적으로 13 bar이상 증가하는 구간도 나타났음. 보조 펌프는 로터리 작업 중 토양이 평평하지 않기 때문에 잦은 조향 작업에 의해 압력이 25~105 bar 범위로 나타났음.



[ A4 조건에서의 로터리 작업 시 메인 및 보조 유압펌프의 압력 ]

- PTO 부하분석

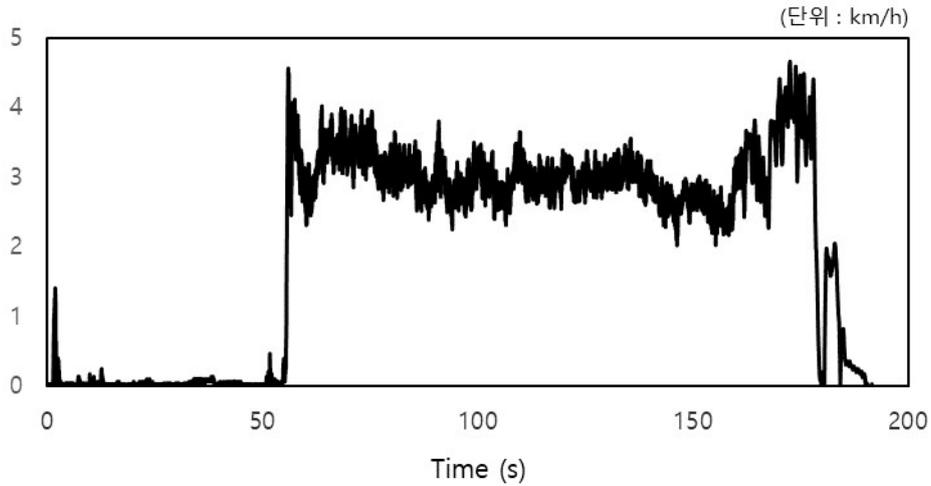
PTO 토크의 최소, 평균, 최대, 표준편차는 각각 0.05, 1.11, 1.87, 0.29 kNm로 나타났음. 회전 속도의 경우 최소, 평균, 최대, 표준편차는 각각 299, 409, 545, 42 rpm으로 나타났음. 출력의 최소, 평균, 최대, 표준편차는 각각 3, 47, 63, 12 kW로 나타났음.



[ A4 조건에서의 로터리 작업 시 PTO 토크 및 회전속도 ]

- 작업 속도

작업 속도 분석결과, 작업 간 평균속도는 3 km/h 으로 나타났음. 작업 속도의 최소, 평균, 최대, 표준편차는 각각 2, 3, 5, 0.3 km/h로 나타났으며, 이는 작업단수를 고려하여 적정한 수준으로 나타났음.



[ A4 조건에서의 로터리 작업 시 작업속도 ]

- A4 조건에서 로터리 경운 시 엔진, 차축, 메인 및 보조 유압 데이터 분석 결과는 아래 표와 같음.

표 79. A4 조건에서 로터리 경운 시 부하 분석 결과

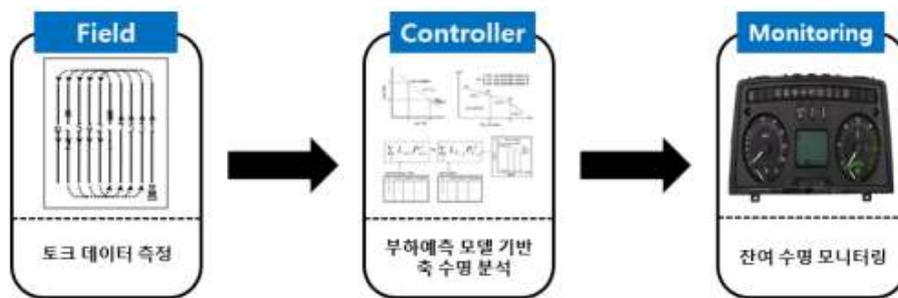
		Minimum	Average	Maximum	Standard deviation
Engine	Torque (Nm)	102.8	337.53	379.82	35.23
	Rotational speed (rpm)	1342	1723	2248	169
	Power (kW)	24.17	60.39	66.48	5.2
Torque_front (Nm)		0.16	1,824.05	5,101.46	610.72
Torque_rear (Nm)		0.02	596.03	9,990.29	714.97
Main pressure (bar)		6.39	9.16	13.18	1.15
Sub pressure (bar)		25.42	45.73	105.39	13.22
PTO	Torque (Nm)	53.35	1108.22	1,872.73	290.03
	Rotational speed (rpm)	299.29	408.83	545.13	42.3
	Power (kW)	3	47	63	12
Velocity (km/h)		2.02	2.99	4.56	0.33

㉞ 트랙터 모니터링 시스템을 이용한 고장 예지 기술 개발

㉠ 트랙터 주요 부품 고장 예지 항목 선정

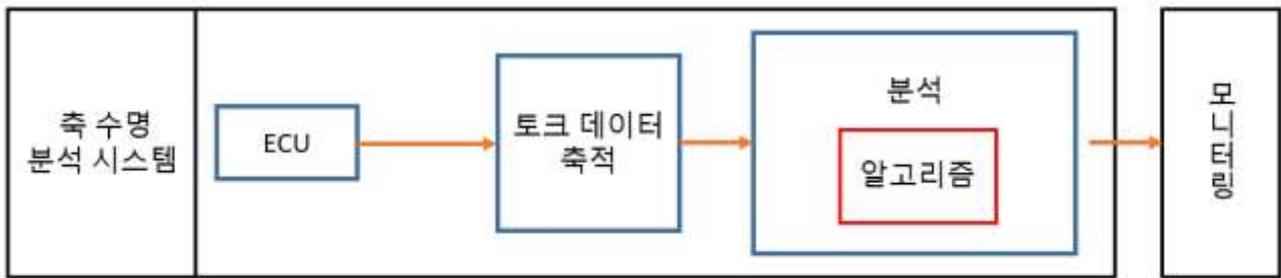
- 변속기 축 수명 분석

1. 트랙터 동력전달계 중 변속기는 토양 특성, 작업조건 등에 따라 파손 및 고장 영향이 가장 높음.  
동력전달계 중 변속기가 아닌 다른 부품은 별도의 센서가 필요하며 별도의 센서 설치의 번거로움과 센서를 계측하기 위한 별도의 시스템을 구축해야 하지만 변속기는 엔진의 ECU 데이터를 통해 분석이 가능함
2. 트랙터의 동력전달은 엔진의 동력이 단수에 따라 주변속 기어와 부변속 기어에 전달되며, PTO의 경우 엔진 축과 직결되어 있어 바로 전달됨
3. 본 연구에서는 위 사항들을 고려하여 엔진의 ECU 데이터 중 토크 데이터를 활용하여 변속기 축 수명을 분석하며 변속기의 주변속 축, 부변속 축, PTO 축의 부하를 계산하여 각각의 축에 대한 수명을 계산함
4. 트랙터 사용 간, 엔진 ECU를 통해 토크 데이터를 측정하고 부하예측 알고리즘을 개발하여 각 축에 대한 수명을 분석하여 분석한 축 수명을 사용자가 모니터링 할 수 있음



[트랙터 수명 모니터링 시스템 프로세스]

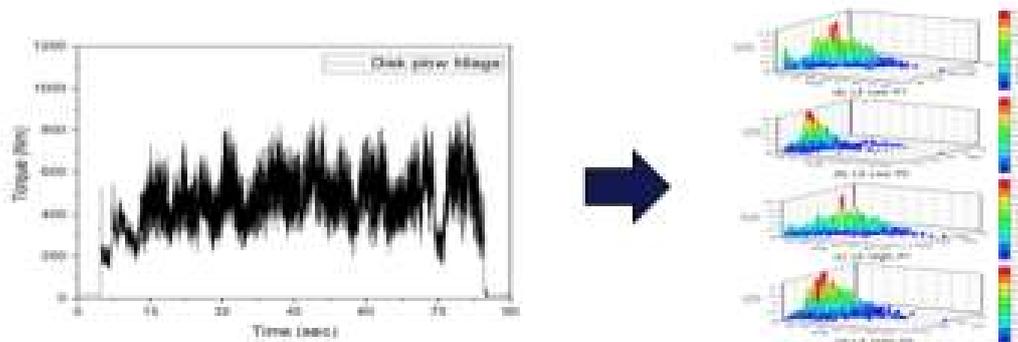
5. 부하 분석 절차를 통해 축 잔여수명을 계산하였으며 사용자가 트랙터 사용 시, ECU의 엔진 토크 데이터를 축적하며 축적된 토크 데이터를 트랙터 작업 단수, 작업 종류에 따라 토크를 부하 분석 절차를 통해 각 축에 대해 손상합을 계산하여 축의 잔여 수명을 모니터로 전달함



[트랙터 측 수명 분석 시스템]

- 변속기 수명 해석 프로세스

1. 하중 파형 사이클 계산법에서는 일반적으로 레인지 페어법과 낙수계수법이 합리적인 것으로 알려져 있음. 본 연구에서는 낙수계수법을 이용하여 시간 함수의 부하 신호를 부하의 크기와 빈도수로 나타냄. 이를 부하 스펙트럼으로 작성하고 부하 스펙트럼은 측정된 부하의 최대 및 최소를 고려하여 32등분 하여 나타냄



[낙수계수법을 이용하여 부하 스펙트럼으로 표현]

2. HBM 社の nCode software를 통해 측정된 데이터를 낙수계수법으로 표현 가능하며 부하 스펙트럼으로 나타냄



[HBM 社の nCode software를 통한 낙수계수법]

3. 레인플로우법을 통해 나타낸 부하 스펙트럼은 스미스-왓슨-토퍼(Smith-Watson-Topper)식을 이용하여 평균 토크와 기어비를 고려함

$$T_e = \sqrt{(t_r + t_m)t_r}$$

여기서,  $t_r$  : 평균토크(Nm)

$t_m$  : 엔진 토크의 진폭(Nm)

$T_e$  : 공칭토크(Nm)

4. 가속도 계산을 위해 토크를 응력으로 변환해야 하며, 앞서 계산한 공칭토크를 응력 계산식에 대입하여 응력으로 변환함. 계산식은 아래와 같음

$$S = \frac{16 T_e}{\pi d^3}$$

여기서,  $S$ : 응력(MPa)

$T_e$  : 공칭토크(Nm)

$d$ : 축의 직경(mm)

5. 재료의 S-N 곡선을 구하기 위해 선행연구에서 제시된 SCM420H 재질을 선정했으며 계산식을 통해 선형화된 S-N 곡선을 표현함. 계산식은 아래와 같음

Material	SCM420H
Ultimate strength of test specimen (Mpa)	2300
Fatigue strength of test specimen (Mpa)	700
Surface factor	0.580
Size factor	0.876
Load factor	0.577
Temperature factor	1.010
Miscellaneous-effects factor	0.840

[입력축의 피로강도 및 수정계수 값]

$$N = 10^{(6 - 6.097 \log \frac{S}{223})}$$

여기서,  $N$ : 재료가 허용할 수 있는 빈도수(Cycles)

$S$ : 응력(Mpa)

6. S-N 곡선을 통해 최종적으로 재료의 파손 시기를 파악함. 수정된 마이너 법칙(Miner's rule)에 따르면 재료의 손상합이 1이 되면 재료는 파손되며 계산식으로 표현함. 계산식은 아래와 같음

$$D = \sum_{i=1}^k \frac{n_i}{N_i}$$

여기서,  $D$ : 손상합

$n_i$  : 빈도수(Cycle)

$N_i$  : 피로수명(Cycle)

7. 부하 스펙트럼에 앞서 개발한 프로세스와 같이 계산을 진행하였음. 필드 테스트를 통해 수집한 시간은 약 260초 정도로 아래 식을 이용하여 획득한 부하 데이터 사이클을 트랙터의 전체수명 시간으로 환산하여 연장하였으며, 선행연구 및 통계자료를 기준으로 트랙터의 수명은 10년, 연간 사용시

간은 311.2 시간으로 설정함 (KAMICO and KSAM, 2017). 사용자에게 잔여 수명을 표시하기 위해 계산식을 통해 손상합을 퍼센트로 표현함. 계산식은 아래와 같음

$$N_T = 3600NLh$$

여기서,  $N_T$ : 빈도수(Cycles)

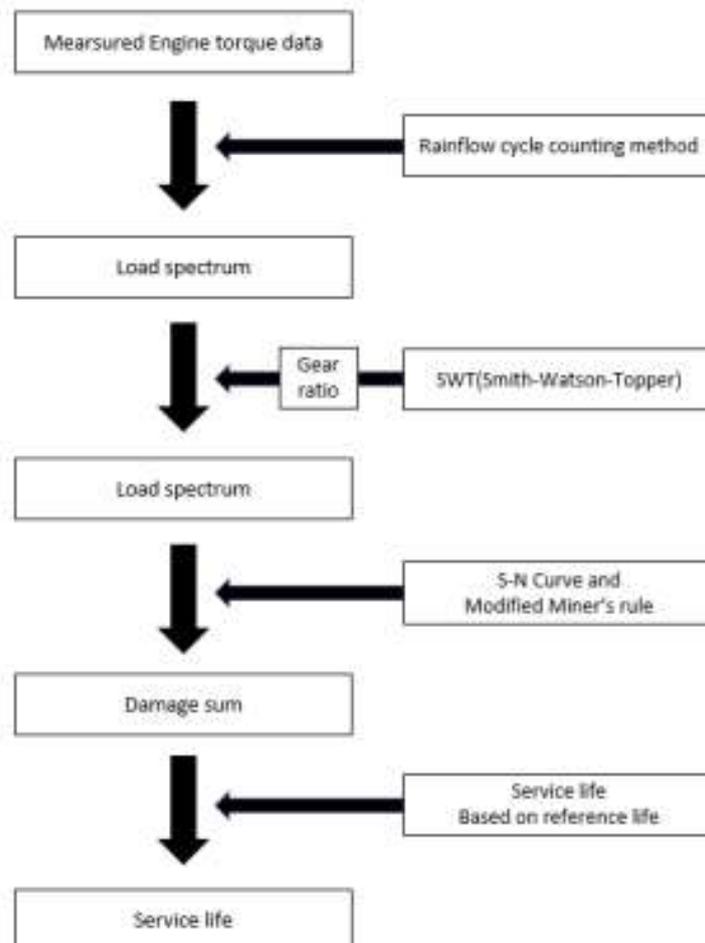
$L$ : 트랙터 내용연수(Year)

$h$ : 작동시간(h/year)

$$100(1 - D) = L\%$$

여기서,  $L$ : 축의 잔여 수명

8. 트랙터의 ECU에서 토크 데이터를 측정하여 측정된 토크 데이터를 낙수계수법을 이용하여 부하 스펙트럼으로 표현함. 기어비와 평균 토크를 고려하기 위해 스미스-왓슨-토퍼법을 적용했으며 S-N 곡선과 마이너 법칙을 적용하여 재료의 파손 시기를 파악함. 최종적으로 손상합을 통해 잔여 수명을 퍼센트로 나타냄. 위 과정들을 하나의 절차로 통합했으며 아래와 같음



[부하 분석을 통한 축 잔여 수명 해석 절차]

㉔ 트랙터 주요 부품 고장 예지 기술 개발 및 성능평가

- 필드 테스트 조건

1. 수명 해석 절차를 기반으로 트랙터의 주요 구동 축 고장을 예지하는 데 필요한 부하 데이터를 측정하기 위해 필드 테스트를 수행하였음.
2. 트랙터의 필드 테스트는 쟁기 및 로타리 작업을 수행했으며 이때의 엔진 데이터를 계측하였음. 필드 테스트는 서산에 위치한 논에서 수행하였으며, 쟁기 작업을 수행한 후에 로타리 작업을 실시하

였음

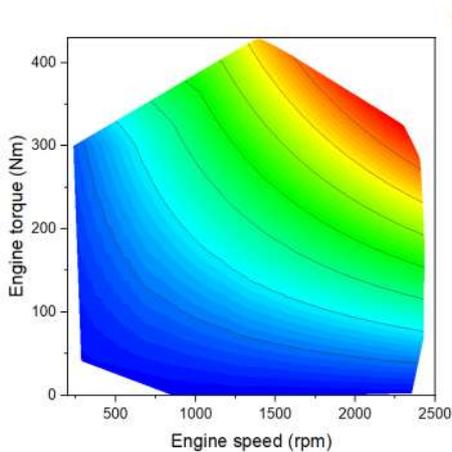
- 3. 작업 단수는 로타리 작업 시 L3 High P1 단, 쟁기 작업 시 M2 High 단계에서 수행하였으며, 각 작업 당 3반복 수행하였음



