Journal of KIIT. Vol. 23, No. 2, pp. 253-261, Feb. 28, 2025. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2025.23.2.253

머신 러닝 워크플로우 자동화를 위한 분산파일 통합관리 시스템

김진우*[1], 박준동**[1], 정은희**[2], 김성렬*[2]

A Distributed File Management System for Machine Learning Workflow Automation

Jin-Woo Kim*[1], Jun-Dong Park**[1], Eun-Hui Jeong**[2], and Sung-Ryul Kim*[2]

이 연구는 국립금오공과대학교 대학 연구과제비로 지원되었음(2022~2024)

요 약

기존 머신 러닝 솔루션은 간편한 데이터 분석을 제공하지만 방대한 파일에서 특정 부분을 추출하여 병합하는 작업은 여전히 수작업으로 진행되며 분산된 파일의 개수가 많을수록 이는 연구자의 부담을 가중한다. 본 연구에서는 정규표현식으로 파일에서 데이터를 자동으로 추출하고 이를 머신 러닝으로 분석하는 웹 기반 머신 러닝 플랫폼을 제안한다. 제안 시스템은 사용자 친화적 GUI를 통해 쉽게 모델 학습, 평가 및 시각화를 수행하도록 한다. 제안된 시스템의 효용성 검증은 리튬이온 이차전지 소재의 물성 연구를 통해 진행되었으며 그 결과 R2와 RMSLE(Root Mean Squared Logarithmic Error)가 각각 0.954와 0.242로 나타났다. 이는 데이터 공학에 대한 지식이 없는 연구자도 제안 시스템을 통해 효과적으로 머신 러닝을 수행할 수 있음을 시사한다.

Abstract

Although existing machine learning solutions make it easy to analyze data, extracting the interesting parts from massive files and merging them is still a manual process, and the greater the number of distributed files, the more burden this places on researchers. This study proposes a web-based machine learning platform that automatically extracts data from files using regular expressions and analyzes them using machine learning. The proposed system provides a user-friendly GUI allowing users to easily train model and evaluate and visualize the performance. To validate the effectiveness it was applied to the study of predicting the properties of lithium-ion battery materials and it shows that R^2 and Root Mean Squared Logarithmic Error(RMSLE) were 0.954 and 0.094, respectively. This result reveals that researchers without knowledge about data engineering can adopt machine learning using the proposed system.

Keywords
artificial intelligence, autoML, platform, file management

* 국립금오공과대학교 컴퓨터소프트웨어공학과(*[2] 교신저자)
- ORCID[1]: https://orcid.org/0009-0005-7054-4892
- ORCID[2]: https://orcid.org/0009-0005-5218-1477
** 숙명여자대학교 화공생명공학부
- ORCID[1]: https://orcid.org/0000-0003-2918-5339
- ORCID[2]: https://orcid.org/0009-0002-2209-5947

· Received: Dec. 23, 2024, Revised: Feb. 18, 2025, Accepted: Feb. 21, 2025
· Corresponding Author: Sung-Ryul Kim
Dept. of Computer Software Engineering, Kumoh National Institute of Technology, Korea
Tel.: +82-54-478-7549, Email: sungryul@kumoh.ac.kr

Ⅰ. 소 개

인공지능은 설계 자동화, 최적화, 예측 모델링 등 다양한 영역에서 기존의 공학적 한계를 극복하고 있으며, 이는 엔지니어링 프로세스의 효율성 향상과 복잡한 시스템의 데이터 기반 의사결정을 가능케 하여 공학의 패러다임을 재정의하고 있다[1]. 또한 인공지능의 성공적인 적용 사례가 증가함에 따라, 다양한 연구 분야에서 인공지능 기반 데이터 분석이 시도되고 있다.

최신 인공지능 모델 학습, 성능 평가 및 시각화의 과정을 자동으로 수행하는 AutoML의 발전으로 인공지능 기술 접근성이 크게 향상되었다. 그러나 소프트웨어 역량이 부족한 연구자들은 여전히 인공지능을 자신의 연구 분야에 적용하는 데 어려움을 겪고 있다. 이는 데이터 수집 및 전처리부터 인공지능 모델 선정과 학습에 이르는 전 과정에 대한 이론적 지식과 실제적 개발 능력을 요구하기 때문이다[2]-[4].

대부분의 기존 AutoML 연구에서는 주어진 데이터를 분석하는 데 초점이 맞춰져 있으나 분석을 위한 데이터 수집 과정에 대한 고려는 거의 없는 실정이다[5]-[7]. 특히, 현대의 데이터 분석 환경에서는 실험 및 관측 데이터가 다양한 형식의 파일로 저장되는 경우가 잦다. 다양한 형식에서의 데이터 추출은 상당히 번거로운 작업이며 휴먼 에러 발생에 대한 위험도 존재한다. 더욱이, 물리적으로 분산된 원격지 데이터를 통합관리하는 것은 인공지능 기술 적용에 있어 해결해야 할 어려움이지만 이에 대한 필요성 제기나 해결책은 거의 없다. 이에 본 연구는 웹 기반 분산파일 통합관리 시스템 및 머신 러닝 워크플로우 자동화 플랫폼을 제안한다. 본 플랫폼의 구성은 크게 두 가지로 구분된다.

