

객관적 기후예측에 적합한 인공지능 활용 기반기술 개발(II)

부제: 기후예측에 적합한 인공지능모델의 민감도 분석 및 개선

**The Development of Artificial Intelligence Base
Technologies for Objective Climate Predictions
(II)**

김미애, 박경원, 이성규, 이운영, 정유란

기후사업본부 예측기술과

2023.12.

APEC기후센터

Executive Summary

Recently, artificial intelligence (AI)-based climate prediction models within 14 days have been developed by global companies and research centers such as NVIDIA (FourCastNet), Google DeepMind (GraphCast), HUAWEI (Pangu-Weather), ECMWF AIFS (European Centre for Medium-Range Weather Forecasts Artificial Intelligence/Integrated Forecasting System). These models are in experimental operation via the ECMWF website. However, scaling to sub-seasonal and seasonal timescales is still lacking as these models focus on a forecast horizon of 14 days or less. In addition, AI research requires a large amount of training data, but the ECMWF ERA-5 reanalysis, the climate data currently used by global companies for training, has only about 74 years of data from 1950 to present. If the data is divided into training data, validation data, and test data, the dataset used for actual learning are very scarce.

Due to the climate crisis, the importance of highly accurate sub-seasonal prediction data in the applied climate field has grown significantly. Sub-seasonal prediction data faces challenges in predicting the diverse interactions of the atmosphere and oceans solely through numerical models based on physical principles. To overcome these challenges, an Attention U-Net model and a seasonal prediction model were developed to give more weight to important patterns and features in the input data, aiming to improve accuracy. Additionally, various sensitivity analyses were conducted using techniques such as Filters, Wrappers, and Embedded methods based on observational data, identifying features inherent in variables. By selecting specific combinations of variables from both the model and observational data, it was confirmed that accuracy improved compared to ECMWF model predictions. However, in many combinations of variables, statistical results did not consistently demonstrate optimized outcomes. Therefore, it is inferred that future improvements in artificial intelligence models in the climate field, considering the characteristics of variables and recent advancements, could lead to the utilization of more accurate prediction data.

The research was carried out to explore methods to enhance the prediction accuracy of Seasonal to Sub-seasonal (S2S) climate variables (such as maximum daily temperature and daily total precipitation) by extending the U-Net architecture using Attention and Residual mechanism while optimizing hyperparameters. Attention measures the relevance between inputs and subsequent outputs to focus on specific information, while Residual learning addresses the vanishing gradient problem. In this study, Attention U-Net and Residual U-Net were constructed by adding these two mechanisms to the existing U-Net, respectively, and Attention based-on Residual U-Net was constructed by combining the two mechanisms. Grid search algorithm was employed to optimize hyperparameters, with epochs, batch size, and learning rate set as key parameters. Training data utilized S2S prediction data. The optimized hyperparameter combinations showed similar trends in most S2S climate models. In the

extended U-Net models of predictions of daily maximum temperature and daily total precipitation during the test period, the models with added Attention or Residual demonstrated improved accuracy, with the Attention-based Residual U-Net, incorporating both mechanisms, exhibiting superior performance. Specifically, the Residual mechanism appeared to influence temperature prediction, while Attention effectively improved precipitation prediction. However, limitations persisted in enhancing the week 1 prediction of temperature and the week 2-3 prediction of precipitation. Subsequent research is expected to propose suitable methods for S2S temperature and precipitation predictions by leveraging ensemble techniques to combine high-performing models.

Historical climate data are often not enough to train a deep learning model, especially when working with monthly data. A specific climate phenomenon such as Madden-Julian Oscillation (MJO) only accounts for a part of long-term climate data that have been collected over decades. Semi-supervised learning (SSL) is ideally suited to such situations, as it can effectively leverage a small amount of data while maintaining accuracy. In this study, we developed a SSL-based deep learning model for MJO phase classification using decades of climate anomaly images. First of all, a supervised learning approach was employed with all labeled data to classify MJO phases. Various input variables, data split strategies, model architectures were tested, and a optimal model was chosen based on test accuracy. The selected model was then trained in semi-supervised learning framework. The sensitivity of the SSL-based model, developed with Mean Teacher SSL algorithm, was analyzed with respect to data augmentation methods, model training process, consistency loss weight differences, and the number of labeled data. Comparison with results from using most or all labeled data (supervised learning) showed that the SSL-based model achieved comparable or even better performance with only about half of the whole labeled data. In addition to MJO phase classification, we also tested spatiotemporal predictive AI models to forecast MJO input fields on the next day or consecutive 7 days using timeseries of input fields with various pre-sequence lengths. Different sampling methods such as excluding inactive MJO cases or summer seasons were investigated.

In the climate field, research using deep learning techniques increases rapidly. However, research on improving the model by understanding a deep learning model is insufficient. In this study, XAI, loss surface analysis, and model internal structure analysis and improvement were performed to better understand deep learning models and derive improvements to the models. We analyzed whether the models built with supervised and semi-supervised learning through XAI techniques extracted similar information from the input data and used it to make predictions. Also, we conducted a study to determine whether the structure of the model is advantageous for finding the global minimum by analyzing the loss surface. In addition, we evaluated the predictability of models designed by attention mechanisms, dataset expansion, increasing the number of layer filters, and improving the model structure by analyzing the internal structure and feature collapse of the deep learning model.

국문 요약

최근 NVIDIA(FourCastNet), 구글 딥마인드(GraphCast), 화웨이(Pangu-Weather), ECMWF(AIFS) 등 글로벌 기업과 글로벌 연구소를 중심으로 14일 이내의 인공지능 기후예측모델이 개발되고 있다. 이들 모델은 ECMWF 웹사이트를 통해 시범 운영 중에 있다. 그러나 여전히 예측기간이 14일 이내로 계절내 및 계절 시간규모로의 확장은 아직 미흡하다. 또한, 인공지능 연구에는 대량의 학습자료가 필요하지만 현재 글로벌 기업들이 학습에 사용한 기후자료인 ECMWF ERA-5 재분석자료는 1950년부터 현재까지 약 74년의 자료만 구축되어 있다. 이 자료를 학습자료, 검증자료, 테스트자료로 나눠서 사용하면 실제 학습에 사용되는 자료는 매우 부족하다.

기후위기로 인하여 응용기후분야에 정확도 높은 계절 내 예측자료의 중요성이 커지고 있다. 이러한 계절 내 예측자료는 물리적 기반으로 다양한 대기 및 해양의 상호작용을 수치모델만으로 예측하기 어려운 문제에 직면하고 있다. 이러한 어려움을 극복하기 위하여 입력자료에서 보다 중요한 패턴과 특징에 가중치를 줘서 정확도를 개선하고자 하는 Attention U-Net 모델과 계절 내 예측 모델 뿐만 아니라 관측자료를 기반으로 Filter, Wrapper, Embedded 등의 기법을 이용하여 변수가 가지고 있는 특징들을 찾아내어 다양한 민감도 분석을 실시하였다. 모델/관측자료의 변수의 선택으로 일부 변수조합으로 계절내 예측보다 정확도가 향상됨을 확인할수 있었다. 그러나 많은 변수들의 조합에서는 통계적 결과가 최적화된 결과를 보여주지 못한다는 것을 알수가 있었다. 따라서 향후 변수가 가지는 특성과 최근 발전한 기후분야의 인공지능 모델을 개선한다면 보다 정확도가 향상된 예측자료를 활용할 수 있을 것으로 판단된다.

