



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

빅데이터를 활용한 인공지능 기반
에너지 관리 시스템 개발



국립부경대학교대학원

인공지능융합학과

누머너브 일리요스벡 라크힘전 우글리

공 학 석 사 학 위 논 문

빅데이터를 활용한 인공지능 기반
에너지 관리 시스템 개발
(Development of an AI-based
energy management system using
Big data)

지도교수 김 태 국

이 논문을 공학석사 학위논문으로 제출함.
2025년 08월

국립부경대학교 대학원

인공지능융합학과

누머너브 일리요스백 라크힘전 우글리

누머너브 일리요스벡 라크힘전
우글리의 석사 학위논문을 인준함.

2025년 08월 22일



위원장 공학박사 장원두 (인)

위원 공학박사 김훈희 (인)

위원 공학박사 김태국 (인)

목차

I. 서론	1
1.1 연구 배경 및 목적	1
1.2 관련 연구	3
1.3 논문 개요	5
II. 이론적 배경	6
2.1 기존 에너지 관리 시스템의 한계	6
2.2 머신러닝 기반 전력 사용량 예측 연구 동향	7
2.3 기존 연구의 문제점 및 해결 필요성	8
2.4 머신러닝 모델 구현 및 아키텍처	9
2.4.1 LSTM	9
2.4.2 XGBoost	10
2.4.3 Prophet	12
2.4.4 RNN	13
2.4.5 Transformer	13
2.4.6 Time-MoE	14
2.4.7 PatchTST	15
2.4.8 TSMixer	16
2.5 본 연구의 차별성 및 기여점	18
III. 연구 방법	19
3.1 데이터 수집 및 구성	19
3.1.1 데이터 출처 및 특성	19
3.1.2 데이터 전처리 및 정규화 기법	23
3.2 모델 개발 및 학습 과정	29
3.2.1 전력 예측을 위한 최적의 머신러닝 모델 선정	29
3.2.2 하이퍼파라미터 튜닝 및 성능 개선 전략	30
3.2.3 모델 구현	33
3.3 시스템 설계 및 구현	43
3.3.1 전력 사용량 실시간 모니터링 시스템 구축	43
3.3.2 유저 시나리오	44
3.3.3 에너지 관리 대시보드 UI 설계	46
3.3.4 하이퍼파라미터 최적화 기능	51

3.3.5 결과 확인 테이블 및 대시보드	54
3.3.6 데이터 시각화	57
3.3.7 앙상블 모드	59
3.3.8 전 모델 재사용	62
IV. 연구 결과	64
4.1 모델 성능 평가 및 비교 분석	64
4.2 개발 및 실험 환경	73
V. 결론 및 향후 연구	75
5.1 결론	75
5.2 향후 연구	77
참고문헌	78



표 목차

표 1. LSTM_a 모델을 위한 최적의 하이퍼파라미터	35
표 2. LSTM_b 모델을 위한 최적의 하이퍼파라미터	36
표 3. LSTM_c 모델을 위한 최적의 하이퍼파라미터	37
표 4. LSTM_d 모델을 위한 최적의 하이퍼파라미터	37
표 5. XGBoost_a 모델을 위한 최적의 하이퍼파라미터	39
표 6. XGBoost_b 모델을 위한 최적의 하이퍼파라미터	40
표 7. XGBoost_c 모델을 위한 최적의 하이퍼파라미터	41
표 8. XGBoost_d 모델을 위한 최적의 하이퍼파라미터	42
표 9. 제주삼다수 공장 데이터세트 LSTM 모델 성능지표	64
표 10. 제주삼다수 공장 데이터세트 기반 다변량 LSTM 모델의 입력 변수 최적 조합	66
표 11. 제주삼다수 공장 데이터세트 XGBoost 모델 성능지표	66
표 12. 제주삼다수 공장 데이터세트 기반 다변량 XGBoost 모델의 입력 변수 최적 조합	68
표 13. 모로코 지역1 LSTM 모델 성능지표	69
표 14. 모로코 지역1 데이터세트 기반 다변량 LSTM 모델의 입력 변수 최적 조합	70
표 15. 모로코 지역1 XGBoost 모델 성능지표	70
표 16. 모로코 지역1 데이터세트 기반 다변량 XGBoost 모델의 입력 변수 최적 조합	72
표 17. HW 실험 환경	73
표 18. SW 개발 환경	74

그림 목차

그림 1. AI FEMS 구성도	2
그림 2. 제주 삼다수 공장 데이터셋의 특성 히스토그램	20
그림 3. 제주 삼다수 공장 데이터셋의 전력 사용량 히스토그램 및 박스플롯	20
그림 4. 제주 삼다수 공장 데이터셋의 전력 사용량 시각화	21
그림 5. 테투안시 데이터셋의 특성 히스토그램	22
그림 6. 테투안시 구역1의 전력 사용량 히스토그램 및 박스플롯	22
그림 7. 테투안시 구역1의 전력 사용량 시각화	23
그림 8. Z-Score 범위 시각화	24
그림 9. 제주 삼다수 공장 데이터의 변수 간 상관관계 히트맵	25
그림 10. 테투안시 데이터의 변수 간 상관관계 히트맵	26
그림 11. 제주 삼다수 공장 데이터셋의 특성 간 페어 플롯	27
그림 12. 테투안시 데이터셋의 특성 간 페어 플롯	28
그림 13. 제주 삼다수 공장 데이터셋의 변수 조합	31
그림 14. 테투안시 구역1의 변수 조합	31
그림 15. 하이퍼파라미터 조합	32
그림 16. 전체 프로젝트 구성 다이어그램	44
그림 17. 사용자 시나리오	45
그림 18. 데이터베이스 스키마	46
그림 19. 사용자 안내 투어 기능	47
그림 20. 사용자 안내 투어 기능	47
그림 21. 로그인 화면	48
그림 22. 메인 화면	48
그림 23. AI 존 화면 1	49
그림 24. AI 존 화면 2	49
그림 25. AI 존 화면 4	50
그림 26. AI 존 화면 3	51
그림 27. 하이퍼파라미터 최적화 화면 1	52
그림 28. 하이퍼파라미터 최적화 화면 2	52
그림 29. 하이퍼파라미터 최적화 화면 3	53
그림 30. 하이퍼파라미터 최적화 화면 4	54
그림 31. 대시보드 화면 1	55
그림 32. 대시보드 화면 2	55
그림 33. 대시보드 화면 3	56

그림 34. 대시보드 화면 4	57
그림 35. 데이터 시각화 및 분석 화면 1	58
그림 36. 데이터 시각화 및 분석 화면 2	58
그림 37. 데이터 시각화 및 분석 화면 3	59
그림 38. 앙상블 모드 화면 1	60
그림 39. 앙상블 모드 화면 2	60
그림 40. 앙상블 모드 화면 3	61
그림 41. 앙상블 모드 화면 4	61
그림 42. 앙상블 모드 화면 5	62
그림 43. 전 모델 사용 기능	63



수식 목차

수식 1	9
수식 2	9
수식 3	9
수식 4	10
수식 5	10
수식 6	10
수식 7	11
수식 8	13



감사의 글

본 연구를 마무리하는 이 시점에서, 지금까지 제 곁에서 아낌없는 지도와 따뜻한 응원을 보내주신 모든 분들께 진심으로 감사의 말씀을 드립니다.

먼저, 본 연구가 올바른 방향으로 나아갈 수 있도록 깊은 통찰과 따뜻한 배려로 지도해 주신 김태국 교수님께 가장 큰 감사를 드립니다. 교수님께서서는 연구 뿐만 아니라 인생의 여러 방향에서도 귀중한 조언을 아끼지 않으셨으며, 외국인 유학생인 제가 새로운 환경에 잘 적응하고 성장할 수 있도록 많은 이해와 격려를 보내주셨습니다. 교수님의 학문적 열정과 진심 어린 관심은 제 연구 여정에 있어 가장 큰 등불이었습니다.

또한, 논문 심사에 참여해주신 장원두 교수님과 김훈희 교수님께도 깊은 감사의 말씀을 전합니다. 연구실에서 함께 생활하며 서로에게 자극과 위로가 되어준 모든 동료들에게도 진심으로 감사드립니다. 평보 박사님은 제 아이디어가 실현 가능한 방향으로 나아갈 수 있도록 많은 기술적 조언을 해줬으며, 티러, 태오, 아크누라, 리엔시아를 비롯한 모든 선후배 연구생들과 나눈 일상의 대화와 협업은 단순한 연구를 넘어 깊은 유대감과 배움의 시간을 만들어주었습니다. 여러분과 함께한 시간은 이 논문을 넘어, 제 인생에 있어 소중한 자산입니다.

무엇보다도, 이 연구 여정을 묵묵히 응원해주신 저의 가족에게 가장 큰 감사를 드립니다. 특히, 멀리서도 늘 믿음과 사랑으로 저를 지지해주신 아버님, 어머님, 그리고 아내에게 이 자리를 빌려 깊은 마음을 전하고 싶습니다. 힘든 순간마다 가족의 믿음은 제게 다시 일어설 수 있는 힘이 되었고, 이 논문이 완성되기까지 가장 큰 원동력이었습니다.

이 논문이 저를 도와주신 모든 분들께 작은 보답이 되기를 바라며, 앞으로의 길에서도 받은 은혜를 잊지 않고 학문적 성실함과 겸손함으로 보답해 나가겠습니다.

빅데이터를 활용한 인공지능 기반 에너지 관리 시스템 개발

누머너브 일리요스백 라크힘전 우글리

국립부경대학교 대학원 인공지능융합학과

요 약

본 연구는 기업의 전력 사용량 데이터를 기반으로 인공지능(AI)을 활용한 예측 및 통합 에너지 관리 시스템을 개발하는 것을 목적으로 한다. 이를 통해 기업은 전력 소비를 효율적으로 조절함으로써 경제적 부담을 경감하고, ESG(Environmental, Social, and Governance) 목표 달성에도 기여할 수 있다. 특히, 피크 시간대의 전력 사용을 최적화하여 높은 기본요금을 피하는 것이 중요하며, 이를 위해 사무실 및 공장에 에너지 관리 시스템을 도입하고 있다. 본 연구에서 제안한 시스템은 피크 전력 예측 및 제어 기능을 통해 비용 절감, 이산화탄소(CO₂) 배출 저감, 이상 전력 사용 감지 등의 기능을 수행한다.

본 연구에서는 제주 삼다수 공장과 모로코 테투안시의 구역별 전력 소비 데이터를 대상으로, LSTM, XGBoost, Prophet, RNN, Transformer 등 다양한 시계열 예측 모델을 구현하고 그 성능을 비교하였다. 성능 평가는 결정계수(R^2), 평균절대백분오차(MAPE), 대칭평균절대백분율오차(sMAPE)를 기준으로 수행되었으며, 각 모델이 기존 기본 모델 대비 우수한 예측 정확도를 보임을 확인하였다. 제주 삼다수 공장 데이터에서는 LSTM 모델이 최대 68%, XGBoost 모델이 36%까지 MAPE를 감소시켰으며, 모로코 지역 1번 데이터셋에서도 LSTM은 최대 58%, XGBoost는 60%까지 MAPE 감소 효과를 보였다. 반면, XGBoost 모델은 일부 지역 및 데이터 특성에 따라 예측 안정성에 한계를 드러내는 등 모델별 성능 편차도 관찰되었다. 아울러, 본 연구에서는 웹 기반 및 온프레미스(on-premise) 환경에서 작동 가능한 플랫폼을 개발하여, 사용자 회원가입 후 실제 현장에 적용할

수 있도록 구현하였다. 향후 연구에서는 기능 선택(feature selection)을 최적화하거나 고도화된 모델링 기법을 도입하여 예측 정확도를 더욱 향상시키는 방향으로 나아갈 예정이다.

본 연구는 에너지 관리 분야의 혁신을 도모하며, 기업들이 에너지 비용을 절감하고 환경적 책임을 실현하는 동시에 경제적 이익을 극대화할 수 있는 실질적 솔루션을 제시한다.



Development of an AI-based energy management system using Big data

Ilyosbek Numonov Rakhimjon ugli

Department of Artificial Intelligence Convergence,
Pukyong National University Graduate School

Abstract

This study aims to develop an AI-based predictive and integrated energy management system using corporate electricity consumption data. By doing so, companies can efficiently control their power usage, thereby reducing economic burdens and contributing to the achievement of ESG (Environmental, Social, and Governance) goals. In particular, optimizing electricity usage during peak hours is essential to avoid high base charges, and many offices and factories are adopting energy management systems to address this issue. The proposed system in this study performs functions such as peak power prediction and control, cost reduction, CO₂ emissions reduction, and detection of abnormal electricity usage.

The study applied various time series forecasting models, including LSTM, XGBoost, Prophet, RNN, and Transformer, to electricity consumption data from the Jeju Samdasoo factory and different zones in Tetouan, Morocco. Model performance was evaluated using R² (coefficient of determination), MAPE (mean absolute percentage error), and sMAPE (symmetric mean absolute percentage error). The results showed that each model demonstrated superior predictive accuracy compared to the baseline models. For the Jeju Samdasoo factory dataset, the LSTM model reduced MAPE by up to 68%, while the XGBoost model achieved a reduction of up to 36%. In the first dataset from the Morocco region, the LSTM model achieved a MAPE reduction of up to 58%, and XGBoost

reached up to 60%. However, performance variations across models were observed, with the XGBoost model showing limitations in prediction stability depending on regional and data characteristics.

In addition, a platform operable in both web-based and on-premise environments was developed, allowing user registration and real-world deployment. Future research will aim to improve predictive accuracy by optimizing feature selection and incorporating more advanced modeling techniques.

This study contributes to innovation in the field of energy management, offering a practical solution that enables companies to reduce energy costs, fulfill environmental responsibilities, and maximize economic benefits.



I. 서론

1.1 연구 배경 및 목적

4차 산업혁명이 가속되면서 산업 현장은 사물인터넷(IoT), 빅데이터, 인공지능(AI) 기술을 융합한 데이터 기반 의사결정 체계로 급속히 전환되고 있다. 전력 소비 또한 예외가 아니며, 특히 피크 시간대에 발생하는 과도한 전력 사용은 기업의 경제적 부담을 증폭시키는 핵심 요인으로 지목된다. 대한민국은 1978년 1월부터 기본요금 피크연동제¹⁾를 시행하여, 기업이 월간 최고 부하뿐 아니라 직전 12개월 하계(7-9월)·동계(12-2월) 최대부하 중 더 높은 값을 기준으로 기본요금을 산정하도록 규정한다. 이 제도는 단 한 번의 피크 초과가 이후 1년간 지속적인 고요금 부담으로 이어지는 구조다. 따라서 기업은 피크 전력 한도를 넘어서는 순간을 관리·예측하는 일이 곧 비용 절감과 ESG(Environment, Social, Governance) 경영의 필수 조건으로 자리 잡는다.

이에 따라 FEMS(Facility Energy Management System)[그림 1]와 같은 에너지 관리 솔루션의 필요성이 부각된다. FEMS는 실시간 모니터링, 피크 예측, 자동 제어 기능을 통해 피크 부하를 분산시키고 설비 효율을 높이는 역할을 수행한다. 그러나 기존 상용 시스템은 폐쇄적 구조, 고가의 라이선스, 한정된 맞춤형 기능 탓에 중소·중견기업이 쉽게 도입하기 어렵다. 정확한 예측 알고리즘과 재현 가능한 오픈소스 기반 플랫폼이 결합된 확장 가능한 에너지 관리 솔루션이 필요한 이유가 여기에 있다.

본 연구는 다음 세 가지 목표를 달성하고자 한다.

모델 구조 개선을 통한 예측 성능 향상. LSTM(Long Short-Term Memory), XGBoost(eXtreme Gradient Boosting), Prophet, Transformer 등 다양한 시계열 모델을 기반으로, 배치 정규화·이상치 처리·하이퍼파라미터 최적화 기법을 결합해 모델의 구조적 한계를 극복한다.

실데이터 기반 전력 소비 예측으로 산업 적용성 입증. 제주 삼다수 공장, 모로코 테투안 지역 전력 사용 데이터를 활용해 여러 산업·지역 환경에서 알고리즘의 실효성을 검증하고, 피크 부하 감소 효과를 정량화한다.

오픈소스 기반의 재현·확장 가능한 예측 플랫폼 제공: 웹·온프로미스 환경 모두에서 구동 가능한 플랫폼을 설계해 기업이 손쉽게 도입·커스터마이즈할 수 있도록 지원하며, 향후 SaaS 형태로 확장할 수 있는 기술적 토대를 마련한다.

이상의 연구를 통해 본 시스템은 피크 전력 제어, 비용 절감, CO₂ 배출 저감이라는 다층적 가치를 동시에 실현하고, 나아가 에너지 산업의 디지털 전환과 탄소 중립 달성을 촉진하는 핵심 인프라로 자리매김할 것을 목표로 한다.

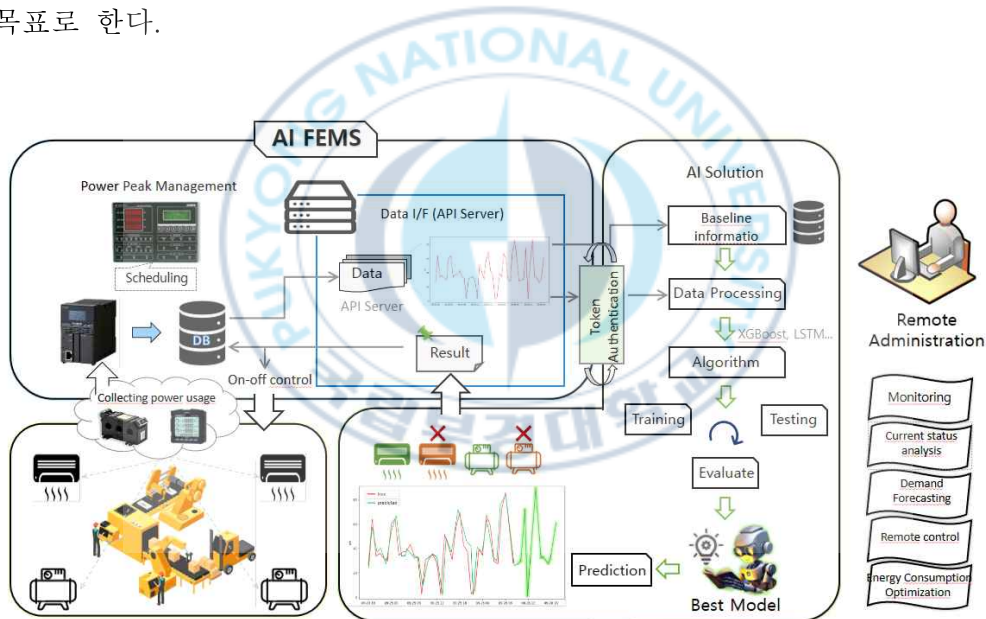


그림 1. AI FEMS 구성도

1.2 관련 연구

에너지 관리 시스템과 관련하여 국내에서는 다양한 연구가 활발히 진행되고 있다. 하지만 현재까지 무료로 공개된 솔루션은 없으며, 기업들은 이 기술의 초기 비용 문제 등으로 도입에 어려움을 겪고 있다. 이전 연구들을 살펴보면 다음과 같다. 중앙대학교 지능형 에너지산업융합학과 연구진들은 머신러닝의 통계적 학습 방법인 릿지 회귀와 라쏘 회귀를 사용하여 6개월 데이터를 기반으로 3개월 스팀 유량을 예측하는 연구를 수행했다. 이 연구는 적정 스팀 매입량을 예측하는 데 유용하다는 결론을 내렸다[2]. 고등기술연구원 에너지환경 IT융합그룹에스는 에너지 절감을 목적으로 LSTM 기반 전력부하 예측 시스템을 설계했다. 데이터 전처리 및 전력 데이터 분류 과정을 거친 후 LSTM 모델을 학습시켰으며, 이 모델은 평균 오차가 약 0.26으로 높은 예측 성능을 보였고, RNN(Recurrent Neural Network, 순환 신경망)보다 정확도가 높다는 결과를 얻었다[3]. 한국전자통신연구원에서는 LSTM을 이용해 시계열 데이터의 주기성과 패턴을 학습하여 전력 소비를 예측하는 연구를 진행했다[4]. 국립 창원대학교에서 진행된 연구는 마트 공장을 위한 전력 소비 예측 모델을 LSTM을 사용하여 개발했다. 연구 결과, 이 모델은 스마트 공장의 전력 소비 예측에 매우 효과적일 것으로 기대된다고 밝혔다[5]. 국립 순천대학교는 LSTM과 XGBoost 모델을 기반으로 전력 소비량 예측 모델을 비교 분석하는 연구를 수행했다[6]. 이안테크놀로지(주)는 건물 에너지 관리를 위해 에너지 소비량을 예측하는 시스템을 연구했다. VLSW(Variable Length Sliding Window) 알고리즘을 활용하여 적은 양의 데이터로 학습에 충분한 데이터를 생성한 후 SSIM(Sequence-to-Sequence Imputation Model)에 적용하여 예측 모델을 구현했다. 연구 결과, SSIM 모델은 기존 LSTM 모델보다 우수한 예측 성능을 보였으며, 결측치가 포함된 건물 에너지 데이터 세트의 예측 정확도를 향상시켰다[7]. “연결된 건물의 에너지 관리를 위한

IoT 시스템”에 관한 연구에서는 모든 건물이 운영 데이터를 중앙 집계자와 공유하고, 심층 강화 학습을 사용하여 HVAC(heating, ventilation, & air conditioning) 시스템의 에너지 최적화를 달성하는 방법을 제시한다[그림 2]. 물리 기반 모델과 학습 기반 모델을 결합하여 태양광 발전과 실제 건물 간의 편차를 최소화하고, ESS(Energy Storage System)를 통해 전력 피크 시간대에 에너지를 공급한다. Pecan Street 프로젝트의 실제 건물 데이터를 사용한 시뮬레이션을 통해 시스템을 검증했다[8]. 스마트 홈을 위한 에너지 최적화된 IoT 아키텍처 모델링 및 구현 논문에서는 자동 외부 조명, 자동 급수 및 자동 계단 조명을 포함한 시스템을 모델링하고, Arduino 카드 및 WiFi 모듈을 사용하여 구현한다. 테스트 결과는 리소스 사용을 효과적으로 최적화한다는 것을 보여준다[9]. AIoT(Artificial Intelligence of Things) 기반 스마트 홈 에너지 관리 시스템 논문에서는 IoT 기반의 관리 시스템을 통해 스마트 홈의 에너지 효율성을 향상시킨다. 센서 데이터를 머신러닝 기술로 분석하여 가전제품과 조명의 작동을 결정하며, 시스템은 에너지 소비를 줄이는 데 탁월한 예측 결과를 보인다[10]. 스마트 홈을 위한 IoT ML(Machine learning) 기반 시스템 논문에서는 에너지 효율성에 중점을 두고, 에어컨, TV, 조명 등을 자동으로 제어하는 시스템을 제안한다. 센서 데이터를 기계 학습 기술로 분석하여 가전제품의 작동을 결정하고, 시스템의 하드웨어 및 소프트웨어 구성을 논의한다[11]. IoT 활성화 스마트 홈을 위한 AI 기반 자동화의 잠재력 검토 논문에서는 AI 기반 자동화가 스마트 홈에 혁명을 일으킬 가능성이 있으며, 사용자 경험과 스마트 홈의 효율성을 향상시킬 수 있다. 규칙 기반 시스템과 기계 학습 알고리즘은 스마트 온도 조절 장치와 조명을 자동화하는 데 사용될 수 있다[12].

1.3 논문 개요

본 논문은 총 다섯 개의 장으로 구성되어 있으며, 각 장의 주요 내용을 다음과 같이 요약한다.

제1장 서론에서는 본 연구의 배경과 목적을 설명하고, 에너지 소비 최적화의 필요성과 인공지능 기반 예측 시스템의 중요성을 강조한다. 또한 관련 선행 연구를 검토하고, 본 논문의 구성 및 주요 기여점을 간략히 소개한다.