1) 웹 기반 파일 관리 및 데이터 통합 관리: 분산 저장된 파일들을 웹 기반 시스템에 업로드하고 저장하여 중앙 집중식으로 관리한다. 파일 원본과 더불어 정규표현식을 이용해 복잡한 파일 형식에서 분석에 사용될 데이터를 자동으로 구분하여 데이터 추출 및 병합 과정에서 발생하는 휴먼 에러를 최소화한다. 추가적으로 GUI(Graphical User Interface) 기반 필터링을 통해 연구자의 다각적인 데이터 분석을 도모하였다.

2) GUI 기반 모델 학습 및 시각화: 웹 UI를 통해 연구자가 모델 학습에 사용할 특성(Feature)과 목표변수(Target)를 선택할 수 있게 한다. 또한, 학습에 사용할 모델을 UI 기반으로 간단히 선택할 수 있게 하여 학습 과정을 용이하게 한다. 학습이 완료된 모델의 성능 평가 결과는 자동으로 시각화되어 제공된다. 이를 통해 연구자는 다양한 데이터에 머신 러닝을 적용할 수 있다.

개발한 플랫폼의 효용성을 검증하기 위해 리튬이온 이차전지 소재의 물성을 예측하는 연구에 해당 플랫폼을 적용하였다. 실험 데이터는 2024년 1월 1일부터 2024년 11월 30일까지 두 개의 연구그룹, 총 14명의 연구자로부터 수집되었으며 총 파일의 개수는 400여 건이다. 해당 연구의 핵심 목표인 전극에 포함되는 슬러리(Slurry)의 탄성을 예측하기 위해 회귀 분석모델을 적용했으며 가장 좋은 성능을 보인 LGBM(Light Gradient Boosting Machine) 모델의 경우 R2와 RMSLE(Root Mean Squared Logarithmic Error)가 각각 0.954와 0.242로 나타났다. 이는 본 시스템이 연구자의 최소개입만으로도 유의미한 분석이 가능함을 방증한다. 또한, 실험환경에 따른 탄성 추이를 시각적으로 표현하고 스케일 조정 및 데이터 선별 기능을 통해 연구의 통찰력을 제공할 수 있음을 확인하였다.

본 연구의 구성은 다음과 같다. 2장에서는 제안하는 전체 시스템 아키텍처 및 핵심모듈의 기능을 소개한다. 3장에서는 각 모듈의 세부 구현사항을 설명한다. 4장에서는 사례 연구를 통해 개발된 플랫폼의 효용성을 검증한다. 마지막으로 5장에서는 본 연구의 결론을 요약하고 향후 연구 방향을 제시한다.

Ⅱ. 시스템 아키텍처

본 연구에서 제안하는 시스템은 인공지능 학습 자동화와 웹 기반 플랫폼의 장점을 결합한 구조를 갖추고 있다. 이 시스템은 그림 1과 같이 크게 네 가지 주요 모듈로 구성되어 있으며, 각 모듈은 유기적으로 연계되어 효율적인 머신 러닝 워크플로우를 지원한다.

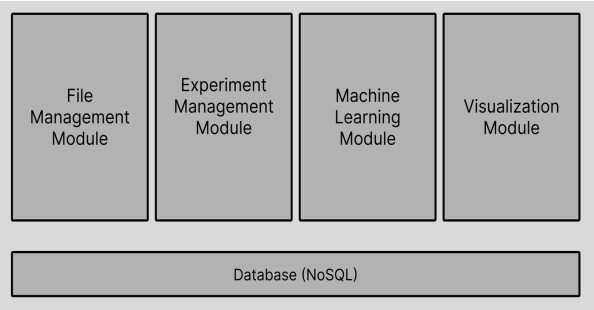


그림 1. 시스템 구조도
Fig. 1. System architecture

특히, 각 기능의 모듈화 및 이들 간의 인터페이스 설계를 통해 기능의 자유로운 추가/확장을 지원하여 플랫폼의 범용성을 도모하였다. 각 모듈의 핵심기능은 다음과 같다.

파일 관리 모듈(File management module)은 시스템의 기초적인 데이터 처리를 담당한다. 이 모듈은 다양한 형식의 파일에 대한 업로드 및 다운로드 기능을 제공한다. 특히, 파일로부터 데이터를 자동으로 추출하는 기능은 본 시스템의 핵심적인 요소로, 분산된 데이터 소스로부터 정보를 효과적으로 통합하는 기능을 수행한다. 이를 통해 사용자는 데이터 전처리 과정에서 발생할 수 있는 시간과 노력을 크게 절감할 수 있으며 원본 파일에서 직접 데이터를 추출할 때 자주 발생하는 휴먼 에러를 예방할 수 있다.