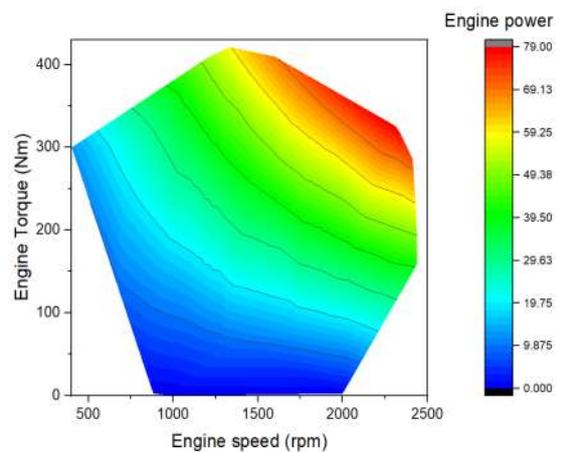
[부하 데이터 측정을 위한 쟁기/로타리 필드테스트 수행]

- 필드 테스트 결과

- 1. 필드 테스트 진행한 토양의 입자 밀도는  $2.33 \text{ g}\cdot\text{cm}^{-3}$ , Sand, Silt, Clay 각각 16.00% ,62.00%, 22.00%이며 필드 테스트를 통해 엔진 토크, 엔진 스피드, 출력 3가지를 측정함. 트랙터 로타리 작업 시 발생한 엔진 토크의 최대, 최소, 평균, 표준편차는 각각 407.94, 178.06, 322.79, 26.89 Nm로 나타남. 엔진 스피드의 최대, 최소, 평균, 표준편차는 각각 2430, 1611, 2241.66, 132.24 rpm으로 나타남. 작업 간 엔진 출력의 최대, 최소, 평균, 표준편차는 각각 78.82, 45.11, 75.45, 3.52 kW로 나타남. 트랙터 쟁기 작업 시 발생한 엔진 토크의 최대, 최소, 평균, 표준편차는 각각 428.69, 0.56, 262.67, 72.95 Nm로 나타남. 엔진 스피드의 최대, 최소, 평균, 표준편차는 각각 2431, 722.50, 2052.65, 493.01 rpm으로 나타남. 엔진 출력의 최대, 최소, 평균, 표준편차는 각각 78.34, 0.06, 58.89, 20.47 kW로 나타남. 그래프 개형은 아래와 같음



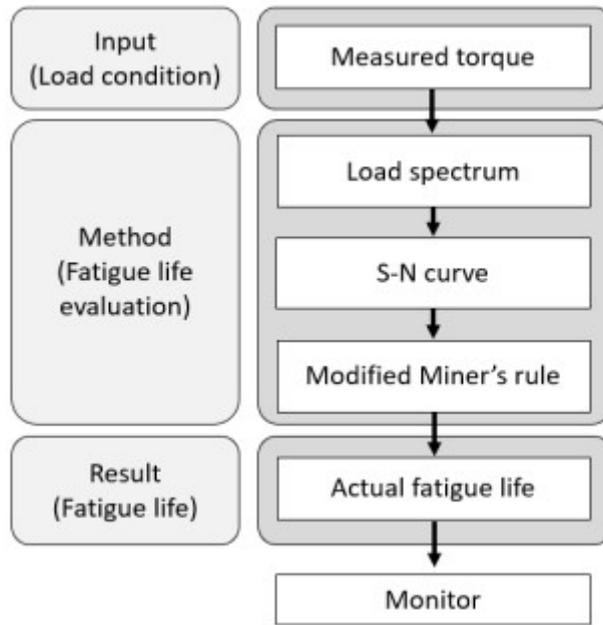
[쟁기 작업 시 발생한 부하 데이터]



[로타리 작업 시 발생한 부하 데이터]

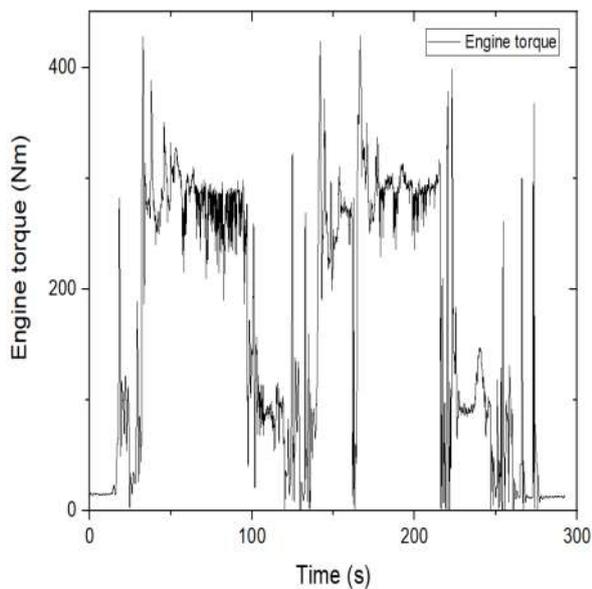
- 필드 테스트 부하 데이터 기반 축 수명 분석

- 1. 엔진 CAN 통신에서 계측된 실시간 토크 데이터는 아래 그림과 같이 부하 스펙트럼, S-N 커브, 손상합 계산 과정을 통해 트랙터 주요 축(주변속, 부변속, PTO) 수명으로 계산됨

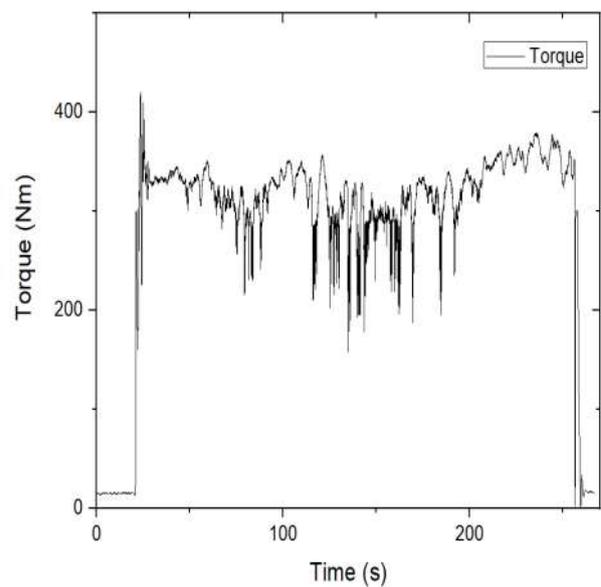


[축 수명 분석 프로세스]

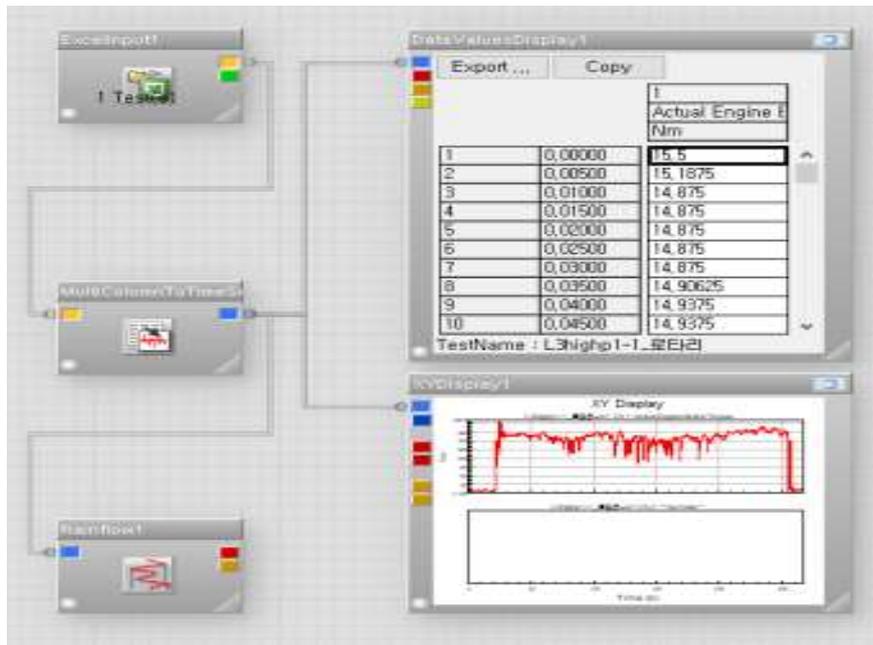
2. 필드 테스트를 통해 ECU에서 계측된 실시간 토크 데이터를 바탕으로 축 수명 프로세스를 이용하여 축 수명의 데이터를 분석해본 결과 농기계의 실사용 중 연료 소모량 및 작업부하와 큰 차이가 없는 것으로 판단됨. 토크 데이터를 축 수명 프로세스를 기반으로 축 수명을 분석 쟁기 작업 시 발생한 엔진 토크 데이터를 낙수계수법을 이용하여 시간 함수의 부하를 빈도수의 함수로써 나타냄. 아래 그림은 쟁기 작업 간 측정된 토크 데이터에 낙수계수법을 적용하는 과정을 나타냄



[쟁기 작업 시 발생한 엔진 토크 데이터]

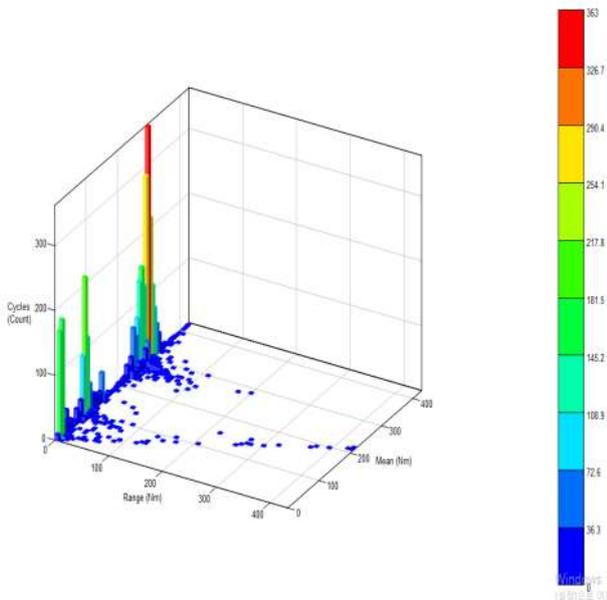


[로타리 작업 시 발생한 엔진 토크 데이터]

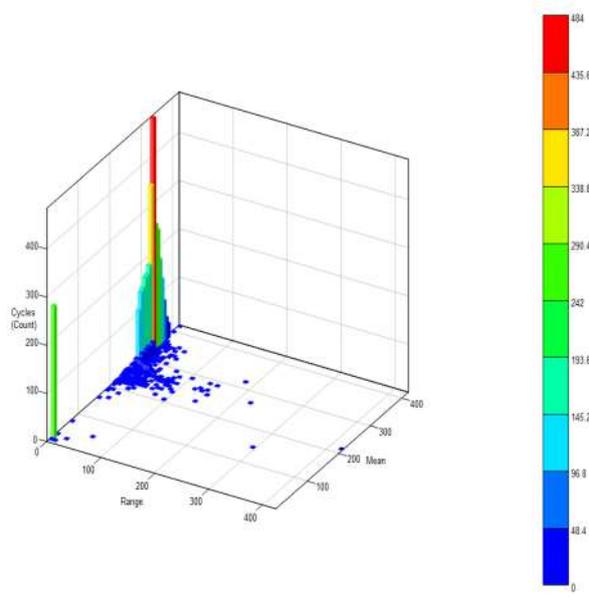


[ S/W를 활용한 농작업 데이터 3D map 변환 프로세스 ]

3. S/W를 통해 계측된 엔진 토크 데이터는 진폭(Range), 평균(Mean), 빈도수(CycleS)의 3D map으로 표현하였다. 그 결과, 쟁기 작업 간 토크의 평균과 진폭은 각각 0 ~ 405 Nm, 0 ~ 400 Nm의 범위를 보임. 가장 높은 빈도수를 나타내는 평균 부하의 범위는 270~330 Nm로 나타났으며, 로타리 작업 간 토크의 평균과 진폭은 각각 0 ~ 400 Nm, 0 ~ 390 Nm로 나타남. 가장 높은 빈도수를 나타내는 평균 부하의 범위는 310 ~ 350 Nm로 나타남



[ 쟁기 작업 간 토크 데이터 3D map ]



[ 로타리 작업 간 토크 데이터 3D map ]

4. 3D map으로 계산된 엔진 데이터는 부하스펙트럼으로 계산됨. 부하 스펙트럼은 농작업시 발생하는 변동 부하 조건을 고려하기 위한 것으로 SWT (Smith-Watson Topper)법을 이용하여 공칭토크로 계산함. 공칭 토크를 계산한 후, 주요 축(주변속, 부변속, PTO) 수명을 각각 산출하기 위해 기어비가 적용되며 S-N 선도를 이용하여 응력으로 계산됨. 그 다음, S-N 선도 기반의 손상합(Damage sum)을 계산하고 이때 트랙터 농작업 간 사용 비율을 적용하였으며(쟁기 작업: 0.25, 로타리 작업: 0.3) 이를 기준수명에 적용하여 잔여수명이 산출됨. 아래 그림은 위의 계산 프로세스를 나타냄



### 3. 연구개발과제의 수행 결과 및 목표 달성 정도

#### 1) 연구수행 결과

##### (1) 정성적 연구개발성과

###### <주식회사 긴트 >

- 농기계 이력관리 시스템을 개발하고 이를 블록체인 네트워크와 연계 개발하여 안정성을 확보하였음. 이를 바탕으로 공동 연구기관과 연계될 수 있도록 API를 제공하고, 기타 부대 시스템을 개발하였음. 또한 정부, 제조사 등에서 활용 가능한 웹서비스 시스템을 개발하여 제공 가능해짐
- 사업화를 위한 양산에 적합한 환경시험을 수행하였으며, 이와 더불어 실증을 통한 서비스의 개선도 이루어 나감

###### <경상대학교 산학협력단>

- 농기계 이력/운행데이터 API 인증 및 데이터 특징 분석
- 농기계 이력/운행 트레이닝 데이터 AI 알고리즘 개발
- 통계 및 머신러닝 기반 기법을 적용한 고도화
- 연동된 데이터 API와 설문조사를 바탕으로 농기계 이력 데이터를 수집함. 이를 바탕으로 농기계 이력 관리에 사용될 수 있는 알고리즘을 개발함, 또한 외부에서 활용 가능한 웹서비스 시스템을 개발하여 API를 통해 서비스 제공이 가능해짐.

###### <충남대학교 산학협력단>

- 부하 계측 시스템 설계 및 개발 과정에서 관련 노하우를 확보하고, 67 kW 트랙터의 부하 계측 시스템을 구축하였음. 확보한 관련 노하우를 활용하여 차년도에는 모니터링 시스템 개선을 수행할 예정이며, 이를 이용하여 농기계 이력 데이터 블록체인 시스템의 성능 최적화 및 개선을 수행함.
- 트랙터 변속기 축 수명 계산 프로세스를 이용하여 트랙터 변속기의 수명 진단 기술을 개발하였음. 또한, 작업중에 변속기 수명을 모니터링 할 수 있는 기술을 개발하였음. 고장 인지 기술 개발을 통해 고장 예방이 가능하도록 농기계 이력 데이터 블록체인 시스템을 구축하였음.