제2장 이론적 배경에서는 기존 에너지 관리 시스템의 한계와 머신러닝 기반 전력 예측 연구의 동향을 살펴본다. 이어서 본 연구가 사용하는 다양한 시계열 예측 모델(LSTM, XGBoost, Prophet, RNN, Transformer, Time-MoE, PatchTST, TSMixer 등)의 구조 및 특징을 소개하고, 기존 연구와의 차별성과 기여점을 정리한다.

제3장 연구 방법에서는 본 연구에 사용된 데이터의 출처와 특성, 전처리 및 정규화 기법을 설명한다. 또한 각 예측 모델의 개발 및 학습 과정, 하이퍼파라미터 최적화 전략, 실험 설계 방법을 구체적으로 기술하며, 실시간 모니터링 시스템 구축, 사용자 시나리오, 대시보드 UI 설계 등 실제 구현된 에너지 관리 플랫폼의 구성 요소들을 상세히 설명한다.

제4장 연구 결과에서는 제안한 각 모델의 예측 성능을 MAPE, sMAPE, R^2 등의 평가 지표를 기준으로 분석하고, 다양한 모델 간의 성능 비교 결과를 제시한다. 또한 모델 학습 및 실험 환경을 포함한 구현 세부 사항을 통해 실제 적용 가능성을 논의한다.

제5장 결론 및 향후 연구에서는 연구의 주요 결과를 요약하고, 본 연구가 학술적·실무적으로 갖는 의의를 제시한다. 마지막으로 기능 고도화, 고급 AI 모델의 추가 적용, 산업 현장 맞춤형 응용 확대 등 향후 연구의 발전 방향을 제안한다.

II. 이론적 배경

2.1 기존 에너지 관리 시스템의 한계

기존 에너지 관리 시스템은 주로 과거 데이터 기반의 규칙(rule based) 또는 단일 최적화 모델에 의존하기에 실시간 부하 변화와 외부 변수(기상·시장 가격 등)에 유연하게 대응하지 못한다. 특히 가정·상업용 프로슈머(prosumer) 환경에서 재생에너지의 간헐성을 통합하는 과정에서 최적인 스케줄링이 부정확하게 이루어져 과잉 생산 또는 공급 부족이 빈번하게 발생한다[13].

또한 분산형 에너지자원(DER) 관리 시 복수의 제어 전략 간 충돌이 발생해 시스템 전체의 에너지 균형을 저해하며, 이는 다양한 제어 알고리즘을 비교·분석한 연구에서도 지적된 바 있다[14]. 다른 논문에서는 계산 자원 및 통신 대역폭의 제약으로 인해 고도화된 머신러닝·딥러닝 모델을 실시간으로 구동하기 어려우며, 이로 인해 예측 정확도가 제한되는 문제가 발생한다고 지적하고 있다[15].

전력망 전반을 아우르는 표준화된 아키텍처가 부재하여 신규 설비나 스마트 미터 연동 시 통합 비용·시간 부담이 크다는 한계도 존재한다[16].

마지막으로 사이버 보안 취약성 해결과 사용자 친화적 UI/UX(User Interface, User Experience) 제공이 미흡해 운영자 및 최종 사용자의 의사결정 지원이 부족하며, 이는 전력 배전시스템 관점에서의 한계를 종합적으로 다른 논문에서도[17] 강조된다.

2.2 머신러닝 기반 전력 사용량 예측 연구 동향

최근 에너지 관리 분야에서는 머신러닝과 딥러닝 기술을 활용한 전력 사용량 예측 연구가 활발히 진행되고 있다. 예측 연구가 전통적 기계학습에서 심층학습, 그리고 트랜스포머 계열로 진화해 온 과정을 살펴볼 수 있다. 초기에는 선형 회귀, 결정트리, 랜덤포레스트, XGBoost 같은 전통적 알고리즘이 해석 용이성과 낮은 계산 비용 덕분에 널리 활용되었으나 MAE(Mean Absolute Error)와 RMSE(Root Mean Square Error)가 10 - 20% 수준에 머물며 비선형성과 장기 시계열 의존성 학습에는 한계를 보였다[18].

RNN 계열인 LSTM과 GRU(Gated Recurrent Unit)는 단기 변동 포착에 강점을 드러냈으나, 예측 구간이 길어질수록 기울기 소실로 성능 저하가 관찰되었다[19]. 이에 Transformer 기반 모델인 Informer, Autoformer, H-Transformer 등이 희소 어텐션을 도입해 장기 의존성 학습 능력을 강화함으로써, 수십 시간 이상의 예측에서도 RNN 대비 10 - 25% 낮은 오차를 달성한 논문도 있다[20],[21].

최신 연구 동향으로는 LSTM-Informer 앙상블, ISt-LSTM-Informer, WNPS-LSTM-Informer 등 스택킹 기법을 적용한 하이브리드 모델이 단·장기 예측 모두에서 단일 모델 대비 평균 MSE·MAE를 15 - 35% 개선하는 성과를 보인다[22],[23]. 전이학습을 적용해 적은 데이터 환경에서도 초기 성능을 확보하고 학습 비용을 절감하는 연구가 활발히 진행되고 있다[24]. 앞으로는 그래프 기반 트랜스포머, 연속 학습, 페더레이티드 러닝을 통해 공간·시간적 상호 의존성을 동시에 고려하고, 개인정보 보호 및 분산 데이터 처리를 강화하는 방향으로 연구가 확장될 것으로 보인다.

2.3 기존 연구의 문제점 및 해결 필요성

기존의 에너지 관리 시스템은 주로 단순한 통계적 기법에 의존하여 전력 소비를 예측하는 데 한계가 있다. 이러한 방법들은 비선형성, 이상치, 계절성 등 복잡한 전력 사용 패턴을 충분히 반영하지 못하며, 이로 인해 에너지 효율성 향상과 비용 절감에 제약이 발생한다. 특히, 산업 및 상업용 건물에서는 다양한 외부 요인(예: 기상 조건, 설비 가동률, 인원 밀집도 등)이 전력 소비에 영향을 미치므로, 보다 정교한 예측 모델이 요구된다.

최근에는 머신러닝과 딥러닝 기술의 발전으로 이러한 한계를 극복하려는 연구가 활발히 진행되고 있다. 이러한 기술들은 대규모의 시계열 데이터를 분석하여 전력 소비 패턴을 학습하고, 이를 기반으로 높은 정확도의 예측을 수행할 수 있다. 특히, LSTM, XGBoost, Prophet, RNN, Transformer, Time-MoE, PatchTST, TSMixer 등 다양한 모델들이 전력 예측에 적용되어 그 성능을 입증하고 있다. 예를 들어, LSTM과 XGBoost 모델을 활용하여 건물의 에너지 소비를 예측하였으며, LSTM 모델이 단기 예측에서 우수한 성능을 보였고, XGBoost는 장기 예측에서 높은 정확도를 나타냈다[25]. 또한, 다양한 머신러닝 알고리즘을 비교 분석하여, 딥러닝 기반 모델들이 전통적인 통계 모델보다 전력 예측에 더 효과적임을 확인하였다[26]. 이러한 연구들은 머신러닝 기반의 전력 예측 모델이 기존의 통계적 방법보다 우수한 성능을 보이며, 에너지 관리 시스템의 효율성을 향상시킬 수 있음을 시사한다. 따라서, 본 연구에서는 다양한 머신러닝 모델을 비교 분석하여 최적의 전력 예측 모델을 선정하고, 이를 기반으로 한 에너지 관리 시스템의 개발 및 구현을 목표로 한다.

2.4 머신러닝 모델 구현 및 아키텍처

2.4.1 LSTM

LSTM 네트워크의 학습 과정은 RNN의 한계를 극복하기 위한 단계적인 절차로, 시퀀스 데이터에서 장기 의존성을 효과적으로 학습할 수 있도록 개발되었던 모델이다[27],[28].

$$\text{Input gate } (i_1) = \text{Sigmoid}(W_i \cdot [h_0, x_1] + b_i) \quad (\text{수식 1})$$

$$\text{State update } (\hat{c}_1) = \tanh(W_g \cdot [h_0, x_1] + b_g) \quad (\text{수식 2})$$

첫째, 입력 시퀀스 [수식 1], [수식 2] 데이터가 네트워크에 제공되면, 각 시점의 입력은 입력 게이트, 포갯 게이트, 출력 게이트를 통해 처리된다[29]. 입력 게이트는 현재 입력 값과 이전 시점의 은닉 상태를 기반으로 새로운 정보를 얼마나 반영할지를 결정하며, 포갯 게이트는 이전 시점의 셀 상태를 얼마나 유지할지를 결정한다. 출력 게이트는 현재 시점의 셀 상태를 기반으로 최종 은닉 상태를 계산한다.

$$\text{Forget gate } (f_1) = \text{Sigmoid}(W_f \cdot [h_0, x_1] + b_f) \quad (\text{수식 3})$$

둘째, 셀 상태와 은닉 상태의 업데이트가 이루어진다 [수식 3]. 셀 상태는 LSTM의 핵심 요소로, 중요한 정보를 장기간 유지할 수 있도록 돕는다. 현재 시점의 새로운 정보와 이전 셀 상태를 결합하여 새로운 셀 상태를 계산하며, 이 과정에서 포갯 게이트와 입력 게이트의 값이 반영된다. 은닉 상태는 셀 상태의 축약된 형태로, 출력 게이트를 통해 계산되며, 이는 다음 시점의 입력으로 전달된다[30].

$$Cell\ state(C_1) = f_1 \odot C_0 + i(t_1) \odot \hat{c}(t_1) \quad (\text{수식 4})$$

셋째, LSTM 네트워크의 출력 값과 실제 목표 값 간의 손실을 계산하고, 이를 통해 역전파 과정을 수행한다 [수식 3]. 손실 함수는 주로 MSE 나 교차 엔트로피(CE)와 같은 함수가 사용되며, 손실 값이 최소화되도록 네트워크의 가중치를 업데이트한다. 역전파 과정은 그래디언트 소실 문제를 완화하기 위해 게이트 단위로 그래디언트를 계산하여 가중치를 수정한다[31].

$$Output\ gate\ (o_1) = Sigmoid(W_o \cdot [h_0, x_1] + b_o) \quad (\text{수식 5})$$

$$Hidden\ state\ (h_1) = o_1 \odot \tanh(C_1) \quad (\text{수식 6})$$

마지막으로, 반복 학습을 통해 네트워크의 성능을 향상시킨다 [수식 5], [수식 6]. 다수의 에폭(epoch)을 통해 학습 데이터를 여러 번 반복하여 네트워크가 점진적으로 학습하도록 하며, 학습률, 배치 크기 등의 하이퍼파라미터를 조정하여 최적의 학습 결과를 도출한다. 이러한 단계적인 학습 과정을 통해 LSTM 네트워크는 시퀀스 데이터의 장기 의존성을 효과적으로 학습할 수 있으며, 시계열 예측, 자연어 처리 등 다양한 분야에서 뛰어난 성능을 발휘한다.

2.4.2 XGBoost

XGBoost는 eXtreme Gradient Boosting의 약자로 Tianqi Chen이 개발한 그래디언트 부스팅 프레임워크의 매우 효율적이고 확장 가능한 프레임워크 중 하나다. 그래디언트 부스팅 자체에서 앙상블에 예측 변수를 순차

적으로 추가하여 각 예측 변수는 이전 변수의 오류를 수정한다. 이를 통해 모델 성능을 향상시키며, 또한, XGBoost는 다음과 같은 몇 가지 고급 기능을 통해 이 예측 성능을 향상시킨다[32].

- 1) Regularization: L1 및 L2 정규화를 통합하여 과적합을 방지한다.
- 2) Sparsity Awareness: 실제 데이터셋에서 흔히 볼 수 있는 희소 데이터를 효율적으로 처리한다.
- 3) Weighted Quantile Sketch: 가중 데이터를 관리하고 대규모 데이터 세트의 대략적인 분위수를 계산한다.
- 4) Parallel Learning: 최신 멀티코어 CPU 및 분산 환경을 활용하기 위한 병렬 처리에 최적화되어 있다.
- 5) Tree Pruning: 최대 깊이 매개변수를 사용하여 나무를 가지치기하여 일반화를 개선한다.

$$L(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (\text{수식 7})$$

XGBoost의 학습 단계는 다음과 같이 진행된다. 먼저, 데이터 전처리 과정을 통해 결측값 처리, 범주형 데이터 인코딩, 특성 스케일링 등을 수행하여 데이터를 학습에 적합한 형태로 변환한다. 이후, 기본 학습기로 사용할 결정 트리를 초기화하고, 손실 함수를 정의하여 모델의 예측 성능을 평가한다[33], [수식 7]. 각 단계에서 새로운 결정 트리를 추가하여 이전 모델의 오차를 보완하며, 이를 통해 예측 값을 업데이트한다. 이 과정에서 학습률을 통해 모델의 업데이트 강도를 조절하고, 과적합을 방지하기 위해 정규화 및 가지치기 작업을 수행한다. 이러한 과정을 반복하여 일정 횟수만큼 트리를 추가하거나 더 이상 손실 함수의 개선이 없을 때까지 학습을 진행한다. 마지막으로, 교차 검증을 통해 최적의 하이퍼파라미터 조합을 찾아내어 모델의 성능을 최적화한다. Xgboost도 위에 언급된 LSTM 모델의 특성 및 조합 자동 선택 방식으로 개발하였다.

2.4.3 Prophet

Prophet는 Meta에서 개발한 시계열 데이터 예측을 위한 모델로, 비전문가도 쉽게 사용할 수 있도록 설계되었다.

- 1) Trend ($g(t)$): Logistic 모델을 기반으로 한 포화성장 모델과 Piecewise 선형 모델의 두 가지 구성 방식으로 비주기적인 변화를 반영한다.
- 2) 계절성($s(t)$): 푸리에 계열을 사용하여 일반적으로 매주 또는 매년 주기적인 변화를 캡처한다.
- 3) 휴일($h(t)$): 불규칙한 휴일 효과를 설명하고 각 휴일을 지수 함수로 모델링된 영향을 갖는 독립 모델로 처리한다.
- 4) 오류($\epsilon(t)$): 모델에서 다루지 않는 무작위 효과를 나타내며 일반적으로 정규 분포를 따른다.

이러한 구성 요소를 장착함으로써 Prophet은 시계열 데이터에 대한 최종 예측 값을 제공하여 변경 사항, 이상 현상 및 휴일과 같은 특별 이벤트를 효과적으로 처리한다[34].

Prophet의 학습 단계는 다음과 같이 진행된다. 먼저, 시계열 데이터를 준비 및 전처리하여 이 과정에서는 결측값 처리, 데이터 형식 변환 등을 수행하여 데이터를 Prophet 모델에 적합한 형태로 만든다[35]. 이후, 데이터의 전반적인 추세(trend)와 주기적 패턴(seasonality)을 모델링한다. 추세는 데이터의 장기적인 변화 방향을 나타내며, Prophet은 이를 선형 또는 비선형 성장 함수로 표현한다. 주기적 패턴은 연간, 주간, 일간 등의 주기를 가지는 변동성을 나타내며, Fourier series를 사용하여 주기성을 모델링한다. Prophet은 또한 공휴일 및 이벤트와 같은 외생 변수의 영향을 고려한다. 사용자는 특정 날짜나 기간을 지정하여 이들 이벤트가 예측에 미치는 영향을 모델에 반영할 수 있다.

2.4.4 RNN

순환 신경망(RNN)은 시계열 데이터의 시간적 순서를 고려하여 숨겨진 상태(hidden state)를 갱신하는 구조이다. 시간 t 에서 입력 x_t 과 이전 은닉 상태 h_{t-1} 를 받아 새로운 은닉 상태 h_t 를 계산하는 식은 일반적으로 다음과 같다[36]:

$$h_t = \phi(W_h h_{t-1} + W_x x_t + b) \quad (\text{수식 8})$$

여기서 ϕ 는 활성화 함수(예: tanh, ReLU 등)이며, W_h, W_x, b 는 학습 가능한 파라미터이다. RNN은 이렇게 unfold된 형태로 나타내며, 과거 정보를 h_{t-1} 를 통해 순차적으로 전달함으로써 먼 과거의 이벤트 간 상관관계를 학습할 수 있다.

- 1) 시간적 종속성 처리: RNN은 은닉 상태를 통해 순차적으로 정보를 전파하여 멀리 떨어진 시점 간의 의존성을 학습할 수 있다.
- 2) 그래디언트 소실 및 폭주: 역전파 시의 곱셈 연쇄로 인해 장기 의존성을 학습할 때 그래디언트가 기하급수적으로 작아지거나 커지는 문제가 발생한다. 즉, 시퀀스 길이가 길어질수록 장기 기여도는 0으로 수렴해 버려 그래디언트 손실 현상이 나타나 학습이 어려워진다. 반대로 폭주 현상은 드물지만 기울기가 매우 커지는 문제를 말하며, 이는 그래디언트 클리핑 등으로 완화할 수 있다[37].

2.4.5 Transformer

트랜스포머는 순환이나 합성곱 없이 자기-어텐션(self-attention) 메커니즘만으로 구성된 모델이다. 입력 시퀀스의 모든 위치 간에 직접적인 연결을 형성하며, 각 위치 간의 연관성을 쿼리-키-값 메커니즘으로 계산한다.

또한, 멀티 헤드 어텐션을 사용해 Q,K,V를 여러 개의 하위 공간으로 선형 투영하여 병렬적으로 어텐션을 수행하고, 이들 출력을 다시 합성하는 구조를 갖는다. 이로써 모델은 서로 다른 표현 공간(subspace)에서 동시에 주의(attend)할 수 있다[38].

- 1) 병렬 처리 구조: 트랜스포머는 순차 연산 없이도 전체 시퀀스의 모든 위치를 동시 계산할 수 있어, RNN보다 훨씬 병렬화가 용이하다. 예를 들어 RNN은 각 시간 단계마다 연속적으로 처리해야 하지만, 트랜스포머는 모든 토큰의 어텐션을 동일 레이어에서 한 번에 계산한다. 이로 인해 학습 및 추론 속도가 크게 향상된다.
- 2) Positional Encoding (위치 인코딩): 트랜스포머는 순서 정보를 직접 다루지 않기 때문에, 입력 토큰에 위치 정보를 더해 주어야 한다. 원본 논문에서는 사인/코사인 함수를 이용한 위치 인코딩을 사용하여 연속된 시간 스텝 간의 상대적 거리를 선형 함수를 통해 표현할 수 있도록 하였다. 실제 구현에서는 학습 가능한 위치 임베딩과 비교하여 유사한 성능을 보였지만, 사인·코사인 방식은 훈련에서 보지 못한 더 긴 시퀀스 길이에 대해서도 일반화 가능하다고 제안되었다.
- 3) 계산 복잡도: 자기-어텐션의 시간 복잡도는 시퀀스 길이 n 에 대해 $O(n^2 d)$ 이다. 이는 RNN $O(n d^2)$ 보다 시퀀스가 짧을 때 효율적이지만, 시퀀스 길이가 길어지면 메모리와 연산 비용이 급증한다. 논문에서는 시퀀스 길이가 표현 차원보다 작을 때 어텐션이 RNN보다 빠르고 기술한다. 또한 각 레이어가 모든 위치를 직접 연결하여 최대 경로 길이를 1로 줄이므로, 장기 의존성 학습이 쉬워지고 경로 길이 측면에서 RNN보다 유리하다[39]. 긴 시퀀스의 경우 계산 비용을 줄이기 위해 국소 어텐션처럼 이웃 제한(self-attention restricted) 구조를 활용할 수도 있다.

2.4.6 Time-MoE

Time-MoE는 시계열 예측을 위한 거대 모델(foundation model)로, 혼합 전문가(Mixture-of-Experts, MoE) 구조를 접목한 Transformer 기반 아키텍처이다. 각 MoE 레이어는 다수의 “전문가” 네트워크(예: 피드포워드 네트워크) 풀을 보유하며, 스파스 게이팅(sparse gating)에 따라 입력 토큰마다 일부 전문가만 활성화한다. 예를 들어, 입력 벡터가 들어오면 게이트 네트워크가 해당 토큰에 가장 적합한 전문가를 선택(또는 복수 선택)하여 연산을 수행하고, 나머지 전문가는 비활성 상태로 둔다. 이렇게 하면 모델 용량은 크면서도, 실제로 활성화되는 파라미터 수는 적게 유지되어 추론 비용을 줄일 수 있다. Time-MoE는 디코더 전용 Transformer 구조를 사용하여 auto-regressive 방식으로 다음 시점을 예측한다. 토큰화는 시계열 값을 시점별로 임베딩하는 방식으로 진행되며, 여러 전문가로 구성된 스파스 FFN이 숨겨진 상태마다 적용된다. 이 모델은 방대한 시계열 데이터(Time-300B)로 사전학습되어 수십억 개의 파라미터 규모에도 불구하고 효율적인 추론을 달성하며, 동급의 밀집 모델보다 예측 정확도가 크게 향상된다[40].

2.4.7 PatchTST

PatchTST(Patch Time Series Transformer)는 멀티변수 시계열 예측을 위해 고안된 효율적인 Transformer 계열 모델이다. 입력 시퀀스를 일정 길이의 패치(patch)로 분할하여 각 패치를 Transformer의 입력 토큰으로 처리한다. 구체적으로, 연속된 시간 구간(예: 길이 64)의 데이터를 하나의 패치로 묶어 임베딩한 후 어텐션 레이어에 전달한다. 이러한 패치 분할 방식의 장점은 세 가지이다: (1) 각 패치 내 국소적 시계열 패턴(“지역 의미”)이 임베딩 단계에서 보존되고, (2) 동일한 look-back 윈도우에 대해 토큰 수가 줄어들어 어텐션 계산 비용이 제공적으로 감소하며, (3) 더 긴 과거 정보를 한 번에 다룰 수 있다는 점이다[41].

또한 채널 독립성(Channel Independence)을 유지한다는 점이 특징이다. 즉, 다변수 시계열의 각 채널(변수)은 자체의 단일 시계열로 간주하고, 모든 채널에 대해 동일한 임베딩 함수와 Transformer 가중치를 공유한다. 이를 통해 모델 파라미터를 크게 절감하면서도 각 채널의 특징을 학습한다. 예를 들어, M개의 채널이 있을 때 임베딩 및 Transformer 계층을 채널 수만큼 복제하지 않고 하나만 사용하되, 각 채널의 데이터를 독립적으로 처리한다. 결과적으로 PatchTST는 전통적인 Transformer보다 메모리 사용량과 연산량을 현저히 줄이면서도 장기 예측 능력을 갖춘 모델로 평가된다[42].

2.4.8 TSMixer

TSMixer(Time Series Mixer)은 순수 MLP 기반의 패치-기반 시계열 예측 모델로, Vision MLP-Mixer 아이디어를 시계열에 적용한 구조다. 입력 시계열을 패치로 분할하여 각 패치를 토큰으로 사용하며, 각 레이어는 패치 간(token) 및 채널(feature) 간의 정보를 교차 혼합한다. 구체적으로 토큰-믹서(token-mixing)와 채널-믹서(channel-mixing) MLP를 교차 사용한다. 한 믹서 블록에서는 먼저 패치 인덱스 방향으로 정보를 혼합하여 서로 다른 패치 간 상관관계를 학습하고(Inter-Patch Mixer), 이후 패치 내부의 각 특징 차원들 사이를 혼합(Intra-Patch Mixer)한다. 또한 다변수 시계열의 경우 채널 간 믹서(Inter-Channel Mixer)를 추가로 두어 서로 다른 변수들 간 상호작용을 포착할 수 있다[43].

1) 구조의 단순성과 학습 효율성: 모든 연산이 기본적인 선형계층과 MLP로 구성되어 있어 구조가 매우 단순하며, GPU/TPU 병렬처리에 최적화되어 학습과 추론이 빠르다. 실제로 TSMixer는 복잡한 자기-어텐션 블록 없이도 Transformer 및 기타 MLP 기반 모델보다 우수하거나 동등한 예측 성능을 보여준다.