U-Net 아키텍처를 개선하기 위해 Attention과 Residual 장치를 활용하여 모델을 확장하고, 초매개변수를 최적화하여 S2S (Sub-seasonal to Seasonal) 기후 변수 (예, 최고기온 및 강수)의 예측 정확도를 향상시키는 방법을 탐색하였다. 먼저, Attention은 입력과 다음 결과 간의 연관성을 측정하여 특정 정보에 집중하는 기법이고, Residual learning은 기울기 소실 문제를 해결하기 위한 기법이다. 본 연구에서는 이 두 가지를 기존 U-Net에 각각 추가하여 Attention U-Net 및 Residual U-Net을 구축하고, 두 장치를 결합한 Attention based-on Residual U-Net을 구축하였다. 또한, 초매개변수를 최적화하기 위해 그리드 탐색(Grid search) 알고리즘을 사용하였다. 에포크, 배치 사이즈, 학습률에 대한 주요 초매개변수에 대해서 최적화를 수행하였다. 훈련 데이터는 계절 내 기후 예측자료를 활용했는데, 대부분의 기후 모델에서 최적화된 초매개변수 조합은 유사한 경향을 보였다. S2S 최고기온과 강수 예측에서, Attention 혹은 Residual이 추가된 U-Net에서 ACC가 향상되었으며, 두 장치가 모두 장착된 Attention based-on Residual U-Net이 가장 우수한 결과를 보였다. 특히, Residual 장치가 기온 예측에 영향을 준 것으로 보이며, Attention은 강수 예측을 개선하는 데 효과적으로 작용한 것으로 추측된다. 그러나, 여전히 기온의 1주 예측 성능과 강수의 2-3주 예측 성능을 향상시키는 데 한계가 있었다. 이에 따라 후속 연구에서는 앙상블 기법을 활용하여 높은 성능을 보이는 모델들을 조합하고, S2S 기온 (최고 및 최저기온) 및 강수 예측에 적합한 방법을 제시할 것으로 기대된다.

기후 자료는 수십 년간 축적되어 왔음에도 MJO와 같은 특정 기후 현상은 그중 일부에 해당하기 때문에 딥러닝 모델에 활용되기에 자료의 양이 충분하지 않은 경우가 많다. 이는 월별

로 자료를 활용하는 경우 더욱 제한적이다. 준지도학습은 적은 양의 라벨링된 자료를 효율적으로 활용해 높은 성능의 모델을 구현할 수 있다. 본 연구에서는 준지도학습 기법을 기반으로 MJO 위상을 분류하는데 수십 년의 이미지를 활용하여 딥러닝 모델을 개발하였다. 우선 모든 자료를 라벨링하여 MJO 위상을 분류하는 지도학습 모델을 개발하였다. 다양한 변수, 훈련 자료 분리 방법, 모델 구조에 따라 다양하게 개발하고 최적의 모델을 선정하였다. 다음으로 이 모델을 준지도학습 환경에서 학습하였다. Mean Teacher 준지도학습 기법 기반 모델을 자료 증강 방법, 모델 학습 방법, Consistency 손실 가중치, 라벨링된 자료의 개수 등을 조정하여 다양한 민감도 실험을 수행하였다. 그 결과로 전체 자료 중 절반의 양을 가지고도 대부분의 라벨링 자료를 사용하거나 모든 자료를 라벨링하여 학습한 지도학습 결과에 비해 비슷하거나 더 나은 결과를 보여주었다. 추가로 다양한 이전 시계열 길이를 가지는 입력장 자료를 학습하여 다음날 혹은 연속 7일의 입력장(U850, U200, OLR)을 예측하는 모델링을 수행하였다. Inactive MJO 발생 사례 제거, 여름철 제거 등의 샘플링을 통해서 모델 테스트를 수행하였다.

기후분야에서 딥러닝 기법을 적용한 연구가 많이 늘어나고 있지만 딥러닝 모델의 이해를 통한 모델의 개선 방향 도출은 미흡하다. 본 연구에서는 딥러닝 모델의 이해도를 높이고 모델의 개선점을 도출하기 위해 설명가능한 인공지능(eXplainable Artificial Intelligence, XAI), 손실 지형 분석, 모델 내부 구조 분석 및 개선을 수행하였다. XAI 기법을 통해 지도학습과 준지도학습으로 구축된 모델이 입력자료에서 유사한 정보를 추출하여 예측에 활용했는지 분석하였다. 손실지형 분석을 이용하여 딥러닝 모델의 구조가 전역최저점을 찾는데 유리한 지형인지 판단하는 연구를 수행하였다. 또한, 딥러닝 모델의 내부 구조와 피쳐 붕괴를 분석하여 어텐션 메카니즘, 데이터셋 확장, 레이어 필터 수 증가, 모델 구조 개선 등을 고려하여 모델을 설계하고 모델의 예측성을 평가하였다.

목차

1. 서론	1
2. 방법 및 자료	8
2.1. 자료 확장에 따른 모델 성능의 민감도 분석	8
2.1.1. 자료	8
2.1.2. 방법	12
2.2. 훈련기법, 초매개변수 조정 및 불확실성 분석을 통한 아키텍처 개선	18
2.2.1. 확장 U-Net 모델 계절내 기후예측 훈련자료	18
2.2.2. 그리드 탐색 알고리즘	28
2.2.3. 계절내 기후 예측 훈련자료	30
2.2.4. 연구 지역 및 평가 방법	31
2.3. 준지도학습 기법 적용 및 모델의 민감도 분석	33
2.3.1. 사용된 자료 및 전처리 방법	33
2.3.2. 준지도학습 기준 모델 구축 방법	35
2.3.3. 준지도학습 기반 MJO 위상 분류 모델 구축 및 민감도 테스트 방법	37
2.3.4. MJO 입력장 예측을 위한 시공간 이미지 예측 인공지능 모델 개발 방법	39
2.4. 설명 가능한 인공지능(XAI), 모델 구조 분석 및 모델 개선	41
2.4.1. XAI 기법 적용을 통한 모델 분석	41
2.4.2. 딥러닝 모델 내부 구조 분석 및 개선	42
3. 결과	48
3.1. 자료 확장에 따른 모델 성능의 민감도 분석	48
3.1.1. 강수자료 학습결과	48
3.1.2. Filter 기법에 따른 민감도 분석 결과	48
3.1.3. Wrapper 기법에 따른 민감도 분석 결과	49
3.1.4. Embedded 기법에 따른 민감도 분석 결과	53
3.2. 훈련기법, 초매개변수 조정 및 불확실성 분석을 통한 아키텍처 개선	54
3.2.1. 확장 U-Net 모델 구축	55
3.2.2. 그리드 탐색: 초매개변수 조합 학습 성능	55
3.2.3. 선정된 최적 초매개변수 조합의 예측 성능 평가	59
3.3. 준지도학습 기법 적용 및 모델의 민감도 분석	63
3.3.1. 다양한 변수 조합, 자료 분리 방법, 모델 구조에 따른 모델 민감도 테스트 결과	63