###### <한국농기계협동조합>

- 여러 시군구와 MOU를 맺어 실증의 기반을 다짐. 이를 바탕으로 실증 대상 농기계를 확대해 나감으로서 연구에 필요한 실데이터를 수집가능하게 해줌.
- 실증 내용을 바탕으로 차년도에는 정책제안에 충실하게 수행함

##### (2) 정량적 연구개발성과(해당 시 작성하며, 연구개발과제의 특성에 따라 수정이 가능합니다)

구분	평가항목 (주요 성능 Spec)	단위	전체 항목 에서 차지 하는 비중 (%)	세계최고 수준 보유국/ 보유기업 ( / )	연구개발 전 국내수준	개발 목표치			평가방법	
				성능수준		성능수준	1차 년도	2차 년도		3차 년도

농기계 장착형 단말기	내환경 신뢰성	℃, %, g	-40~85(보관), -35~75(동작), 90%RH, 4g (미국/Caterpillar)	-40~85(보관), -35~75(동작), 90%RH, 4g	-	-20~70(보관) -15~60(동작) 90%RH, 4g	-40~85(보관) -35~75(동작) 90%RH, 4g 전자기장해(EMC) 시험	공인시험성적서 (KS R 0015, KS R 1034, 75/322/EEC)
	통신 성공률	%	95 (한국/통신3사)	95	-	90	95	공인기관 입회시험
데이터 수집 서버	수용 가입자	가입자 수	- (-/-)	-	500	20,000	100,000	공인기관 입회시험
	서버 응답속도	초	- (-/-)	-	3≥	2≥	1≥	공인기관 입회시험
블록체인 네트워크	초당 트랜잭션 처리	초당 트랜잭션 수(TPS)	- (-/-)	-	-	200	500	공인기관 입회시험
	블록체인 프록시 업로드 속도	초당 업로드 수	- (-/-)	-	-	500	2000	공인기관 입회시험
농업인용 서비스	수용 가입자	가입자 수	- (-/-)	-	500	20,000	100,000	공인기관 입회시험
	서버 응답속도	초	- (-/-)	-	7≥	5≥	3≥	공인기관 입회시험
	분산원장 기여 데이터 가짓수	개	- (-/-)	-	2	4	6	공인기관 입회시험
	AI 알고리즘 연동 서비스 개수	개	- (-/-)	-	-	1	1	공인기관 입회시험
제조사용 서비스	수용 가입자	가입자 수	- (-/-)	-	-	500	20,000	공인기관 입회시험
	서버 응답속도	초	- (-/-)	-	-	5≥	3≥	공인기관 입회시험
	AI 알고리즘 연동 서비스 개수	개	- (-/-)	-	-	-	1	공인기관 입회시험
정부지원 서비스	수용 가입자	가입자 수	- (-/-)	-	-	500	20,000	공인기관 입회시험
	서버 응답속도	초	- (-/-)	-	-	5≥	3≥	공인기관 입회시험
	AI 알고리즘 연동 서비스 개수	개	- (-/-)	-	-	1	1	공인기관 입회시험
	농기계 관리지표 개수	개	- (-/-)	-	-	3	5	공인기관 입회시험

상기 정량적 목표들은 아래와 같은 시험을 통하여 공인기관 입회하에 진행되어, 이를 확인하였다.

## 1. 개요

이 시험결과는 (주)긴트에서 진행한 “블록체인 기술기반의 농업기계 관리기술 개발” 연구 과제를 의뢰자제시규격에 따라 시험한 시험결과임

## 2. 시험항목

- 1) 단말기 통신 성공률
- 2) 데이터 수집/농업인/제조사/정비지원 서버 수용 가입자
- 3) 데이터 수집/농업인/제조사/정비지원 서버 평균 응답 속도
- 4) 블록체인 네트워크 초당 트랜잭션 수
- 5) 블록체인 프록시 초당 업로드 속도
- 6) 분산원장 기여 데이터 가짓수
- 7) 각 서비스별 AI 알고리즘 연동 서비스 개수
- 8) 정부지원 서비스 농기계 관리 지표 개수

### 3. 시험대상

No.	명 칭	분류	비 고	
			식별자	해쉬값 (SHA256)
1	블록체인 기반 텔레매틱스 데이터 수집 서버 (Gint Connect Server v1.1)	소프트웨어	GC-LD (AWS Lambda Function)	1b4c1214560859988956553b08b88cc 8bfee72d4527b7ef8afb57bfd2b2b9b 04
			GC-DB (AWS DynamoDB)	fda685f4814a5a8c70158ddba5c7aac 38b8378c7e279d4574a90bf36aa52d6 72
2	블록체인 기반 텔레매틱스 게이트웨이 단말기 (Gint Connect Gateway v1.0)	하드웨어	GC-GW	
3	블록체인 기반 텔레매틱스 게이트웨이 애플리케이션 (Gint Connect Gateway App v1.0)	소프트웨어	GC-GW-APP	54273a3d259e4c1b8136c584c702de8 835a64ca8f1313ecc71fe0d285e5f48 fd

### 4. 시험결과

#### 1) 단말기 통신 성공률

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 텔레매틱스 서버에 더미 데이터를 전송하도록 만들어진 텔레매틱스 단말기를 통해 데이터를 일정 시간동안 업로드</li> <li>- 텔레매틱스 단말기는 0 부터 10000 개의 데이터를 송신함을 확인</li> <li>- 텔레매틱스 서버에 0부터 10000 개의 데이터를 수신함을 확인</li> <li>- 수신율이 정량적 목표(95% 이상)를 만족하면 시험 통과</li> </ul>	적합	※붙임1 참고

#### 2) 데이터 수집/농업인/제조사/정비지원 서버 수용 가입자

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 텔레매틱스 서버에 더미 데이터를 전송하도록 만들어진 가상 단말기 소프트웨어(nGrinder Script)를 노드 1000대에 배포</li> <li>- 가상 단말기 100,000대(노드 1개 당 100대)를 서버에 접속시켜 데이터 전송 시작</li> <li>- 각 단말기가 서버로부터 응답받는 데 걸린 시간(Round Trip Time)을 기록</li> <li>- 1시간 동안 시험을 진행하고 데이터를 취합하여 평균 응답 속도를 계산</li> <li>- 평균 응답 속도가 정량적 목표(1초)를 만족하면 시험 통과</li> </ul>	적합	※붙임2 참고

3) 데이터 수집/농업인/제조사/정부지원 서비스 서버 평균 응답 속도

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 가상 단말기 100,000대를 서버에 접속시켜 데이터 전송 시작</li> <li>- 평가 대상 가상 단말기를 지정하여 서버로부터 응답받는 데 걸린 시간을 기록</li> <li>- 10,000회 데이터를 전송하고 평균 응답 속도를 계산</li> <li>- 평균 응답 속도가 정량적 목표(3초)를 만족하면 시험 통과</li> </ul>	적합	※붙임3 참고

4) 블록체인 네트워크 초당 트랜잭션 수

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 텔레매틱스 애플리케이션에서 서버에 있는 블록체인 네트워크에 가상의 트랜잭션 연속으로 발생</li> <li>- 1초간 발생한 트랜잭션의 수가 정량적 목표(500개 이상)를 만족하면 시험 통과</li> </ul>	적합	※붙임4 참고

5) 블록체인 프록시 초당 업로드 속도

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 텔레매틱스 애플리케이션에서 서버에 있는 블록체인 네트워크에 가상의 업로드를 연속으로 발생</li> <li>- 1초간 발생한 업로드 수가 정량적 목표(2000개 이상)를 만족하면 시험 통과</li> </ul>	적합	※붙임5 참고

6) 분산원장 기여 데이터 가짓수

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 시험 PC에 더미 데이터 발생용 시뮬레이터를 설치하여 분산원장에 기여 할 데이터인 Engine Speed와 Engine Torque Percent 외 2종 총 6종이상 데이터 수집 서버에 업로드</li> <li>- 데이터가 서버에 업로드되는 것을 확인하면 데이터 수집 서버에서 정량적 목표(6개) 이상의 가짓수의 데이터를 수집한 것이므로 시험 통과</li> </ul>	적합	※붙임6 참고

7) 각 서비스별 AI 알고리즘 연동 서비스 개수

시 험 기 준 및 방 법	결과	비고
<ul style="list-style-type: none"> <li>- 텔레매틱스 어플리케이션에서 연동 서비스 개수 확인</li> <li>- 연동 서비스의 개수가 정량적 목표(3개)를 만족하면 시험 통과</li> </ul>	적합	※붙임7 참고

8) 정부지원 서비스 농기계 관리 지표 개수

시 험 기 준 및 방 법	결 과	비 고
- 텔레매틱스 어플리케이션에서 농기계 관리 지표의 개수 확인 - 농기계 관리지표의 개수가 정량적 목표(5개)를 만족하면 시험 통과	적합	※붙임8 참고

5. 시험환경

o 시험환경 구성1 (시험 1)

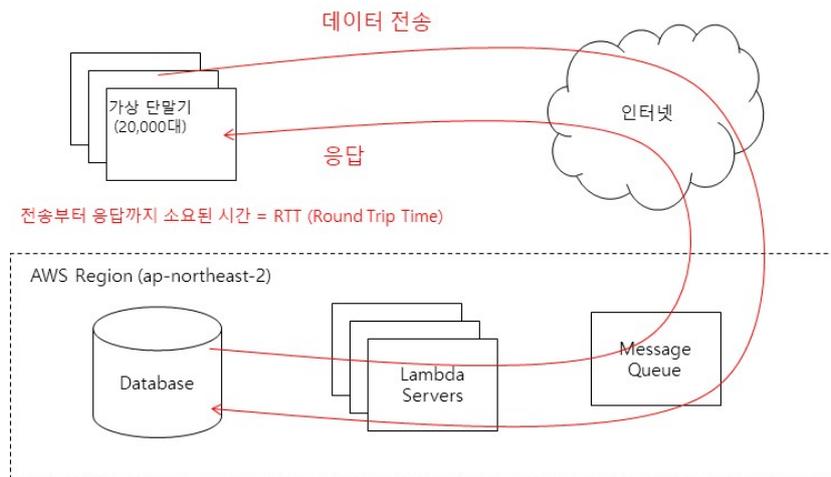
- ※ 텔레매틱스 게이트웨이 단말기에서 AWS IoT Core를 이용하여 AWS 서비스 까지 데이터를 0부터 1000까지 1초단위로 KeepAlive 신호를 보낸다.



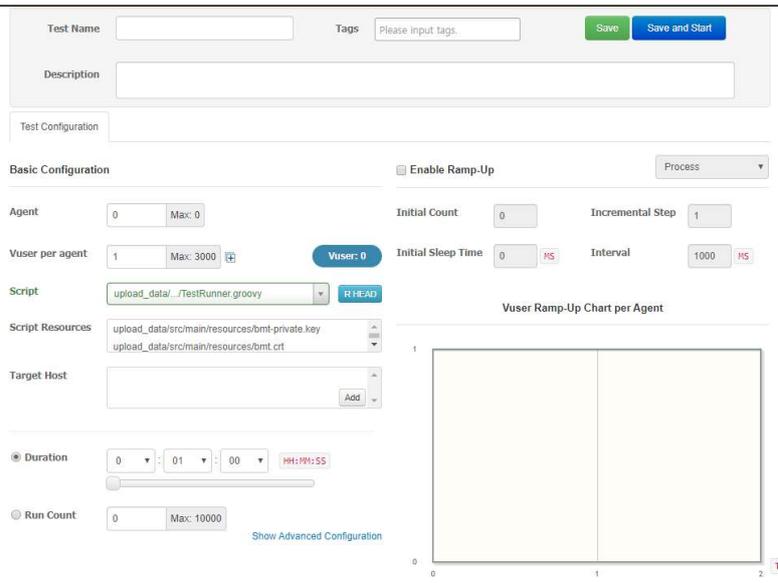
- ※ 서버에서 이를 순서적으로 빠짐이나 틀림없이 수신하였는지 로그를 통하여 확인한다.

o 시험환경 구성2 (시험 2 ~ 3)

- ※ AWS EC2의 AP-2 리전에 인스턴스를 20대 생성하여 nGrinder Agent(부하 생성기)를 설치
- ※ AWS EC2의 AP-2 리전에 인스턴스를 1대 생성하여 nGrinder Controller(Web Console) 설치

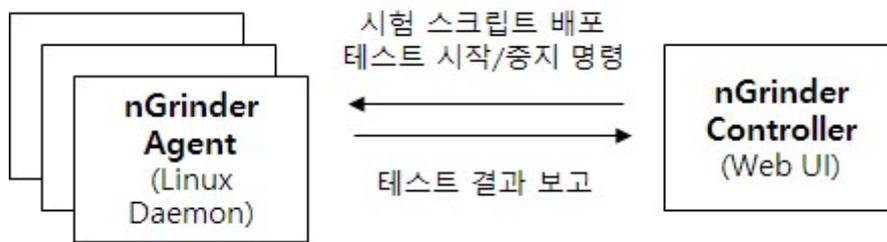


nGrinder 라는 오픈 소스 성능 테스트 툴을 이용하여 시험 환경을 구성한다.



<nGrinder 시험 준비 화면>

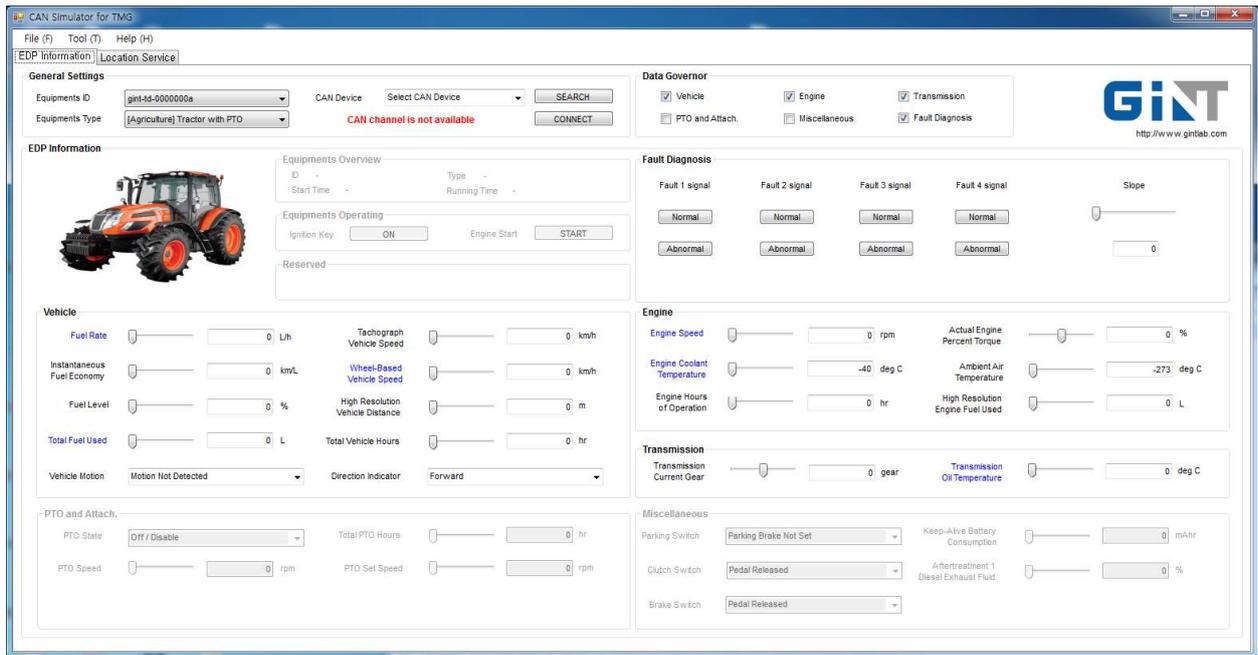
- nGrinder는 지정된 개수의 가상 단말기(VUser)를 스레드로 구현하여 분산된 시험 장비(시험PC-2)에 배포하여 실행하고 결과를 수집한다.
- nGrinder는 시험 수행 결과로 MTT(Mean Test Time)를 출력하며, 이는 측정하고자 하는 값인 RTT(Round Trip Time)를 포함한다. (MTT>=RTT가 언제나 성립) 따라서 MTT가 시험 기준을 만족하면 RTT 또한 시험 결과를 만족했다고 볼 수 있다.
- nGrinder는 controller와 agent 두 가지 모듈로 구성되어 있다. controller는 사용자의 입력을 받아 시험을 수행하고 결과를 출력하는 web UI로 구성되어 있고, agent는 controller로부터 명령을 받아 시험을 직접 수행한다.
- nGrinder는 무료이며 Apache 2.0 라이선스를 따르기 때문에 조건 없이 사용/재배포가 가능하다.