- 2) 토큰-믹싱 vs 채널-믹싱: 각 레이어마다 토큰-믹서(패치 간 혼합)와 채널-믹서(특징 축 혼합)가 번갈아 수행된다. 즉, 토큰-믹서 블록이 n_{pn_p} 개의 패치 간 상관관계를 모델링하면, 채널-믹서 블록은 각 패치의 피쳐 축(또는 입력 채널)간 상관성을 모델링한다. 이런 이중 믹싱으로 시계열의 시간적 특징과 변수 간 관계를 모두 학습할 수 있다. 또한 IC-TSMixer 변형에서는 별도의 채널 간 믹서 모듈로 변수들 간의 명시적 상호작용을 추가로 처리한다.
- 3) 추가 기능: TSMixer는 중요한 정보를 강조하기 위해 각 믹서 블록 후에 간단한 게이티드 어텐션(Gated Attention) 계층을 둔다. 이 게이트는 중요한 특징의 크기를 키우고 불필요한 특징은 억제하여 장기적 상호작용 학습을 돕는다. 또한 패치의 계층적 구조와 채널 상관관계를 고려하여 예측 출력을 조정하는 온라인 재조정 헤드(reconciliation head)를 활용해 예측 성능을 높인다[44].



2.5 본 연구의 차별성 및 기여점

본 연구가 제시하는 인공지능 기반 통합 데이터 분석 플랫폼은 기존 연구들과 구별되는 다음과 같은 차별성과 기여점을 가진다.

첫째, 다양한 데이터 공학 기법을 통합하여 시계열 데이터의 전처리, 이상치 탐지, 결측치 보정, 정규화 등을 체계적으로 수행함으로써 모델 학습의 신뢰성과 정확도를 높인다[45],[46],[47]. 이러한 통합적 접근은 기존 연구들이 개별적으로 다루었던 전처리 과정을 하나의 체계적인 파이프라인으로 구성하여, 데이터 품질 향상과 모델 성능 개선에 기여한다[48].

둘째, 하이퍼파라미터 최적화를 위해 Optuna 기법을 활용하여 모델의 성능을 극대화하며, 특히 앙상블 모델에 적용하여 예측 정확도를 향상시킨다[49],[50].

셋째, 다양한 딥러닝 모델을 조합한 앙상블 학습 구조를 지원하여 단일 모델의 한계를 극복하고[51], 예측의 안정성과 일반화 능력을 강화한다. 이를 통해 단일 모델 대비 일관된 성능 향상을 달성하며, 노이즈가 많은 데이터 환경에서도 안정적인 예측 결과를 제공한다.

넷째, 온프로미스 환경에서 AI모델 예측이 가능하도록 하며, 이는 데이터 보호를 위해 민감한 정보 유출의 방지를 한다. 사용자의 데이터가 사용자 기기에서만 처리되기 때문에, 개인정보 유출 위험을 줄이고 보안성을 높이는 데 기여한다.

이러한 요소들은 기존 연구들과 차별화된 접근으로, 기술적 혁신성과 실용성을 동시에 확보하며, 전력 예측 및 피크 관리뿐만 아니라, 제조업, 금융, 헬스케어, 스마트시티 등 다양한 응용 분야에서의 활용 가능성을 제시한다.

III. 연구 방법

3.1 데이터 수집 및 구성

3.1.1 데이터 출처 및 특성

본 연구에서는 전력 사용량 예측을 위해 국내 제주특별자치도의 제주삼다수 공장 데이터[52], 제주지방기상청에서 제공되는 날씨 데이터[53]와 테투안 지역의 에너지 소비 데이터[54]를 활용하였다.

제주삼다수공장 시간별 전력사용량 데이터세트는 2023년 1월 1일부터 12월 31일까지 12개월간의 시간별 전력 사용량(kWh)으로 기록되어 있다. 제주지방 기상청 기상 정보도 같은 기간의 데이터를 활용하였으며, 기온, 누적 강수량, 풍향, 풍속, 현지 기압, 해면 기압, 습도 등 8개의 주요 변수를 포함한다. 기상 데이터는 분 단위로 수집되었으며, 전력 데이터와 정합을 위해 시간 단위 평균값으로 변환하여 사용하였다.

해외 데이터는 ‘양상불 학습을 통한 LSTM-Informer를 사용한 장기 전력 부하 예측’ 논문[55]에서 사용되었던 모로코 테투안 지역의 에너지 공급 기관 Amendis가 제공한 데이터를 사용한다. 위 데이터세트는 2002년부터 SCADA 시스템을 통해 수집된 52,416개의 10분 간격 에너지 소비 기록을 포함한다. 이 데이터는 온도, 습도, 풍속, 복사량, 구역별 전력 소비 등 총 9개의 변수로 구성되어 있다. 두 데이터세트는 서로 다른 기후와 사용 패턴을 포함하고 있어 모델의 일반화 성능 향상에 기여한다.

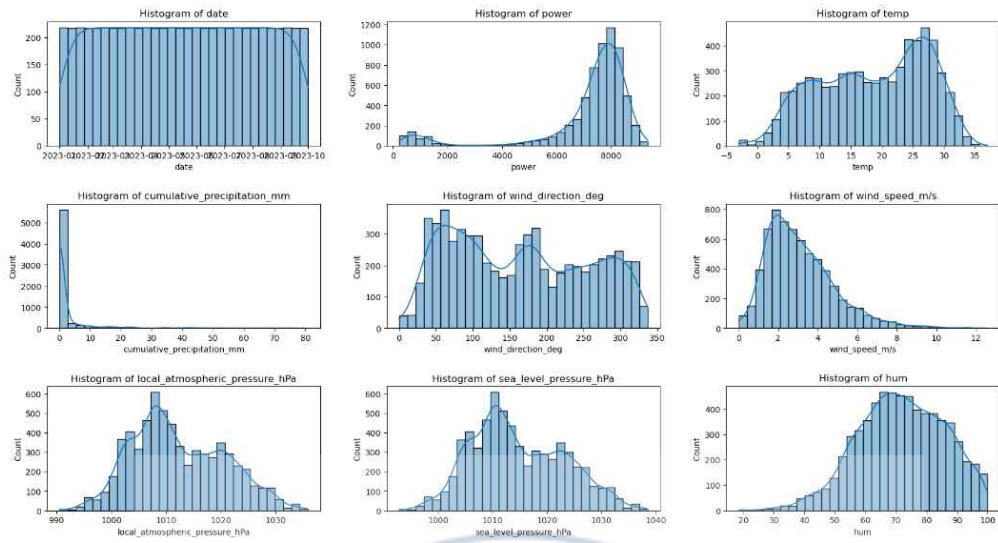


그림 2. 제주 삼다수 공장 데이터셋의 특성 히스토그램

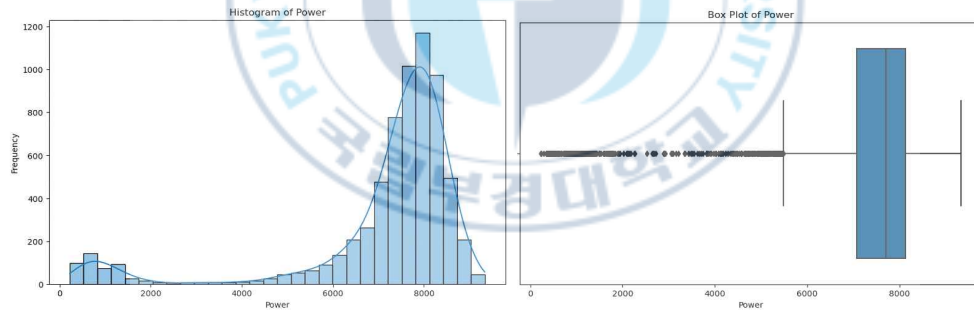


그림 3. 제주 삼다수 공장 데이터셋의 전력 사용량 히스토그램 및 박스플롯

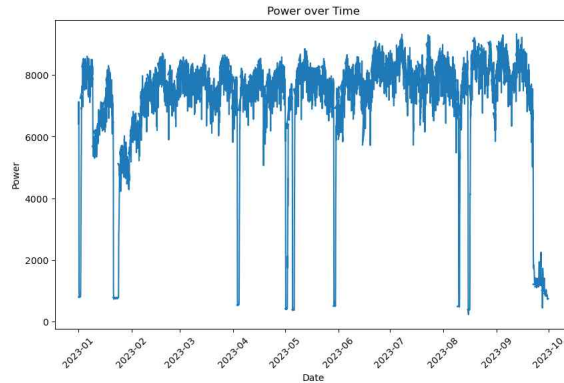


그림 4. 제주 삼다수 공장 데이터셋의
전력 사용량 시각화

전력사용량 데이터의 특징을 [그림 3], [그림 4] 보면 특정 날짜(한국 명절)에 사용량이 급격하게 내려가는 현상이 있는 것을 확인할 수 있다. 전통적인 방법으로는 이상치로 처리하면 모델의 예측력을 올릴 수 있다. 하지만 이 논문에서는 이상치 제거된 데이터와 원본 데이터를 각각 학습 및 예측할 수 있는 모델을 개발한다. 이상치로 생각하는 명절 데이터는 내년에도 명절이 있기에 원본 데이터 학습도 학술적으로 의미 있는 시도라고 생각한다.

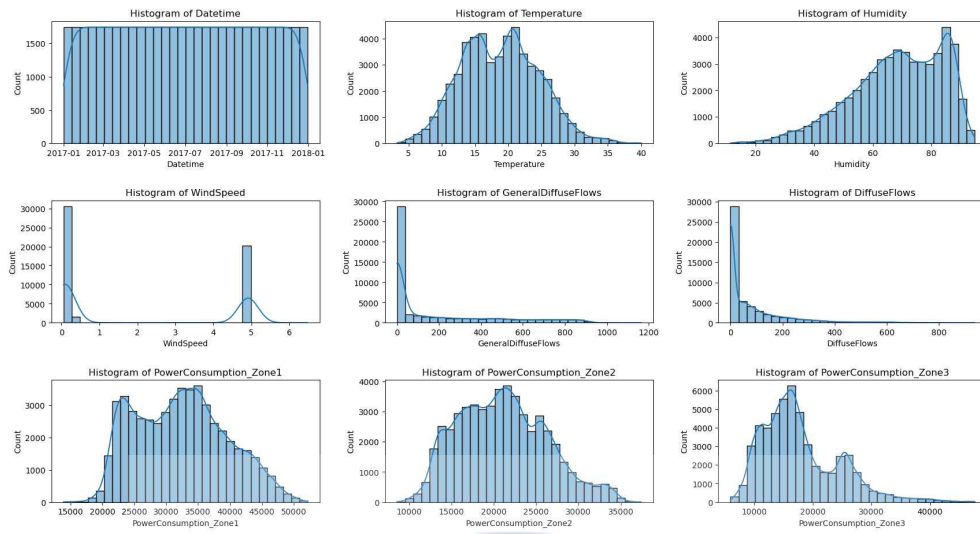


그림 5. 테투안시 데이터셋의 특성 히스토그램

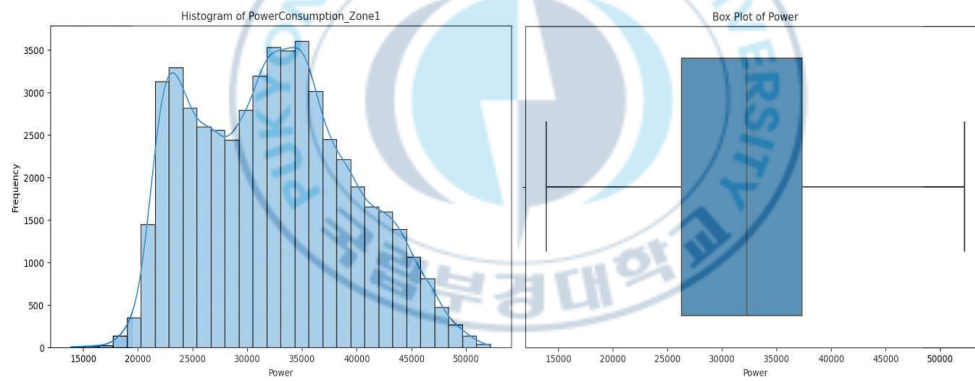


그림 6. 테투안시 구역1의 전력 사용량 히스토그램 및 박스플롯

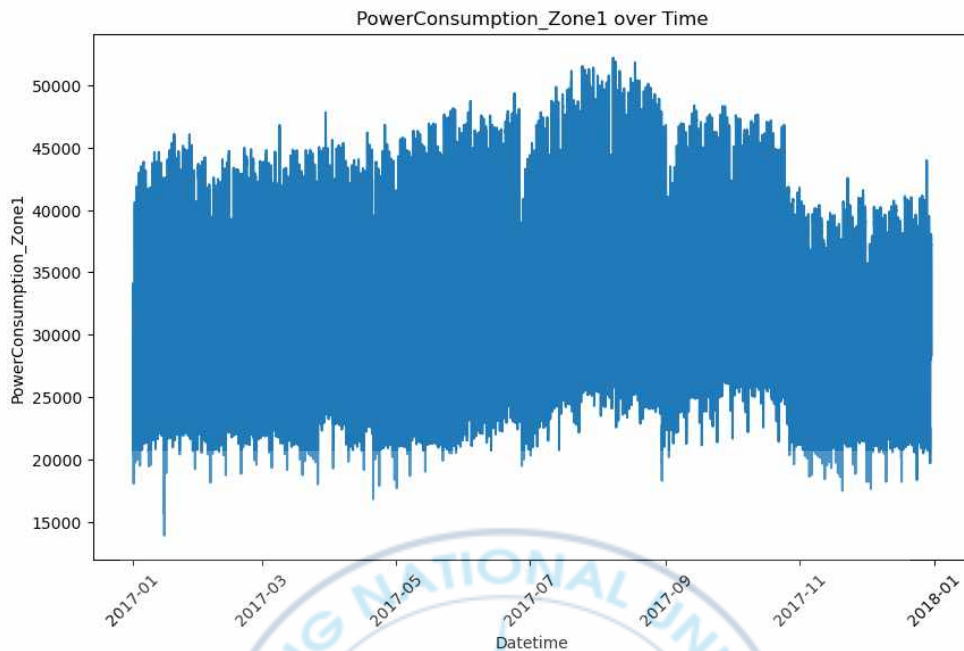


그림 7. 테투안시 구역1의 전력 사용량 시각화

모로코 데이터세트는 날짜 시간, 온도, 습도, 풍속, 일반 확산 흐름, 확산 흐름, 1구역의 전력 소비, 2구역의 전력 소비, 3구역의 전력 소비 총 9 가지 특징 데이터를 포함되어 있다[그림 5], [그림 6], [그림 7].

3.1.2 데이터 전처리 및 정규화 기법

이상치 제거는 데이터 전처리 과정에서 중요한 단계 중 하나이다. 이상치는 데이터 분석과 모델 학습 과정에서 잘못된 결과를 초래할 수 있기 때문에, 이를 적절히 처리하는 것이 필요하다. 이상치를 제거하는 방법에는 여러 가지가 있으며, 각 방법은 데이터의 특성과 분석 목적에 따라 선택될 수 있다. 대표적인 이상치 제거 기법으로는 IQR(Interquartile Range) 방법, 박스 플롯, 사분위수 방법, 그리고 z-score 방법이 있다. z-score 방

법은 대규모 데이터 분석 작업에서 특히 유용하다. z-score는 각 데이터 포인트가 평균으로부터 몇 표준편차 떨어져 있는지를 나타내며, 데이터의 정규성을 가정한다. z-score는 데이터의 스케일에 영향을 받지 않으므로 다양한 데이터셋에 적용 가능하다는 장점이 있다[그림 8].

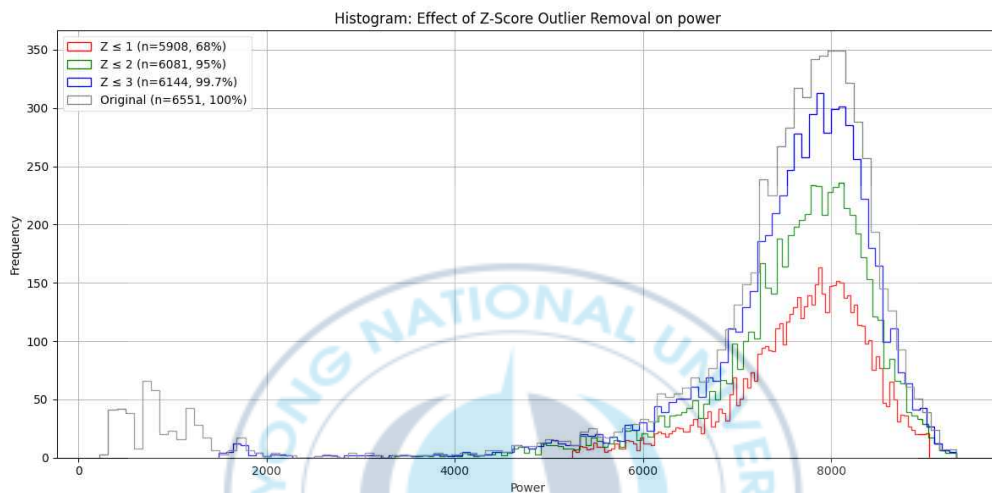


그림 8. Z-Score 범위 시각화

본 논문에서는 z-score 방법을 사용하여 z-1, 2, 3을 각각 모델이 직접 학습 및 예측하여 최적의 값을 찾아낼 수 있게 구현하였다. 모든 데이터 세트 기준으로 z-score 3 즉 데이터의 99.7%가 아닌 값들을 이상치로 간주했을 때 모델의 성능이 우수하다는 것을 확인하였다.

인공지능 모델의 성능을 최적화하기 위해서는 입력 데이터들의 유의미성을 평가하는 과정이 필수적이다. 이를 위해 각 변수들과 전력 사용량의 상관계수를 계산할 필요가 있다. 상관계수는 변수 간의 선형 관계를 정량적으로 나타내며, 이를 통해 어떤 데이터가 전력 사용량에 유의미한 영향을 미치는지 판단할 수 있다. 상관계수의 절대값이 클수록 두 변수 간의 관계가 강하다는 것을 의미한다.



그림 9. 제주 삼다수 공장 데이터의 변수 간 상관관계 히트맵

그림 9번을 통해 알 수 있듯이, 전력사용량과 가장 높은 상관관계를 보이는 변수는 기온(temp)이다. 이는 기온이 전력 사용량에 유의미한 영향을 미칠 가능성이 높음을 시사한다. 반면, 풍속(wind_speed_m/s)과 해수면 기압(sea_level_pressure_hPa)은 전력 사용량과 음의 상관관계를 보였으며, 상관계수의 절대값이 상대적으로 큰 편이다. 이는 이러한 변수들이 전력 사용량에 영향을 미칠 수 있는 중요한 요인임을 의미한다.

따라서 모로코 데이터 세트의 상관 관계분석을 해보면 아래와 같다[그림 10].

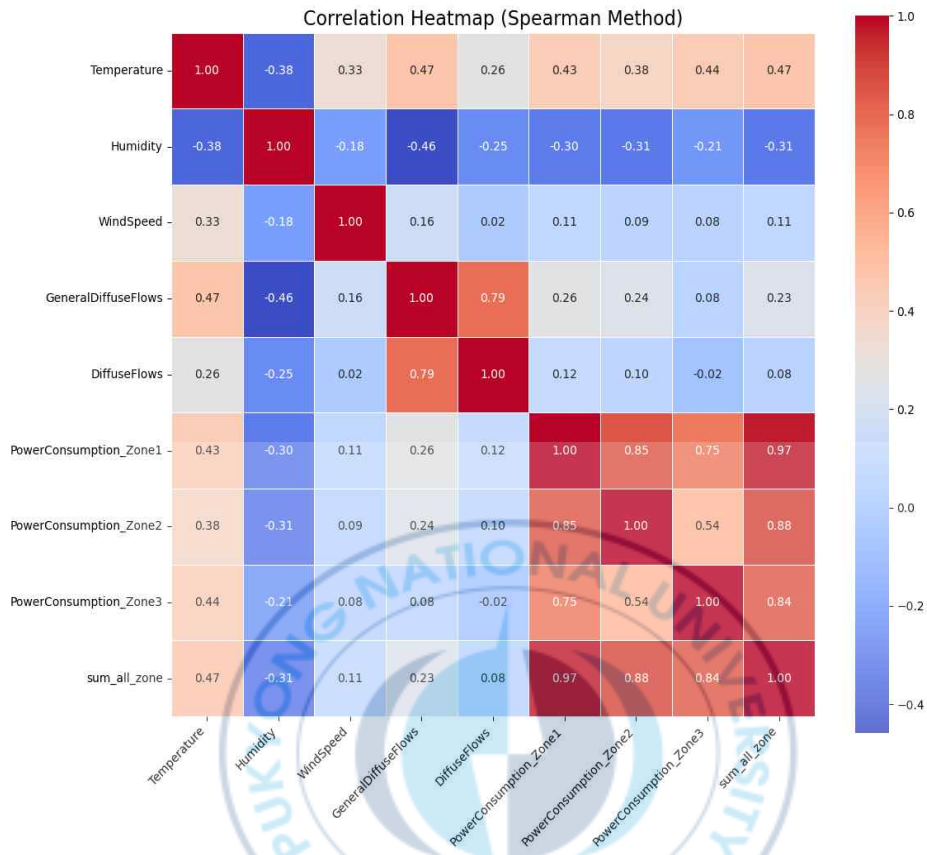


그림 10. 테투안시 데이터의 변수 간 상관관계 히트맵

각 변수별 특징들을 조금 더 분석을 해보면 페어플롯(pair plot)을 그려 보는 것이 있다[그림 11], [그림 12]. 이는 다변량 데이터의 각 변수 간의 관계를 시각적으로 표현하는 유용한 도구이다. 주로 수치형 데이터를 다룰 때 사용되며, 변수 간 상관관계와 분포를 한눈에 파악할 수 있게 한다. 이 도구를 사용하면 변수 간의 상관관계와 데이터의 분포 패턴을 직관적으로 이해할 수 있어, 모델링 과정에서 유용한 인사이트를 제공한다.