실험 관리 모듈(Experiment management module)은 데이터의 체계적인 관리와 조직화를 담당한다. 이 모듈에서는 '실험'이라는 개념을 도입하여, 업로드된 파일, 추출된 데이터, 그리고 사용자가 추가로 입력한 정보를 하나의 단위로 묶어 관리한다. 이러한 접근 방식은 데이터의 일관성을 유지하고, 실험의 재현성을 높이는 데 크게 기여한다. 또한, 실험 조회 및 검색 기능을 통해 사용자는 과거의 실험 결과를 쉽게 참조하고 비교할 수 있다.

머신 러닝 모듈(Machine learning module)은 실험 관리 모듈에서 준비된 데이터를 기반으로 모델을 생성하고, 그 결과를 확인하는 기능을 수행한다. 이 모듈은 파일 관리 모듈과 실험 관리 모듈에서 준비된 데이터에서 특징과 목표 변수를 손쉽게 선택할 수 있도록 설계되었다. 사용자는 다양한 머신 러닝 알고리즘을 선택하고 하이퍼 파라미터를 조정할 수 있으며, 모델의 학습 과정을 모니터링하고 결과를 분석할 수 있다.

시각화 모듈(Visualization module)은 머신 러닝 과정의 다양한 측면을 직관적으로 이해할 수 있도록 지원한다. 이 모듈은 성과 지표, 데이터 분포, 모델의 예측 결과 등을 그래프, 차트, 히트맵 등 다양한 형태로 시각화한다. 이를 통해 사용자는 복잡한 데이터와 모델의 특성을 쉽게 파악하고, 의사결정에 필요한 통찰을 얻을 수 있다.

한편, 각 모듈에서 사용하는 데이터는 스키마리스(Schema less)를 지원하는 NoSQL(Not only SQL) 데이터베이스를 통해 관리된다. 스키마를 가지는 관계형 데이터베이스를 사용할 경우, 파일의 형태 변경에 유연하게 대처하지 못한다는 제약이 있다. 반면 NoSQL 데이터베이스의 스키마리스 특성은 다양한 구조의 데이터를 효율적으로 저장하고 관리할 수 있다는 장점을 지닌다. 특히 장기간에 걸쳐 실험 데이터를 수집할 때 데이터 구조의 변경은 종종 발생하므로 플랫폼의 범용성과 유연성 측면을 고려한 NoSQL 데이터베이스 적용이 적합하다[8][9].

본 시스템은 데이터 과학자뿐만 아니라 다양한 분야의 연구자들도 쉽게 머신 러닝 모델을 개발하고 적용할 수 있는 환경을 제공한다. 특히, 웹 기반 플랫폼의 특성을 활용하여 사용자들은 시간과 장소에 구애받지 않고 시스템에 접근하여 실험을 수행하고 결과를 확인할 수 있다.

III. 시스템 구현

파일 관리 모듈과 실험 관리 모듈은 그림 2와 같이 데이터 통합 관리를 위해 유기적으로 연계되어 작동한다. 파일 관리 모듈에서는 .pdf, .xlsx, .csv 등 다양한 형식의 파일을 입력받아 저장하고 데이터를 추출하는 역할을 수행한다. 추출된 데이터는 실험 관리 모듈로 전달되고, 이 모듈에서 추가적인 사용자 입력을 받아, 추출된 데이터와 사용자 입력을 통합하여 '실험'을 생성한다. 두 모듈은 모두 NoSQL 데이터베이스와 연결되어 있어 데이터를 효율적으로 저장하고 관리할 수 있다. 이러한 구조를 통해 다양한 소스에서 얻은 데이터가 통합되어 체계적으로 관리되어, 연구의 효율성과 데이터의 일관성이 향상된다.

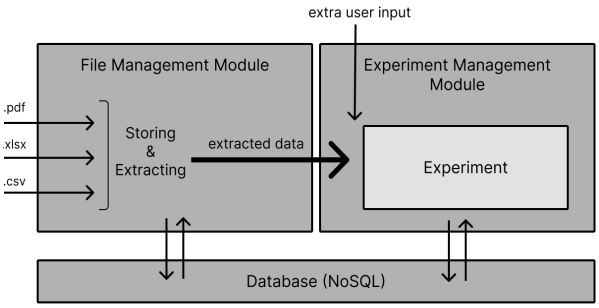


그림 2. 파일 관리 모듈과 실험 관리 모듈 간의 상호 작용
Fig. 2. Interaction of file management module and experiment managment module

본 시스템의 구현에 사용된 기술은 표 1과 같다. 스프링 프레임워크(Spring framework)를 사용하여 메인 서버를 개발하였으며, 데이터베이스는 MongoDB를 사용하였다. SSR(Server Side Rendering)방식으로 구현하기 위해 리액트(React)를 도입하였고 웹 서버(Web server)와 리버스 프록시(Reverse Proxy)의 역할로는 Nginx가 사용되었다. 추가적으로, 머신 러닝 서버(ML server)는 파이썬(Python) 기반의 FastAPI 프레임워크를 사용하여 개발하였다.