3.3.2. 준지도학습 기반 MJO 위상 분류 모델 민감도 테스트 결과	64
3.3.3. MJO 입력장 예측을 위한 시공간 이미지 예측 인공지능 모델 개발 결과	68
3.4. 설명 가능한 인공지능(XAI), 모델 구조 분석 및 모델 개선	70
3.4.1. XAI 기법 적용을 통한 모델 분석	70
3.4.2. 딥러닝 모델 내부 구조 분석 및 개선 결과	75
4. 요약 및 결론	84
사사	88
References	89
부록 A	96

표 차례

Table	page
Table 2.1. Data description of ERA5.	11
Table 2.2. Overview of the characteristics of ERA5.	12
Table 2.3. The U-Net models applied in the study.	19
Table 2.4. Types of hyperparameters in neural network models.	29
Table 2.5. Information of hyper-parameters and their ranges set in the study.	30
Table 2.6. S2S (Sub-seasonal to Seasonal) climate models used in this study, number of ensembles and period of hindcast of each models.	30
Table 2.7. List of experiments for model sensitivity analysis on input variables, data split methods, model architectures.	35
Table 2.8. List of experiments for sensitivity analysis of SSL-based AI model on model learning strategy, loss weights, and data augmentation methods.	39
Table 3.1. Index and variables of subsets for filter method.	49
Table 3.2. Index and variables of subsets for wrapper method.	52
Table 3.3. Index and variables of subsets for embedded method.	54
Table 3.4. Accuracy of SSL-based model in the test data for MJO phase classification for different data augmentation methods.	65
Table 3.5. Summary of SSL-based model skill of the experiments (cf. Table 2.2) for MJO phase classification with the test data for MJO index and different labeled samples.	65

그림 차례

Figure	page
Figure 1.1. (a) Supervised learning, (b) semi-supervised learning, (c) unsupervised learning. Colored circles are labeled data and gray ones are unlabeled data.	3
Figure 1.2. Diagram of Pi model (Source: Laine and Aila, 2016).	4
Figure 1.3. Comparison of Pseudo Labels (left) and Meta Pseudo Labels (right) with the feedback from the Student model (Source: Pham et al., 2021).	5
Figure 1.4. Overview of this study. (a) Development and sensitivity analysis of semi-supervised learning-based deep learning model for MJO phase classification, (b) Prediction of MJO-related climate input fields using spatiotemporal predictive learning models.	6
Figure 2.1. NOAA NDVI(Normalized Difference Vegetation Index data form AVHRR satellite.	8
Figure 2.2. GPM IMERG Final data.	9
Figure 2.3. Sea surface temperature data from NOAA.	10
Figure 2.4. ERA5 Maximum temperature at 2 meters.	11
Figure 2.5. A block diagram of Attention U-Net architecture (Source: Oktay et al., 2018).	13
Figure 2.6. Schematic of the attention gate (Source: Oktay et al., 2018).	13
Figure 2.7. A diagram of input data from observation and ECMWF model data for Attention U-Net model.	14
Figure 2.8. A diagram of trained datasets for sensitivity test.	14
Figure 2.9. A diagram for filter feature selection scheme (Source: Hall, M., 1999).	15
Figure 2.10. Correlation heatmap of observation and ECMWF model variables.	16
Figure 2.11. Scatterplot of correlation of variables.	16
Figure 2.12. A diagram for wrapper feature selection scheme (Source: Hall, M., 1999).	17
Figure 2.13. A diagram for embedded method for feature selection (Source: Trabelsi et al., 2021).	18
Figure 2.14. The U-Net which is used in the 2022 study. (a) is the original U-Net (Source: Ronneberger et al., 2015), and (b) is the U-Net schematized briefly in the study.	19

Figure 2.15. Attention U-Net: (a) is the original Attention U-Net (Source: Oktay et al., 2018), and (b) is the Attention U-Net schematized briefly in the study.	20
Figure 2.16. Attention Gate: (a) is the Attention Gate presented in the original paper (Oktay et al., 2018), and (b) is the Attention Gate expressed through coding in the study.	22
Figure 2.17. Residual block: (a) represents a general Residual U-Net, and (b) and (c) show residual blocks in encoding and decoding, respectively.	25
Figure 2.18. Residual U-Net: (a) is the Residual U-Net briefly expressed in the study, and (b) shows the residual block mounted on the Residual U-Net of the study.	26
Figure 2.19. Attention gate based-on Residual U-Net of the study.	28
Figure 2.20. Grid Search: expressing the method of finding optimal hyperparameters as a function.	29
Figure 2.21. Steps of data preprocessing for filtered anomaly data.	34
Figure 2.22. Composite images of the filtered OLR anomaly (W/m ²) for each MJO phase for training (left), validation (center), and test (right) datasets, respectively. The number of days available for each phase are given in the bottom right corner. Negative (Positive) anomalies indicate more (less) rain.	34
Figure 2.23. Data split strategy. (a) Temporal split, (b) Random split method.	35
Figure 2.24. Diagram of modified VGG16 model used in this study.	36
Figure 2.25. Residual learning structure of ResNets model (Source: He et al., 2016).	36
Figure 2.26. Diagram of modified Resnet18 model and different versions of fully connected layers used in this study.	37
Figure 2.27. Diagram of Mean Teacher algorithm (Source: Tarvainen and Valpola, 2018).	38
Figure 2.28. Data augmentation methods used in Mean Teacher algorithm.	38
Figure 2.29. Diagrams of SimVP model architecture and its gSTA module (Source: Tan et al., 2022).	40
Figure 2.30. Diagrams of PredRNN version 2 (Source: Wang et al., 2022).	40
Figure 2.31. Perturbation-based occlusion in XAI.	42
Figure 2.32. Training dataset.	43
Figure 2.33. Architecture of deep learning-based 1-month T2M prediction model.	43
Figure 2.34. Visualization of activation map.	45
Figure 2.35. Example of scaling the number of layer filters.	45