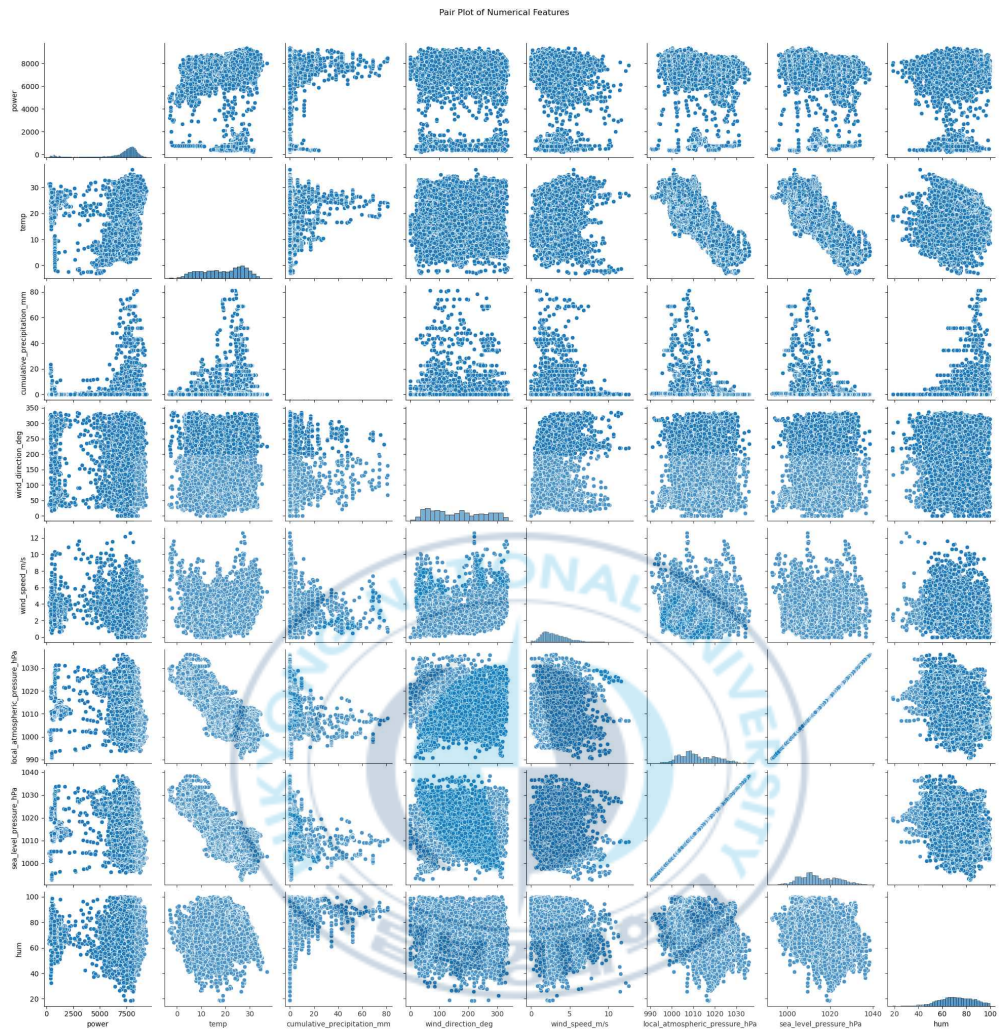


그림 11. 제주 삼다수 공장 데이터셋의 특성 간 페어 플롯

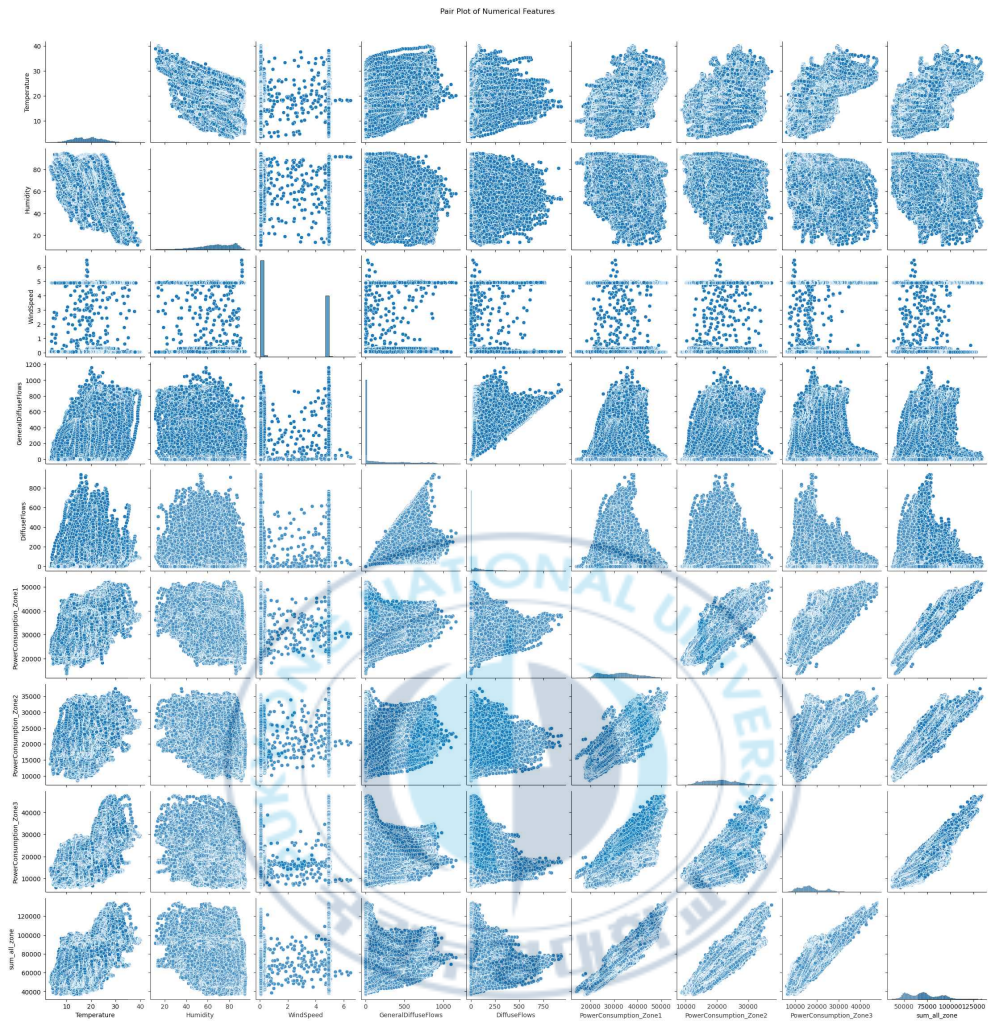


그림 12. 테투안시 데이터셋의 특성 간 페어 플롯

3.2 모델 개발 및 학습 과정

3.2.1 전력 예측을 위한 최적의 머신러닝 모델 선정

본 연구에서 선정된 데이터셋에 가장 적합한 예측 모델을 도출하기 위해 다양한 시계열 기반 머신러닝 및 딥러닝 기법을 비교 분석하였다. 각 모델은 데이터의 특성과 예측 목적에 따라 상이한 구조적 장점을 보유하고 있으며, 본 연구에서는 모델별 성능 차이를 정량적으로 평가하고, 실제 적용 가능성을 고려한 맞춤화 작업을 병행하였다.

분석에 활용된 주요 예측 모델은 다음과 같다:

- 1) LSTM: 시계열 데이터의 장기 의존성을 효과적으로 학습할 수 있는 순환 신경망 기반 모델로, 전력 사용량의 주기성과 추세성을 반영하는데 유리하다.
- 2) XGBoost: 강력한 성능과 높은 해석력을 바탕으로 다양한 특성 변수를 포함한 회귀 예측에 적합한 결정 트리 기반의 앙상블 학습 기법이다.
- 3) Prophet: Facebook에서 개발한 시계열 예측 모델로, 비정상성과 시계열성을 갖는 시계열 데이터를 빠르게 모델링할 수 있는 장점을 지닌다.
- 4) RNNs: 시계열 데이터를 순차적으로 처리하며 시간적 패턴을 포착하지만, 장기 의존성 처리에 한계가 있어 비교 모델로 포함하였다.
- 5) Transformer: 자연어 처리 분야에서 강력한 성능을 보였으며, 최근에는 시계열 예측에서도 self-attention 메커니즘을 활용해 복잡한 시간 패턴을 효과적으로 학습하는 모델로 주목받고 있다.
- 6) Time-MoE: 다양한 시계열 패턴에 대한 적응력을 갖춘 전문가 집단 구조 기반 모델로, 데이터 구간별로 최적화된 예측을 가능하게 한다.
- 7) PatchTST: 시계열 데이터를 패치 단위로 분할하여 처리하는 최신 Transformer 아키텍처로, 병렬 연산과 전체 시계열 구조의 학습이 동시에 가능하다.

8) TSMixer: 최근 제안된 시계열 전용 MLP 기반 모델로, 단순한 구조임에도 불구하고 우수한 예측 정확도를 보이며, 계산 효율성과 경량화 측면에서 장점을 가진다.

모든 모델은 동일한 전처리 및 학습 환경에서 실험되었으며, 주요 성능 지표로 R^2 , MAPE, sMAPE를 기준으로 정량 평가를 수행하였다. 또한, 각 모델의 예측 안정성, 이상치에 대한 민감도, 실제 시스템에의 통합 가능성 등을 종합적으로 고려하여 최적 모델을 선정하였다. 이를 통해 도출된 최적 모델은 향후 예측 정확도를 기반으로 한 에너지 소비 최적화 전략의 핵심 도구로 활용될 수 있으며, 본 시스템의 성능을 좌우하는 결정적인 요소로 작용한다.

3.2.2 하이퍼파라미터 튜닝 및 성능 개선 전략

시계열 데이터 분석에서 Lag Feature는 특정 시점의 데이터와 과거의 데이터를 사용하여 새로운 변수를 생성하는 방법이다. 이는 시계열 데이터를 다룰 때 중요한 기법으로, 과거 정보가 현재 또는 미래에 어떤 영향을 미치는지를 모델이 학습할 수 있게 한다. 예를 들어, 오늘의 전력 사용량이 어제와 그제의 전력 사용량에 영향을 받을 수 있기 때문에, 이러한 과거 데이터를 lag feature로 추가하면 모델 성능을 향상시킬 수 있다.

Lag Feature 생성 방법:

- 1단계 Lag: (X_{t-1}) (이전 시점의 데이터)
- 2단계 Lag: (X_{t-2}) (2단계 이전의 데이터)
- n단계 Lag: (X_{t-n}) (n단계 이전의 데이터)

장점은 반복되는 패턴이나 주기를 효과적으로 포착할 수 있다. 과거 데이터가 미래 데이터를 예측하는 데 중요한 경우, lag feature를 사용하면 모델의 예측 성능이 향상될 수 있다. 데이터의 연속성과 시간 의존성을

모델에 반영할 수 있다.

모델의 성능을 최적화하기 위해 다양한 변수 조합을 고려하는 것이 필요하다. 이러한 조합을 통해 모델이 어떤 변수들이 예측에 중요한 역할을 하는지 학습할 수 있다. 다음은 다양한 변수 조합을 정의한 리스트이다. 각 조합은 시계열 데이터의 여러 특성을 반영하며, 모델이 최적의 조합을 선택할 수 있도록 구현하였다[그림 13],[그림 14].

```

1 feature_combinations = [
2   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s'],
3   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'holiday'],
4   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'holiday'],
5   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'power_lag2', 'holiday'],
6   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'power_lag2', 'temp_lag1', 'holiday'],
7   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'power_lag2', 'temp_lag1',
8    'local_atmospheric_pressure_hPa_lag1', 'holiday'],
9   ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'power_lag2', 'power_lag3', 'temp_lag1',
10  'temp_lag2', 'local_atmospheric_pressure_hPa_lag1', 'local_atmospheric_pressure_hPa_lag2', 'wind_speed_m/s_lag1', 'wind_speed_m/s_lag2', 'holiday'],
11  ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'holiday', 'power_lag3'],
12  ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'power_lag1', 'power_lag2', 'power_lag3', 'temp_lag1',
13  'temp_lag2', 'local_atmospheric_pressure_hPa_lag1',
14  'sea_level_pressure_hPa', 'wind_speed_m/s', 'local_atmospheric_pressure_hPa_lag2',
15  'wind_speed_m/s_lag2'],
16  ['power', 'temp', 'local_atmospheric_pressure_hPa', 'sea_level_pressure_hPa', 'wind_speed_m/s', 'holiday', 'power_lag1', 'local_atmospheric_pressure_hPa',
17  'local_atmospheric_pressure_hPa_lag1']
18 ]

```

그림 13. 제주 삼다수 공장 데이터셋의 변수 조합

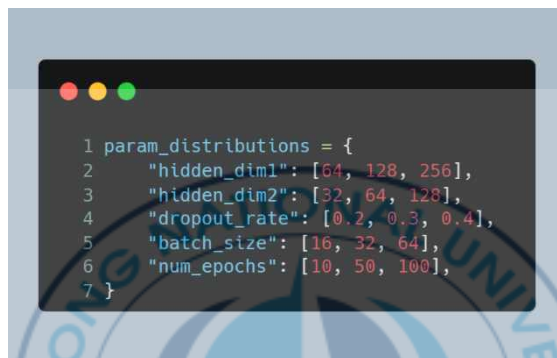
```

feature_combinations = [
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'sum_all_zone'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag1'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag1', 'PowerConsumption_Zone2_lag1'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag1', 'PowerConsumption_Zone2_lag1', 'PowerConsumption_Zone1_lag2'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag1', 'PowerConsumption_Zone2_lag1', 'PowerConsumption_Zone1_lag2'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'sum_all_zone', 'PowerConsumption_Zone1_lag3'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag1', 'PowerConsumption_Zone2_lag1', 'PowerConsumption_Zone1_lag2',
  'PowerConsumption_Zone1_lag3'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'PowerConsumption_Zone1_lag2', 'PowerConsumption_Zone2_lag2'],
  ['PowerConsumption_Zone1', 'Temperature', 'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
  'sum_all_zone', 'PowerConsumption_Zone1_lag1', 'PowerConsumption_Zone1_lag2']
]

```

그림 14. 테투안시 구역1의 변수 조합

다양한 feature 조합 중에서 최적의 조합을 자동으로 선택하여 검증 데이터셋에 대한 성능을 평가함으로써 이루어진다. 최적의 조합은 모델의 예측 성능을 극대화하는 조합으로 선택된다. 최종적으로, 모델이 선택한 최적의 feature 조합은 전력 사용량 예측에 있어 가장 유의미한 변수를 포함하게 된다.



```

1 param_distributions = {
2     "hidden_dim1": [64, 128, 256],
3     "hidden_dim2": [32, 64, 128],
4     "dropout_rate": [0.2, 0.3, 0.4],
5     "batch_size": [16, 32, 64],
6     "num_epochs": [10, 50, 100],
7 }

```

그림 15. 하이퍼파라미터 조합

모델 튜닝을 위해 `param_distributions`를 설정하는 과정은 모델의 성능을 최적화하는 데 중요한 역할을 한다[그림 15]. 주어진 `param_distributions`는 하이퍼파라미터의 다양한 조합을 실험하여 모델의 최적 성능을 찾는 데 사용된다. 각 하이퍼파라미터의 역할은, `hidden_dim1`은 모델의 첫 번째 은닉층(hidden layer)에서의 유닛(뉴런) 수를 정의한다. 이는 모델의 표현력에 영향을 미치며, 값이 클수록 더 복잡한 패턴을 학습할 수 있지만, 과적합의 위험도 증가할 수 있다. `hidden_dim2`은 모델의 두 번째 은닉층에서의 유닛 수를 정의한다. 이는 첫 번째 은닉층과 유사한 역할을 하며, 모델의 깊이와 표현력에 영향을 미친다. 복잡한 시계열 데이터의 경우 여러 은닉층이 필요할 수 있다. `dropout_rate` 드롭아웃은 과적합을 방지하기 위해 학습 중에 무작위로 일부 뉴런을 비활성화하는 기법이다. 비율이 높을수록 모델의 일반화 능력

이 향상될 수 있지만, 너무 높으면 학습이 어려워질 수 있다. `batch_size`: 한 번의 학습 단계에서 모델에 공급되는 데이터 샘플의 수를 정의한다. 배치 크기가 클수록 학습 안정성이 높아질 수 있지만, 메모리 사용량이 증가한다. 작은 배치는 더 자주 파라미터 업데이트를 하여 훈련 속도를 높일 수 있지만, 더 많은 노이즈가 포함될 수 있다. `num_epochs`: 전체 데이터셋을 몇 번 반복 학습할 것인지 정의한다. 에포크 수가 너무 적으면 모델이 충분히 학습되지 않을 수 있고, 너무 많으면 과적합의 위험이 커질 수 있다.

3.2.3 모델 구현

기본 LSTM과 XGBoost 계열 모델을 기반으로 데이터셋의 통계적 특성과 실험 시나리오의 요구사항에 맞춰 구조와 하이퍼파라미터를 재구성한 파생 모델들을 추가로 구현한다. 모델별로 입력 전처리 전략, 이상치 처리 방식, 손실 함수 선택, 그리고 Optuna 기반 탐색 공간을 차별화하여 예측 정확도와 학습 효율을 극대화한다.

이하에서는 수정된 각 모델을 제시된 순서에 따라 설명한다. 먼저 LSTM 변형 모델들을 비교한 뒤, XGBoost 파생 모델들의 최적화 절차와 성능 차이를 기술한다. 각 설명은 모델 구조, 하이퍼파라미터 탐색 방법, 성능 지표, 그리고 관찰된 성능 향상의 원인을 중심으로 서술한다.

1) LSTM_basic 모델:

LSTM_basic 모델은 특별한 구조적 개선 없이 기본적인 LSTM 구조를 사용한 시계열 예측 모델이다. 입력 시퀀스를 받아 LSTM 계층을 통과시키고, 최종 시점의 출력을 Dense 레이어를 통해 회귀값으로 변환한다. 이 모델은 다른 실험 모델들과의 성능 비교를 위한 기준점(baseline) 역할을 수행한다.

2) LSTM_a 모델:

LSTM_a 모델은 LSTM_basic의 구조를 기반으로 하여 다음과 같은 주요 개선을 포함한다. 첫째, LSTM 출력에 배치 정규화(Batch Normalization)를 적용하여 내부 공변량 변화(Internal Covariate Shift)를 줄이고 학습 안정성과 수렴 속도를 향상시킨다. 둘째, 잔차 연결(Residual Connection)이 포함된 Dense 네트워크 구조를 채택하여 정보 손실을 줄이고, 두 단계의 Dense 레이어와 ReLU 활성화 함수 및 Dropout을 적용하여 비선형성을 확보하고 과적합을 방지한다.

또한, 다양한 랙(lag) 특성을 자동 생성하여 입력 시계열 데이터의 시차 특성을 반영한다. 이때 전력(power), 온도(temp), 풍속(wind_speed_ms), 기압(local_atmospheric_pressure_hPa) 등의 주요 변수에 대해 다양한 조합의 랙을 생성하고, Optuna 기반 하이퍼파라미터 최적화를 통해 최적의 조합을 선택한다.

데이터 전처리 단계에서는 z-score 기반 이상치 제거 기법을 도입하여 노이즈로 인한 성능 저하를 방지하며, 임계값은 자동으로 조정된다. 학습 과정에서는 Adam 옵티마이저와 ReduceLRonPlateau 스케줄러를 활용하여 손실 감소가 정체될 경우 학습률을 동적으로 조정한다. 하이퍼파라미터 탐색 범위에는 hidden size, 레이어 수, dropout, batch size, learning rate, Huber Loss의 delta, 랙 조합, 시퀀스 길이, 이상치 임계값, 배치 정규화 사용 여부 등이 포함된다.

표 1. LSTM_a 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	LSTM_a	
	Jeju	Morocco zone 1
Dataset		
Metric_Directional_Accuracy	46.21087315	60
Param_use_batch_norm	1	1
Param_use_scheduler	1	1
Param_z_thresh	3	2
Param_huber_delta	0.719748694	0.879646532
Param_epochs	50	100
Param_sequence_length	60	60
Param_learning_rate	0.005	0.001
Param_batch_size	32	64
Param_dropout	0.1	0.1
Param_num_layers	1	2
Param_hidden_size	32	128

3) LSTM_b 모델

LSTM_b 모델은 LSTM_a 모델과 구조적으로 유사하며, 배치 정규화, 잔차 연결 기반 Dense 레이어, 다양한 lag 조합 자동 생성, Adam 옵티마이저와 ReduceLRonPlateau 스케줄러 적용, Optuna 기반 하이퍼파라미터 탐색 등 동일한 주요 기능을 포함한다. 차이점은 이상치 제거 기법이 명시적으로 포함되어 있지 않다는 점이다. 따라서 LSTM_b는 정규화와 구조적 개선을 통해 성능을 높이는 데 집중하며, 전처리 단계에서의 이상치 제거 없이도 안정적인 학습이 가능하도록 설계되었다.

표 2. LSTM_b 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	LSTM_b	
Dataset	Jeju	Morocco zone 1
Metric_Directional_Accuracy	45.77572965	60
Param_use_batch_norm	1	1
Param_use_scheduler	1	0
Param_z_thresh	no	no
Param_huber_delta	1.738347657	1.879465132
Param_epochs	100	100
Param_sequence_length	30	60
Param_learning_rate	0.01	0.01
Param_batch_size	32	64
Param_dropout	0.1	0.1
Param_num_layers	1	1
Param_hidden_size	64	256

4) LSTM_c 모델

LSTM_c 모델은 동적 이상치 제거 기법과 Huber Loss 함수를 도입한 것이 주요 특징이다. 입력 데이터에서 z-score 기반 이상치 제거를 수행하며, 이때 사용되는 임계값(z_thresh)은 Optuna를 통해 자동 최적화된다. 손실 함수로는 Huber Loss를 사용하여, MSE 대비 이상치에 덜 민감한 학습이 가능하도록 한다. delta 매개변수 역시 하이퍼파라미터로 설정하여, 학습의 안정성을 높인다.

모델 구조는 기본적인 LSTM을 따르며, Optuna를 통해 은닉층 크기, 레이어 수, 드롭아웃 비율 등 총 9개 하이퍼파라미터를 최적화한다.

표 3. LSTM_c 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	LSTM_c	
	Jeju	Morocco zone 1
Dataset		
Metric_Directional_Accuracy	0	0
Param_use_batch_norm	0	0
Param_use_scheduler	0	0
Param_z_thresh	3	3
Param_huber_delta	0.658877426	0.894153263
Param_epochs	50	100
Param_sequence_length	30	60
Param_learning_rate	0.001	0.01
Param_batch_size	32	64
Param_dropout	0.2	0.1
Param_num_layers	1	2
Param_hidden_size	128	32

5) LSTM_d 모델

LSTM_d 모델은 LSTM_c와 거의 동일한 구성을 갖지만, z-score 기반 이상치 제거를 포함하지 않는다. Huber Loss를 손실 함수로 사용하며, delta 값을 최적화한다. Optuna를 통해 핵심 하이퍼파라미터(은닉층 크기, 레이어 수, 드롭아웃 비율 등 9개)를 탐색한다.

표 4. LSTM_d 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	LSTM_d	
	Jeju	Morocco zone 1
Dataset		
Metric_Directional_Accuracy	0	0
Param_use_batch_norm	0	0
Param_use_scheduler	0	0
Param_z_thresh	no	no
Param_huber_delta	0.628548216	0.549681153
Param_epochs	100	100
Param_sequence_length	10	60
Param_learning_rate	0.001	0.0005
Param_batch_size	32	32
Param_dropout	0.5	0.5
Param_num_layers	1	1
Param_hidden_size	128	64

6) XGBoost_basic 모델

XGBoost_basic 모델은 특별한 구조 개선 없이 기본적인 XGBoost^[56] 회귀 알고리즘을 사용하는 시계열 예측 모델로, 후속 실험에서의 비교 기준점 역할을 한다.

7) XGBoost_a 모델

XGBoost_a 모델은 기본 XGBoost 모델을 기반으로 하여 여러 고급 기법들을 통합하였다. Optuna를 통해 하이퍼파라미터를 자동으로 최적화하며, z-score 기반 이상치 제거를 통해 노이즈를 억제한다. 특징 공학 단계에서는 power, temp, wind_speed_ms, local_atmospheric_pressure_hPa 변수에 대해 다양한 시차(lag) 특징, 이동 평균(rolling mean), 이동 표준편차(rolling std)를 생성하고, 상호작용 항(wind_temp_interaction, pressure_temp_interaction)을 도입한다.

특징 선택 단계에서는 상호 정보(mutual information)를 기반으로 한 SelectKBest 기법을 통해 의미 있는 입력 변수를 선별한다. Huber Loss를 손실 함수로 채택하여 이상치에 강건한 모델을 학습하며, 교차 검증은 시계열 특성을 반영한 TimeSeriesSplit 방식을 사용한다. 최적화 목표는 R^2 점수이며, 모델의 전처리-학습-평가의 전체 파이프라인의 재현성과 성능 향상에 기여한다.

표 5. XGBoost_a 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	XGBoost_a	
	Jeju	Morocco zone 1
Dataset	Jeju	Morocco zone 1
Param_reg_lambda	2.701	0.101
Param_random_state	42	42
Param_min_child_weight	9	2
Param_colsample_bylevel	0.87523013	0.849873691
CV_Std_R2	0.013406319	0.000733917
Param_z_thresh	3	3
Param_reg_alpha	9.101	8.101
Param_n_estimators	1000	1000
Param_learning_rate	0.1	0.1
Param_gamma	4	0
Param_colsample_bytree	1	1
Param_subsample	0.6	0.9
Param_max_depth	4	10

8) XGBoost_b 모델

XGBoost_b 모델은 XGBoost_a 모델과 거의 동일한 구조 및 기법을 사용하나, z-score 기반 이상치 제거가 적용되지 않는다. 나머지 구성 요소 - Optuna 기반 하이퍼파라미터 최적화, 고급 특징 공학(시차, 이동 통계, 상호작용), SelectKBest를 통한 특징 선택, Huber Loss 적용, TimeSeriesSplit 기반 교차 검증, R^2 점수 최적화 동일하게 유지된다.