표 1. 시스템 구성 환경
Table 1. System configuration environment

Main server	Java 17, Spring boot 3.14
Web frontend	NodeJS 20.8.1, React 18
Web server	Nginx 1.18.0
DBMS	MongoDB (Community Edition 6.0.1)
AutoML	Pycaret 3.3.2
ML server	Python 3.9, FastAPI 0.114

3.1 파일 관리 모듈

파일 관리 모듈은 파일 조작 모듈과 파일 추출 모듈로 구성되어 있으며, 이는 효율적인 파일 처리와 데이터 추출을 위한 프레임워크를 제공한다. 파일 조작 모듈은 업로드된 파일을 저장하고 파일 추출 모듈로 전달하는 역할을 수행한다.

파일 추출 모듈은 그림 3과 같이, ExtracterManager, Extracter, Parser 세 부분으로 나뉜다. 파일 추출 모듈의 핵심 요소인 ExtracterManager는 다양한 파일 형식에 대응하는 Extracter를 관리하며, 각 파일 형식에 적합한 Extracter에게 파일을 전달한다. Extracter는 파일의 구조에 따라 실제 데이터 추출 로직을 포함하는 Parser를 관리하고, 해당 파일 유형에 맞는 Parser에게 파일을 전달한다. 이러한 구조를 통해 모든 입력 파일의 데이터를 효과적으로 추출하고, 추출된 데이터를 실험관리 모듈로 이동시킬 수 있다. 만약 새로운 형태의 파일 관리가 필요할 경우 사용자는 해당 파일에 맞는 Parser를 구현하고 이를 Extracter에 등록만 하면 된다. 이러한 유연성을 통해 사용자는 본인의 도메인 특성에 맞게 시스템을 최적화(Customizing)할 수 있다.

3.2 실험 관리 모듈

실험 관리 모듈은 파일로부터 추출된 데이터와 사용자 입력 데이터를 '실험'이라는 단일 단위로 통합하여 관리하는 것을 주요 목적으로 한다.

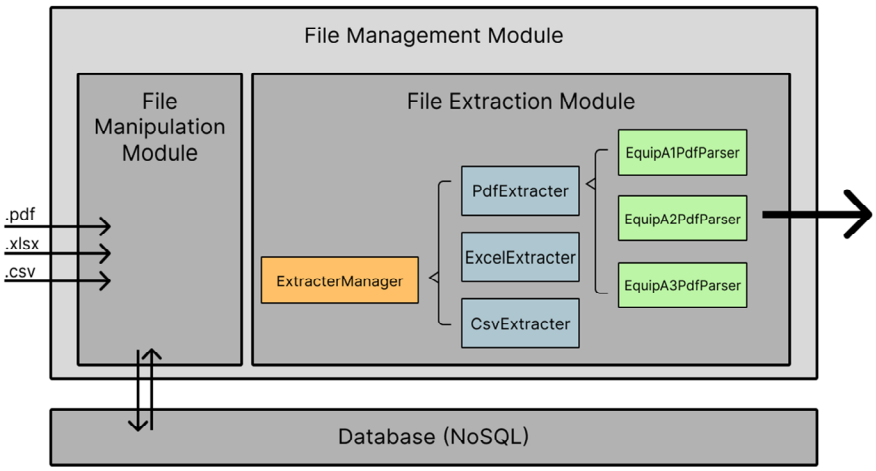


그림 3. 파일 관리 모듈 구조도
Fig. 3. Architecture of file management module

이러한 통합관리 방식은 파일에서 추출한 데이터와 관련 정보의 효율적인 조회 및 검색을 가능하게 하며, 일관된 데이터 소스를 제공함으로써 머신 러닝 학습에도 최적화된 환경을 제공한다. 특히, 본 모듈은 사용자의 도메인 특성에 따른 높은 수준의 개인화를 지원하는데, '실험'의 구조와 구성요소를 사용자의 요구사항에 맞게 유연하게 재구성할 수 있다.

이러한 유연성은 스키마리스 특성을 가진 NoSQL 데이터베이스를 통해 더욱 강화되어, '실험' 데이터의 스키마 변경에도 데이터 영속성을 보장한다. 또한, 사용자가 별도의 개인화를 수행하지 않더라도, 시스템은 파일에서 추출한 데이터를 '실험' 단위로 자동 구성하여 추가적인 변형 없이도 머신 러닝 학습까지 한 번에 수행할 수 있다.