<nGrinder 부하 생성기 구조>

< 시스템 사양 >

구분	시스템 사양	운영SW 및 시험도구	구현 모듈
시험PC-1 (컨트롤러)	<ul style="list-style-type: none"> <li>• CPU : 16 vCore</li> <li>• RAM : 32 GB</li> <li>• HDD : EBS</li> <li>• OS : CentOS 7.4</li> </ul>	<ul style="list-style-type: none"> <li>• nGrinder v3.4.2</li> <li>• openjdk 1.8.0</li> </ul>	<ul style="list-style-type: none"> <li>• test script (groovy)</li> </ul>
시험PC-2 (부하 생성기)	<ul style="list-style-type: none"> <li>• CPU : 16 vCore</li> <li>• RAM : 32 GB</li> <li>• HDD : EBS</li> <li>• OS : CentOS 7.4</li> </ul>	<ul style="list-style-type: none"> <li>• nGrinder v3.4.2</li> <li>• openjdk 1.8.0</li> </ul>	<ul style="list-style-type: none"> <li>• test script (groovy)</li> </ul>
시험PC-3 (더미 데이터 발생 시뮬레이터)	<ul style="list-style-type: none"> <li>• CPU : 8 core</li> <li>• RAM : 8 GB</li> <li>• HDD : 500 GB</li> <li>• OS : Windows 10</li> </ul>	<ul style="list-style-type: none"> <li>• 시뮬레이터(C#)</li> <li>• Windows 10</li> </ul>	<ul style="list-style-type: none"> <li>• 더미 데이터 발생 시뮬레이터</li> </ul>



<더미 데이터 발생 시뮬레이터(시험PC-3) 준비 화면>

## 6. 시험 수행

- 텔레메틱스 단말기를 이용하여 다음과 통신 성공률 시험을 수행한다.

### 시험 항목 1. 단말기 통신 성공률

- ① 텔레메틱스 단말기가 KeepAlive 신호를 1초 단위로 0 ~ 10000 개 서버로 송신한다.
- ② 서버에서는 수신된 KeepAlive 신호의 로그를 확인한다.
- ③ 송신수와 수신수를 비교하여 통신성공률(%)을 분석한다. 95%가 넘으면 정량적 목표를 만족하여 시험 성공으로 본다.

- nGrinder를 이용하여 다음과 같이 부하 시험을 수행한다.

### 시험 항목 2. 데이터 수집/농업인/제조사/정비지원 서버 수용 가입자

- ① AWS EC2 서비스를 이용하여 ap-northeast-2 region에 nGrinder controller를 1대, agent를 20대 배포한다.
- ② nGrinder에 시험 스크립트를 등록하고 실행한다. 실행 설정은 다음과 같다.
  - Agent: 20
  - VUser per agent: 1000
  - Duration: 1h
- ③ 시험 종료 후 결과값(MTT)을 분석한다. 평균 MTT가 1초를 넘지 않으면 정량 목표를 만족하여 시험 성공으로 본다.

### 시험 항목 3. 데이터 수집/농업인/제조사/정비지원 서버 평균 응답 속도

- ① AWS EC2 서비스를 이용하여 ap-northeast-2 region에 nGrinder controller를 1대, agent를 51대 배포한다.
- ② nGrinder에 시험 스크립트를 등록하고 실행한다. 실행 설정은 다음과 같다.
  - Agent: 20
  - VUser per agent: 1000
  - Duration: 2h
- ③ 측정 대상 가상 단말기를 생성하여 이어서 실행한다. 실행 설정은 다음과 같다.
  - Agent: 1

---

- VUser per agent: 1

- Run Count: 10,000

- ④ 시험 종료 후 측정 대상 가상 단말기의 결과값(MTT)을 분석한다. 평균 MTT가 1초를 넘지 않으면 정량 목표를 만족하여 시험 성공으로 본다.

시험 항목 4. 블록체인 네트워크 초당 트랜잭션 수

- ① 텔레매틱스 애플리케이션에서 서버에 있는 블록체인 네트워크에 가상의 트랜잭션 연속으로 발생
- ② 블록체인 모니터링 시스템에서 초당 트랜잭션 수 확인
- ③ 1초간 발생한 트랜잭션의 수가 정량적 목표(500개 이상)를 만족하면 시험 성공으로 본다.

시험 항목 5. 블록체인 프록시 초당 업로드 속도

- ① 텔레매틱스 애플리케이션에서 서버에 있는 블록체인 네트워크에 가상의 업로드를 연속으로 발생
- ② 블록체인 모니터링 시스템에서 초당 업로드 수 확인
- ③ 1초간 발생한 업로드 수가 정량적 목표(2000개 이상)를 만족하면 시험 성공으로 본다.

시험 항목 6. 분산원장 기여 데이터 가짓수

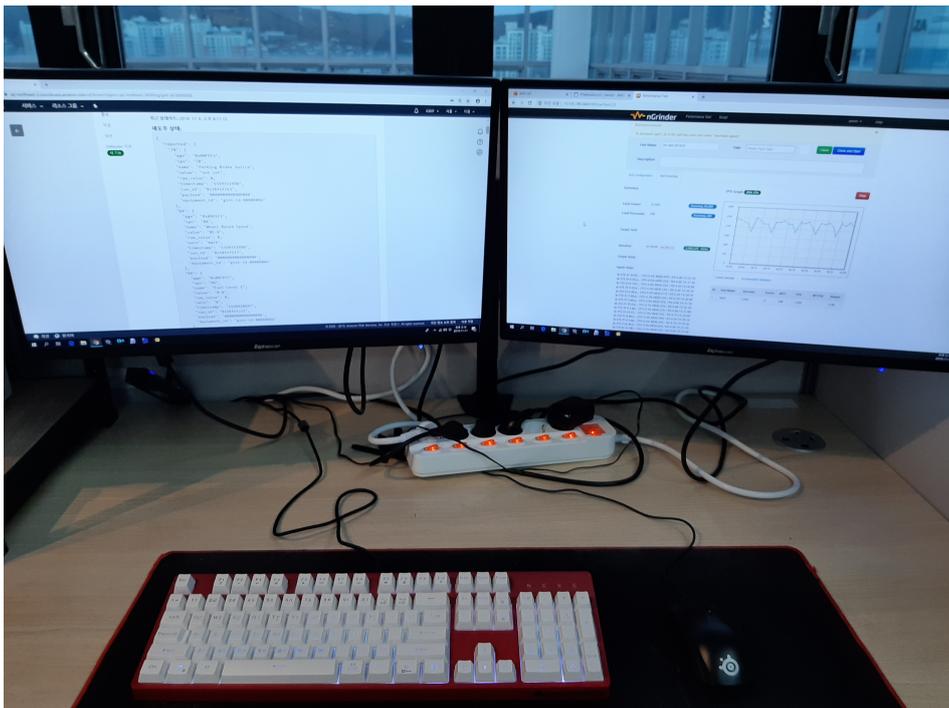
- ① 시험 PC에 더미 데이터 발생용 시뮬레이터를 설치하여 분산원장에 기여할 데이터인 Engine Speed 와 Engine Torque Percent 외 2종 총 6종 이상 데이터 수집 서버에 업로드
- ② 데이터가 서버에 업로드되는 것을 확인하면 데이터 수집 서버에서 정량적 목표(6개) 이상의 가짓수의 데이터를 수집한 것이므로 시험 성공으로 본다.

시험 항목 7. 각 서비스별 AI 알고리즘 연동 서비스 개수

- ① 텔레매틱스 애플리케이션에서 연동 서비스 개수 확인
- ② 연동 서비스의 개수가 정량적 목표(3개)를 만족하면 시험 성공으로 본다.

시험 항목 8. 정부지원 서비스 농기계 관리 지표 개수

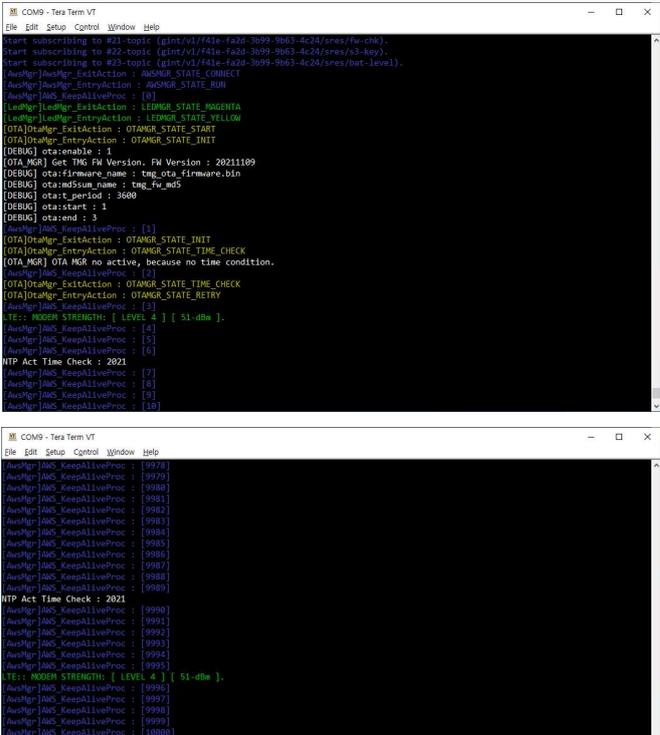
- ① 텔레매틱스 애플리케이션에서 농기계 관리 지표의 개수 확인
- ② 농기계 관리지표의 개수가 정량적 목표(5개)를 만족하면 시험 성공으로 본다.



<시험 항목 1 ~ 8 시험 사진>

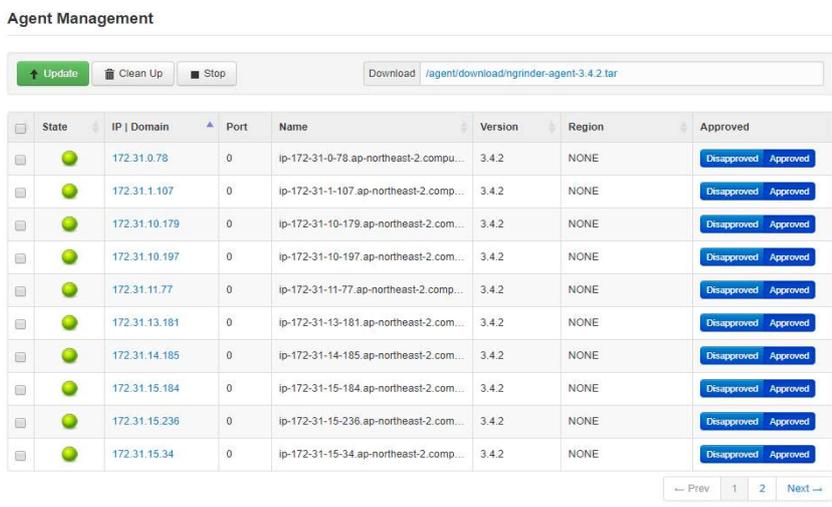
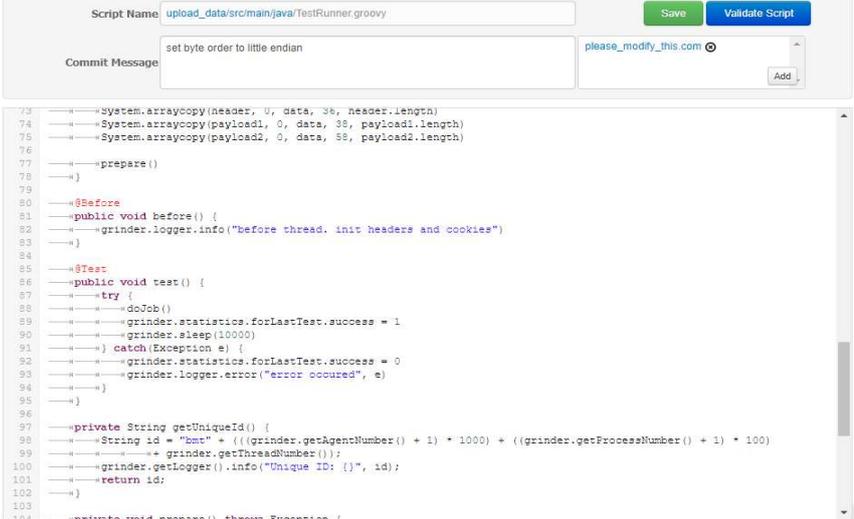
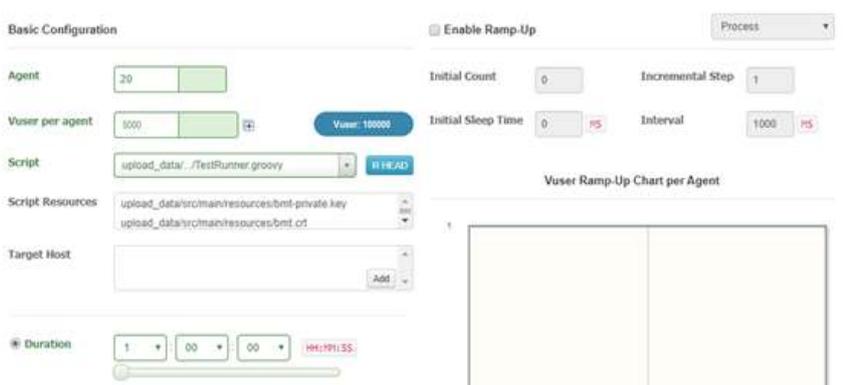
※붙임1. 단말기 통신 성공률

o 세부시험결과

단 계	시험 사진	비 고
<p>단말기에서 KeepAlive 신호 생성</p>		<p>총 1000개의 KeepAlive 신호를 생성한다.</p>
<p>업로드된 신호 확인</p>		<p>AWS 콘솔에서 업로드 여부를 확인한다.</p>
<p>결과</p>	<p>1000개 모두 수신 하였으므로 통신성공률 100%로 시험기준(95이상)을 만족함</p>	

※붙임2. 데이터 수집/농업인/제조사/정비지원 서버 수용 가입자

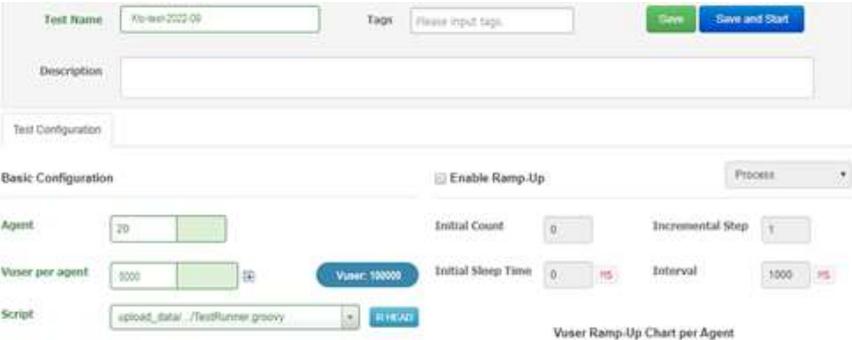
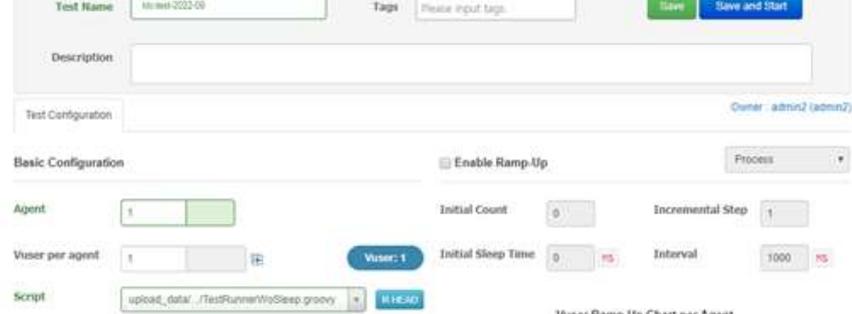
o 세부시험결과

단 계	시험 사진	비 고
테스트 에이전트 생성		총 20개의 테스트 에이전트를 생성한다.
테스트 스크립트 업로드		테스트를 위한 스크립트를 업로드한다.
Task & Thread 설정		User가 100,000이 되도록 20개의 Task와 5000개의 Thread를 설정한다.
단 계	시험 사진	비 고
테스트 시작		테스트를 시작하고 약 1시간정도 진행상황을 확인한다.