Figure 2.36. The overview of SE block (adapted from Hu et al. (2020)).	46
Figure 2.37. The overview of CBAM (adapted from Woo et al. (2018)).	46
Figure 2.38. Visualization of activation map (input images obtained from https://github.com/ismailuddin/gradcam-tensorflow-2).	46
Figure 3.1. Comparison of the result of Attention U-Net and ECMWF model.	48
Figure 3.2. Trained results obtained from variables based on filter method.	49
Figure 3.3. Visualization of the result of Attention U-Net by wrapper method.	50
Figure 3.4. Comparison of the result of Attention U-Net by Wrapper method.	51
Figure 3.5. Comparison of the result of all variables by Wrapper method.	53
Figure 3.6. Correlation coefficient of the variables from ECMWF model by embedded method.	53
Figure 3.7. Comparison of the result of all variables by Embedded method.	54
Figure 3.8. Comparison of loss function changes to learning rate according to hyperparameter combinations in the study.	56
Figure 3.9. Comparison of loss function changes to batch size according to hyperparameter combinations in the study.	57
Figure 3.10. Comparison of loss function changes for the four extended U-Net models applied in this study in hyperparameter combinations.	58
Figure 3.11. Comparison of changes in learning rate in the extended U-Net applied according to the batch size by extracting a low loss function: the x-axis shows the four extended U-Nets, the y-axis is the loss function. (a) is the S2S maximum temperature of ECMWF and (b) is the S2S precipitation of ECMWF.	59
Figure 3.12. Comparison of ACC of ECMWF S2S daily maximum temperature tested in the four extended U-Net models selected hyperparameter combinations in the study: (a) ACC for the entire 46-day forecast frequency of ECMWF, and (b) is zoomed that ACC from 7th to 21th day out of the total 46-day period, and (c) is zoomed which the ACC from the 21st to the 35th day.	60
Figure 3.13. Comparison of ACC of ECMWF S2S daily total precipitation tested in the four extended U-Net models with selected hyperparameter combinations in the study: (a) ACC for the entire 46-day forecast frequency of ECMWF, and (b) is zoomed that ACC from 7th to 21th day out of the total 46-day period, and (c) is zoomed which the ACC from the 21st to the 35th day.	61

Figure 3.14. Comparison of weekly PCCs for ECMWF S2S daily maximum temperature (a) and daily total precipitation (b) tested on the four extended U-Net models with selected hyperparameter combinations in the study.62

Figure 3.15. Maximum temperature maps selected for 7 days within each week from the 46-day prediction frequency of ECMWF S2S maximum temperature predicted during the test period in the four expanded U-Nets with selected hyperparameter combinations. (a) to (d) are from week 1 to week 4.62

Figure 3.16. Precipitation maps selected for 7 days within each week from the 46-day prediction frequency of ECMWF S2S total precipitation predicted during the test period in the four expanded U-Nets with selected hyperparameter combinations. (a) to (d) are from week 1 to week 4.63

Figure 3.17. Summary of model skill for MJO phase classification with the test data. Colored bars of each panel show the test accuracy (%) of the models. Left panels are the results with RMM input variables (U850, U200, OLR), center ones with VPM variables (U850, U200, VP), and the right ones with OLR only. The upper row shows the results with temporal split data and the lower ones with random split data.64

Figure 3.18. Bar charts of the SSL-based model skill of the experiments in the test data with comparison to supervised learning results (baseline) shown as horizontal dashed gray lines. Colored bars indicate the experiments (cf. Table 2.2).66

Figure 3.19. Monthly and seasonal SSL-based model performances with the test data for MJO phase classification grouped by active (orange) and inactive (blue) MJO cases. Error cases on the y-axis indicate MJO events misclassified by the model.67

Figure 3.20. Monthly and seasonal SSL-based model performance heat maps with the test data for MJO phase classification. The number of errors in each grid shows the MJO events misclassified by the model.68

Figure 3.21. Predictions of OLR from PredRNN v2 model for pre-sequence length of 7 or 30 days and after-sequence length of 1 or consecutive 7 days with reference (Trues) from NOAA interpolated OLR data (left column).69

Figure 3.22. Predictions of OLR from SimVP gSTA model for pre-sequence length of 7 or 30 days and after-sequence length of 1 or consecutive 7 days with reference (Trues) from NOAA interpolated OLR data (left column).69

Figure 3.23. Predictions of OLR for different sampling strategies from SimVP gSTA model with pre-sequence length of 120 days and after-sequence length of 1 day with reference from NOAA interpolated OLR data (left columns).	70
Figure 3.24. Composite data of the filtered OLR anomalies (training data).	71
Figure 3.25. Occlusion analysis using results predicted from test data with the resnet18 model (window 10x10 and strides 7x7).	71
Figure 3.26. Occlusion analysis using results predicted by test data with the resnet18 model and the deep learning model after applying semi-supervised learning: (a) XAI result of the resnet 18 model with window 10x10 and strides 7x7, (b) window 10x10 and strides 7x7, (c) window 5x5 and strides 3x3.	72
Figure 3.27. Visualization of loss surface of Densenet121 model with 0.1 learning rate, 64 batch size and SGD (Stochastic Gradient Descent) optimizer.	73
Figure 3.28. Visualization of loss surface of Resnet56_noshort with 0.1 learning rate, 64 batch size and SGD optimizer.	73
Figure 3.29. Visualization of loss surface of Resnet56 model with 0.1 learning rate, 128 batch size and SGD optimizer.	74
Figure 3.30. Visualization of loss surface of 1-month T2M prediction model with 0.001 learning rate, 16 batch size and ADAM (adaptive moment estimation) optimizer.	74
Figure 3.31. Plot of model training results by batch size.	75
Figure 3.32. Plot of the comparison of the ACC by season.	76
Figure 3.33. Plot of the comparison of the ACC by year.	76
Figure 3.34. Summary of the HSS assessment of lead times F1(W+3) and F2(W+4) for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2	77
Figure 3.35. Summary of the HSS assessment of lead times F1(W+3) by year for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2.	78
Figure 3.36. Summary of the HSS assessment of lead times F2(W+4) by year for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2.	78
Figure 3.37. Line chart of training loss of deep learning model.	79
Figure 3.38. Feature pair-wise cosine distance analysis of deep learning model.	79

Figure 3.39. The results of the feature pair-wise cosine distance analysis of the deep learning models with the SE block.80

Figure 3.40. The results of the feature pair-wise cosine distance analysis of the deep learning models with the CBAM block.80

Figure 3.41. Design of CCA block.81

Figure 3.42. The results of the feature pair-wise cosine distance analysis of the deep learning models with the SE block after changing the bottleneck block.81

Figure 3.43. The results of the feature pair-wise cosine distance analysis of the deep learning models with the CBAM block after changing the bottleneck block.81

Figure 3.44. Summary of the HSS assessment of lead times F1(W+1) and F2(W+4) for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.82

Figure 3.45. Summary of the HSS assessment of lead times F1(W+1) by year for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.83

Figure 3.46. Summary of the HSS assessment of lead times F2(W+2) by year for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.83

Figure A1. Comparison of probability forecasts from KMA ASOS, ECMWF S2S model, and deep learning models on February 8, 2021.96