표 6. XGBoost_b 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	XGBoost_b	
Dataset	Jeju	Morocco zone 1
Param_reg_lambda	3.601	4.501
Param_random_state	42	42
Param_min_child_weight	6	11
Param_colsample_bylevel	0.832910675	0.93936747
CV_Std_R2	0.023023602	0.000875738
Param_z_thresh	no	no
Param_reg_alpha	8.801	0.301
Param_n_estimators	600	1000
Param_learning_rate	0.1	0.1
Param_gamma	4	0
Param_colsample_bytree	1	1
Param_subsample	0.5	0.7
Param_max_depth	4	9

9) XGBoost_c 모델

XGBoost_c 모델은 이상치 제거, 고급 특징 엔지니어링, 하이퍼파라미터 최적화, 연산 가속화 등 다양한 기법을 통합한 고성능 XGBoost 기반 모델이다. 전처리 단계에서 z-score 기반 이상치 탐지를 수행하고, 임계값은 Optuna를 통해 자동으로 탐색된다. 특징 엔지니어링에서는 날짜를 연·월·일·요일로 분해하여 시간 정보를 반영하며, 핵심 변수에 대한 시차 및 통계 특징을 생성한다.

표 7. XGBoost_c 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	XGBoost_c	
	Jeju	Morocco zone 1
Dataset		
Param_reg_lambda	7	5
Param_random_state	0	0
Param_min_child_weight	0	0
Param_colsample_bylevel	0	0
CV_Std_R2	0	0
Param_z_thresh	3	3
Param_reg_alpha	1	5
Param_n_estimators	500	500
Param_learning_rate	0.01	0.01
Param_gamma	5	1
Param_colsample_bytree	0.9	0.7
Param_subsample	0.5	0.8
Param_max_depth	3	4

10) XGBoost_d 모델

XGBoost_d 모델은 XGBoost_c 모델과 유사한 구조를 가지며, z-score 이상치 제거를 제외한 모든 구성 요소가 동일하게 적용된다. 즉, Optuna 기반 하이퍼파라미터 최적화, 날짜 기반 시간 특성 추출, 고급 특징 생성 및 연산 가속화 기법이 포함되며, 실험 효율성과 재현성 모두를 달성하는데 기여한다.

표 8. XGBoost_d 모델을 위한 최적의 하이퍼파라미터

Hyperparameters	XGBoost_d	
	Jeju	Morocco zone 1
Dataset		
Param_reg_lambda	9	10
Param_random_state	0	0
Param_min_child_weight	0	0
Param_colsample_bylevel	0	0
CV_Std_R2	0	0
Param_z_thresh	no	no
Param_reg_alpha	1	9
Param_n_estimators	800	600
Param_learning_rate	0.01	0.01
Param_gamma	4	1
Param_colsample_bytree	0.6	0.9
Param_subsample	1	0.9
Param_max_depth	3	4



3.3 시스템 설계 및 구현

3.3.1 전력 사용량 실시간 모니터링 시스템 구축

본 연구에서 개발되는 통합 시스템은 업로드된 데이터를 분석하여 에너지 소비 패턴을 예측하고, 최적의 에너지 사용 전략을 제안하는 데 중점을 둔다[그림 16]. 인공지능 알고리즘을 활용한 데이터 분석은 통합 시스템의 핵심으로, 여러 기법을 통해 데이터의 패턴을 학습하고 예측 모델을 생성한다. 이를 통해 에너지 관리의 효율성을 높이고, 장기적인 에너지 소비 계획을 수립하는 데 기여할 수 있다.

최근 데이터 유출 사고와 개인정보 침해에 대한 우려가 커지면서, 클라우드 기반 AI 시스템의 한계를 극복할 수 있는 온프레미스 AI 시스템의 필요성이 대두되고 있다. 본 연구에서 개발된 시스템은 온프레미스 웹 앱이 사용자의 데이터가 외부 서버로 전송되지 않고 내에서 직접 처리될 수 있도록 설계되었다. 이로써 개인정보와 기업 기밀정보의 유출 위험을 획기적으로 줄일 수 있으며, 특히 의료, 금융, 국방 등 민감한 정보를 다루는 분야에서 그 중요성이 더욱 강조된다.



그림 16. 전체 프로젝트 구성 다이어그램

3.3.2 유저 시나리오

온프로미스 AI 시스템의 전체적인 아키텍처와 데이터 흐름을 시각화하면, 사용자는 본 시스템을 통해 데이터셋을 업로드하고 다양한 데이터 전처리 및 공학 기법을 적용할 수 있다. 사용자가 업로드한 데이터는 내부적으로 저장된 후, 표준화나 결측치 처리 등 다양한 전처리 모듈을 거쳐 정제된다. 이후 사용자는 인공지능 모델을 선택하고 하이퍼파라미터를 세부적으로 커스터마이징할 수 있다. 선택된 모델은 설정된 하이퍼파라미터 값을 기반으로 학습 작업을 수행하며, 학습된 모델은 저장되어 추후 예측 및 분석에 사용된다[그림17].

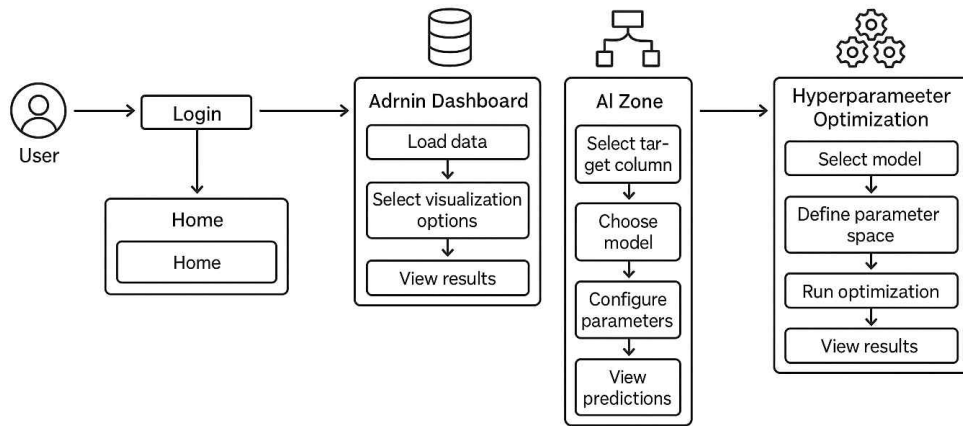


그림 17. 사용자 시나리오

데이터 전송과 모델 학습은 모두 장치 내에서 이루어지므로 데이터 유출 위험이 줄어들고, 실시간 처리가 가능하다. 또한, 본 시스템은 사용자의 편의를 위해 데이터 업로드부터 모델 선택, 하이퍼파라미터 설정, 학습 및 예측까지의 전 과정을 하나의 파이프라인으로 제공한다. 이러한 흐름을 통해 최종적으로 사용자는 업로드된 데이터를 기반으로 빠르고 안전하게 예측 결과를 생성할 수 있다. 그림 18에서는 데이터베이스 스키마를 표현해 본다.

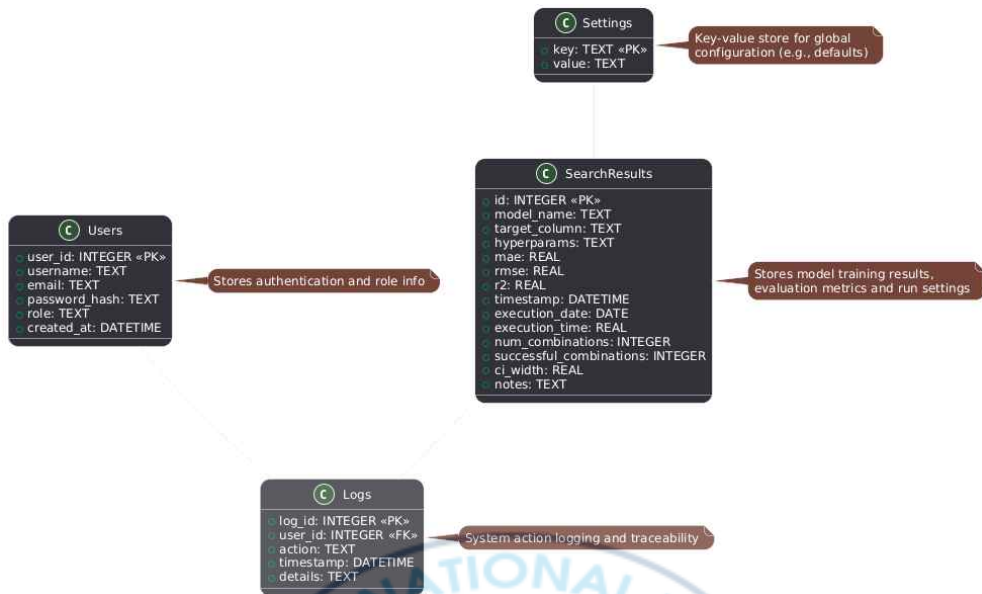


그림 18. 데이터베이스 스키마

3.3.3 에너지 관리 대시보드 UI 설계

사용자 중심 설계 측면에서는 인터랙티브 투어[그림 19], [그림 20] 가이드와 자동화 워크플로우 빌더를 도입해 사용자가 복잡한 모델 설정할 필요가 없으며 4단계(데이터 선택 → 모델 구성 → 임계값 설정 → 예측 결과 확인)으로 시스템을 운영할 수 있도록 간소화했다.

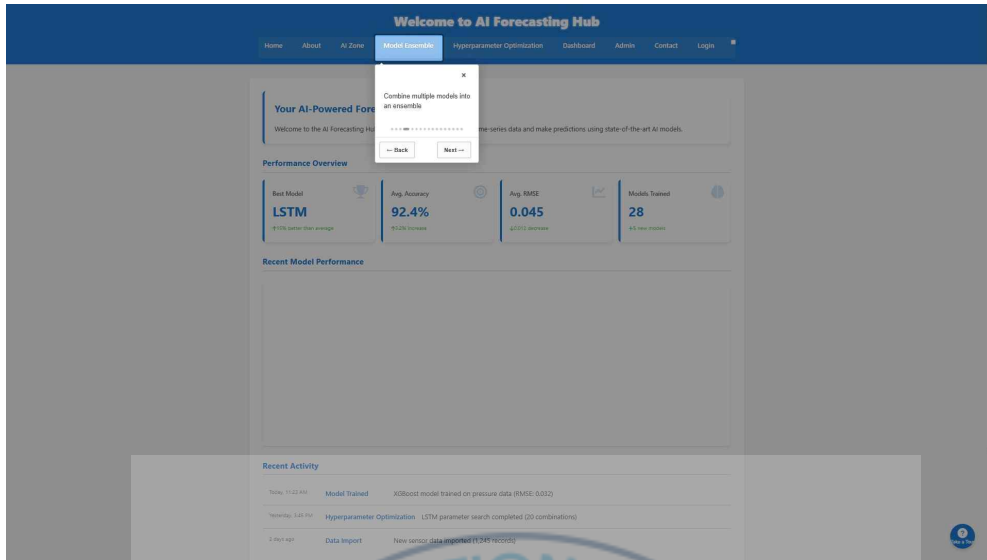


그림 19. 사용자 안내 투어 기능



그림 20. 사용자 안내 투어 기능

본 시스템을 이용하는 사용자는 먼저 계정 인증 절차를 거쳐 AIZone 플랫폼에 로그인한다[그림 21].

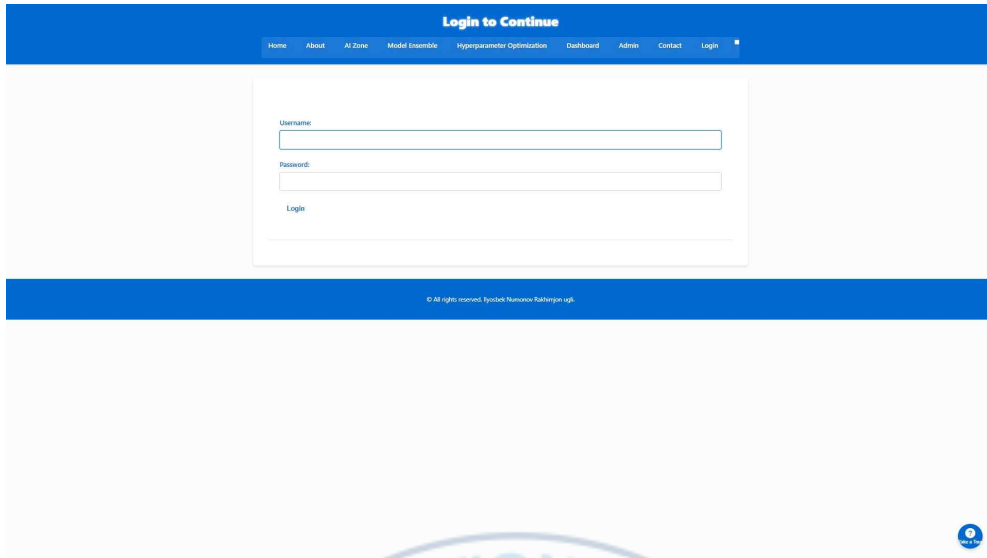


그림 21. 로그인 화면

로그인 후 메인 대시보드에서 사용자 계정을 확인하고, 학습에 사용할 데이터셋을 업로드한다[그림 22].

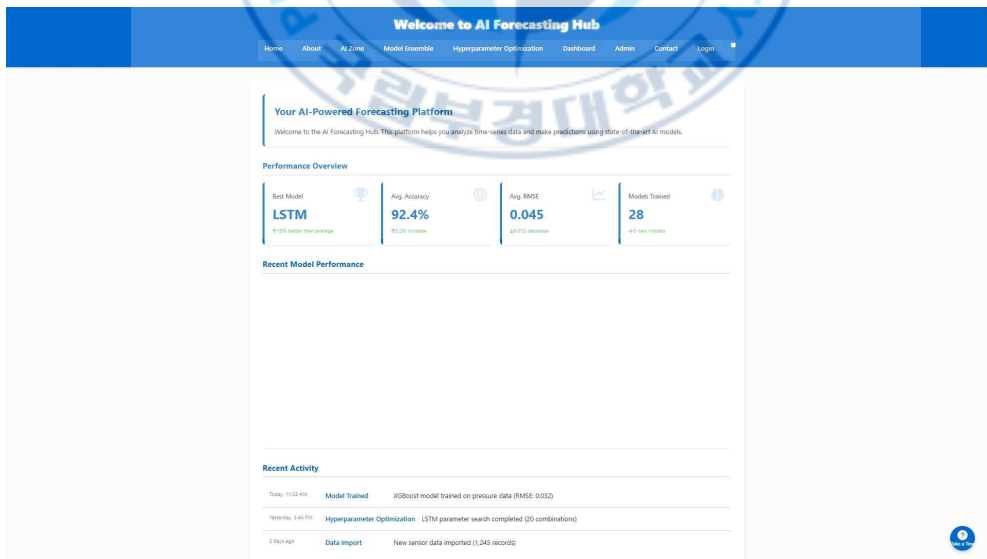


그림 22. 메인 화면

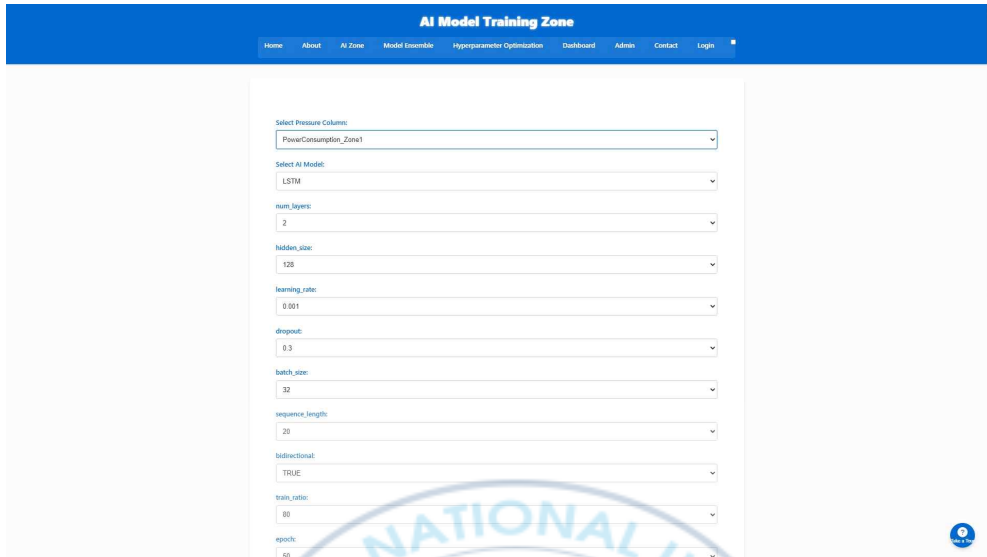


그림 23. AI 존 화면 1



그림 24. AI 존 화면 2

이때 사용자는 입력값, 학습률(learning rate), 결정 트리의 깊이(depth)

와 같은 다양한 하이퍼파라미터 값을 직접 설정하거나 미리 정의된 값 중에서 선택할 수 있다[그림 23], [그림 24]. 하이퍼파라미터 설정이 완료되면, 사용자 명령에 따라 시스템은 즉시 학습을 시작하며 진행 상황을 실시간으로 대시보드에 표시한다. 이후 해당 데이터셋에 적합한 모델을 선택하고, 각 모델의 하이퍼파라미터 설정 화면으로 진입한다.

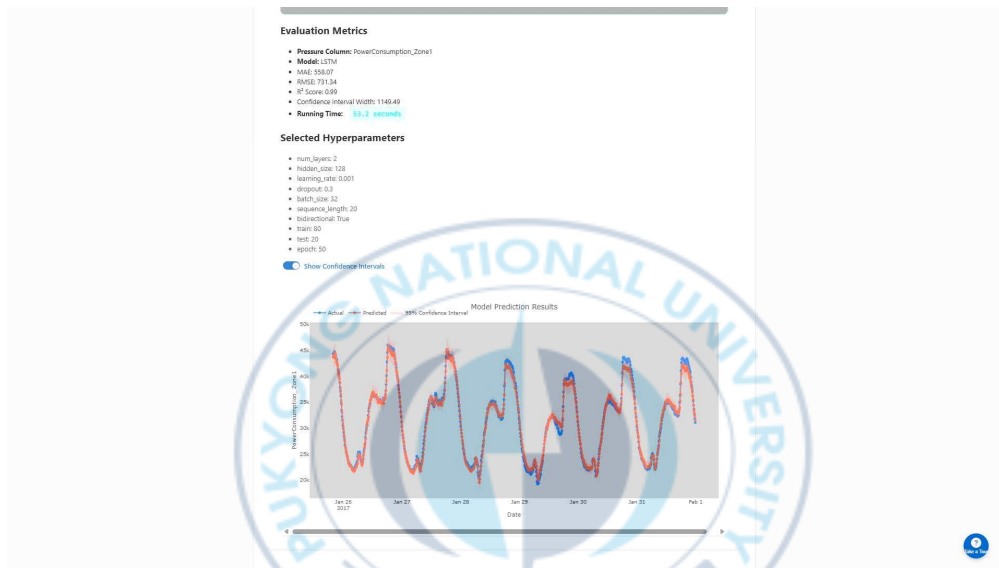


그림 25. AI 존 화면 4

학습이 완료되면, 사용자 인터페이스에서는 실험 결과를 확인할 수 있는 화면이 제공된다[그림 25]. 과거에 수행한 학습 및 추론 결과를 열람하여 모델의 성능 지표(예: 정확도, 손실 함수 값)와 하이퍼파라미터 조합을 비교할 수 있다. 특히 이전 결과에 사용되었던 하이퍼파라미터 조합을 선택하여 같은 조건으로 재학습하거나 새로운 데이터에 대한 예측을 진행할 수 있다.

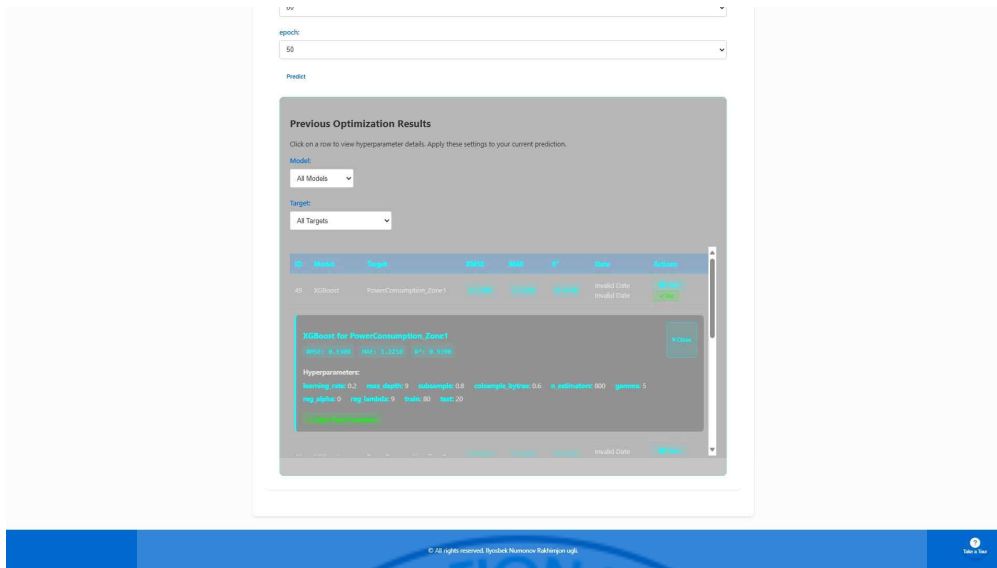


그림 26. AI 존 화면 3

이러한 기능을 통해 사용자는 과거 실험의 결과를 재활용하고, 모델 튜닝 과정을 효율적으로 수행할 수 있다[그림 26].

3.3.4 하이퍼파라미터 최적화 기능

본 시스템은 하이퍼파라미터 최적화를 통해 모델의 성능을 극대화한다. 사용자는 미리 정의된 하이퍼파라미터 값의 범위나 조합을 지정하면, 시스템은 그에 따른 다양한 조합의 학습 작업을 자동으로 실행한다[그림 27], [그림 28], [그림 29].