테스트 종료		테스트가 완료되면 테스트 결과를 CSV파일로 export시킨다.																																																																								
MTT 최대값 확인	<table border="1" data-bbox="355 943 1155 1384"> <tbody> <tr><td>100,000</td><td>4560</td><td>0</td><td>532.123</td><td>82.334</td><td>2152</td></tr> <tr><td>100,000</td><td>5315</td><td>0</td><td>489.655</td><td>86.844</td><td>1899</td></tr> <tr><td>100,000</td><td>4896</td><td>0</td><td>512.284</td><td>88.332</td><td>1344</td></tr> <tr><td>100,000</td><td>2331</td><td>0</td><td>511.223</td><td>84.212</td><td>2111</td></tr> <tr><td>100,000</td><td>8589</td><td>0</td><td>546.215</td><td>89.224</td><td>2354</td></tr> <tr><td>100,000</td><td>5531</td><td>0</td><td>612.234</td><td>85.331</td><td>1897</td></tr> <tr><td>100,000</td><td>2288</td><td>0</td><td>488.367</td><td>81.212</td><td>1356</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td>MAX</td><td>673.65</td><td></td><td></td></tr> <tr><td></td><td></td><td>MID</td><td>543.659</td><td></td><td></td></tr> <tr><td></td><td></td><td>AVG</td><td>533.98</td><td></td><td></td></tr> </tbody> </table>	100,000	4560	0	532.123	82.334	2152	100,000	5315	0	489.655	86.844	1899	100,000	4896	0	512.284	88.332	1344	100,000	2331	0	511.223	84.212	2111	100,000	8589	0	546.215	89.224	2354	100,000	5531	0	612.234	85.331	1897	100,000	2288	0	488.367	81.212	1356															MAX	673.65					MID	543.659					AVG	533.98			Mean Test Time의 최대값은 약 674 ms, 평균값은 약 534ms로 확인함.
100,000	4560	0	532.123	82.334	2152																																																																					
100,000	5315	0	489.655	86.844	1899																																																																					
100,000	4896	0	512.284	88.332	1344																																																																					
100,000	2331	0	511.223	84.212	2111																																																																					
100,000	8589	0	546.215	89.224	2354																																																																					
100,000	5531	0	612.234	85.331	1897																																																																					
100,000	2288	0	488.367	81.212	1356																																																																					
		MAX	673.65																																																																							
		MID	543.659																																																																							
		AVG	533.98																																																																							
결과	Mean Test Time의 최대값은 약 674 ms, 평균값은 약 534ms로 확인되어 1초 이하의 시험기준을 만족함																																																																									

※붙임3. 데이터 수집/농업인/제조사/정비지원 서버 평균 응답 속도

o 세부시험결과

단 계	시험 사진	비 고
테스트 에이전트 시작		1번 시험항목과 동일하게 조건을 설정하고 100,000 개의 Vuser 를 실행한다.
추가 테스트 에이전트 생성		100,000 개의 Vuser 가 실행된 조건에서 추가적인 에이전트를 생성한다.
테스트 스크립트 업로드	<pre data-bbox="316 1263 1168 1554"> 97  private String getUniqueId() { 98  String id = "bmc" + ((grinder.getAgentsNumber() + 1) * 1000) + ((grinder.getProcessNumber() + 1) * 100) 99  grinder.grinder.getThreadNumber(); 100 grinder.getLogger().info("Unique ID: {}", id); 101 return id; 102 } 103 104 private void prepare() throws Exception { 105 //data.equipment_id = getUniqueId() 106 KeyStore ks = KeyStore.getInstance(KeyStore.getDefaultType()) 107 ks.load(new FileInputStream(keyStorePath), keyStorePassword.toCharArray()) 108 109 AmazonMQtoClient client = new AmazonMQtoClient(endpoint, getUniqueId(), ks, keyPassword) 110 AmazonMQtoDevice device = new AmazonMQtoDevice(getUniqueId()) 111 client.attach(device) 112 client.connect() 113 clientMap.put(getUniqueId(), client) 114 } 115                     </pre>	테스트를 위한 스크립트를 업로드한다.

단 계	시험 사진	비 고
테스트 시작		서버에 데이터를 10,000회 업로드 하도록 설정하여 테스트를 실행한다.

테스트 종료		테스트가 완료되면 테스트 결과를 CSV파일로 export 시킨다.																																			
MTT 최대값 확인	<table border="1"> <tr> <td>I</td> <td>Q</td> <td>U</td> <td>509.23</td> <td>42.705</td> </tr> <tr> <td>1</td> <td>3</td> <td>0</td> <td>561.667</td> <td>84.909</td> </tr> <tr> <td>1</td> <td>4</td> <td>0</td> <td>526</td> <td>57.892</td> </tr> <tr> <td>1</td> <td>2</td> <td>0</td> <td>633</td> <td>82</td> </tr> <tr> <td>1</td> <td>4</td> <td>0</td> <td>477.25</td> <td>80.976</td> </tr> <tr> <td>0</td> <td>3</td> <td>0</td> <td>544.333</td> <td>140.794</td> </tr> <tr> <td></td> <td></td> <td>AVG</td> <td>840.5592</td> <td></td> </tr> </table>	I	Q	U	509.23	42.705	1	3	0	561.667	84.909	1	4	0	526	57.892	1	2	0	633	82	1	4	0	477.25	80.976	0	3	0	544.333	140.794			AVG	840.5592		Mean Test Time의 평균값은 약 840ms로 확인함.
I	Q	U	509.23	42.705																																	
1	3	0	561.667	84.909																																	
1	4	0	526	57.892																																	
1	2	0	633	82																																	
1	4	0	477.25	80.976																																	
0	3	0	544.333	140.794																																	
		AVG	840.5592																																		
결과	Mean Test Time의 평균값은 약 840ms로 확인되어 3초 이하의 시험기준을 만족함																																				

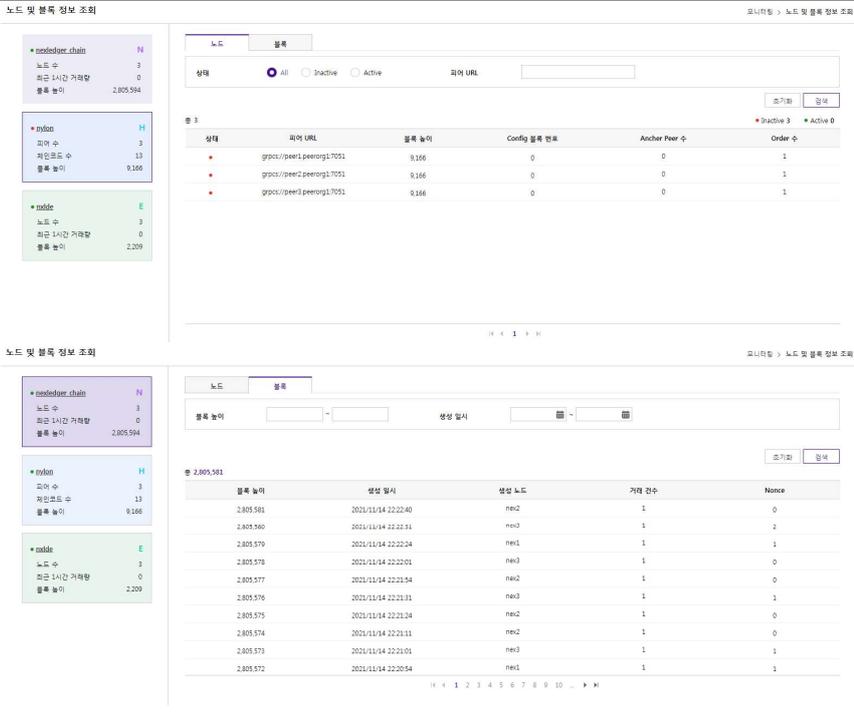
※붙임4. 블록체인 네트워크 초당 트랜잭션 수

o 세부시험결과

단 계	시험 사진	비 고																																
블록 체인 모니터링 시스템에서 관련 자료 확인	<p>The screenshots show a monitoring interface with filters for '거래 일시' (Transaction Time), '제한' (Limit), and '연관 구분' (Related Category). The first two screenshots show a list of transactions for 'neledger_chain' with columns for '거래 일시(전)', '블록 높이' (Block Height), '노드 이름' (Node Name), '계좌' (Account), and '거래 ID' (Transaction ID). The third screenshot shows transactions for 'nside' with columns for '거래 일시(전)', '블록 번호' (Block Number), '계좌' (Account), '거래 ID' (Transaction ID), and 'Gas'.</p>	초당 트랜잭션 수를 확인한다.																																
초당 TPS 확인	<table border="1" style="width: 100%; text-align: center;"> <tr><td>1</td><td>4</td><td>0</td><td>2000</td></tr> <tr><td>1</td><td>2</td><td>0</td><td>1800</td></tr> <tr><td>0</td><td>3</td><td>1</td><td>664</td></tr> <tr><td>1</td><td>3</td><td>0</td><td>212</td></tr> <tr><td>1</td><td>3</td><td>0</td><td>1532</td></tr> <tr><td>1</td><td>3</td><td>0</td><td>855</td></tr> <tr><td></td><td>3</td><td>0</td><td>468</td></tr> <tr><td colspan="3">AVG</td><td>1075.857</td></tr> </table>	1	4	0	2000	1	2	0	1800	0	3	1	664	1	3	0	212	1	3	0	1532	1	3	0	855		3	0	468	AVG			1075.857	TPS 는 약 1075 개로 확인함
1	4	0	2000																															
1	2	0	1800																															
0	3	1	664																															
1	3	0	212																															
1	3	0	1532																															
1	3	0	855																															
	3	0	468																															
AVG			1075.857																															
결과	TPS가 정량적 목표(500개 이상)을 만족함																																	

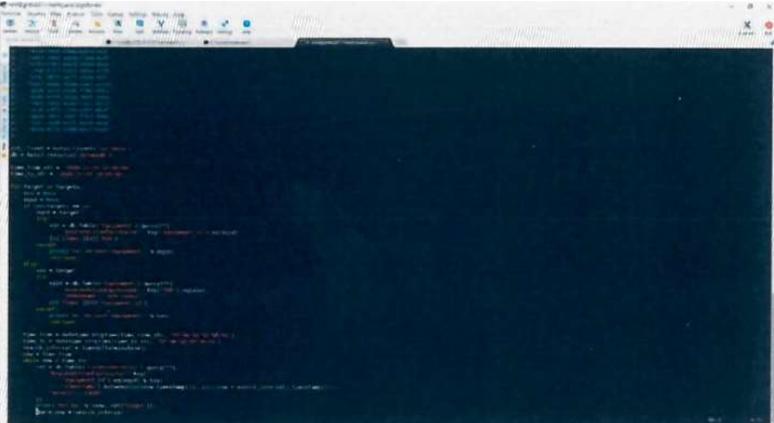
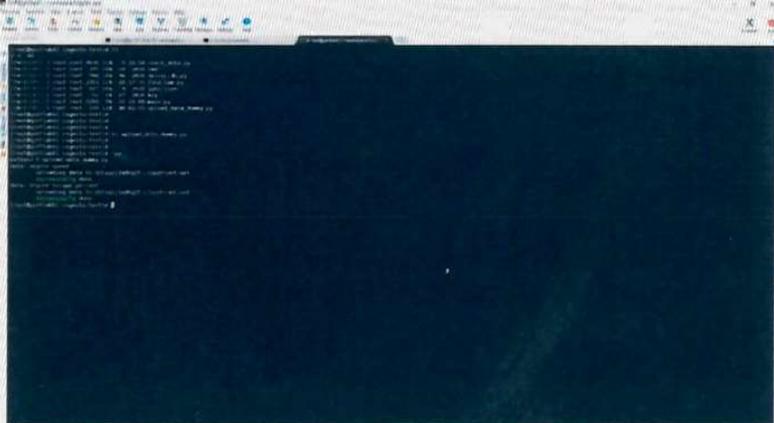
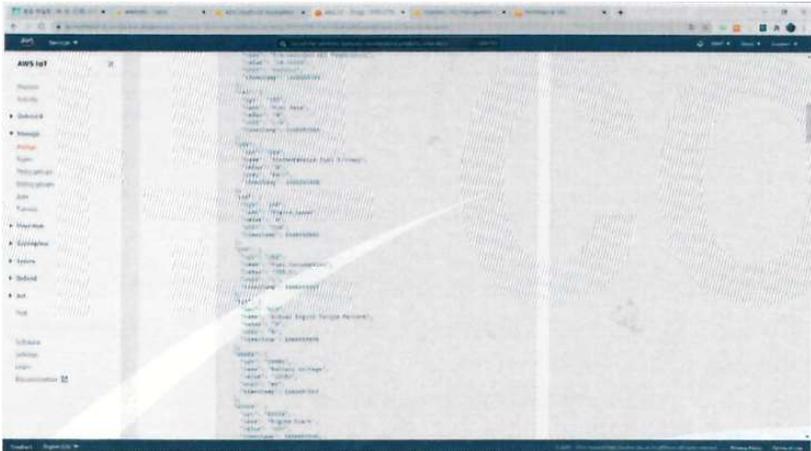
※붙임5. 블록체인 네트워크 초당 업로드 속도

o 세부시험결과

단 계	시험 사진	비 고																																							
블록체인 모니터링 시스템에서 관련 자료 확인		초당 업로드 수를 확인한다.																																							
초당 업로드 수를 확인	<table border="1" data-bbox="336 1108 1190 1574"> <tbody> <tr><td>f1afd43cae4ab8892ae0cef69262e689f1c9689</td><td>0</td><td>2111</td></tr> <tr><td>:03b6ba96121472b2292393b56afa52049a5c6e</td><td>0</td><td>1958</td></tr> <tr><td>f1854aadec4bea98738788f02420366a3e566</td><td>1</td><td>1580</td></tr> <tr><td>l75fdf0522018e01ecb540e3858d451dc60a055f</td><td>1</td><td>1423</td></tr> <tr><td>362b1195db59167adf5d5fa5fa0e059b250e52</td><td>1</td><td>2500</td></tr> <tr><td>f3c56ffb93cf196b7e4b9db7286daf1490766d</td><td>0</td><td>1333</td></tr> <tr><td>8aaf1868fc683d14932aec0f9198a1ebd205c</td><td>1</td><td>3232</td></tr> <tr><td>630e9f9a78c97209def6cd8daba2d133d920c9</td><td>1</td><td>2231</td></tr> <tr><td>ae7d3decdd641e918a3280c5928aae13db37a7</td><td>1</td><td>2266</td></tr> <tr><td>i81077d28342c60ea7497ca4963ca0a768b7bc2</td><td>0</td><td>2131</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>AVG</td><td>2076.5</td></tr> </tbody> </table>	f1afd43cae4ab8892ae0cef69262e689f1c9689	0	2111	:03b6ba96121472b2292393b56afa52049a5c6e	0	1958	f1854aadec4bea98738788f02420366a3e566	1	1580	l75fdf0522018e01ecb540e3858d451dc60a055f	1	1423	362b1195db59167adf5d5fa5fa0e059b250e52	1	2500	f3c56ffb93cf196b7e4b9db7286daf1490766d	0	1333	8aaf1868fc683d14932aec0f9198a1ebd205c	1	3232	630e9f9a78c97209def6cd8daba2d133d920c9	1	2231	ae7d3decdd641e918a3280c5928aae13db37a7	1	2266	i81077d28342c60ea7497ca4963ca0a768b7bc2	0	2131								AVG	2076.5	초당업로드수는 약 2076개로 확인함
f1afd43cae4ab8892ae0cef69262e689f1c9689	0	2111																																							
:03b6ba96121472b2292393b56afa52049a5c6e	0	1958																																							
f1854aadec4bea98738788f02420366a3e566	1	1580																																							
l75fdf0522018e01ecb540e3858d451dc60a055f	1	1423																																							
362b1195db59167adf5d5fa5fa0e059b250e52	1	2500																																							
f3c56ffb93cf196b7e4b9db7286daf1490766d	0	1333																																							
8aaf1868fc683d14932aec0f9198a1ebd205c	1	3232																																							
630e9f9a78c97209def6cd8daba2d133d920c9	1	2231																																							
ae7d3decdd641e918a3280c5928aae13db37a7	1	2266																																							
i81077d28342c60ea7497ca4963ca0a768b7bc2	0	2131																																							
	AVG	2076.5																																							
결과	업로드 수가 정량적 목표(2000개 이상)을 만족함																																								