3.3 머신 러닝 모듈

머신 러닝 모듈은 머신 러닝 파이프라인을 포괄하며, 사용자 친화적인 웹 기반 GUI를 통해 사용자 편의성을 높였다. 제안된 시스템의 핵심 구성 요소인 머신 러닝 모듈은 학습 데이터 선정, 전처리, 모델 생성 및 파라미터 튜닝, 모델 및 메타데이터 저장 순의 작업 순서를 가진다. 가장 먼저 이전에 저장된 '실험' 데이터셋에서 학습에 필요한 데이터를 선정하고, 선정된 데이터는 전처리 과정을 거쳐 모델 학습에 적합한 형태로 변환된다. 이 과정에서 데이터 정제, 특성 선택, 스케일링 등 다양한 전처리 기법이 적용될 수 있으며, 이는 모델의 성능 향상에 중요한 역할을 한다. 모델 학습 단계에서는 전처리된 데이터를 사용하여 다양한 머신 러닝 알고리즘을 적용, 최적의 모델을 생성한다. 학습이 완료된 모델과 관련 메타데이터는 NoSQL 데이터베이스에 저장/관리된다.

그림 4에서 보는 바와 같이, 머신 러닝 모듈은 크게 데이터 전처리 모듈과 모델 관리 모듈로 구성되며, 이들은 GUI Wrapper 내에서 통합되어 작동한다. 데이터 전처리 모듈은 '실험'에서 생성된 데이터를 입력받아 사전 정의된 형식으로 변환하는 역할을 수행한다. 사용자는 이 모듈을 통해 특징 변수와 목적 변수를 선택하고, 데이터의 유효 범위를 지정하는 등 기본적인 전처리 작업을 수행할 수 있다.

전처리된 데이터는 모델 관리 모듈로 전달되어 다양한 머신 러닝 모델을 탐색하고 학습시킬 수 있다. 본 시스템에서는 모델의 생성 및 성능평가를 위해 AutoML(Automated Machine Learning) 라이브러리인 Pycaret을 포함하고 있다. Pycaret의 compare_models 함수는 주어진 데이터에 널리 알려진 머신 러닝 알고리즘을 적용하여 성능 비교를 수행하므로 연구자가 개별 모델을 일일이 생성하고 성능을 비교하는 수고로움을 대폭 감소할 수 있다는 장점이 있다[10]. 이 과정에서 사용자는 예측 결과를 확인하며, 학습된 모델을 저장할 수 있다. 전체 시스템은 Pycaret 라이브러리 위에 구축되어 있으며, 데이터베이스와 연동되어 데이터와 모델을 효율적으로 관리한다.

이러한 구조를 통해 본 머신 러닝 모듈은 기존 Pycaret 라이브러리의 강력한 기능을 유지하면서도, 연구자들이 복잡한 코딩 없이도 쉽게 머신 러닝 워크플로우를 수행할 수 있도록 지원한다. 웹 기반 GUI를 통해 데이터 전처리부터 모델 학습, 평가, 저장에 이르는 전과정을 직관적으로 관리할 수 있다.

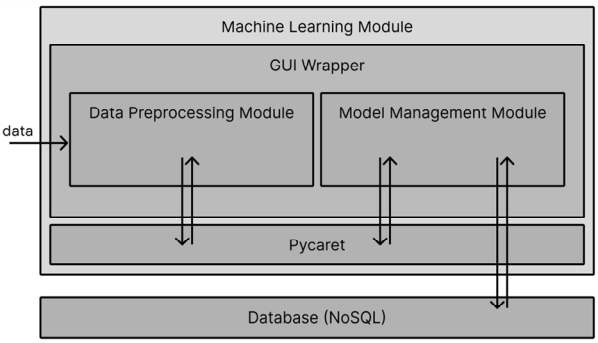


그림 4. 머신 러닝 모듈 구조도
Fig. 4. Architecture of machine learning module

3.4 시각화 모듈

시각화 모듈은 학습된 모델의 성능 평가지표와 메타데이터를 직관적으로 표현한다. 이를 통해 사용자는 모델의 정확도 및 정밀도, 재현율 등 다양한 평가 메트릭을 시각적으로 분석할 수 있으며, 모델의 특성과 행동을 더 깊이 이해할 수 있다.