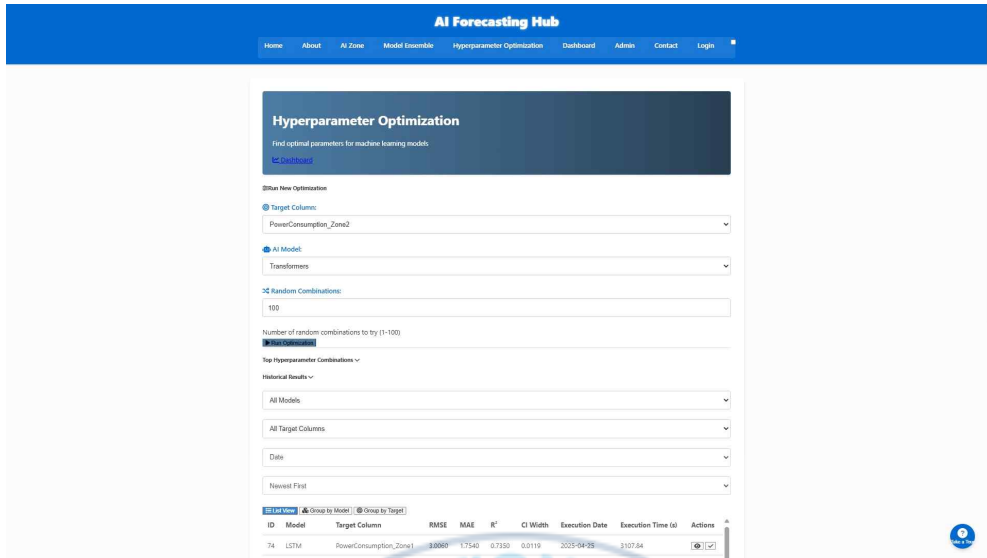


그림 27. 하이퍼파라미터 최적화 화면 1

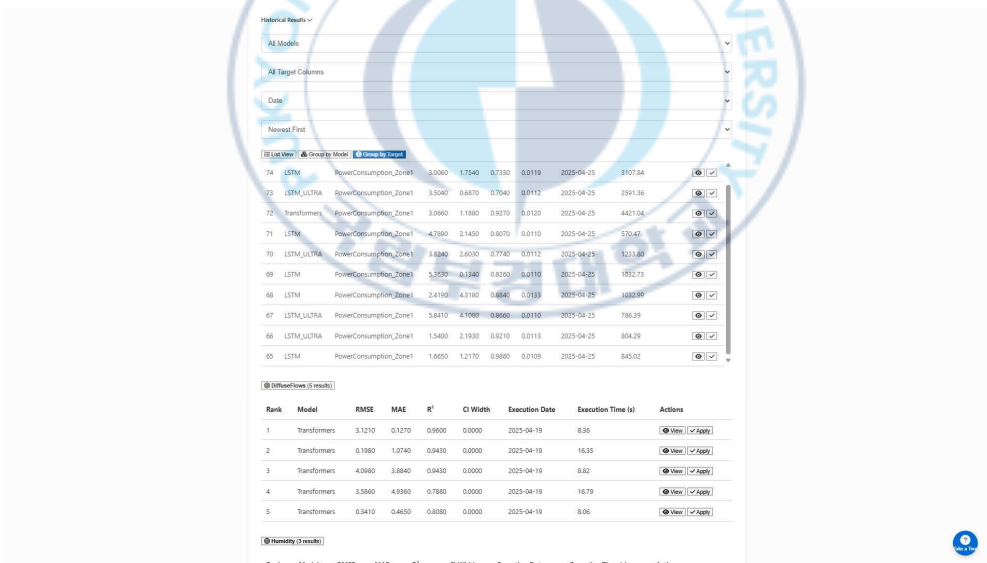


그림 28. 하이퍼파라미터 최적화 화면 2

1	PowerConsumption_Zone1	3.4530	4.4910	0.9140	434898175299	2025-04-25	27.24	View Apply
2	PowerConsumption_Zone1	3.9600	4.0560	0.8650	327811440287	2025-04-25	9.57	View Apply
3	PowerConsumption_Zone1	5.1100	2.7430	0.7750	26334157319	2025-04-25	13.60	View Apply
4	PowerConsumption_Zone1	3.9330	4.4740	0.7470	231162484519	2025-04-25	8.70	View Apply
5	PowerConsumption_Zone1	4.5040	1.0140	0.7520	38955237425	2025-04-25	24.44	View Apply

RMN (5 results)								
Rank	Target Column	RMSE	MAE	R ²	CI Width	Execution Date	Execution Time (s)	Actions
1	Temperature	5.3930	2.1950	0.7650	0.0000	2025-04-25	4.81	View Apply
2	Temperature	4.3830	2.6610	0.9310	0.0000	2025-04-25	5.75	View Apply
3	Temperature	3.8150	2.8500	0.8080	0.0000	2025-04-25	4.67	View Apply
4	Temperature	0.6490	0.6860	0.7090	0.0000	2025-04-25	10.41	View Apply
5	Temperature	3.8310	1.4300	0.9970	0.0000	2025-04-18	6.28	View Apply

Transformers (5 results)								
Rank	Target Column	RMSE	MAE	R ²	CI Width	Execution Date	Execution Time (s)	Actions
1	PowerConsumption_Zone1	3.0660	1.1880	0.9270	0.0120	2025-04-25	4421.04	View Apply
2	Temperature	2.1600	2.5200	0.8570	0.0000	2025-04-25	6.56	View Apply
3	Temperature	2.1000	3.8760	0.7380	0.0000	2025-04-25	11.04	View Apply
4	Temperature	5.4120	3.8280	0.9470	0.0000	2025-04-25	12.05	View Apply
5	Temperature	3.2250	4.4470	0.7860	0.0000	2025-04-25	41.39	View Apply

MLBoost (5 results)								
Rank	Target Column	RMSE	MAE	R ²	CI Width	Execution Date	Execution Time (s)	Actions
1	PowerConsumption_Zone1	4.1820	0.8900	0.7240	638.9155	2025-04-25	11.98	View Apply
2	PowerConsumption_Zone1	3.3980	1.5200	0.8320	931.3814	2025-04-25	4.21	View Apply
3	PowerConsumption_Zone1	1.0480	0.4770	0.7810	821.3123	2025-04-25	33.85	View Apply
4	PowerConsumption_Zone1	0.1300	1.2210	0.9390	1025.1853	2025-04-25	19.26	View Apply
5	PowerConsumption_Zone1	4.8150	2.8020	0.7970	134.7194	2025-04-25	5.47	View Apply

그림 29. 하이퍼파라미터 최적화 화면 3

예를 들어, 학습률, 배치 크기, 정규화 계수 등의 하이퍼파라미터 조합을 자동으로 탐색하여 각 모델별 성능을 평가한다. 이러한 자동화된 탐색 과정을 통해 최적의 하이퍼파라미터 조합을 찾아낼 수 있으며, 사용자는 대시보드에서 각 조합에 대한 학습 결과를 비교 분석할 수 있다. 이를 통해 모델의 성능 향상 가능성을 체계적으로 검토할 수 있다[그림 30].

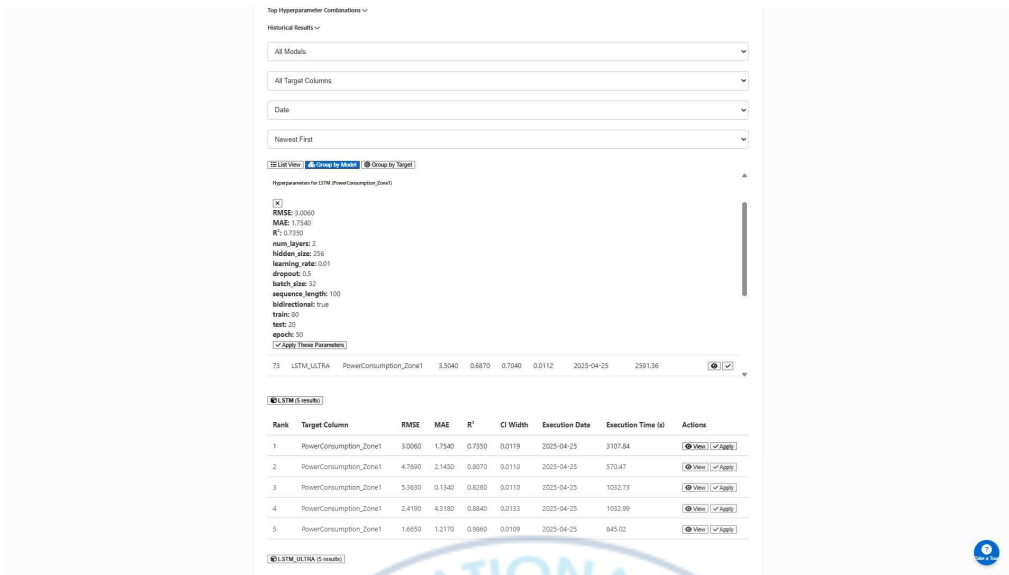


그림 30. 하이퍼파라미터 최적화 화면 4

3.3.5 결과 확인 테이블 및 대시보드

모델별, 변수별 성능 결과는 테이블 형식으로 정리되어 시각적으로 제공된다. 각 모델의 예측 정확도, 손실 값 등의 지표가 변수별로 정리된 테이블이 생성되며, 사용자는 이를 통해 특정 변수나 하이퍼파라미터 설정이 모델 성능에 미치는 영향을 쉽게 파악할 수 있다[그림 31], [그림 32].

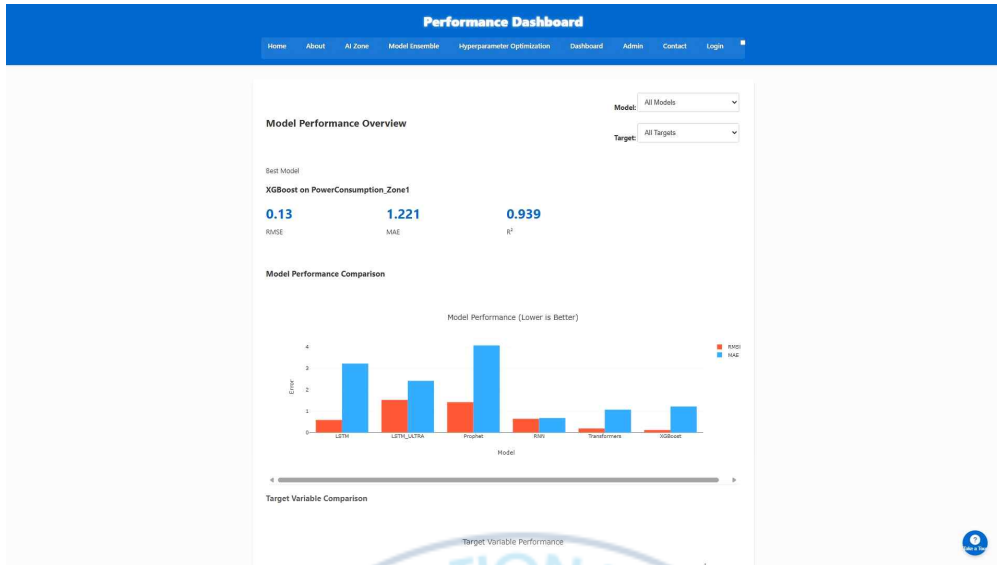


그림 31. 대시보드 화면 1

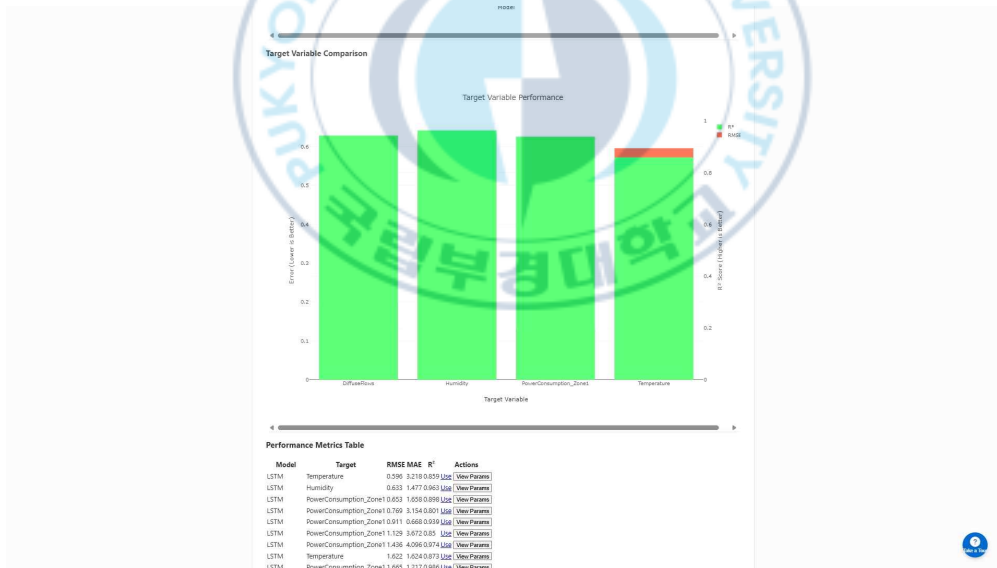


그림 32. 대시보드 화면 2

LSTM	PowerConsumption_Zone1	1.439	4.096	0.974	View Params
LSTM	Temperature	1.622	1.624	0.873	View Params
LSTM	PowerConsumption_Zone1	1.655	1.217	0.983	View Params
LSTM	PowerConsumption_Zone1	2.112	4.325	0.967	View Params
LSTM	PowerConsumption_Zone1	2.237	0.639	0.784	View Params
LSTM	PowerConsumption_Zone1	2.382	0.463	0.895	View Params
LSTM	PowerConsumption_Zone1	2.478	4.318	0.864	View Params
LSTM	PowerConsumption_Zone1	3.006	1.754	0.735	View Params
LSTM	Humidity	3.261	1.848	0.855	View Params
LSTM	Temperature	3.376	3.226	0.993	View Params
LSTM	Humidity	3.838	4.161	0.742	View Params
LSTM	PowerConsumption_Zone1	3.907	3.564	0.853	View Params
LSTM_ULTRA	Temperature	1.527	2.418	0.877	View Params
LSTM_ULTRA	PowerConsumption_Zone1	1.54	2.192	0.821	View Params
LSTM_ULTRA	PowerConsumption_Zone1	3.504	0.687	0.704	View Params
LSTM_ULTRA	PowerConsumption_Zone1	3.824	2.262	0.774	View Params
Prophet	PowerConsumption_Zone1	1.423	4.06	0.965	View Params
Prophet	PowerConsumption_Zone1	1.5	4.37	0.908	View Params
Prophet	PowerConsumption_Zone1	2.02	4.038	0.823	View Params
Prophet	PowerConsumption_Zone1	3.433	4.481	0.918	View Params
Prophet	PowerConsumption_Zone1	3.833	4.474	0.747	View Params
Prophet	PowerConsumption_Zone1	3.847	4.558	0.708	View Params
RNN	Temperature	0.649	0.668	0.709	View Params
RNN	Temperature	2.37	2.002	0.863	View Params
RNN	Temperature	3.831	1.43	0.997	View Params
RNN	Temperature	3.835	2.85	0.808	View Params
Transformers DiffusionFlow		0.186	1.075	0.943	View Params
Transformers DiffusionFlow		0.341	0.465	0.808	View Params
Transformers DiffusionFlow		1.2	4.031	0.911	View Params
Transformers DiffusionFlow		1.993	0.791	0.909	View Params
Transformers Temperature		2.1	3.873	0.718	View Params
Transformers Temperature		2.16	2.52	0.837	View Params
Transformers PowerConsumption_Zone1		3.066	1.188	0.977	View Params
Transformers DiffusionFlow		0.121	0.127	0.96	View Params
Transformers Temperature		3.223	4.447	0.766	View Params
Transformers DiffusionFlow		3.386	4.938	0.788	View Params
XGBoost	PowerConsumption_Zone1	1.13	1.221	0.938	View Params
XGBoost	PowerConsumption_Zone1	0.154	1.323	0.955	View Params
XGBoost	PowerConsumption_Zone1	0.699	2.111	0.835	View Params
XGBoost	PowerConsumption_Zone1	1.049	4.477	0.781	View Params
XGBoost	PowerConsumption_Zone1	1.054	1.54	0.963	View Params
XGBoost	PowerConsumption_Zone1	3.338	1.52	0.832	View Params
XGBoost	PowerConsumption_Zone1	3.678	4.547	0.918	View Params
XGBoost	Temperature	3.888	0.254	0.718	View Params

그림 33. 대시보드 화면 3

또한, 통합 대시보드에서는 각 모델의 성능 기록과 하이퍼파라미터 최적화 과정의 상세 내역을 확인할 수 있다[그림 33]. 대시보드는 시간 흐름에 따른 모델 성능 추이를 그래프로 나타내어 모델 개선 과정을 직관적으로 보여준다. 이를 통해 사용자는 과거 실험의 히스토리를 살펴보며 최적화 과정의 결과를 체계적으로 관리할 수 있다[그림 34].

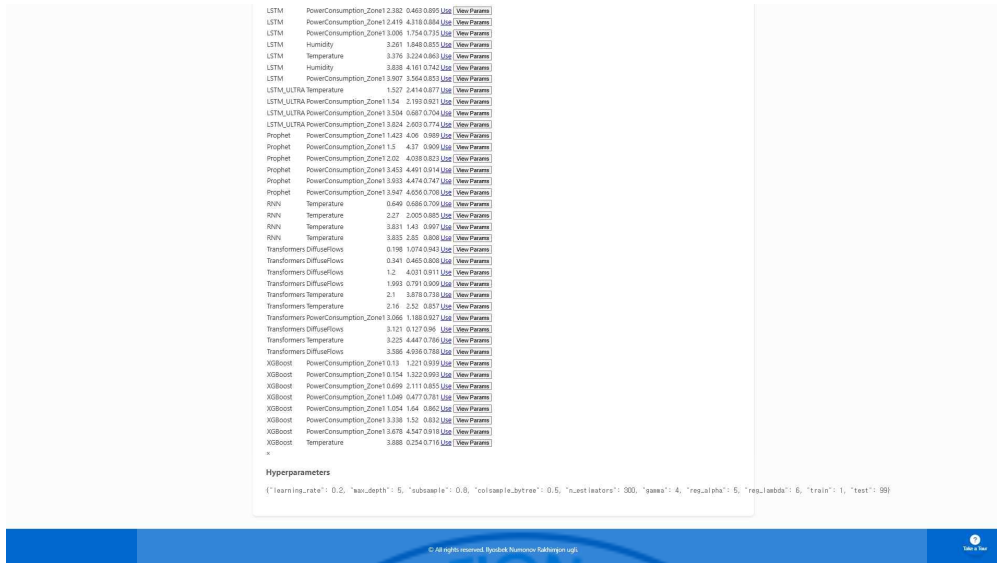


그림 34. 대시보드 화면 4

3.3.6 데이터 시각화

AIZone 플랫폼은 분석 및 의사결정을 돕기 위해 다양한 데이터 시각화 기법을 제공한다. 사용자는 산점도, 상자도(box plot), 분포도(histogram), 히트맵과 같은 시각화 방법을 선택하여 데이터의 특성을 파악할 수 있다. 예를 들어, 데이터셋의 개별 변수 분포나 변수 간 상관관계를 시각화하여 이상치나 편향을 발견할 수 있다[그림 35], [그림 36].

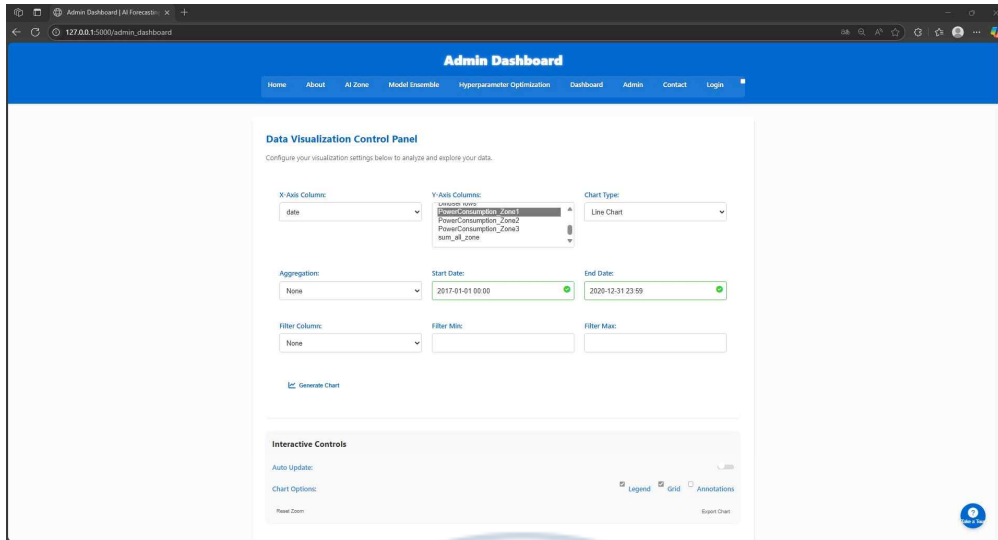


그림 35. 데이터 시각화 및 분석 화면 1

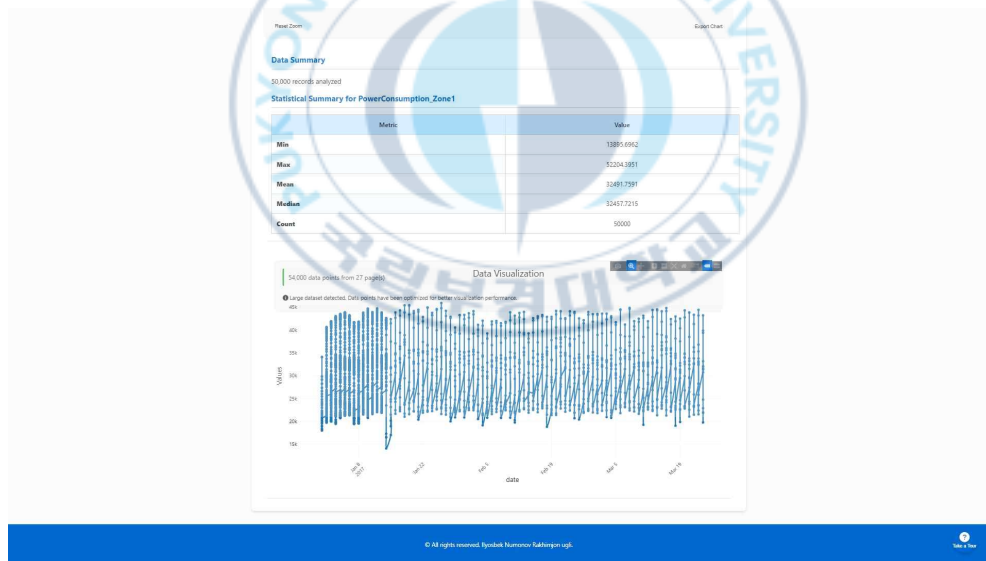


그림 36. 데이터 시각화 및 분석 화면 2

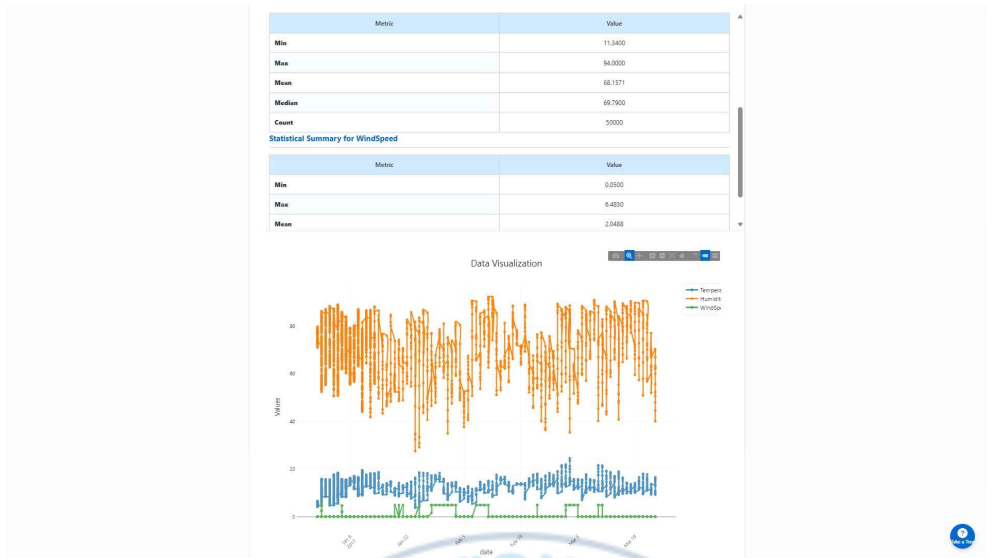


그림 37. 데이터 시각화 및 분석 화면 3

또한, 예측 결과를 시각화하여 모델의 예측 분포와 실제 분포를 비교할 수 있으며, 사용자는 그래프 형태(PNG, SVG 등)를 자유롭게 선택할 수 있다. 이러한 다양한 시각화 기능은 데이터에 대한 이해도를 높여 주며, 모델 성능 평가 시 통찰력을 제공한다[그림 37].

3.3.7 앙상블 모드

본 시스템은 앙상블 모드를 지원하여 여러 모델의 예측을 결합함으로써 더욱 높은 예측 성능을 제공한다[그림 38], [그림 39]. 사용자는 단일 모델 뿐만 아니라 다양한 모델을 동시에 선택하여 예측에 활용할 수 있으며, 시스템은 선택된 모델들의 예측 결과를 단순 가중치 값 반영으로 통합한다[그림 40], [그림 41].