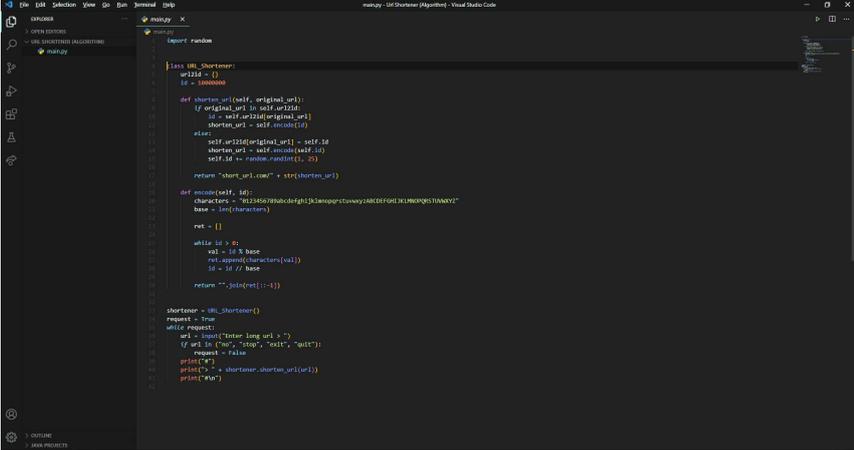
※붙임6. 분산원장 기여 데이터 가짓수

o 세부시험결과

단 계	시험 사진	비 고
분산원장 기여 데이터 가짓수 시험 준비		<p>더미 데이터 (Engine Speed, Engine Torque Percent 외 2종을 업로드할 python 스크립트를 준비한다.</p>
분산원장 기여 데이터 모사 더미 데이터 업로드		<p>스크립트를 실행하여 준비된 데이터를 서버에 업로드 한다.</p>
데이터 수집 서버에서 업로드된 데이터 확인		<p>데이터 수집 서버에 접근하여 데이터가 올바르게 업로드 되었는지 확인한다.</p>
결과	6 가지 데이터 모두 업로드 되었으므로 시험 기준(6개 이상)을 만족함	

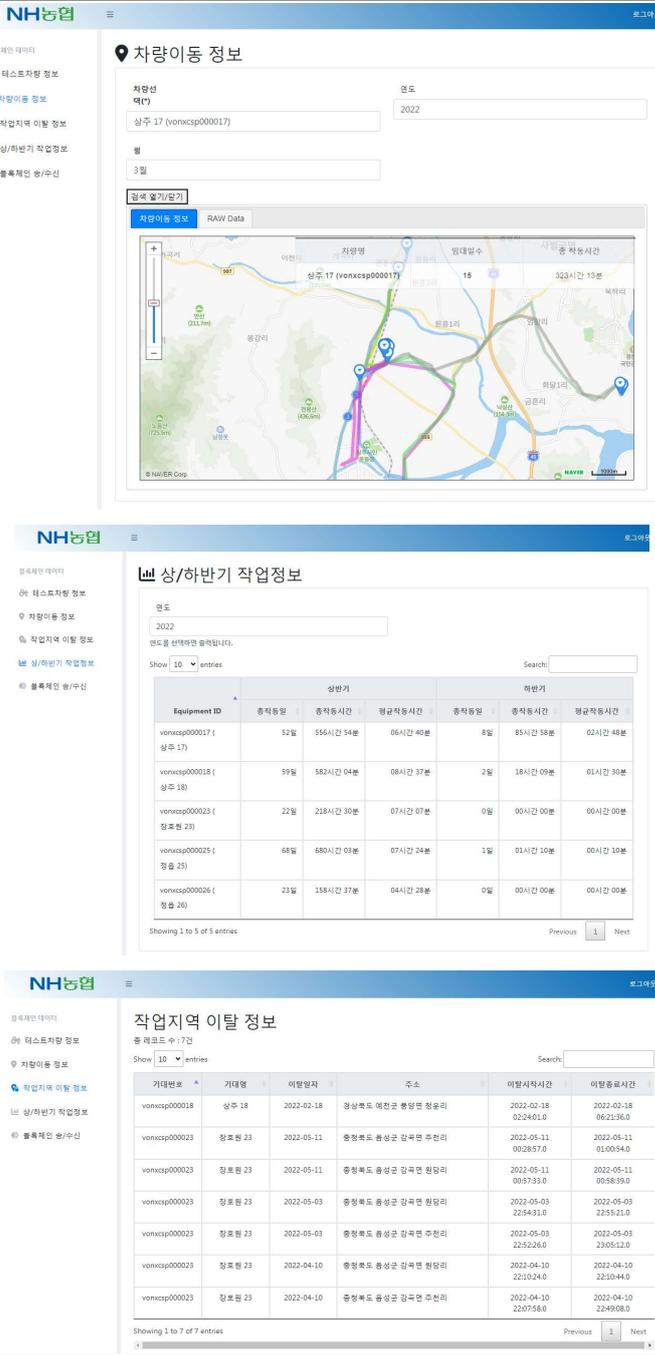
※붙임7. 각 서비스별 SI 알고리즘 연동 서비스 개수

o 세부시험결과

단 계	시험 사진	비 고																																																																														
연동 서비스를 위한 스펙 확인	<pre> API 요청시 필요한 정보 URL : /operation url http://192.1.168.188:8000/predict_fuel_torque  Method : Get Params - key에 따라이러 이음, value에 따라이러 값  1. 트래커 기술기 값에 : tractor_warning (트래커의 기술을 실시간으로 수집하여 해당 트래커의 최근 위험 기록을 반환) Params - equipment(장치 아이디) 1. 장비 아이디 - lg300-protor-00000001 - lg300-protor-00000002 - lg300-protor-00000003 Result - type(operation 종류) - data(반대 날짜) - equipment(장치 아이디) - date_warning(경고 일자) - warning(경고 타입)  2. 트래커 기술기 값에 : tractor_monitoring (트래커의 상태변화를 통해 트래커의 기능을 감지) Params - equipment(장치 아이디) 1. 장비 아이디 - lg300-protor-00000001 - lg300-protor-00000002 - lg300-protor-00000003 Result - type(operation 종류) - data(반대 날짜) - equipment(장치 아이디) - monitoring(감지 타입)  3. 트래커 로그 기반 연도 소모량 예측 : predict_fuel_torque (트래커의 로그 기록을 이용하여 연도 소모량 예측) ● 입력받은 데이터를 해당에서 사용  Params - input(로그 기록 데이터) 1. Format - [날, 시간, 속도, 온도, ...]를 기준으로 데이터를 api로 요청한 도한 데이터에 " " 입력으로 요청 2. Result - type(operation 종류) - data(반대 날짜) - predicted_fuel(예측된 연도소모량) 1. Format - %d (소수점 2째 번째 자리까지 출력)  4. 트래커 작업량 기반 연도 소모량 예측 : predict_fuel_operation (트래커의 정해진 시간동안 작업량 기록을 이용하여 연도 소모량 예측) Params - fuel(작업량 단위) : Second - start(시작 시간) : Second 2. Result - type(operation 종류) - data(반대 날짜) - predicted_fuel(예측된 연도소모량) 1. Format - %d (소수점 2째 번째 자리까지 출력) </pre>	Restful API 형식의 연동 서비스 스펙을 문서로 확인																																																																														
연동 서비스를 사용		스크립트를 실행하여 데이터를 서버와 연동 확인.																																																																														
데이터 수집 서버에서 다운로드된 데이터 확인	<table border="1"> <thead> <tr> <th>Name</th> <th>CAN ID</th> <th>SPN</th> <th>Payload</th> <th>Value</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td>Engine Coolant Temperature</td> <td>18feee00</td> <td>110</td> <td>423eb025f</td> <td>26</td> <td>Celsius</td> </tr> <tr> <td>Voltage</td> <td>18fef700</td> <td>168</td> <td>ffffff1801</td> <td>14</td> <td>V</td> </tr> <tr> <td>Fuel Temperature</td> <td>18feee00</td> <td>174</td> <td>423eb025f</td> <td>22</td> <td>Celsius</td> </tr> <tr> <td>Engine Throttle Position</td> <td>18fef200</td> <td>51</td> <td>730000000</td> <td>31.6</td> <td>%</td> </tr> <tr> <td>Wheel Based Speed</td> <td>18fef113</td> <td>84</td> <td>8E+13</td> <td>1.5</td> <td>km/h</td> </tr> <tr> <td>Engine Coolant Temperature</td> <td>18feee00</td> <td>110</td> <td>433ec825f</td> <td>27</td> <td>Celsius</td> </tr> <tr> <td>Voltage</td> <td>18fef700</td> <td>168</td> <td>ffffff1901</td> <td>14.05</td> <td>V</td> </tr> <tr> <td>Fuel Temperature</td> <td>18feee00</td> <td>174</td> <td>433ec825f</td> <td>22</td> <td>Celsius</td> </tr> <tr> <td>Fuel Rate</td> <td>18fef200</td> <td>183</td> <td>730000000</td> <td>5.75</td> <td>L/h</td> </tr> <tr> <td>Instantaneous Fuel Economy</td> <td>18fef200</td> <td>184</td> <td>730000000</td> <td>0</td> <td>km/L</td> </tr> <tr> <td>Engine Speed</td> <td>cf00400</td> <td>190</td> <td>f1959f7029</td> <td>1326</td> <td>rpm</td> </tr> <tr> <td>Actual Engine Torque Percent</td> <td>cf00400</td> <td>513</td> <td>f1959f7029</td> <td>34</td> <td>%</td> </tr> </tbody> </table>	Name	CAN ID	SPN	Payload	Value	Unit	Engine Coolant Temperature	18feee00	110	423eb025f	26	Celsius	Voltage	18fef700	168	ffffff1801	14	V	Fuel Temperature	18feee00	174	423eb025f	22	Celsius	Engine Throttle Position	18fef200	51	730000000	31.6	%	Wheel Based Speed	18fef113	84	8E+13	1.5	km/h	Engine Coolant Temperature	18feee00	110	433ec825f	27	Celsius	Voltage	18fef700	168	ffffff1901	14.05	V	Fuel Temperature	18feee00	174	433ec825f	22	Celsius	Fuel Rate	18fef200	183	730000000	5.75	L/h	Instantaneous Fuel Economy	18fef200	184	730000000	0	km/L	Engine Speed	cf00400	190	f1959f7029	1326	rpm	Actual Engine Torque Percent	cf00400	513	f1959f7029	34	%	데이터 수집 서버에 접근하여 데이터가 올바르게 다운로드되었는지 확인한다.
Name	CAN ID	SPN	Payload	Value	Unit																																																																											
Engine Coolant Temperature	18feee00	110	423eb025f	26	Celsius																																																																											
Voltage	18fef700	168	ffffff1801	14	V																																																																											
Fuel Temperature	18feee00	174	423eb025f	22	Celsius																																																																											
Engine Throttle Position	18fef200	51	730000000	31.6	%																																																																											
Wheel Based Speed	18fef113	84	8E+13	1.5	km/h																																																																											
Engine Coolant Temperature	18feee00	110	433ec825f	27	Celsius																																																																											
Voltage	18fef700	168	ffffff1901	14.05	V																																																																											
Fuel Temperature	18feee00	174	433ec825f	22	Celsius																																																																											
Fuel Rate	18fef200	183	730000000	5.75	L/h																																																																											
Instantaneous Fuel Economy	18fef200	184	730000000	0	km/L																																																																											
Engine Speed	cf00400	190	f1959f7029	1326	rpm																																																																											
Actual Engine Torque Percent	cf00400	513	f1959f7029	34	%																																																																											
결과	연동 서비스의 개수가 시험 기준(3개 이상)을 만족함																																																																															

※붙임8. 정부지원 서비스 농기계 관리 지표 개수

o 세부시험결과

단 계	시험 사진	비 고																																																																																										
<p>관리지표를 서비스 페이지에서 확인</p>	 <p><b>차량이동 정보</b></p> <p>차량선역(차) : 상주 17 (vonncsp000017)   연도 : 2022</p> <p><b>상/하반기 작업정보</b></p> <table border="1"> <thead> <tr> <th>Equipment ID</th> <th>총작동일</th> <th>총작동시간</th> <th>평균작동시간</th> <th>총작동일</th> <th>총작동시간</th> <th>평균작동시간</th> </tr> </thead> <tbody> <tr> <td>vonncsp000017 (상주 17)</td> <td>52일</td> <td>556시간 54분</td> <td>06시간 40분</td> <td>8월</td> <td>85시간 58분</td> <td>02시간 46분</td> </tr> <tr> <td>vonncsp000018 (상주 18)</td> <td>59일</td> <td>582시간 04분</td> <td>08시간 37분</td> <td>7월</td> <td>18시간 09분</td> <td>01시간 30분</td> </tr> <tr> <td>vonncsp000023 (정포원 23)</td> <td>22일</td> <td>218시간 30분</td> <td>07시간 07분</td> <td>9월</td> <td>00시간 00분</td> <td>00시간 00분</td> </tr> <tr> <td>vonncsp000025 (정읍 25)</td> <td>68일</td> <td>680시간 03분</td> <td>07시간 24분</td> <td>1월</td> <td>01시간 30분</td> <td>00시간 10분</td> </tr> <tr> <td>vonncsp000026 (정읍 26)</td> <td>23일</td> <td>158시간 37분</td> <td>04시간 28분</td> <td>9월</td> <td>00시간 00분</td> <td>00시간 00분</td> </tr> </tbody> </table> <p><b>작업지역 이탈 정보</b></p> <table border="1"> <thead> <tr> <th>기대번호</th> <th>기대명</th> <th>이탈일자</th> <th>주소</th> <th>이탈시작시간</th> <th>이탈종료시간</th> </tr> </thead> <tbody> <tr> <td>vonncsp000018</td> <td>상주 18</td> <td>2022-02-18</td> <td>경상북도 예천군 흥업면 청운리</td> <td>2022-02-18 02:24:01.0</td> <td>2022-02-18 06:21:36.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-05-11</td> <td>충청북도 음성군 장곡면 주현리</td> <td>2022-05-11 09:28:57.0</td> <td>2022-05-11 01:00:54.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-05-11</td> <td>충청북도 음성군 장곡면 현달리</td> <td>2022-05-11 09:57:33.0</td> <td>2022-05-11 09:58:39.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-05-03</td> <td>충청북도 음성군 장곡면 현달리</td> <td>2022-05-03 22:54:31.0</td> <td>2022-05-03 22:55:21.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-05-03</td> <td>충청북도 음성군 장곡면 주현리</td> <td>2022-05-03 22:52:26.0</td> <td>2022-05-03 23:05:12.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-04-10</td> <td>충청북도 음성군 장곡면 현달리</td> <td>2022-04-10 22:10:24.0</td> <td>2022-04-10 22:10:44.0</td> </tr> <tr> <td>vonncsp000023</td> <td>정포원 23</td> <td>2022-04-10</td> <td>충청북도 음성군 장곡면 주현리</td> <td>2022-04-10 22:07:58.0</td> <td>2022-04-10 22:49:08.0</td> </tr> </tbody> </table>	Equipment ID	총작동일	총작동시간	평균작동시간	총작동일	총작동시간	평균작동시간	vonncsp000017 (상주 17)	52일	556시간 54분	06시간 40분	8월	85시간 58분	02시간 46분	vonncsp000018 (상주 18)	59일	582시간 04분	08시간 37분	7월	18시간 09분	01시간 30분	vonncsp000023 (정포원 23)	22일	218시간 30분	07시간 07분	9월	00시간 00분	00시간 00분	vonncsp000025 (정읍 25)	68일	680시간 03분	07시간 24분	1월	01시간 30분	00시간 10분	vonncsp000026 (정읍 26)	23일	158시간 37분	04시간 28분	9월	00시간 00분	00시간 00분	기대번호	기대명	이탈일자	주소	이탈시작시간	이탈종료시간	vonncsp000018	상주 18	2022-02-18	경상북도 예천군 흥업면 청운리	2022-02-18 02:24:01.0	2022-02-18 06:21:36.0	vonncsp000023	정포원 23	2022-05-11	충청북도 음성군 장곡면 주현리	2022-05-11 09:28:57.0	2022-05-11 01:00:54.0	vonncsp000023	정포원 23	2022-05-11	충청북도 음성군 장곡면 현달리	2022-05-11 09:57:33.0	2022-05-11 09:58:39.0	vonncsp000023	정포원 23	2022-05-03	충청북도 음성군 장곡면 현달리	2022-05-03 22:54:31.0	2022-05-03 22:55:21.0	vonncsp000023	정포원 23	2022-05-03	충청북도 음성군 장곡면 주현리	2022-05-03 22:52:26.0	2022-05-03 23:05:12.0	vonncsp000023	정포원 23	2022-04-10	충청북도 음성군 장곡면 현달리	2022-04-10 22:10:24.0	2022-04-10 22:10:44.0	vonncsp000023	정포원 23	2022-04-10	충청북도 음성군 장곡면 주현리	2022-04-10 22:07:58.0	2022-04-10 22:49:08.0	<p>관리지표의 개수 확인</p>
Equipment ID	총작동일	총작동시간	평균작동시간	총작동일	총작동시간	평균작동시간																																																																																						
vonncsp000017 (상주 17)	52일	556시간 54분	06시간 40분	8월	85시간 58분	02시간 46분																																																																																						
vonncsp000018 (상주 18)	59일	582시간 04분	08시간 37분	7월	18시간 09분	01시간 30분																																																																																						
vonncsp000023 (정포원 23)	22일	218시간 30분	07시간 07분	9월	00시간 00분	00시간 00분																																																																																						
vonncsp000025 (정읍 25)	68일	680시간 03분	07시간 24분	1월	01시간 30분	00시간 10분																																																																																						
vonncsp000026 (정읍 26)	23일	158시간 37분	04시간 28분	9월	00시간 00분	00시간 00분																																																																																						
기대번호	기대명	이탈일자	주소	이탈시작시간	이탈종료시간																																																																																							
vonncsp000018	상주 18	2022-02-18	경상북도 예천군 흥업면 청운리	2022-02-18 02:24:01.0	2022-02-18 06:21:36.0																																																																																							
vonncsp000023	정포원 23	2022-05-11	충청북도 음성군 장곡면 주현리	2022-05-11 09:28:57.0	2022-05-11 01:00:54.0																																																																																							
vonncsp000023	정포원 23	2022-05-11	충청북도 음성군 장곡면 현달리	2022-05-11 09:57:33.0	2022-05-11 09:58:39.0																																																																																							
vonncsp000023	정포원 23	2022-05-03	충청북도 음성군 장곡면 현달리	2022-05-03 22:54:31.0	2022-05-03 22:55:21.0																																																																																							
vonncsp000023	정포원 23	2022-05-03	충청북도 음성군 장곡면 주현리	2022-05-03 22:52:26.0	2022-05-03 23:05:12.0																																																																																							
vonncsp000023	정포원 23	2022-04-10	충청북도 음성군 장곡면 현달리	2022-04-10 22:10:24.0	2022-04-10 22:10:44.0																																																																																							
vonncsp000023	정포원 23	2022-04-10	충청북도 음성군 장곡면 주현리	2022-04-10 22:07:58.0	2022-04-10 22:49:08.0																																																																																							
<p>결과</p>	<p>관리 지표의 개수가 시험 기준(3개 이상)을 만족함</p>																																																																																											