1. 서론

최근 NVIDIA, 구글 딥마인드, 화웨이 등 글로벌 기업을 중심으로 14일 이내의 인공지능 기후예측모델이 개발되고 있다. NVIDIA는 SFNO (Spherical Fourier Neural Operators)를 적용한 트랜스포머 기반 FourCastNet, 구글 딥마인드는 GNN (Graph Neural Network) 기반 GraphCast, 화웨이는 개선된 Swin-Transformer 기반 Pangu-Weather, 그리고 유럽중기예보센터(European Centre for Medium-Range Weather Forecasts, 이하 ECMWF)는 GNN 기반의 AIFS (Artificial Intelligence/Integrated Forecasting System)를 개발했으며 ECMWF 웹사이트를 통해 시범 운영 중에 있다(Bi et al., 2023; Bonev et al., 2023; ECMWF, 2023; Lam et al., 2023). 그러나 여전히 예측기간이 14일 이내로 계절내 및 계절 시간규모로의 확장은 아직 미흡하다. 또한, 인공지능 연구에는 대량의 학습자료가 필요하지만 현재 글로벌 기업들이 학습에 사용한 기후자료인 ECMWF ERA5 (ECMWF Reanalysis v5) 재분석자료는 1950년부터 현재까지 약 74년의 자료만 구축되어 있다. 이 자료를 학습자료, 검증자료, 테스트자료로 나눠서 사용하면 실제 학습에 사용되는 자료는 매우 부족하다.

우리는 이러한 한계를 극복하기 위해 기후예측 연구에 필요한 다양한 학습자료 구축 및 자료 조합별 민감도 분석, 계절내 ECMWF S2S 예측모델의 예측성 개선, 기후예측모델의 준지도학습기법, 딥러닝 모델 내부 구조 및 설명가능한 인공지능 분석을 통한 모델 개선 등 연구를 수행하고자 한다.

최근 고온 등 이상기후에 따른 기후위기가 고조되고 있기 때문에 날씨와 같은 단기예보와 기후변화 시나리오의 장기예보를 잇는 3~4주의 계절 내 예측자료가 아시아 태평양 지역에서도 식량안보, 에너지 분야 등 다양한 기후응용분야 활용에 갈수록 중요성이 증대되고 있다. 이에 따라 계절 내 예측의 정확도 향상을 위한 연구가 활발히 진행되고 있다. 하지만 기후모델에 기반한 예측자료는 기상 패턴의 단기적이고 장기적인 모든 변동성을 고려해야 할 뿐만 아니라 대규모 대기현상과 대기-해양간의 상호작용 등의 다양한 영향을 모델에 반영해야 하기 때문에 계절내 예측에 어려움이 많은 것이 사실이다.

ChatGPT의 등장으로 생성형 딥러닝 모델의 등장으로 인공지능 기법에 기반하여 계절 내 예측자료의 오차를 최대한 줄이기 위한 연구가 지속적으로 진행되고 있다. 최근 ClimaX (Nguyen et al, 2003)와 같은 날씨, 기후, 자료상세화 및 실시간에서 계절 예측까지 다양한 분야에서 예측가능한 모델이 개발되었고, NVIDIA의 FourCastNet V2 (Bonev et al., 2023)와 화웨이(Huawei)의 Pangu-Weather (Bi et al., 2023)와 같은 글로벌 기업의 기후모델이 최근 정확도를 높이며 연구가 발전하고 있다. 하지만 예측 리드타임이 1~2주에서만 예측이 가능한 단점이 발생한다. 따라서 본 연구에서는 여전히 부족한 자료는 관측자료를 추가하고 입력자료에서 중요한 부분에 가중치를 더 부여하여 좀 더 중요한 부분에서 깊게 학습하도록 해주는 Attention U-Net 모델을 기반으로 다양한 민감도 기법을 활용하여 정확도 개선에 필요한 변수를 선택하여 신뢰성 있는 예측자료를 생산하고자 한다.

활발하고 다양한 신경망 모델 개발과 산업분야 적용 사례 및 연구가 진행되면서 어떤 신경망 구조 및 관련 환경이 적합한가에 대한 연구도 활발하다. 신경망 모델에서 각각의 weight