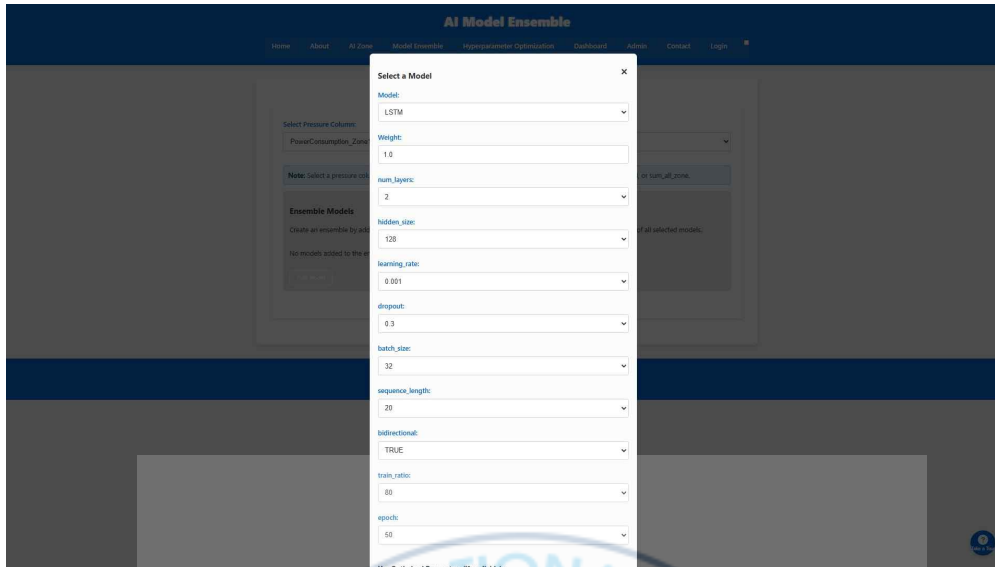


그림 38. 앙상블 모드 화면 1

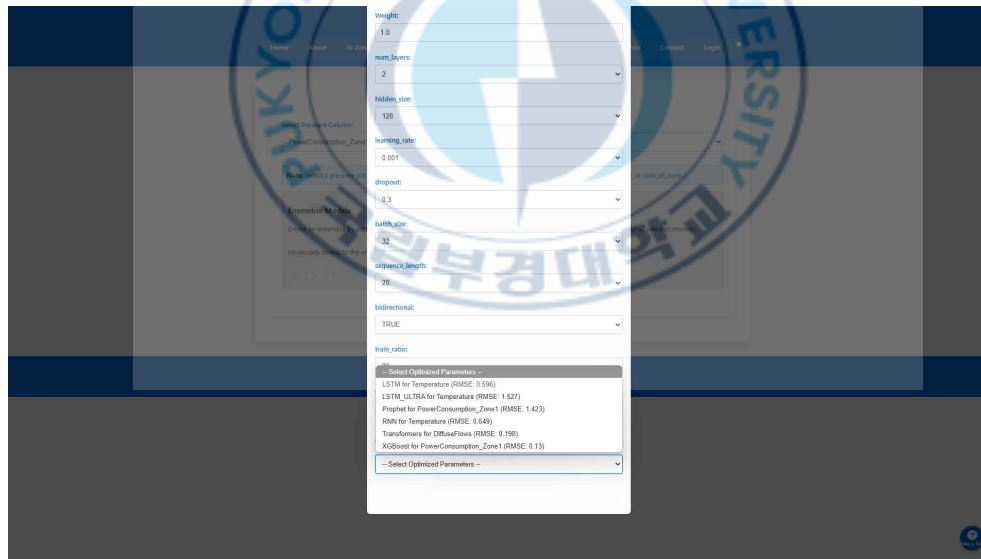


그림 39. 앙상블 모드 화면 2

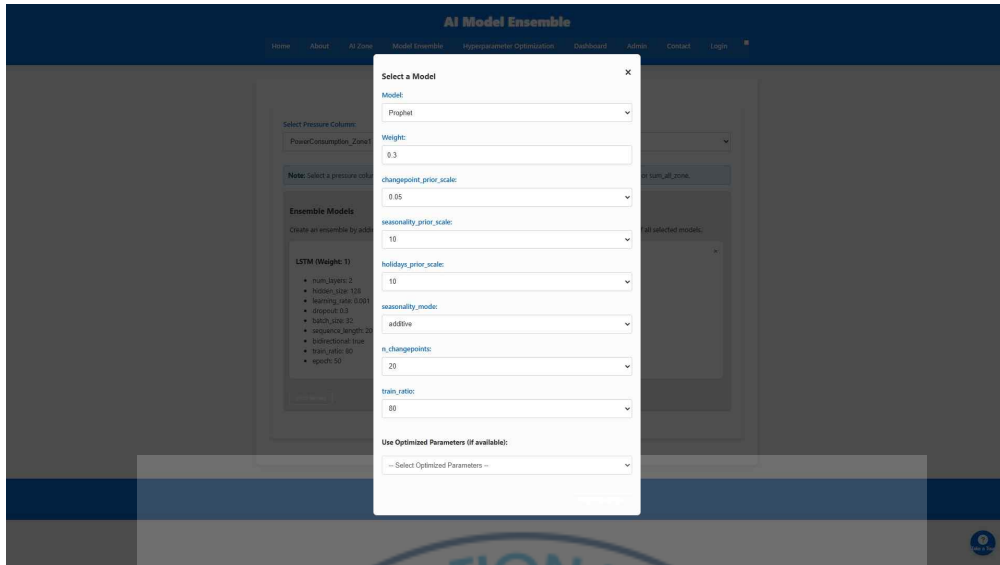


그림 40. 앙상블 모드 화면 3



그림 41. 앙상블 모드 화면 4



그림 42. 앙상블 모드 화면 5

이러한 앙상블 기법을 통해 단일 모델보다 성능이 향상된 예측 결과를 도출할 수 있다[그림 42].

3.3.8 전 모델 재사용

모델별 예측 결과는 검증 페이지에서 상세히 제공된다. 사용자는 각 모델별로 산출된 지표를 비교함으로써 가장 적합한 모델을 선택할 수 있다 [그림 43].

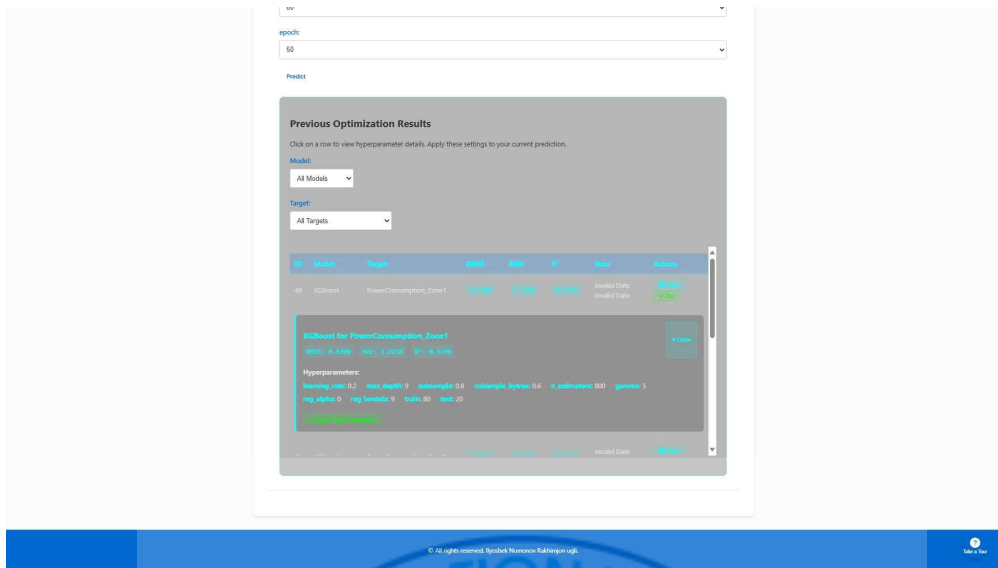


그림 43. 전 모델 사용 기능



IV. 연구 결과

4.1 모델 성능 평가 및 비교 분석

본 논문에서 구현한 AI-Zone 플랫폼은 LSTM, XGBoost, Prophet, RNN, Transformer 등 다양한 시계열 예측 모델을 통합적으로 지원하여, 사용자의 목적과 데이터 특성에 따라 유연하게 모델을 선택하고 구성할 수 있는 맞춤형 예측 환경을 제공한다. 본 장에서는 이 중에서도 LSTM과 XGBoost 모델을 중심으로 예측 성능을 정량적으로 비교 분석하며, 타 모델들에 대한 성능 논의는 생략한다.

실험은 모든 모델의 대해서 동일한 데이터셋을 기반으로 진행되었으며, 모델 성능 평가는 R^2 , MAPE, sMAPE를 주요 지표로 활용하였다. 특히, 시계열 데이터의 시차 특성을 효과적으로 반영하기 위해 다양한 랙(lag) 특성을 자동 생성하였고, 주요 입력 변수로는 전력, 온도, 풍속, 기압 등을 사용하였다. 각 변수 조합과 모델 파라미터는 Optuna 기반의 하이퍼파라미터 최적화 프레임워크를 통해 탐색되었으며, 이때 최적화 목표는 R^2 값을 최대화하는 것이었다.

모델 비교 결과, LSTM과 XGBoost는 데이터의 통계적 특성과 전처리 조건에 따라 상이한 성능 양상을 보였으며, 동일한 조건에서도 성능 편차가 존재함을 확인하였다. 이는 예측 목적, 데이터의 계절성과 이상치 분포, 변수 간 상호작용 특성에 따라 모델 선택 전략이 달라져야 함을 시사한다. 따라서 AI-Zone 플랫폼은 단일 모델에 의존하기보다, 다양한 예측 모델 간 성능을 비교하고 최적 조합을 찾아갈 수 있는 실험 기반 환경으로서 유의미한 역할을 할 수 있다.

표 9. 제주삼다수 공장 데이터세트 LSTM 모델 성능지표

Jeju LSTM						
Model name	R ²	Compare Base Model	MAPE	Compare Base Model	sMAPE	Compare Base Model
LSTM_basic	0.87	0%	12.85	0%	9.981	0%
LSTM_a	0.825	-5.172%	6.716	47.735%	6.248	37.401%
LSTM_b	0.917	5.402%	11.739	8.646%	9.252	7.304%
LSTM_c	0.919	5.632%	4.013	68.77%	3.83	61.627%
LSTM_d	0.913	4.943%	5.348	58.381%	5.115	48.753%

표 9에서는 LSTM_basic(기준 모델) $R^2 = 0.870$, MAPE = 12.85%, sMAPE = 9.98%로 기본적인 LSTM 구조만으로는 전체 추세 설명력은 준수하지만, 오차 지표(MAPE, sMAPE)가 비교적 높아 세부 예측에 취약하다. LSTM_a 모델은 $R^2 = 0.825$ (-5.17%p), MAPE = 6.72% (-47.7%), sMAPE = 6.25% (-37.4%). 배치 정규화, 잔차 연결 기반 Dense, 자동 렉 생성, z-score 이상치 제거를 도입하면서 평균 절대 백분율 오차가 크게 줄었다. 특히 이상치 제거가 강력히 작용해 큰 오차 샘플이 없어진 이유로 MAPE·sMAPE가 크게 개선됐다. 이상치 제거와 강력한 정규화 기법이 극단적인 값의 영향을 차단했으나, 전체 분산 설명력(R^2)이 소폭 하락한 것은 모델이 극단 예측보다는 중앙값 예측에 치중하면서 변동성 설명을 일부 희생했기 때문이다.

LSTM_b 모델은 $R^2 = 0.917$ (+5.40%p), MAPE = 11.74% (-8.65%), sMAPE = 9.25% (-7.30%)로 LSTM_a와 마찬가지로 배치 정규화와 잔차 연결 등을 쓰지만, 이상치 제거를 빼니 R^2 가 크게 올랐다. 중간값이 아닌 전체 추세와 변동성을 더 잘 포착한 결과다. 원인은 이상치 제거를 하지 않아 모든 데이터 패턴을 학습하며 분산 설명력이 개선됐지만, 여전히 큰 오차 샘플이 남아 MAPE·sMAPE 개선 폭은 제한적이다. LSTM_c 모델은 $R^2 = 0.919$ (+5.63%p), MAPE = 4.01% (-68.8%), sMAPE = 3.83% (-61.6%)로 동적 z-score 이상치 제거와 Huber Loss를 결합해, 이상치에 덜 민감하면서도 전체 변동성까지 잘 설명했다. 원인은 Huber Loss가 작은 오차에는 MSE처럼, 큰 오차에는 MAE처럼 동작해 안정적인 수렴을 유도하며, 동적 이상치 임계값 최적화가 각 시점별 노이즈를 효과적으로 제어했다. 이로 인해 R^2 , MAPE, sMAPE가 모두 최상 성능을 기록했다. LSTM_d 모델은 $R^2 = 0.913$ (+4.94%p), MAPE = 5.35% (-58.4%), sMAPE =

5.12% (-48.8%)로 Huber Loss는 유지하되 이상치 제거를 빼면 LSTM_b보다 오차 지표가 훨씬 낮아진다. Huber의 견고함 덕분에 큰 오류가 줄었으나, 동적 이상치 제거가 빠진 만큼 LSTM_c만큼은 아니다. 원인은 Huber Loss가 큰 오차를 로버스트하게 처리해 MAPE·sMAPE를 크게 낮췄지만, 여전히 일부 극단적인 값이 학습되어 LSTM_c 대비 오차 지표가 소폭 상승하고 R²도 약간 낮아졌다. 오차 축소 목적으로는 ‘이상치 제거 + 정규화 구조(LSTM_a)’ 혹은 ‘Huber Loss(LSTM_d)’가 효과적이다. 전체 변동성 설명력(R²)과 오차 지표의 균형을 최적화하려면 ‘동적 이상치 제거 + Huber Loss(LSTM_c)’ 조합이 가장 우수하다. 이상치 처리 없이 정규화·잔차 연결만 적용하면 R²가 크게 오르고(LSTM_b), 중간 수준의 오차 개선이 가능하다.

표 10. 제주삼다수 공장 데이터셋 기반 다변량 LSTM 모델의 입력 변수 최적 조합

Optimal combination of variables for Jeju	
LSTM_a	LSTM_b
power_lag1	power_lag1
power_lag2	power_lag2
power_lag3	power_lag3
power_lag4	temp_lag1
power_lag5	temp_lag2
temp_lag1	local_atmospheric_pressure_hPa_lag1
temp_lag2	local_atmospheric_pressure_hPa_lag2
local_atmospheric_pressure_hPa_lag1	power_rolling_mean_3
local_atmospheric_pressure_hPa_lag2	power_rolling_std_3
power_rolling_mean_3	power_rolling_mean_6
power_rolling_std_3	power_rolling_std_6
power_rolling_mean_6	power_rolling_mean_12
power_rolling_mean_12	power_rolling_std_12
power_rolling_std_12	wind_temp_interaction
pressure_temp_interaction	pressure_temp_interaction

표 11. 제주삼다수 공장 데이터셋 XGBoost 모델 성능지표

Jeju XGBoost						
Model name	R ²	Compare Base Model	MAPE	Compare Base Model	sMAPE	Compare Base Model
XGBoost_basic	0.88	0%	9.38	0%	6.52	
XGBoost_a	0.9	2.273%	5.502	41.343%	5.502	15.613%
XGBoost_b	0.9	2.273%	5.943	36.642%	5.595	14.187%
XGBoost_c	0.189	-78.523%	10.663	-13.678%	8.787	-34.77%
XGBoost_d	0.07	-92.045%	71.953	-667.09%	20.306	-211.442%

표 11에는 XGBoost_basic 모델은 기본적인 XGBoost 회귀 알고리즘만을 적용한 모델로, 후속 실험의 기준점으로 활용되었다. 해당 모델은 R² = 0.88, MAPE = 9.38, sMAPE = 6.52로 비교적 우수한 예측 정확도를 보였다.

반면, XGBoost_a 모델은 Optuna 기반의 하이퍼파라미터 자동 탐색, z-score 이상치 제거, 시차 및 이동 통계 기반의 특징 생성(lag, rolling), 변수 간 상호작용 항(interaction term), SelectKBest를 통한 특징 선택, Huber Loss 적용 등 다양한 고급 기법을 통합하였다. 그 결과, R²는 0.90으로 2.27% 향상되었고, MAPE는 5.582로 40.45% 감소, sMAPE 역시 5.502로 15.61% 감소하였다. 이는 모델이 이상치와 노이즈에 강건하게 반응하며, 정보성이 높은 특징을 중심으로 효과적으로 학습되었음을 의미한다.

XGBoost_b 모델은 XGBoost_a와 동일한 구조이나 이상치 제거 단계만 생략하였다. R²는 동일하게 0.90이지만, MAPE와 sMAPE는 각각 5.943, 5.595로 소폭 상승하였다. 이는 이상치 제거가 예측 정확도에 긍정적인 영향을 미쳤음을 시사하며, 특히 MAPE 측면에서 성능 저하가 두드러졌다. 이상치 제거의 유무가 모델의 민감도를 조절하는 요소로 작용함을 확인할 수 있다.

한편, XGBoost_c 모델은 z-score 임계값을 Optuna를 통해 자동 탐색하며 이상치를 제거하고, 날짜 정보를 연, 월, 일, 요일 등의 변수로 분해하는 시간 기반 특징을 추가하였다. 그러나 결과적으로 R²는 0.189로 급감(-78.5%), sMAPE는 8.787로 증가(+34.77%)하였다. 이는 자동 임계값 탐색 과정에서 실제로는 유의미한 데이터를 이상치로 잘못 분류했을 가능성이 있으며, 날짜 기반 특성이 오히려 예측 노이즈로 작용했음을 시사한다. 즉, 잘못된 전처리와 특징 엔지니어링이 오

히려 성능을 저하시키는 원인으로 작용하였다.

마지막으로 XGBoost_d 모델은 XGBoost_c에서 이상치 제거만 생략한 구조이
나, $R^2 = 0.07$ 로 가장 낮은 성능을 보였으며, MAPE는 71.953로 기본 모델 대비
무려 667.09% 악화되었다. 이는 고차원 특징 추가 및 복잡한 하이퍼파라미터 탐
색이 오히려 과적합(overfitting)이나 정보 손실로 이어졌을 가능성이 높으며, 특
히 이상치 제거 없이 시계열에 취약한 날짜 분해 특성만 추가된 구조가 예측 불
안정성을 증폭시킨 것으로 분석된다.

요약하면, XGBoost 기반 모델의 성능은 단순히 복잡한 구조의 도입보다도 적
절한 이상치 제거, 유의미한 특징 선택, 손실 함수의 강건성 확보 여부에 따라 큰
차이를 보인다. 특히, XGBoost_a 모델은 전체 구조의 균형을 적절히 유지하며
가장 안정적이고 우수한 성능을 달성하였다.

표 12. 제주삼다수 공장 데이터셋 기반 다변량 XGBoost 모델의 입력
변수 최적 조합

Optimal combination of variables for Jeju	
XGBoost_a	XGBoost_b
power_lag1	power_lag1
power_lag2	power_lag2
power_lag3	power_lag3
power_lag4	temp_lag1
power_lag5	temp_lag2
temp_lag1	local_atmospheric_pressure_hPa_lag1
temp_lag2	local_atmospheric_pressure_hPa_lag2
local_atmospheric_pressure_hPa_lag1	power_rolling_mean_3
local_atmospheric_pressure_hPa_lag2	power_rolling_std_3
power_rolling_mean_3	power_rolling_mean_6
power_rolling_std_3	power_rolling_std_6
power_rolling_mean_6	power_rolling_mean_12
power_rolling_mean_12	power_rolling_std_12
power_rolling_std_12	wind_temp_interaction
pressure_temp_interaction	pressure_temp_interaction

표 13. 모로코 지역1 LSTM 모델 성능지표

Morocco zone1 LSTM						
Model name	R ²	Compare Base Model	MAPE	Compare Base Model	sMAPE	Compare Base Model
LSTM_basic	0.983	0%	1.95	0%	1.93	0%
LSTM_a	0.986	0.305%	1.92	1.538%	1.89	2.073%
LSTM_b	0.959	2.442%	4.027	-106.513%	3.915	-102.85%
LSTM_c	0.997	1.424%	0.807	58.615%	0.804	58.342%
LSTM_d	0.851	-13.428%	6.904	-254.051%	6.6	-241.969%

표 13에는 LSTM_basic 기준 모델 R² = 0.983, MAPE = 1.95%, sMAPE = 1.93% 기본 LSTM만으로도 이미 높은 설명력과 낮은 예측 오차를 보인다. LSTM_a R² = 0.986 (+0.31%), MAPE = 1.92% (-1.54%), sMAPE = 1.89% (-2.07%)로 배치 정규화·잔차 연결·자동 레 생성·이상치 제거를 도입해 모든 지표가 근소하게 개선됐다. 원인은 이상치 제거로 노이즈가 줄고, 배치 정규화가 학습 안정성을 높여 예측 정밀도가 소폭 상승했다. LSTM_b R² = 0.959 (-2.44%), MAPE = 4.03% (+106.5%), sMAPE = 3.92% (+102.9%)로 정규화·잔차 연결만 적용하고 이상치 제거를 뺀 결과, 설명력과 오차 지표가 크게 악화되었다. 원인은 모든 데이터 패턴을 학습하며 극단치가 그대로 반영돼 오차가 배 이상 커졌다. LSTM_c R² = 0.997 (+1.42%), MAPE = 0.81% (-58.6%), sMAPE = 0.80% (-58.3%)로 동적 z-score 이상치 제거와 Huber Loss 결합으로 모든 지표에서 최상의 결과를 달성했다. 원인인 Huber Loss가 작은 오차엔 MSE처럼, 큰 오차엔 MAE처럼 작용하며 안정적인 학습을 유도하고, 동적 이상치 임계값 최적화가 노이즈를 효과적으로 제어했다.

LSTM_d R² = 0.851 (-13.4%), MAPE = 6.90% (+254.1%), sMAPE = 6.60% (+242.0%)로 Huber Loss만 썼지만 이상치 제거를 제외하니 예측 성능이 최악으로 떨어졌다. 원인은 Huber Loss의 강건성만으로는 극단치로 인한 왜곡을 충분히 막지 못해, 오차가 크게 확대됐다. 소폭 개선이 목표라면 LSTM_a(이상치 제거 + 정규화), 설명력과 오차 최소화의 균형이 필요하다면 LSTM_c(동적 이상치 제거 + Huber Loss), 가장 민감한 오차 방어가 중요하다면 LSTM_c 데이터 전 범위 학습을 원하되 어느 정도 오차 허용이 가능하다면 LSTM_b 를 고려할 만

하다.