(3) 세부 정량적 연구개발성과(해당되는 항목만 선택하여 작성하되, 증빙자료를 별도 첨부해야 합니다)

[과학적 성과]

논문(국내외 전문 학술지) 게재

번호	논문명	학술지명	주저자명	호	국명	발행기관	SCIE 여부 (SCIE/비SCIE)	게재일	등록번호 (ISSN)	기여율
1	Cultivation Environment Control and Plant Growth Monitoring in Greenhouse using Sensors, Communication Network and Data Processing Technologies : A review	Precision Agriculture Science and Technology	백승운	1	한국	PAST	비SCIE	2022.03.30	2672-0086	100%
2	Statistical-based tractor fuel prediction machine learning model	Journal of Agriculture & Life Science	이호민	게제확정	한국	경상대학교 농업생명과학연구원	비SCIE	23년2월말	-	100%

국내 및 국제 학술회의 발표

번호	회의 명칭	발표자	발표 일시	장소	국명
1	한국농업기계학회	김태민, 박재성, 문병은, 이덕현, 김현태	2020.10.26	온라인	대한민국
2	한국농업기계 학회 2021년 추계공동학술대회	이호민	2021.10.29.(금)	소노벨 제주	대한민국
3	2022 한국자동차공학회 춘계학술대회	백승민	2022.06.03	부산 BEXCO	대한민국
4	International Symposium on Machinery and Mechatronics for Agriculture and Biosystems Engineering (ISMAB 2022)	국중후	2022.11.16	가오슝	대만

기술 요약 정보

연도	기술명	요약 내용	기술 완성도	등록 번호	활용 여부	미활용사유	연구개발기관 외 활용여부	허용방식

보고서 원문

연도	보고서 구분	발간일	등록 번호

생명자원(생물자원, 생명정보)/화합물

번호	생명자원(생물자원, 생명정보)/화합물 명	등록/기탁 번호	등록/기탁 기관	발생 연도

[기술적 성과]

지식재산권(특허, 실용신안, 의장, 디자인, 상표, 규격, 신제품, 프로그램)

번호	지식재산권 등 명칭 (건별 각각 기재)	국명	출원				등록			기여율	활용 여부
			출원인	출원일	출원 번호	등록 번호	등록인	등록일	등록 번호		
1	농사용 차량 또는 그 부품의 수명 알림 시스템 및 그 방법	대한민국	주식 회사 긴트	2020. 11.10.	10-2020- 0149729	-	-	-	-	33%	-
2	농지의 특성에 적합한 농작물을 추천하기 위한 방법 및 장치	대한민국	주식 회사 긴트	2021. 06.01	10-2021- 0070820	-	-	-	-	100	-
3	자율 주행을 수행하고 PTO가 작동되는 농기계에 대하여 잠 재적 사고를 예측하고 잠재적 사고를 회피하기 위한 동작 을 제어하기 위한 방법 및 장치	대한민국	주식 회사 긴트	2022. 04.20	10-2022- 0049020	-	-	-	-	100%	-
4	통신 시스템에서 자율 주행을 수행하고 PTO가 작동되는 농 기계에 대하여 잠재적 사고를 예측하고 잠재적 사고를 회 피하기 위한 동작을 제어하기 위한 방법 및 장치	대한민국	주식 회사 긴트	2022. 04.20	10-2022- 0049021	-	-	-	-	100%	-

지식재산권 활용 유형

※ 활용의 경우 현재 활용 유형에 √ 표시, 미활용의 경우 향후 활용 예정 유형에 √ 표시합니다(최대 3개 중복선택 가능).

번호	제품화	방어	전용실시	통상실시	무상실시	매매/양도	상호실시	담보대출	투자	기타
1										√
2										√
3										√
4										√

저작권(소프트웨어, 서적 등)

번호	저작권명	창작일	저작자명	등록일	등록 번호	저작권자명	기여율

신기술 지정

번호	명칭	출원일	고시일	보호 기간	지정 번호

기술 및 제품 인증

번호	인증 분야	인증 기관	인증 내용		인증 획득일	국가명
			인증명	인증 번호		

□ 표준화

○ 국내표준

번호	인증구분 <sup>1)</sup>	인증여부 <sup>2)</sup>	표준명	표준인증기구명	제안주체	표준종류 <sup>3)</sup>	제안/인증일자

- \* 1) 한국산업규격(KS) 표준, 단체규격 등에서 해당하는 사항을 기재합니다.
- \* 2) 제안 또는 인증 중 해당하는 사항을 기재합니다.
- \* 3) 신규 또는 개정 중 해당하는 사항을 기재합니다.

○ 국제표준

번호	표준화단계구분 <sup>1)</sup>	표준명	표준기구명 <sup>2)</sup>	표준분과명	의장단 활동여부	표준특허 추진여부	표준개발 방식 <sup>3)</sup>	제안자	표준화 번호	제안일자

- \* 1) 국제표준 단계 중 신규 작업항목 제안(NP), 국제표준초안(WD), 위원회안(CD), 국제표준안(DIS), 최종국제표준안(FDIS), 국제표준(IS) 중 해당하는 사항을 기재합니다.
- \* 2) 국제표준화기구(ISO), 국제전기기술위원회(IEC), 공동기술위원회1(JTC1) 중 해당하는 사항을 기재합니다.
- \* 3) 국제표준(IS), 기술시방서(TS), 기술보고서(TR), 공개활용규격(PAS), 기타 중 해당하는 사항을 기재합니다.

[경제적 성과]

□ 시제품 제작

번호	시제품명	출시/제작일	제작 업체명	설치 장소	이용 분야	사업화 소요 기간	인증기관 (해당 시)	인증일 (해당 시)

□ 기술 실시(이전)

번호	기술 이전 유형	기술 실시 계약명	기술 실시 대상 기관	기술 실시 발생일	기술료 (해당 연도 발생액)	누적 징수 현황
1	직접실시	농사용 차량 또는 그 부품의 수명 알람 시스템 및 그 방법	(주)긴트	2021.01.01	16,165,682원	0

- \* 내부 자금, 신용 대출, 담보 대출, 투자 유치, 기타 등

□ 사업화 투자실적

번호	추가 연구개발 투자	설비 투자	기타 투자	합계	투자 자금 성격*

□ 사업화 현황

번호	사업화 방식 <sup>1)</sup>	사업화 형태 <sup>2)</sup>	지역 <sup>3)</sup>	사업화명	내용	업체명	매출액		매출 발생 연도	기술 수명
							국내 (천원)	국외 (달러)		
1	자가실시	기존 제품 개선	국내	텔레매틱스	매출	네이스코	1,924	-	2021	-
2	자가실시	기존 제품 개선	국내	텔레매틱스	매출	한국생산기술연구원	2,000	-	2022	-

- \* 1) 기술이전 또는 자기실시
- \* 2) 신제품 개발, 기존 제품 개선, 신공정 개발, 기존 공정 개선 등
- \* 3) 국내 또는 국외

□ 매출 실적(누적)

사업화명	발생 연도	매출액		합계	산정 방법
		국내(천원)	국외(달러)		
텔레매틱스 컨트롤러 조합 외	2021	323,099	-	323,099	납품 공급가액 합산
텔레매틱스 컨트롤러 조합-TMS	2022	146,419	-	146,419	납품 공급가액 합산
산업용 텔레매틱스 게이트웨이	2022	3,300	-	3,300	납품 공급가액 합산
합계		472,818	-	472,818	-

□ 사업화 계획 및 무역 수지 개선 효과

성과					
사업화 계획	사업화 소요기간(년)				
	소요예산(천원)				
	예상 매출규모(천원)	현재까지	3년 후	5년 후	
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
		국내 국외			
향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획					
무역 수지 개선 효과(천원)	수입대체(내수)	현재	3년 후	5년 후	
	수출				

□ 고용 창출

순번	사업화명	사업화 업체	고용창출 인원(명)		합계
			2021년	2022년	
1	농기계 데이터 단말기	주식회사 긴트	2	9	11
합계			2	9	11

□ 고용 효과

구분			고용 효과(명)
고용 효과	개발 전	연구인력	
		생산인력	
	개발 후	연구인력	
		생산인력	

□ 비용 절감(누적)

순번	사업화명	발생연도	산정 방법	비용 절감액(천원)
합계				

□ 경제적 파급 효과

(단위: 천원/년)

구분	사업화명	수입 대체	수출 증대	매출 증대	생산성 향상	고용 창출 (인력 양성 수)	기타
해당 연도							
기대 목표							

산업 지원(기술지도)

순번	내용	기간	참석 대상	장소	인원

기술 무역

(단위: 천원)

번호	계약 연월	계약 기술명	계약 업체명	계약업체 국가	기 징수액	총 계약액	해당 연도 징수액	향후 예정액	수출/수입

[사회적 성과]

법령 반영

번호	구분 (법률/시행령)	활용 구분 (제정/개정)	명 칭	해당 조항	시행일	관리 부처	제정/개정 내용

정책활용 내용

번호	구분 (제안/채택)	정책명	관련 기관 (담당 부서)	활용 연도	채택 내용
1	제안	블록체인 기술기반의 농업기계 관리기술 정책활용 제안	한국농기계협동조합	2022	진행중

설계 기준/설명서(시방서)/지침/안내서에 반영

번호	구분 (설계 기준/설명서/지침/안내서)	활용 구분 (신규/개선)	설계 기준/설명서/지침/안내서 명칭	반영일	반영 내용

전문 연구 인력 양성

번호	분류	기준 연도	현황														
			학위별				성별		지역별								
			박사	석사	학사	기타	남	여	수도권	충청권	영남권	호남권	기타				

산업 기술 인력 양성

번호	프로그램명	프로그램 내용	교육 기관	교육 개최 횟수	총 교육 시간	총 교육 인원

다른 국가연구개발사업에의 활용

번호	중앙행정기관명	사업명	연구개발과제명	연구책임자	연구개발비

국제화 협력성과

번호	구분 (유치/파견)	기간	국가	학위	전공	내용

□ 홍보 실적

번호	홍보 유형	매체명	제목	홍보일
1	브로셔	학회지	긴트 커넥티비티 플랫폼	2021.10.27
2	시연회	자체 농민대상 시연회	긴트 커넥티비티 플랫폼 홍보	2022.09.23
3	박람회	KIEMSTA 2022	긴트 제품군 박람회 출품	2022.11.02.~05

□ 포상 및 수상 실적

번호	종류	포상명	포상 내용	포상 대상	포상일	포상 기관

[인프라 성과]

□ 연구시설·장비

구축기관	연구시설/ 연구장비명	규격 (모델명)	개발여부 (○/×)	연구시설·장비 종합정보시스템* 등록여부	연구시설·장비 종합정보시스템* 등록번호	구축일자 (YY.MM.DD)	구축비용 (천원)	비고 (설치 장소)

\* 「과학기술기본법 시행령」 제42조제4항제2호에 따른 연구시설·장비 종합정보시스템을 의미합니다.

[그 밖의 성과](해당 시 작성합니다)

---



---

(4) 계획하지 않은 성과 및 관련 분야 기여사항(해당 시 작성합니다)

---



---

## 2) 목표 달성 수준

추진 목표	달성 내용	달성도(%)
○ 농기계 이력 데이터 블록체인 시스템 구현	○ 넥스레저를 이용하여 농기계 이력데이터 시스템과 연동 개발 완료	○ 100%
○ 농기계 이력 및 운행 데이터 수집 환경과 블록체인 / AI 알고리즘 연동 개발	○ 농기계 이력데이터를 조회할 수 있는 API를 제공하여 연동 완료	○ 100%
○ 농기계 이력 데이터 입력 시스템 개발 및 정부 지원 시스템 개발	○ 농협 및 정부에서 필요한 정보를 조회 또는 다운로드 할 수 있는 시스템 및 서비스 개발 완료	○ 100%
○ 농기계 이력 및 운행 데이터 기반 제조사 웹 서비스 개발	○ 제조사에서 출하한 농기계에 대한 적절한 운행 데이터를 조회 또는 다운로드 가능한 서비스개발 완료	○ 100%
○ 농기계 이력 데이터 확보	○ 이력 확보 완료	○ 100%
○ 농기계 이력/운행 데이터 시스템과 연동 확인 및 트레이닝 데이터 특징 분석	○ 연동 확인 후 데이터 특징 분석 완료	○ 100%
○ 농기계 이력/운행 트레이닝 데이터를 활용한 기본 AI 알고리즘 개발	○ 전문기관 의료 후 수행 완료	○ 100%
○ 양산형 단말기 환경 시험 수행	○ 시군구 및 사용자 피드백 반영 완료	○ 100%
○ 사용자 피드백을 바탕으로 서비스 개선	○ 기본 알고리즘 개발 완료	○ 100%
○ 통계 기법 기반 영향 인자를 실증 데이터를 통하여 도출 후 고도화 반영	○ 통계기법 적용 후 결론 도출 완료	○ 100%
○ 머신러닝 기반 기법을 적용하여 추가적 데이터 특징을 도출 후 고도화 반영	○ 머신러닝 기법을 적용하여 가능성 확인 완료	○ 100%
○ 실증사업화 대상 시군구 및 농업기계 선정	○ (유관기관) 농협경제지주(주), (시군구) 상주시, 남원시와 업무협약 체결,	○ 100%
○ 선정 농업기계와 관련한 정부 (지자체 포함)의 지원현황 파악	○ 대상 농업기계 선정 : 트랙터, 굴착기	○ 100%
○ 선정 시군구 시스템 실증 시험 실시 및 점검	○ 농기계 구입지원사업, 농협 농기계은행사업, 지자체별 농기계 보조사업 및 임대사업 등	○ 100%
○ 실증사업화 추진 현장 점검	○ 실증사업 추진 완료	○ 100%
○ 정부 정책 제안	○ 현장 점검 완료	○ 100%
○ 트랙터 주요 동력 소모원 선정 및 부하계측 시스템 설계·개발	○ 제안 완료	○ 100%
	○ 트랙터의 주요 소모원에 대한 분석을 수행하였고, 각 인터페이스에 적합한 센서들을 선정하여 계측 시스템을 개발하였음.	○ 100%

<ul style="list-style-type: none"> <li>○ 필드 실험을 통한 주요 농작업 부하 데이터 수집 및 분석</li> <li>○ 데이터 분석을 통한 고장 인지 기술 개발</li> <li>○ 현장 실증시험 수행 및 성능평가 수행</li> </ul>	<ul style="list-style-type: none"> <li>○ 쟁기 및 로타리 작업기를 선정하여 계측 시험을 수행하여 부하 데이터를 수집하였음.</li> <li>○ 트랙터의 주요 소모원을 설정하였으며, 모니터링 시스템을 이용한 고장 인지 기술을 개발하였음.</li> <li>○ 현장 실증시험 및 성능평가를 수행하여 개발한 기술의 적용 적합도를 판단하였음.</li> </ul>	<ul style="list-style-type: none"> <li>○ 100%</li> <li>○ 100%</li> <li>○ 100%</li> </ul>
---	---	--

4. 목표 미달 시 원인분석(해당 시 작성합니다)

1) 목표 미달 원인(사유) 자체분석 내용

---

해당 없음.