와 bias를 학습시킬 수 없다라는 문제가 있었으며, 이후 역전파법(backpropagation) 방법, 즉 앞의 진행 방향에서 고쳐가는 것이 아니라 결과를 보고 뒤로 가면서 weight와 bias를 조정하는 역전파법을 고안하게 되면서 weight와 bias를 알맞게 학습하는 방법이 제안되었지만(Rumelhart et al., 1986), 신경망 깊이가 깊어질수록 원하는 결과를 얻을 수 없다라는 문제가 다시 제기되었을 뿐만 아니라 신경망 학습을 위한 파라미터 값의 최적화에 대한 이론적인 근거가 없다라는 염려 등이 제기되었다. 그러나, 신경망을 학습시키기 전에 층 단위의 학습을 거쳐 더 나은 초기값을 얻는 사전 훈련(pre-training) 방법 등이 제안되면서(Hinton et al., 2006), 신경망 학습은 심층학습, 딥러닝(deep learning)으로 발전하게 되었다. 최근에는 딥러닝 학습이 사용되지 않는 분야를 찾기 힘들 정도로 다방면으로 사용되고 있다. 주가 예측은 물론 의료 분야에서도 큰 영향을 끼치고 있고, 이미지 및 음성 인식, 자연어 처리, 자율주행에도 많은 방법이 딥러닝을 활용하는 방식으로 대체되고 있다. 또한 깊은 신경망을 통한 많은 계산량을 빠르게 수행할 수 있는 GPU 컴퓨팅 기술과 빅데이터 등 높은 기술력 및 하드웨어 발전과 맞물려 크게 성장하고 있다. 마찬가지로, 기후 분야에서 딥러닝 모델이나 신경망 모델 활용 및 개발 연구는 매우 다양하게 진행되고 있다. 예를 들어, 원격탐사 분야에서는 딥러닝을 이용한 수질 예측 모델 구성이 연구되고 있으며, 미세먼지 예측, 에어로졸 예측 등에도 활용되고 있다(Chae et al., 2021, 2020). 또한, 기후 변화 예측 분야에서는 딥러닝 알고리즘이 티핑 포인트가 되는 기후 변화인 지진, 가뭄 등이 일어나기 전 조기 경고 신호를 정확히 알려줄 수 있다는 최근 연구 결과도 있다. 또한, 딥러닝 학습을 사용한 기상 예측의 성능은 연구에 따라 다르지만, 최근 딥러닝 기술을 활용한 기상 예측 분야에서 많은 연구가 이루어지고 있다. 예를 들어, 구글의 딥 마인드가 영국 기상청과 협력하여 초단기 날씨 예측 정확성이 더 뛰어난 인공지능 모델 등을 개발하는 등 기상 분야에서 활발히 진행되고 있다(Reed et al., 2022). 계절(seasonal) 예측을 위한 딥러닝 모델 연구도 다양하게 진행되고 있는데, 한국 남부 해역의 SST 예측 연구에 딥러닝 기반의 모델이 개발 및 사용되고 있으며 이를 통해 단기적인 해수면 온도 예측이 가능해졌다(Ju et al., 2022). 그러나, S2S (Sub-seasonal to Seasonal) 기후 예측 개선을 위한 머신러닝(machine learning) 및 딥러닝 활용 연구는 상대적으로 많지 않다. 그 이유는 S2S 기후 예측 입력 자료의 전처리 기법, S2S 시간 규모 예측에 적용할 수 있는 딥러닝 모델이나 기술에 관한 연구 부족 등에서 찾을 수 있다. 또한, 단기나 계절 기후를 위한 예측 모델 결과 자료와 달리, S2S 기후 예측 모델 결과 자료는 예측 자료 생산 시작일의 차이로 인해 달라지는 예측 빈도(forecast frequency), 예측 선행시간 및 기간(forecast range) 등에 대한 예측 정보 제공 기관 간의 합의된 표준이 없어 기관별로 자료의 포맷이 매우 상이하다. 이는 일반적인 사용자가 S2S 예측 자료를 다루거나 후보정을 하는 것에 하나의 제약 요인이 되고 있다. 예를 들어, S2S 기후 예측의 대표적 두 모델인 ECMWF와 GloSea5 (Global Seasonal forecast system version 5)의 기후 변수 출력 산출물은 각 16개와 13개이며, 기후 변수별로 시간 해상도가 다르다. 또한, 예측 자료 생산 시작일을 기준으로 최고 및 최저기온은 6시간 단위로 제공되고 그 외 기후 변수는 일별 단위로 제공되는데 이것 또한 어떤 변수는 전일부터 제공되고 다른 변수는 익일부터 (예, 시작일 다음날) 제공되는 등, 변수 및 저장 시간 규모, 배열 형태 등이 상이하고 복잡하여 전체 자료를 바로 처리해서 활용하기 어렵다. 이처럼 S2S 데이터를 다루기 힘들고 처리하는 문제를 해결하고 그에 따른 학습 및 예측 성능을 테스트하였다(Chung et al., 2023). 그럼에도 불구하고, S2S 1-2주 예측 성능의 개선은 뚜렷하지 않은 문제점이 여전히 남아 있었다. 따라서, 본 아키텍처 개선 연구에서는 선례 연구 U-Net 기반의 학습 및 예측 결과가 미흡한 이유가 아키텍처에 있을 수 있다는 가설에서 시작하여, U-Net 기반으로 가중치 계산 및 전달 방법을 다양하게 할 수 있는 장치를 추가함으로써 다양한

아키텍처를 구축하고, 그 아키텍처의 초매개변수를 최적화하여 S2S 예측 기후변수 (예, 최고기온 및 강수)의 후보정 개선에 적합한 딥러닝 모델을 탐색하고자 수행하였다.

준지도학습(Semi-Supervised Learning, SSL)이란 지도학습과 비지도학습을 함께 사용하는 방법이다(Figure 1.1). 지도학습은 라벨링된 자료로부터 패턴이나 규칙을 학습하는 방식이고, 비지도학습은 라벨링 되지 않은 자료로 학습한다. 라벨링된 자료란 입력 자료(X)가 주어지면 입력 자료에 대한 라벨(Label; Y)이 있는 자료를 말한다. 딥러닝 모델들은 라벨링된 자료를 사용하는 다양한 지도학습 문제에 성공적으로 활용되어져 왔다. 하지만 많은 경우 모든 자료가 라벨링이 되어 있지 않으며 라벨링하는 작업은 많은 시간과 비용이 든다. 따라서 이러한 문제점을 해결하는 하나의 방안으로 준지도학습 기법이 소개되었다. 준지도학습은 적은 양의 라벨링된 자료와 많은 양의 라벨이 없는(Unlabeled) 자료들이 있는 상황에서 활용되는 기법으로, 라벨링된 자료의 양을 줄이면서도 제한된 자료를 효율적으로 활용하여 비슷하거나 향상된 성능을 구현할 수 있다.

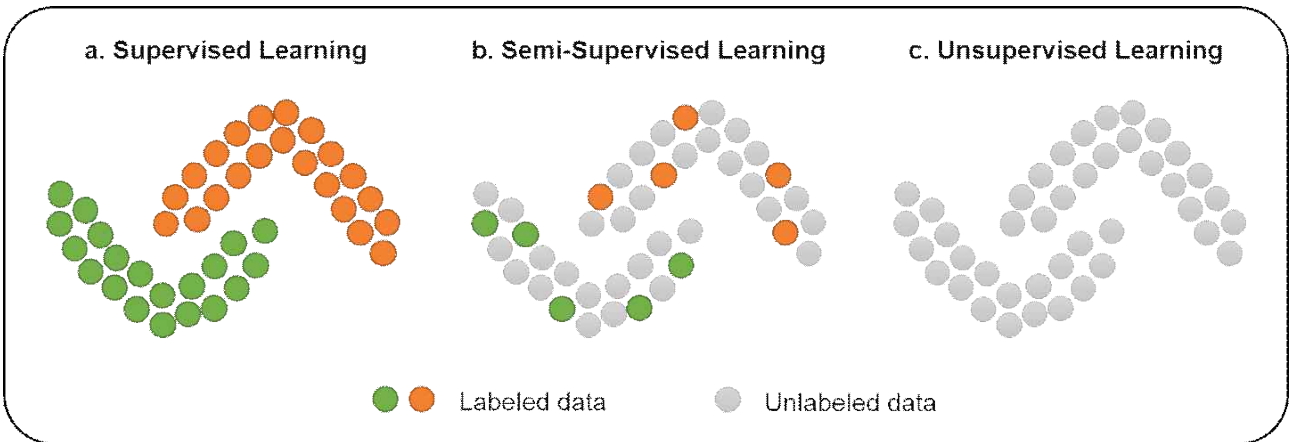


Figure 1.1. (a) Supervised learning, (b) semi-supervised learning, (c) unsupervised learning. Colored circles are labeled data and gray ones are unlabeled data.

이러한 준지도학습 기법은 기후 연구 분야에서 유용하게 활용될 수 있다. 기후 자료는 수십 년의 자료가 축적되어 왔지만 여러 복잡한 기후의 특성을 고려하면 딥러닝 모델에 활용하기에는 부족한 상황이 많고 월별로 자료를 활용하는 경우에는 더욱 제한적이다. Subseasonal to Seasonal (S2S) 규모의 예보에서 중요한 기후 예측 인자인 Madden-Julian Oscillation (MJO)는 대략 30~60일의 주기를 가지는 계절 내 진동으로 장기간의 기후 자료 내에서도 발생한 사례가 그중 일부에 해당하기 때문에 (RMM 인덱스 기준 1974년~현재까지 진폭 1 이상의 MJO 발생 사례는 약 62.5%) 인공지능 모델을 위한 충분한 자료 구축에 어려움이 있다. 따라서 준지도학습 기법을 활용하여 적은 기후 자료를 효과적으로 활용할 수 있는 인공지능 모델을 개발하는 것이 필요하며 향후 제한된 기후 자료를 활용하는 다양한 연구에 유용할 것이다.