표 14. 모로코 지역1 데이터셋 기반 다변량 LSTM 모델의 입력 변수 최적 조합

Optimal combination of variables for Morocco zone1	
LSTM_a	LSTM_b
PowerConsumption_Zone1_lag1	PowerConsumption_Zone1_lag1
PowerConsumption_Zone1_lag2	PowerConsumption_Zone1_lag2
PowerConsumption_Zone1_lag3	PowerConsumption_Zone1_lag3
PowerConsumption_Zone1_lag4	PowerConsumption_Zone1_lag4
PowerConsumption_Zone1_lag5	PowerConsumption_Zone2_lag1
PowerConsumption_Zone2_lag1	PowerConsumption_Zone3_lag1
PowerConsumption_Zone2_lag2	sum_all_zone_lag1
sum_all_zone_lag1	sum_all_zone_lag2
sum_all_zone_lag2	PowerConsumption_Zone1_rolling_m an_3
PowerConsumption_Zone1_rolling_m an_3	PowerConsumption_Zone1_rolling_m an_6
PowerConsumption_Zone1_rolling_m an_6	PowerConsumption_Zone1_rolling_m an_12
PowerConsumption_Zone1_rolling_m an_12	PowerConsumption_Zone3_rolling_m an_3
PowerConsumption_Zone3_rolling_m an_3	PowerConsumption_Zone3_rolling_m an_6
wind_PowerConsumption_Zone2_inter action	wind_PowerConsumption_Zone2_inter action
pressure_PowerConsumption_Zone2_in teraction	pressure_PowerConsumption_Zone2_in teraction

표 15. 모로코 지역1 XGBoost 모델 성능지표

Morocco zone1 XGBoost						
Model name	R ²	Compare Base Model	MAPE	Compare Base Model	sMAPE	Compare Base Model
XGBoost_basic	0.975	0%	2.51	0%	2.5	0%
XGBoost_a	0.991	1.641%	0.918	63.426%	0.817	67.32%
XGBoost_b	0.991	1.641%	0.997	60.279%	0.648	74.08%
XGBoost_c	0.211	-78.359%	16.532	-558.645%	16.031	-541.24%
XGBoost_d	0.202	-79.282%	16.846	-571.155%	16.249	-549.96%

표 15는 기본 모델인 XGBoost_basic은 $R^2 = 0.975$, MAPE = 2.51, sMAPE = 2.5의 준수한 성능을 보였다. 이는 기본적인 구조만으로도 해당 지역의 시계열 예측 문제에 대해 상당히 높은 설명력을 가지며, 비교 기준점으로서 적절함을 나타낸다. XGBoost_a 모델은 Optuna 기반 하이퍼파라미터 탐색, z-score 이상치 제거, 다양한 시차 및 이동 통계 특징, 상호작용 항, SelectKBest, Huber Loss 등 고급 기법을 통합하였다. 해당 모델은 $R^2 = 0.991$ 로 1.641% 향상되었으며, MAPE = 0.918 (-63.4%), sMAPE = 0.817 (-67.32%)로 예측 정확도 측면에서 큰 폭의 개선을 달성하였다. 이는 이상치 제거와 특징 공학의 시너지 효과를 보여주는 결과로, 노이즈 억제 및 학습 안정성 확보에 기여하였음을 알 수 있다.

XGBoost_b 모델은 XGBoost_a와 동일한 구조에서 이상치 제거만 생략한 모델이다. R^2 역시 0.991로 동일하지만, MAPE와 sMAPE는 각각 0.997, 0.648로 약간 상승하였다. 이는 이상치 제거가 없더라도 나머지 구조적 요소만으로도 충분한 성능 확보가 가능하며, 다만 정밀도 관점에서는 약간의 손실이 발생함을 시사한다. 두 모델 모두 기본 모델 대비 현저한 성능 향상을 보였다.

반면, XGBoost_c는 z-score 임계값을 Optuna로 자동 탐색하여 이상치를 제거하고, 날짜 정보를 분해한 time feature들을 추가한 모델이다. 그러나 $R^2 = 0.211$, MAPE = 16.532, sMAPE = 16.031로 성능이 급격히 악화되었다. 이는 자동화된 이상치 필터링 과정에서 중요 데이터를 과도하게 제거하거나, time feature가 예측에 오히려 잡음을 추가한 경우로 해석된다. 특히 시계열 구조가 강한 데이터에서 불필요한 시간 변수의 도입은 모델 일반화 능력을 저해할 수 있음을 보여준다.

XGBoost_d 모델은 XGBoost_c에서 이상치 제거를 생략한 구조이다. $R^2 = 0.202$, MAPE = 16.846, sMAPE = 16.249로 여전히 성능이 저조하다. 이는 날짜 기반 특징과 복잡한 하이퍼파라미터 설정이 단독으로는 오히려 과적합 또는 정보 왜곡을 야기할 수 있으며, 특히 해당 지역에서는 단순한 구조가 오히려 더 효율적임을 나타낸다.

요약하면, 모로코 지역1번 데이터에서는 복잡한 구조보다도 정교한 이상치 제거와 적절한 특징 선택이 예측 성능 향상에 핵심적이며, 특히 XGBoost_a와 XGBoost_b는 비교적 단순한 구성으로도 뛰어난 성능을 보여줌으로써 실용성과 정확성 측면 모두에서 효과적인 접근임을 입증하였다.

표 16. 모로코 지역1 데이터셋 기반 다변량 XGBoost 모델의 입력 변수 최적 조합

Optimal combination of variables for Morocco zone1	
XGBoost_a	XGBoost_b
PowerConsumption_Zone1_lag1	PowerConsumption_Zone1_lag1
PowerConsumption_Zone1_lag2	PowerConsumption_Zone1_lag2
PowerConsumption_Zone1_lag3	PowerConsumption_Zone1_lag3
PowerConsumption_Zone1_lag4	PowerConsumption_Zone1_lag4
PowerConsumption_Zone1_lag5	PowerConsumption_Zone2_lag1
PowerConsumption_Zone2_lag1	PowerConsumption_Zone3_lag1
PowerConsumption_Zone2_lag2	sum_all_zone_lag1
sum_all_zone_lag1	sum_all_zone_lag2
sum_all_zone_lag2	PowerConsumption_Zone1_rolling_mean_3
PowerConsumption_Zone1_rolling_mean_3	PowerConsumption_Zone1_rolling_mean_6
PowerConsumption_Zone1_rolling_mean_6	PowerConsumption_Zone1_rolling_mean_12
PowerConsumption_Zone1_rolling_mean_12	PowerConsumption_Zone3_rolling_mean_3
PowerConsumption_Zone3_rolling_mean_3	PowerConsumption_Zone3_rolling_mean_6
wind_PowerConsumption_Zone2_interaction	wind_PowerConsumption_Zone2_interaction
pressure_PowerConsumption_Zone2_interaction	pressure_PowerConsumption_Zone2_interaction

4.2 개발 및 실험 환경

본 시스템의 개발은 Python 3.10 기반의 Flask 웹 프레임워크 위에서 이루어졌으며, 데이터 처리·분석을 위해 Pandas, NumPy, SciPy, StatsModels를 활용했다. 머신러닝 및 딥러닝 모델 학습에는 TensorFlow 2.10.0, PyTorch 2.3.0, XGBoost 2.1.4, Prophet, Scikit learn 1.6.1, NeuralProphet 0.8.0을 사용했으며, 데이터베이스 연동에는 SQLAlchemy 2.0.38과 Flask SQLAlchemy 3.1.1을 적용했다. 사용자 인증과 보안을 위해 Flask Login 0.6.3, Flask Bcrypt 1.0.1, Flask WTF 1.2.1을 도입했다. 프론트엔드 구현은 HTML5·CSS3·JavaScript 기반의 React(버전 18)와 TailwindCSS, NiceGUI 2.11.1을 사용했으며, 시각화 라이브러리로 Plotly.js, Bokeh 3.7.2, Matplotlib을 활용했다. 개발·배포 파이프라인은 Git 버전 관리, GitHub Actions CI, Vercel 호스팅(vercel.json)으로 구성하였으며, 가상환경(venv)과 pip를 통해 종속성을 관리했다. 운영 체제는 Windows와 Linux를 모두 지원하며, 고성능 연산을 위해 CUDA 지원 GPU 환경을 기반으로 테스트했다[표 17], [표 18].

표 17. HW 실험 환경

CPU	12th Gen Intel(R) Core(TM) i9-12900K 3.20 GHz
GPU	NVIDIA GeForce RTX 4060 Ti
RAM	32.0GB

표 18. SW 개발 환경

Category	Detail	Tech name	Version
Backend Technologies	Core Framework	Flask	
		Python	3.x
	Data Processing & Analysis	Pandas	
		NumPy	
		SciPy	
		StatsModels	
	AI/ML Frameworks & Libraries	TensorFlow	2.10.0
		PyTorch	2.3.0
		Scikit-learn	1.6.1
	Database & ORM	SQLAlchemy	2.0.38
Flask-SQLAlchemy		3.1.1	
Authentication & Security	Flask-Login	0.6.3	
	Flask-Bcrypt	1.0.1	
	Flask-WTF	1.2.1	
Frontend Technologies	Core Technologies	HTML5	
		CSS3	
		JavaScript	
	Visualization	Plotly.js	
		Bokeh	3.7.2
	UI Components	Matplotlib	
	TailwindCSS		
Environment Requirements	Operating System Support	NiceGUI	2.11.1
	System Dependencies	Windows	
Additional Libraries & Tools	API & Web Services	CUDA support	
		GPU acceleration	
		FastAPI	0.115.6
	Data Storage & Processing	Uvicorn	0.34.0
		aiohttp	3.11.12
		PyArrow	19.0.1
		Dask	2025.3.0
Testing & Development	pytest	8.3.5	
	debugpy	1.6.4	

V. 결론 및 향후 연구

5.1 결론

본 연구는 사무실 및 공장 등 기업 현장의 전력 사용 데이터를 기반으로 인공지능 예측 기법을 활용한 에너지 관리 시스템을 설계하고 구현함으로써, 기업의 효율적인 에너지 소비와 비용 절감, 환경적 책임 이행에 기여할 수 있는 가능성을 실증적으로 보여주었다.

본 연구에서는 제주 삼다수 공장 전력 소비 예측과 모로코 테투안시 구역별 전력 소비량 예측을 위해 LSTM, XGBoost, Prophet, RNN, Transformer 등 모델을 각각 구현하고, 모델의 약점을 보완하여 변형된 모델을 각각 구현하였다. 또한, 원본 데이터와 이상치 제거 데이터, 다변수 학습 및 단일변수 학습을 종합적으로 비교·분석하였다. 성능 평가는 R^2 , MAPE, sMAPE 지표를 기준으로 수행되었으며, 구현된 모델들이 각 기본 모델 대비 우수한 예측 정확도를 달성함을 확인하였다. 특히, 제주 삼다수 공장 전력 사용량 데이터셋에서는 LSTM 모델은 최대 68%, XGBoost 모델은 36%까지 MAPE 감소 효과를 나타냈으며, 모로코 지역 1번 전력 사용량 데이터셋에서도 LSTM 모델은 최대 58%과 XGBoost 모델은 60%까지 MAPE 감소 등 높은 성능을 보였다. 반면, XGBoost는 지역 및 데이터 특성에 따라 예측 안정성에 한계를 드러내는 등 모델별 성능 편차가 확인되었다.

또한, 기술적 기여 외에도 실용적 구현을 고려하여, 웹 기반 및 온프로그래밍 형태의 시스템을 함께 개발하였다. 사용자 회원가입을 포함한 실제 운영이 가능하도록 플랫폼을 설계함으로써, 실증 연구를 넘어 실제 활용성까지 확보한 점이 주목할 만하다. 이러한 시스템은 기업의 운영 현장에 직접 도입되어 에너지 소비 패턴 분석, 이상 징후 탐지, 피크 수요 예측, 그리고 에너지 사용 계획 수립 등 다양한 기능을 통해 종합적인 에너지

전략 수립을 가능하게 한다.

결론적으로 본 연구는 기업들이 에너지 자원을 보다 체계적이고 효율적으로 관리함으로써 에너지 비용 절감은 물론, ESG 경영 실천과 지속 가능한 성장을 위한 기반을 마련하는 데 실질적인 기여를 할 수 있는 통합 솔루션을 제시하였다. 이는 에너지 관리 시스템의 상품화 가능성, 새로운 비즈니스 모델의 창출, 나아가 전력망 전체의 안정성과 친환경성 확보 측면에서도 중요한 의미를 가질 것으로 기대한다.



5.2 향후 연구

본 연구는 인공지능 기반 전력 예측·관리 시스템의 실효성을 입증하였으나, 산업 현장에 폭넓게 적용하기 위해서는 추가적인 기술 고도화가 요구된다. 향후 연구는 다음 세 가지 방향으로 구체화한다.

1. 우선순위 기반 스마트 스케줄링 알고리즘 개발: 설비별 중요도, 가동 패턴, 생산성과의 상관성을 반영해 전력 공급 우선순위를 자동 산정하는 알고리즘을 구현한다. 실시간 예측값과 연동하여 피크 시간대 총 소비 전력을 기준 이하로 유지하도록 제어함으로써, 비용 절감과 생산 안정성을 동시에 확보한다.

2. SaaS(Software as a Service)·온프로미스 통합 에너지 플랫폼 고도화: 웹 기반 SaaS 형태로 사용자 맞춤형 대시보드, 자동 리포트, 알림·경고 기능을 제공하고, ERP(Enterprise Resource Planning)·MES(Manufacturing Execution System) API 연동으로 손쉬운 도입을 지원한다. 동시에 자원 제약이 큰 온프로미스 환경을 고려해 TensorFlow Lite, PyTorch Mobile 등 경량 모델을 적용하여 추론 속도와 연산 효율을 극대화한다.

3. 외부 요인 통합형 하이브리드 예측 모델 및 이상 탐지 강화: 기온, 습도, 공휴일, 생산 일정 등 비전력 변수를 정량화해 모델 입력에 통합하고, 이상 탐지 모듈을 고도화하여 비정상 소비나 설비 이상 징후를 조기에 식별한다. 이를 통해 예측 정확도를 높이고 운영 리스크를 최소화한다.

이와 같은 후속 연구는 본 시스템을 산업 현장에 적용 가능한 종합 에너지 솔루션으로 발전시키며, 기업의 비용 절감과 탄소 중립 달성을 동시에 지원하는 핵심 기반 기술로 자리매김할 것이다.

참고문헌

- [1] Korea Electric Power Corporation introduces base rate peak-linking system .
<https://home.kepco.co.kr/kepco/CY/K/htmlView/CYKAHP00301.do?menuCd=FN0207010103>
- [2] Song, J., Ku, D., Bae, S., Seo, G., Yang, H., & Kim, M. (2023). A study on the prediction of steam usage in the factory energy management system (FEMS) through regression analysis model. Korean Society of Mechanical Engineers Spring and Autumn Conference, 1-6, November 3-4. Daejeon, Korea.
- [3] Lee, H.-A., Kim, D.-J., Cho, WW.-J., & Gu, J.-H. (2023). Optimization of energy consumption prediction model of food factory based on LSTM for application to FEMS. Journal of Environmental & Thermal Engineering, 18(1), 7-19.
- [4] Kim, S. H., Heo, T.-W., & Lee, I.-W. (2023). Electricity consumption analysis and prediction with quality control charts in factory energy management systems. Proceedings of the Korean Institute of Electrical Engineers Conference, 1-6, May 24-26. Jeju, Korea.
- [5] Ngo, M.-T., Dinh, M.-C., Dao, V.-Q., Kim, C.-S., & Park, M. (2022). Design of a day-ahead power consumption prediction model for a factory energy management system based on long short-term memory and recurrent neural network. Proceedings of the Korean Institute of Electrical Engineers Conference, 1-6, May 25-27. Busan, Korea.
- [6] Lee, K.-M., Choi, S.-C., Lee, Y.-J., Park, S.-O., Park, J.-W., Cho, Y.-Y., & Shin, C.-S. (2023). Comparison between power consumption prediction models to reduce energy consumption. Proceedings of KIIT Conference,
- [7] Quan, J., Shin, J.-W., Ko, J.-L., & Shin, S.-K. (2021). A study on energy consumption prediction from building energy management system data with missing values using SSIM and VLSW algorithms. The Transactions of The Korean Institute of Electrical Engineers, 70(10), 1540-1547. <https://doi.org/10.5370/KIEE.2021.70.10.1540>

- [8] Gao, Y., Li, S., Xiao, Y., Dong, W., Fairbank, M., & Lu, B. (2022). An iterative optimization and learning-based IoT system for energy management of connected buildings. *IEEE Internet of Things Journal*, 9(21), 21246–21259.
- [9] Alghamdi, M. (2022). Modelling and implementing an energy-optimised IoT architecture for smart homes. *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, 1–6. Karak, Jordan.
- [10] Salama, A. K., & Abdellatif, M. M. (2022). AIoT-based smart home energy management system. *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, 177–181. Alamein New City, Egypt.
- [11] Guenfaf, Y., & Zafoune, Y. (2023). An IoT ML-based system for energy efficiency in smart homes. *2023 IEEE World AI IoT Congress (AIIoT)*, 198–203. Seattle, WA, USA.
- [12] Leong, Y. M., Lim, E. H., & Lim, L. K. (2023). A review of potential AI-based automation for IoT-enabled smart homes. *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, 1–6. Shah Alam, Malaysia.
- [13] Mišljenović, N., Žnidarec, M., Knežević, G., Šljivac, D., & Sumper, A. (2023). A review of energy management systems and organizational structures of prosumers. *Energies*, 16(7), 3179. <https://doi.org/10.3390/en16073179>
- [14] Sharma, P., Mathur, H. D., Mishra, P., & Bansal, R. C. (2022). A critical and comparative review of energy management strategies for microgrids. *Applied Energy*, 327, 120028. <https://doi.org/10.1016/j.apenergy.2022.120028>
- [15] Mischos, S., Dalagdi, E., & Vrakas, D. (2023). Intelligent energy management systems: A review. *Artificial Intelligence Review*, 56, 11635–11674. <https://doi.org/10.1007/s10462-023-10441-3>
- [16] Khafiso, T., Aigbavboa, C., & Adekunle, S. A. (2024). Barriers to the adoption of energy management systems in residential buildings. *Facilities*, 42(15/16), 107–125.

- <https://doi.org/10.1108/F-12-2023-0113>
- [17] Alam, M. S., & Arefifar, S. A. (2019). Energy management in power distribution systems: Review, classification, limitations and challenges. *IEEE Access*, PP, 1-1. <https://doi.org/10.1109/ACCESS.2019.2927303>
- [18] Ji, J., Yu, H., Wang, X., & Xu, X. (2025). Machine learning application in building energy consumption prediction: A comprehensive review. *Journal of Building Engineering*, 104, 112295. <https://doi.org/10.1016/j.jobe.2025.112295>
- [19] Wang, K., Zhang, J., Li, X., & Zhang, Y. (2023). Long-term power load forecasting using LSTM-Informer with ensemble learning. *Electronics*, 12(10), 2175. <https://doi.org/10.3390/electronics12102175>
- [20] Galindo Padilha, G. A., Ko, J., Jung, J. J., & de Mattos Neto, P. S. G. (2022). Transformer-based hybrid forecasting model for multivariate renewable energy. *Applied Sciences*, 12(21), 10985. <https://doi.org/10.3390/app122110985>
- [21] Antonese, G., Cioara, T., Anghel, I., Papias, I., Michalakopoulos, V., & Sarmas, E. (2025). Hybrid transformer model with liquid neural networks and learnable encodings for buildings' energy forecasting. *Energy and AI*, 20, 100489. <https://doi.org/10.1016/j.egyai.2025.100489>
- [22] Cao, Y., Liu, G., Luo, D., Bavirisetti, D. P., & Xiao, G. (2023). Multi-timescale photovoltaic power forecasting using an improved stacking ensemble algorithm based LSTM-Informer model. *Energy*, 283, 128669. <https://doi.org/10.1016/j.energy.2023.128669>
- [23] Li, Y., Liu, G., Cao, Y., Chen, J., Xiao, G., & Tang, J. (2025). WNPS-LSTM-Informer: A hybrid stacking model for medium-term photovoltaic power forecasting with ranked feature selection. *Renewable Energy*, 244, 122687. <https://doi.org/10.1016/j.renene.2025.122687>
- [24] Kamalov, F., Sulieman, H., Moussa, S., Avante Reyes, J., & Safaraliev, M. (2024). Powering electricity forecasting with transfer learning. *Energies*, 17(3), 626. <https://doi.org/10.3390/en17030626>
- [25] Kamoona, A., Song, H., Keshavarzian, K., Levy, K., Jalili, M.,

- Wilkinson, R., Yu, X., McGrath, B., & Meegahapola, L. G. (2023). Machine learning based energy demand prediction. *Energy Reports*, 9, 171-176. <https://doi.org/10.1016/j.egy.2023.09.151>
- [26] Egwim, C., Egunjobi, O., Gomes, A., & Alaka, H. (2022). A comparative study on machine learning algorithms for assessing energy efficiency of buildings. In *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-030-93733-1_41
- [27] Song, W., Gao, C., Zhao, Y., & Zhao, Y. (2020). A time series data filling method based on LSTM—Taking the stem moisture as an example. *Sensors*, 20(18), 5045.
- [28] Ugli, N. I. R., Bo, P., Yanxia, L., Izatilo, Y., Tae-oh, L., & Tae-guk, K. (2024). Development of AI-based energy management system for power saving in factories. *Journal of Internet of Things Convergence*, 10(6), 49-55.
- [29] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [30] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [31] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [32] Wang, W., Chakraborty, G., & Chakraborty, B. (2021). Predicting the risk of chronic kidney disease (CKD) using machine learning algorithm. *Applied Sciences*, 11, 202.
- [33] <https://www.qwak.com/post/xgboost-versus-random-forest>
- [34] Zheng, Y., Liu, Y., Jiang, Z., Tang, Q., & Xiang, Y. (2022). Wind power forecasting based on Prophet model. 2022 IEEE/IAS Industrial and Commercial Power System Asia (IW&CPS Asia), Shanghai, China, 1544-1548.
- [35] https://facebook.github.io/prophet/docs/quick_start.html
- [36] Wang, J., et al. (2022). NGCU: A new RNN model for time-series data prediction. *Big Data Research*, 27, 100296.
- [37] Jin, X., et al. (2020). Prediction for time series with CNN and LSTM. In *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*. Singapore: Springer.
- [38] Kim, D.-K., & Kim, K. (2022). A convolutional transformer model for

- multivariate time series prediction. *IEEE Access*, 10, 101319–101329.
- [39] Zhu, R., et al. (2020). Smart transformer/large flexible transformer. *CES Transactions on Electrical Machines and Systems*, 4(4), 264–274.
- [40] Shi, X., et al. (2024). Time-MoE: Billion-scale time series foundation models with mixture of experts. arXiv preprint arXiv:2409.16040.
- [41] Huang, X., Tang, J., & Shen, Y. (2024). Long time series of ocean wave prediction based on PatchTST model. *Ocean Engineering*, 301, 117572.
- [42] Lin, P., et al. (2025). Multi-timescale short-term urban water demand forecasting based on an improved PatchTST model. *Journal of Hydrology*, 651, 132599.
- [43] Ekambaram, V., et al. (2023). TSMixer: Lightweight MLP-Mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [44] Huang, Z., & He, K. (2024). GRU-TSMixers: Sleep apnea and hypopnea detection based on multi scale MLP-mixers. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE.
- [45] <https://datasciencefromsebi.tistory.com/65>
- [46] <https://uky-note.tistory.com/24>
- [47] Herrera-Barajas, L. A., et al. (2024). Importance of data preprocessing in artificial neural network training for medical diagnosis. In *Innovative Applications of Artificial Neural Networks to Data Analytics and Signal Processing* (pp. 223–238). Springer, Cham.
- [48] <https://pseudo-lab.github.io/Tutorial-Book/chapters/time-series/Ch3-preprocessing.html>
- [49] <https://bommbom.tistory.com/entry/%ED%95%98%EC%9D%B4%ED%8D%BC%ED%8C%8C%EB%9D%BC%EB%AF%B8%ED%84%B0-%ED%8A%9C%EB%8B%9D-%EB%B2%A0%EC%9D%B4%EC%A7%80%EC%95%88-%EC%B5%9C%EC%A0%81%ED%99%94Bayesian-Optimization>
- [50] <https://www.ultralytics.com/ko/glossary/evolutionary-algorithms>

- [51] Oliveira, M., & Torgo, L. (2015). Ensembles for time series forecasting. In Asian Conference on Machine Learning. PMLR.
- [52] <https://www.data.go.kr/data/15124355/fileData.do>
- [53] <https://www.kma.go.kr/jeju/html/main/index.jsp>
- [54] <https://archive.ics.uci.edu/dataset/849/power+consumption+of+tetouan+city>
- [55] Wang, K., Zhang, J., Li, X., & Zhang, Y. (2023). Long-term power load forecasting using LSTM-Informer with ensemble learning. Electronics, 12(10), 2175.
- [56] <https://www.tutorialspoint.com/xgboost/xgboost-architecture.htm>