---

2) 자체 보완활동

---

해당 없음.

---

3) 연구개발 과정의 성실성

---

해당 없음.

---

## 5. 연구개발성과의 관련 분야에 대한 기여 정도

(단위 : 백만원, %)

총괄과제명	세부과제명	기관명	유형	총 연구개발비 (A)	정부지원 연구개발비 (B)	정부지원 연구개발비 비율 (C=B/A)	성과 유형	기술기여도	
								산정 근거	비율
블록체인 기술기반의 농업기계 관리기술 개발	블록체인 기술기반의 농업기계 관리기술 개발	(주)긴트	중소기업 (영리)	1,404	960	68.38%	신규 기술개발	①-①	68.38
계				1,404	960	68.38%	-	-	-

- 블록체인 도입으로 인해 농업기계 이력 데이터의 투명성이 보장되며, 별도의 중간관리자 없이 생애주기별 이력 데이터의 추적이 용이해짐.
- 농기계 이력/운행데이터 API 인증 및 데이터 특징 분석
- 농기계 이력/운행 트레이닝 데이터 AI 알고리즘 개발
- 통계 및 머신러닝 기반 기법을 적용한 고도화
- 기존의 농업기계 연구들은 대부분 경험적 방법에 의해서 수행되었음. 본연구를 통해서 수집된 데이터들은 차후 트랙터 설계간의 경험적 방법이 아닌 수치적 방법으로 접근 가능하게 해 줄 것이며, 이를 통해 농업기계의 최적화 설계가 가능할 것으로 판단됨.
- 또한, 이번 계측 시스템 개발간 활용된 기술 및 노하우들은 차후 다른 형태의 농업기계 및 플랫폼 설계 개선 시 사용 가능할 것으로 판단됨.
- 기존의 농업기계 고장 인지 기술들은 대부분 사용자가 농업기계의 손상률을 모니터링 하기 어려웠음. 본 연구를 통해서 개발한 트랙터 고장 인지 모니터링 시스템을 실제 적용하여 작업자의 편의성과 작업 능률을 높이고, 트랙터의 주요 부품에 대한 고장 예방이 가능할 것으로 판단됨.
- 또한, 이번 모니터링 시스템 개발간 활용된 기술 및 노하우들은 차후 다른 형태의 농업기계 및 플랫폼 설계 개선 시 사용가능할 것으로 판단됨.

## 6. 연구개발성과의 관리 및 활용 계획

구분(정량 및 정성적 성과 항목)		연구개발 종료 후 5년 이내				
		2023	2024	2025	2026	2027
국외논문	SCIE					
	비SCIE					
국내논문	SCIE					
	비SCIE					
특허출원	국내					
	국외					
특허등록	국내					
	국외					
인력양성	학사					
	석사					
	박사					
사업화	시제품개발					
	상품출시		1	1		
	기술이전					
	공정개발					
	매출액(단위 : 천원)	200,000	200,000	200,000	200,000	200,000
	기술료(단위 : 천원)	6,837	6,837	6,837	6,837	6,837
정성적 성과 주요 내용		<ul style="list-style-type: none"> <li>- 농기계의 다양한 커넥티비티를 시도하는 것은 최근 에 벗어날 수 없는 트렌드로서 이에 블록체인까지 적 용하여 보안성을 확보한 제품이 될 것</li> <li>- 부하 계측 데이터의 경우 다양한 활용이 가능할 것으 로 판단됨. 농기계의 부품 최적화 설계 및 내구수명 예 측등에 사용가능할 것으로 판단됨. 차후 농기계 업체 및 유관 기관들과의 협업을 통하여 최적화설계와 관련 된 연구가 가능할 것으로 판단되며, 해당 연구간에는 이번연구를 통해 계측된 데이터들을 활용 가능할 것으 로 판단됨.</li> <li>- 고장 인지 기술의 경우 다양한 활용이 가능할 것으 로 판단됨. 농기계 부품의 고장 예방 및 내구수명 예 측 등에 사용 가능할 것으로 판단됨. 차후 농기계 업 체 및 유관 기관들과의 협업을 통하여 최적화 설계와 관련된 연구가 가능할 것으로 판단되며, 해당 연구 간에는 이번 연구를 통해 개발한 기술을 다양한 농업 기계에 적용 가능할 것으로 판단됨.</li> </ul>				

## 자체평가의견서

### 1. 과제현황

		과제번호	320025-03		
사업구분	첨단농기계산업화기술개발사업				
연구분야			과제구분	단위	
사업명	블록체인 기술기반의 농업기계 관리기술 개발			주관	
총괄과제	기재하지 않음		총괄책임자	기재하지 않음	
과제명	블록체인 기술기반의 농업기계 관리기술 개발		과제유형	개발	
연구개발기관	주식회사 긴트		연구책임자	김용현	
연구기간 연구개발비 (천원)	연차	기간	정부	민간	계
	1차년도	2020. 04. 29 - 2020. 12. 31	400,000	134,000	534,000
	2차년도	2021. 01. 01 - 2021. 12. 31	530,000	177,000	707,000
	3차년도	2022. 01. 01 - 2022. 12. 31	530,000	133,000	663,000
	4차년도	-	-	-	-
	5차년도	-	-	-	-
	계	2020. 04. 29 - 2022. 12. 31	1,460,000	440,000	1,904,000
참여기업	주식회사 긴트, 경상대학교, 한국농기계공업협동조합, 충남대학교				
상대국	-	상대국연구개발기관	-		

※ 총 연구기간이 5차년도 이상인 경우 셀을 추가하여 작성 요망

2. 평가일 : 2023. 02. 09

3. 평가자(연구책임자) :

소속	직위	성명
주식회사 긴트	대표이사	김용현

4. 평가자(연구책임자) 확인 :

본인은 평가대상 과제에 대한 연구결과에 대하여 객관적으로 기술하였으며, 공정하게 평가하였음을 확약하며, 본 자료가 전문가 및 전문기관 평가 시에 기초자료로 활용되기를 바랍니다.

확약	
----	---

## I. 연구개발실적

※ 다음 각 평가항목에 따라 자체평가한 등급 및 실적을 간략하게 기술(200자 이내)

### 1. 연구개발결과의 우수성/창의성

■ 등급 : 우수

연구개발 결과는 계획한 연구 내용들을 모두 다뤘으며, 이에 수행한 내용 또한 우수한 것으로 판단한다. 특히, 논문과 같은 학문적인 연구 뿐만아니라 일부 사업화에서도 실적을 거두었다. 또한 정책 건의, 박람회 참가, 시연회 개최 등을 통한 다양한 루트를 통해 관련기술의 확산에도 기여하였다.

### 2. 연구개발결과의 파급효과

■ 등급 : 우수

농기계의 스마트화와 IoT의 발전으로, 농기계에서 발생하는 데이터의 확산은 명약관화한 요즘의 상황이다. 데이터의 확산에서 가장 중요한 요소중의 하나로 보안이 있으며 이는 암호화가 필수이다. 이에 따라 최근 가장 보안성 높은 암호화 기술인 블록체인 기술이 농기계 데이터 수집에 접목한 본 과제의 파급효과는 매우 클 것으로 생각된다.

### 3. 연구개발결과에 대한 활용가능성

■ 등급 : 우수

대표적인 연구 개발 산출물로 나온 농기계용 단말기와 서비스를 활용한다면, 쉽게 적용이 가능하다. 또한 설치 대상 농기계에도 크게 제약은 없으며, 다양한 농기계에 적용 가능하다. 특히 및 논문, 정책 제언 등은 앞으로 유사 분야 개발 연구에도 큰 도움이 될 것으로 생각된다.

### 4. 연구개발 수행노력의 성실도

■ 등급 : 우수

본 과제에 참여한 기관 모두 연구 개발 목표 달성 및 일정 준수를 위하여 매진하였으며, 이는 모든 연구개발 목표를 달성한 것으로 미루어 보아 충분히 성실하였음을 확인할 수 있었다.

### 5. 공개 발표된 연구개발성과(논문, 지적소유권, 발표회 개최 등)

■ 등급 : 우수

공개된 논문 및 등록되어 공개된 특허, 또는 출원중이라 현재는 미공개지만 곧 공개될 특허들이 연구 개발 결과로 계획 대비 충분히 도출 되었다. 이를 보아 질적, 양적으로 우수하다고 판단할 수 있다. 이에 더하여 본 과제를 수행중 다양한 홍보활동(시연회 개최, 박람회 출품)등을 통하여 성과에 대한 공개에도 충분한 역할을 하였다고 생각된다.

## II. 연구목표 달성도

세부연구목표 (연구계획서상의 목표)	비중 (%)	달성도 (%)	자체평가
특허출원(4건)	10	100	4건 / 보통
기술실시(1건)	5	100	1건 / 보통
기술료(5백만원)	5	0	0 / 미달
제품화(2건)	20	100	2건 / 보통
매출액(200백만원)	10	231	463백만원 / 우수
고용창출(6명)	20	416	25명 / 우수
학술발표(2건)	5	150	3건 / 우수
정책활용(1건)	15	100	1건 / 보통
홍보전시(3건)	10	100	3건 / 보통
합계	100	174	우수

## III. 종합의견

### 1. 연구개발결과에 대한 종합의견

종합적으로 보아 본 연구개발 과제는 계획 대비 그 이상의 실적을 보여주었다. 특히 매출액 부분에서 본 과제의 대표적인 산출물이 계획대비 높은 매출을 보여주었다. 이를 보아 사업화에 충실했던 과제로 판단되며, 사업의 확장에 따른 고용의 창출에도 이바지 하였다.

### 2. 평가시 고려할 사항 또는 요구사항

본 과제의 평가시에는 주관 기관이 회사인 만큼 실제 사업화가 되었고, 산출물로 인한 매출이 발생이 이루어졌는지가 평가의 주요 요건이라고 판단된다.

### 3. 연구결과의 활용방안 및 향후조치에 대한 의견

본 연구결과의 활용방안은 크게 두 가지로 생각된다. 첫 번째는 사업화된 제품의 판매 확장, 두 번째는 특허, 논문 등의 연구 산출물에 대한 확산이다. 이를 위해 주관 기관은 판매 확장에 주력할 것이며 그 외 대학기관에서는 교신저자로서의 활동에 충실할 것을 당부한다. 또한 기술료 납부 부분을 과제 기간내에 실시 하였어야하나, 대내외적인 상황속에서 그러지 못한 부분에 대해서는 추후 성실히 납부하도록 하겠다.

#### IV. 보안성 검토

※ 보안성이 필요하다고 판단되는 경우 작성함.

##### 1. 연구책임자의 의견

##### 2. 연구개발기관 자체의 검토결과

# 연구성과 활용계획서

## 1. 연구과제 개요

사업추진형태	<input checked="" type="checkbox"/> 자유응모과제	<input type="checkbox"/> 지정공모과제	분 야	개발
연구과제명	블록체인 기술기반의 농업기계 관리기술 개발			
주관연구개발기관	주식회사 긴트		주관연구책임자	김용현
연구개발비	정부지원 연구개발비	기관부담연구개발비	기타	총연구개발비
	1,460,000	444,000	0	1,904,000
연구개발기간	2020. 04. 29 - 2022. 12. 31 (2년 8개월)			
주요활용유형	<input checked="" type="checkbox"/> 산업체이전 <input type="checkbox"/> 미활용 (사유: )	<input type="checkbox"/> 교육 및 지도	<input checked="" type="checkbox"/> 정책자료	<input type="checkbox"/> 기타( )

## 2. 연구목표 대비 결과

당초목표	당초연구목표 대비 연구결과
① 정부기관용 농기계 관리서비스 개발	완료
② 제조사/농민용 농기계 관리서비스 개발	완료
③ 관리기술 확산을 위한 정책 제언	완료

\* 결과에 대한 의견 첨부 가능

## 3. 연구목표 대비 성과

(단위 : 건수, 백만원, 명)

성과 목표	사업화지표										연구기반지표								
	지식 재산권				기술 실시 (이전)		사업화				기술 인증	학술성과			교육 지도	인력 양성	정책 활용·홍보		기타 (타연구활용)
	특허 출원	특허 등록	품종 등록	S M A R T F O R E C A S T I N G	건 수	기 술 료	제 품 화	매 출 액	수 출 액	고 용 창 출		투 자 유 치	논 문	S C I			비 S C I	논 문 평 관 I F	
											건				건	건			건
단위	건	건	건	건	건	백만원	건	백만원	백만원	명	백만원	건	건	건	건	명	건	건	건
가중치	10				5	5	20	10		20				5			15	10	
최종 목표	4				1	5	2	200		8			2	2			1	3	
당해 년도	목표	2			1	5	1	200		2			1	1			1	1	
	실적	3			1	0	1	149		11			3	4			1	2	
달성률 (%)	133				100	0	100	75		550			300	400			100	150	

4. 핵심기술

구분	핵심기술명
①	GPS기반 농기계 관리 가능 단말기 및 서비스 기술
②	농민/제조사 트랙터 상태 모니터링 제공 기술
③	정부 기관 임대 농기계 관리 및 모니터링 기술

5. 연구결과별 기술적 수준

구분	핵심기술 수준					기술의 활용유형(복수표기 가능)				
	세계 최초	국내 최초	외국기술 복제	외국기술 소화·흡수	외국기술 개선·개량	특허 출원	산업체이전 (상품화)	현장애로 결	정책 자료	기타
①의 기술					0		0			
②의 기술					0		0		0	
③의 기술					0		0		0	

\* 각 해당란에 v 표시

6. 각 연구결과별 구체적 활용계획

핵심기술명	핵심기술별 연구결과활용계획 및 기대효과
①의 기술	다종의 농기계에 확대 적용 목표
②의 기술	다수의 제조사와 농민들에게 확대 적용 목표
③의 기술	각 지방 자치 단체에 홍보 및 도입 권유

7. 연구종료 후 성과창출 계획

(단위 : 건수, 백만원, 명)

성과 목표	사업화지표										연구기반지표								
	지식 재산권				기술 실시 (이전)		사업화				기술 인증	학술성과			교육 지도	인력 양성	정책 활용·홍보		기타 (타연구활용액)
	특허 출원	특허 등록	품종 등록	SMART	건수	기술료	제품화	매출액	수출액	고용 창출		투자유치	논문 SCI	논문 비SCI			논문 평균 I-F	학술 발표	
단위	건	건	건	건	건	백만원	건	백만원	백만원	명	백만원	건	건	건	건	명	건	건	
가중치	10				5	5	20	10		20				5			15	10	
최종목표	4	3			1	5	4	1200		8			2		2		1	4	
연구기간내 달성실적	4				1		2	463		25			3		4		1	3	
연구종료후 성과창출 계획		3				5	2	1000		2								1	



### 주 의

1. 이 보고서는 농림축산식품부에서 시행한 첨단농기계산업화기술개발사업의 연구보고서입니다.
2. 이 보고서 내용을 발표하는 때에는 반드시 농림축산식품부에서 시행한 첨단농기계산업화기술 개발사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 됩니다.