시각화 모듈은 데이터베이스(NoSQL)에 저장된 다양한 머신 러닝 관련 데이터를 효과적으로 시각화하는 기능을 제공한다. 이 모듈은 학습에 사용된 데이터셋, 학습된 모델의 성능 지표, 그리고 관련 메타데이터를 데이터베이스로부터 조회하여 다양한 형태의 시각적 표현을 생성한다. 시각화 기능은 Pycaret과 scikit-learn 라이브러리의 기본 시각화 도구를 활용하며, 이를 통해 데이터의 분포, 상관관계, 모델 성능 평가 등을 직관적으로 표현할 수 있다. 특히 본 모듈은 웹 기반 GUI를 통해 제공되어, 사용자가 코딩 없이도 데이터 시각화를 수행할 수 있도록 설계되었다. 사용자는 필요에 따라 시각화 파라미터를 조정하거나 새로운 시각화 방식을 추가하는 등의 개인화가 가능하다. 이러한 웹 기반 인터페이스는 데이터 과학 분야의 비전문가들도 복잡한 데이터 패턴을 쉽게 이해하고 분석할 수 있도록 지원한다.

Ⅳ. 사례 연구

개발한 플랫폼의 효용성을 검증을 위해 유변물성 분석 연구에 본 시스템을 도입하였다. 대상 연구의 핵심 목표는 리튬이온 이차전지 음극 전극을 만들기 위한 슬러리의 탄성계수(Storage modulus) 예측에 있다. 이를 위해 슬러리 제조에서 탄성계수에 영향을 미치는 요소인 고분자 바인더 함량(CMC, Carboxymethyl Cellulose), 고분자 바인더 분자량(CMC-M, Carboxymethyl cellulose Molecular weight), 도전재 함량(CB, Carbon Black), 활물질 함량(G, Graphite), 진동 변형률(Oscillation strain)을 특징 변수, 탄성계수를 목적 변수로 설정하여 회귀분석을 수행하였다.

해당 실험 파일은 물리적으로 떨어진 두 연구그룹의 14명의 연구자로부터 수집되었다. 수집기간은 2024년 1월 1일부터 2024년 11월 30일이며 수집된 총 파일의 개수는 400여 건이다. 다양한 형식의 파일로부터 데이터를 효과적으로 추출하기 위해, 파일 형식과 종류에 특화된 Extracter와 Parser를 개발하여 시스템에 추가하였다. 실험 관리 모듈은 유변물성 연구의 특성을 고려하여 다양한 종류의 데이터를 관리할 수 있도록 개인화되었다. ‘실험’ 구조는 파일에서 추출한 데이터뿐만 아니라 연구에 필요한 추가 데이터도 포함하도록 추가했으며, 그림 5와 같이 연구자들의 편의를 위해 상세한 조건 검색 기능을 구현하여 ‘실험’ 데이터의 조회 및 검색의 용이성을 높였다.

개발된 시스템은 두 개의 물리적인 노드(Node)에 나누어 배포되었다. 사용자의 수가 많지 않은 만큼, 메인 노드에는 시스템 전체를 배포하여 불필요한 추가적인 분산처리 환경을 고려하지 않았다. 보조 노드에는 백업용 데이터베이스를 배포하여, 메인 노드에 있는 데이터베이스와 레플리카 셋(Replica set)으로 구성하여 데이터 손실을 방지하였다.

전처리 과정은 크게 유효값 검증과 로그 변환으로 구분된다. 탄성계수의 경우 양수 값 범위가 유효한 값의 범위이므로, 탄성 계수가 음수인 데이터는 제거하였다. 탄성 계수가 큰 범위에 넓게 분포하고 있어서, 학습 과정에서 손실 함수의 편차를 줄이기 위해 $\log_{10}(x+1)$ 함수를 적용하여 데이터를 정규화했다.

다양한 회귀 알고리즘 중 본 연구에서는 LGBM(Light Gradient Boosting Machine), GBR (Gradient Boosting Regressor), RFR(Random Forest Regressor), DTR(Decision Tree Regressor), KNR(K-Neighbors Regressor)을 도입하였다. 모델 학습 전 총 5개의 특징 변수가 예측에 미치는 영향도를 파악하기 위해 순열 중요도(Permutation Importance)를 측정했으며 그 결과 CMC-M과 CMC가 각각 1.6, 1.2로 0.2 이하인 다른 특징 대비 가장 높은 영향도를 보여주었다.