지난 수년 동안 많은 준지도학습 기법들이 개발됐다. 준지도학습 알고리즘은 크게 5가지의 카테고리로 나눌 수 있다 (Ouali et al., 2020): 1) Consistency Regularization, 2) Proxy-label Methods, 3) Generative Models, 4) Graph-based Methods, 5) Entropy minimization. 첫째로 Consistency Regularization 방법은 라벨이 바뀌지 않을 정도로 노이즈(noise)나 자료 증강(data

augmentation) 기법을 통해 자료에 변화를 주더라도 일관된(consistent) 예측을 하도록 학습하는 방법이다. 자료 증강은 자료를 조금씩 변형해서 새로운 자료를 만드는 방법으로 크기 및 배율 조정, 이미지 회전, 좌우/상하 반전(Horizontal Flip, Vertical Flip), 흐림(Blur), 선명도 조절(Sharpness) 등의 다양한 방법들이 있다. Consistency Regularization을 사용하는 알려진 모델로 Pi-model (Laine and Aila, 2016), Mean teachers (Tarvainen and Valpolar, 2018), Virtual Adversarial Training (Miyato et al., 2018) 등의 방법들이 있다. Figure 1.2는 Pi 모델의 모식도를 보여준다. 입력 자료(x)는 자료 증강 및 모델의 dropout(신경망의 뉴런을 부분적으로 제거하여 과적합을 방지하는 방법)을 거쳐 두 가지 버전의 출력값을 만든다. 그중 하나는 원 라벨 결과(y)와 비교하여 정확도 손실을 계산하고 다른 하나는 두 출력값의 차이를 통한 손실을 계산하는 방식으로 학습된다. 컴퓨터 비전(vision) 분야에서는 다양한 자료 증강 기법들이 적용됐지만 도메인의 특성에 따라 적절한 자료 증강을 적용할 필요가 있으며 특히 공간적인 특성이 중요한 기후 분야에서는 주의해서 사용해야 할 것이다.

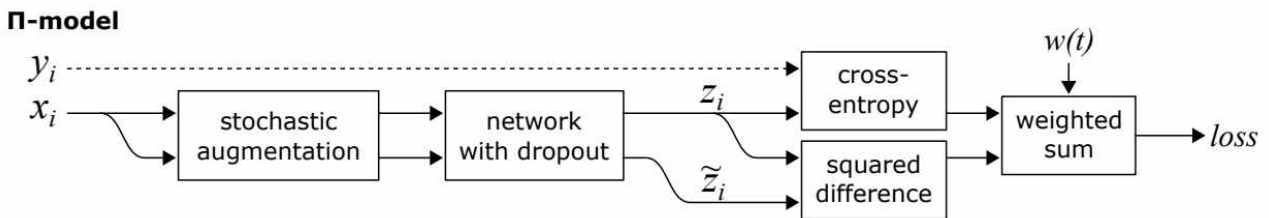


Figure 1.2. Diagram of Pi model (Source: Laine and Aila, 2016).

둘째로 Proxy-label 방법은 먼저 라벨링된 자료로 모델을 학습한 후 비라벨링 자료에 대해 pseudo 라벨을 생성하고 이 중 신뢰도가 높은 결과를 라벨링 자료셋(dataset)으로 포함하여 재훈련 해나가는 방식으로 학습된다. Proxy-label 방법은 크게 self-training과 multi-view learning 방법으로 나뉜다. 전자는 하나의 모델로 라벨링을 해나가는 방식이며 후자는 자료의 여러 특성에 대해 훈련 모델을 여러 가지로 만들어 라벨링을 하는 방식이다. 잘 알려진 방법으로 Self-training의 Pseudo-labeling (Lee, 2013)과 Meta Pseudo Labels (Pham et al., 2021) 방법이 있다. Pseudo-labeling은 비교적 간단한 알고리즘이기 때문에 활용하기 쉽지만, 비라벨링 자료에 대해 생성한 pseudo 라벨이 부정확할 경우 쉽게 성능이 저하되는 한계(confirmation bias; 확증 편향 문제)를 가진다. Meta Pseudo Labels 방법은 pseudo 라벨을 포함한 자료로 학습된 모델을 기존 라벨 자료에 대해 정확도를 계산하고 이를 피드백 받아 모델을 업데이트하는 방식으로 Pseudo-labeling의 문제점을 개선하여 더 높은 성능을 내는 알고리즘이다(Figure 1.3).

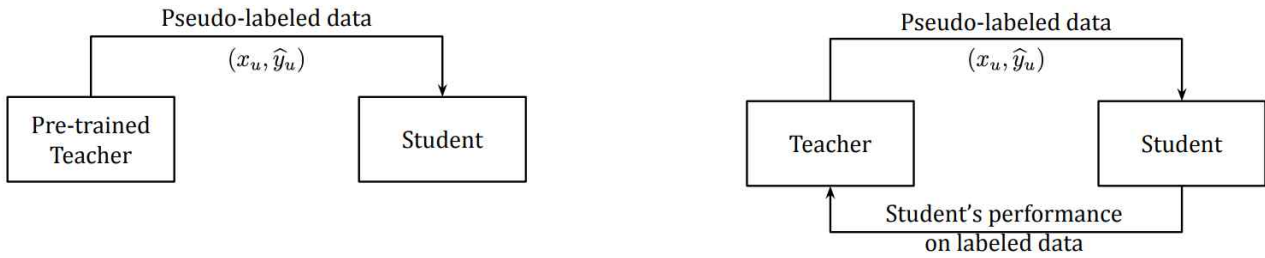


Figure 1.3. Comparison of Pseudo Labels (left) and Meta Pseudo Labels (right) with the feedback from the Student model (Source: Pham et al., 2021).

생성형 모델(Generative Models)은 말 그대로 자료를 새롭게 생성할 수 있는 모델이다. 이는 주어진 자료에 대해서 해당 자료를 생성하는 메커니즘을 역으로 추정하는 방식으로, 분포 자체를 학습한 모델은 새로운 자료를 생성할 수 있게 된다. 준지도학습 환경에서의 생성형 모델은 라벨링 자료뿐만 아니라 비라벨링 자료로부터의 분포를 학습함으로써 라벨링 자료만 가지고 학습한 것보다 더 좋은 분포 학습을 할 수 있다. Graph-based 방법은 노드들과 노드들 사이를 연결하는 에지(edge)로 구성된 그래프(graph) 구조를 기반으로 라벨링과 비라벨링 자료의 각 그리드를 노드(node)로 생각하고 노드 간의 유사도를 활용하여 비라벨링 노드의 라벨을 추정해 나가는 방법이다. 마지막으로 Entropy minimization 방법은 예측값의 엔트로피를 최소화하는 방향으로 모델을 학습한다. 이 외에도 MixMatch, FixMatch 등의 Holistic 준지도학습 기법들은 앞선 다양한 방법들을 통합하여 성능을 향상한 알고리즘들이 있다.