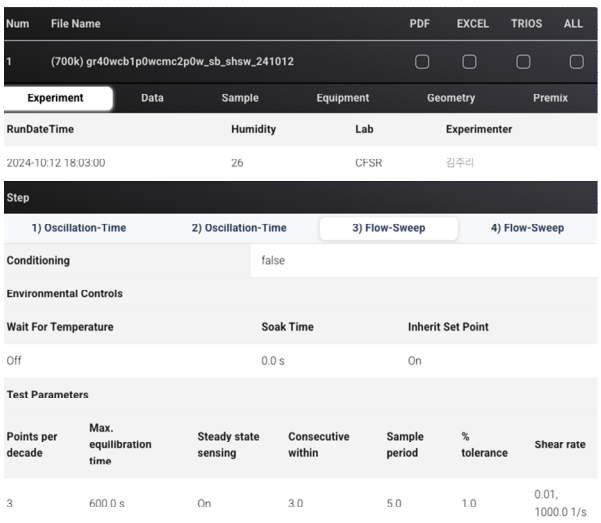


그림 5. 유변 물성 실험 조건 검색 페이지
Fig. 5. Web page for searching experiment of rheometers

해당 각 모델에 대한 성능지표는 표 2와 같다. 1에 가까울수록 모델이 데이터를 잘 설명하고 있음을 나타내는 지표인 R2를 기준으로 가장 성능이 좋은 모델은 LGBM인 것으로 나타났다. 또한, Pycaret의 tune_model 함수를 이용하여 최적화를 수행한 결과 해당 모델에서 과적합/과소적합 조절에 영향을 미치는 핵심 파라미터인 learning_rate, num_leaves, min_child_samples는 각각 0.3, 80, 51로 선정되었음을 확인하였다.

표 2. Storage modulus 예측 성능
Table 2. Storage modulus prediction performances

Model	MAE	MSE	R2	RMSLE	MAPE
LGBM	0.413	0.094	0.954	0.242	0.500
GBR	0.432	0.094	0.953	0.245	0.542
RFR	0.535	0.138	0.939	0.288	0.637
DTR	0.611	0.197	0.908	0.327	0.698
KNR	0.828	0.334	0.857	0.403	0.954

일반적으로 R2가 0.9 이상일 때 해당 모델이 매우 우수한 설명력을 갖는다고 평가되므로 학습된 모델은 데이터를 잘 표현한다고 할 수 있다. 또한, 0.242의 RMSLE는 해당 연구그룹이 유의미하다고 판단되는 수준의 정확도이다. 한편, 해당 연구는 회귀 정확도와 더불어 진동 변형률에 따른 탄성계수의 계형을 로그 스케일(Log-scale)로 확인하는 것이므로 그림 6에 보인 시각화 진행하였다. 그 결과 그림과 같이 예측값 및 그 계형이 실제와 거의 유사함을 확인할 수 있다.

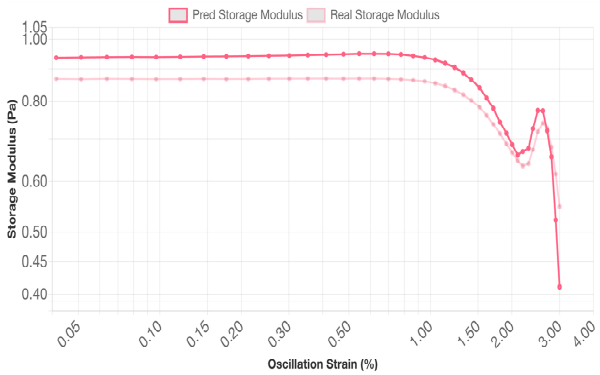


그림 6. 실제 탄성 계수 값과 예측 값 비교 그래프
Fig. 6. Graph for comparing real and predicted values of storage modulus

앞서 보인 성공적인 분석 결과는 데이터 과학에 대한 전문적 지식이 없는 연구자도 손쉽게 머신 러닝 기술을 도입할 수 있음을 시사한다. 또한, 특징 변수와 목적 변수를 선정하여 다양한 머신 러닝 알고리즘을 적용하는 일반적인 워크플로우는 도메인에 구분 없는 공통적인 절차이므로 파일에서 데이터를 추출하는 기능만 개인화할 경우 본 시스템을 다양한 연구에 활용할 수 있을 것으로 기대한다.

V. 결 론

본 연구에서는 웹 기반 분산파일 통합관리 시스템 및 자동화 인공지능 플랫폼을 제안하였다. 개발 시스템은 웹 기반 GUI를 통해 복잡한 머신 러닝 워크플로우를 코딩 없이 수행할 수 있도록 지원함으로써, 데이터 과학 비전문가도 쉽게 인공지능 기술을 활용할 수 있게 하였으며 사례 연구를 통해 이를 입증하였다.

특히 파일 관리 모듈과 실험 관리 모듈을 통한 효과적인 데이터 통합 관리 기능은 다양한 형식의 분산된 파일로부터 정보를 효율적으로 통합하는 데 크게 기여하였다. 뿐만 아니라 NoSQL 데이터베이스의 활용과 모듈화된 시스템 구조를 통한 높은 수준의 확장성과 유연성은 다양한 연구 분야와 데이터 유형에 대응할 수 있는 능력을 제공한다.

본 연구에서는 리튬이온 이차전지 소재 데이터 분석에 관한 사례를 제시했으나 향후 다양한 도메인에 적용하여 개발 시스템의 일반화를 검증해야 한다. 또한, 정형데이터에 대한 분석과 더불어 이미지나 음성과 같은 비정형 데이터에 대한 분석기능을 추가하여 적용 가능한 도메인을 확장할 필요가 있다. 추가적으로 더 많은 머신 러닝 알고리즘과 딥러닝 모델을 지원하고 하이퍼 파라미터 튜닝, 특성 엔지니어링 과정의 자동화 수준을 높여 사용자 편이성 측면의 고도화를 진행할 예정이다.

References

[1] I. K. Nti, A. F. Adekoya, B. A. Weyori, and O. Nyarko-Boateng, "Applications of artificial intelligence in engineering and manufacturing: a

systematic review", Journal of Intelligent Manufacturing, Vol. 33, No. 6, pp. 1581-1601, Aug. 2022. https://doi.org/10.1007/s10845-021-01771-6.

[2] H. J. Kim, B. R. Kwon, J. Y. Han, Y. W. Sohn, and J. E. Lee, "Development and Application of Artificial Intelligence Image Recognition Technology for Innovation in Retail Services", Journal of Korea Service Management Society, Vol. 24, No. 5, pp. 24-39, Dec. 2023. https://doi.org/ 10.15706/jksms.2023.24.5.002.

[3] L. Yang, M. E. Rajab, A. Shami, and S. Muhaidat, "Enabling AutoML for Zero-Touch Network Security: Use-Case Driven Analysis", IEEE Transactions on Network and Service Management, Vol. 21, No. 3, pp. 3555-3582, Mar. 2024. https://doi.org/10.1109/TNSM.2024.3376631.

[4] S. Jain and E. Fallon, "UDNet: A Unified Deep Learning-Based AutoML Framework to Execute Multiple ML Strategies for Multi-Modal Unstructured Data Processing", IEEE Access, Vol. 12, pp. 77959-77975, May 2024. https://doi.org/10.1109/ACCESS.2024.3403724.

[5] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, "Implementing AutoML in Educational Data Mining for Prediction Tasks", Applied Sciences, Vol. 10, No. 1, Jan. 2020. https://doi.org/10.3390/app10010090.

[6] M. Wever, A. Tornede, F. Mohr, and E. Hüllermeier, "AutoML for Multi-Label Classification: Overview and Empirical Evaluation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 43, No. 9, pp. 3037-3054, Sep. 2021. https://doi.org/10.1109/TPAMI.2021.3051276.

[7] A. Y. Seong, S. Y. Kang, Y. G. Song, and G. W. Kim, "Health-AutoML: An Automatic Adaptive Multi-Layer Stacking Ensemble Learning Framework for Analyzing Healthcare Data", Journal of Korean Institute of Information Technology, Vol. 22, No. 1, pp. 23-37, Jan. 2024. https://doi.org/10.14801/jkiit.2024.22.1.23.

[8] S. Palanisamy and P. SuvithaVani, "A survey on RDBMS and NoSQL Databases MySQL vs MongoDB", 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, Jan. 2020. https://doi.org/10.1109/ICCCI48352.2020.9104047.

[9] Z. Aftab, W. Iqbal, K. M. Almustafa, F. Bukhari, and M. Abdullah, "Automatic NoSQL to relational database transformation with dynamic schema mapping", Scientific programming, Vol. 2020, No. 1, pp. 1-13, 2020. https://doi.org/10.1155/2020/8813350.

[10] J. Wu, H. Wang, C. Ni, C. Zhang, and W. Lu, "Data Pipeline Training: Integrating AutoML to Optimize the Data Flow of Machine Learning Models", 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE), Shanghai, China, pp. 730-734, Mar. 2024. https://doi.org/10.1109/ICAACE61206.2024.10549260.

저자소개

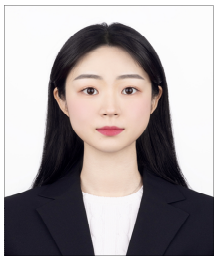
김 진 우(Jin-Woo Kim)

2023년 8월 : 국립금오공과대학교 소프트웨어공학과(학사)
2023년 9월 ~ 현재 : 국립금오공과대학교 소프트웨어공학과 석사과정
관심분야 : 웹, 클라우드, 머신러닝

박 준 동 (Jun-Dong Park)

2010년 2월 : 서울대학교 화학생물공학부(학사)
2016년 2월 : 서울대학교 화공생명공학부(공학박사)
2020년 9월 ~ 현재 : 숙명여자대학교 화공생명공학부 부교수
관심분야 : 유변학, 머신러닝

정 은 희 (Eun-Hui Jeong)



2022년 2월 : 국립금오공과대학교
화학공학과(학사)
2022년 3월 ~ 현재 :
숙명여자대학교
화공생명공학부 박사과정
관심분야 : 유변학, 머신러닝

김 성 렬 (Sung-Ryul Kim)

2010년 2월 : 부산대학교
컴퓨터공학과(학사)
2017년 8월 : 부산대학교 대학원
컴퓨터공학과(공학박사)
2019년 3월 ~ 현재 :
국립금오공과대학교
컴퓨터소프트웨어공학과 조교수
관심분야 : 빅데이터, 머신러닝