Figure 1.4는 본 연구에서 수행한 연구 모식도를 보여준다. 본 연구에서는 준지도학습 기반 MJO 위상(phase)을 분류하는 이미지 기반 인공지능 모델을 개발하였다 (Figure 1.4a). 전통적으로 MJO 위상을 구하는 기존의 방법은 기후 자료의 아노말리(anomaly)에 Empirical Orthogonal Function (EOF) 분석을 적용하여 주성분(Principal Component, PC) 1과 주성분 2를 사용하여 얻는다. 사용되는 기후 변수의 종류와 자료에 적용되는 필터링 등의 전처리 방법에 따라서 Real-time Multivariate MJO 인덱스 (RMM; Wheeler and Hendon, 2004), Velocity Potential MJO 인덱스(VPM; Ventrice et al., 2013), Outgoing Longwave Radiation MJO Index 인덱스(OMI; Kiladis et al., 2014) 등의 다양한 MJO 인덱스들이 개발됐다. 하지만 MJO 인덱스에 따라서 MJO의 다양한 기후 패턴이 제대로 반영되지 않을 수 있다(Kiladis et al., 2014). Kiladis et al.(2014)는 MJO 인덱스 종류에 따라 MJO 위상과 강도를 서로 다르게 진단하는 것을 발견했다. 따라서, 본 연구에서는 MJO 인덱스 종류에 따른 모델 성능을 평가하기 위해 준지도학습 기반 딥러닝 모델을 인덱스 계산에 사용되는 여러 입력장 자료들의 조합에 따라 훈련하였으며, 준지도학습 적용 전후를 비교하여 준지도학습 기법의 효과성에 대해서 평가하였다. 추가로 본 연구에서는 MJO 입력장을 예측하는 인공지능 모델을 개발하였다 (Figure 1.4b). 시계열의 기후 입력장 이미지를 활용하여 기후 입력장을 예측함으로써 MJO 위상 분류에 나아가 MJO 입력장 예측을 위해 수행되었다.

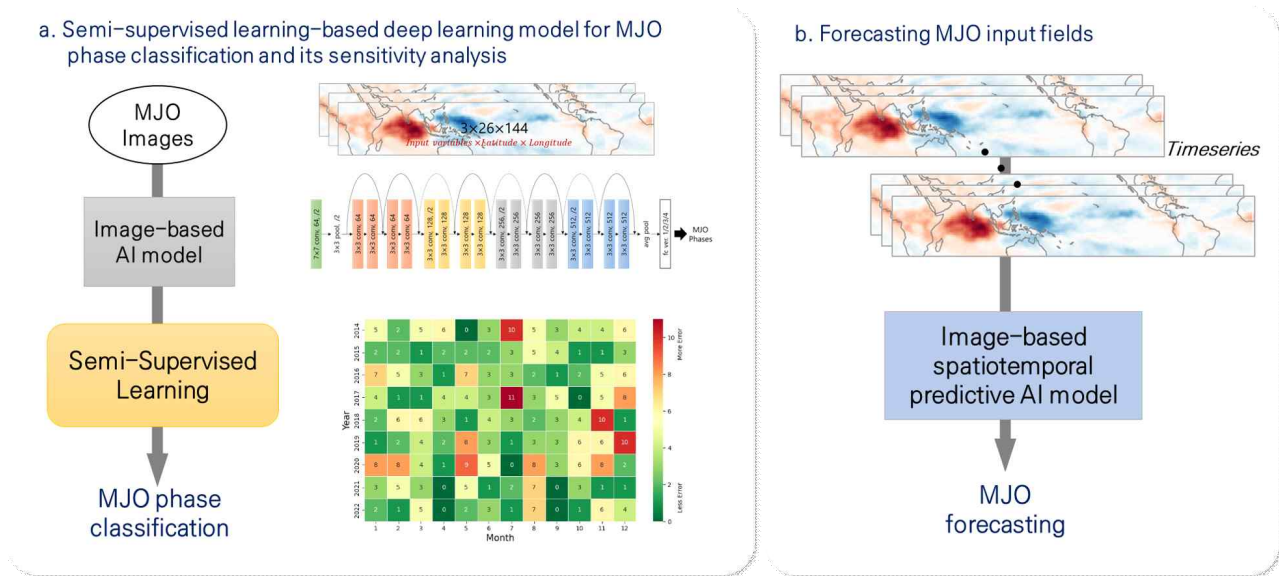


Figure 1.4. Overview of this study. (a) Development and sensitivity analysis of semi-supervised learning-based deep learning model for MJO phase classification, (b) Prediction of MJO-related climate input fields using spatiotemporal predictive learning models.

최근 인공지능 분야에서는 2018년부터 시행 중인 유럽연합(European Union, EU)의 개인정보보호(General Data Protection Regulation, GDPR) 규정에 따라 인공지능 알고리즘에 의해 행해진 결정에 대해 질문하고 결정에 관여한 논리에 대해 의미 있는 설명을 요구할 권리를 포함하는 설명을 요구할 권리(Right to explanation)가 주목받고 있다(NIA, 2018). 그리고 2018년 12월에 첫 번째 초안이 공개된 유럽연합의 AI 윤리 가이드라인(Ethics Guidelines for Trustworthy AI)에는 AI 시스템이 신뢰할 수 있는 7가지 주요 요건으로 인간의 권한과 감독(Human agency and oversight), 기술적 견고성 및 안전성(Technical Robustness and safety), 개인정보 보호 및 데이터 거버넌스(Privacy and data governance), 투명성(transparency), 다양성, 비차별, 공정성(Diversity, non-discrimination and fairness), 사회 및 환경적 복지(Social and environmental well-being), 책무(Accountability)가 제시되어 있다. 이 중에서 투명성은 데이터, 시스템, AI 비즈니스 모델의 투명성과 이해관계자에게 적합한 방식으로 결정과정이 설명되어야 한다고 명시되어 있다(EU, 2019). 이러한 이유로 인공지능 분야에서는 블랙박스인 인공지능 모델의 설명가능한 인공지능(XAI, eXplainable Artificial Intelligence) 기법을 적용하고자 하는 노력이 계속되고 있다. 인공지능 모델 중 이미지 분류(image classification), 객체 탐지(object detection) 등에서 활발하게 적용되고 있다.

인공지능 모델은 순환신경망(Recurrent Neural Network, RNN), 합성곱 신경망(Convolutional Neural Network, CNN) 등을 토대로 인코더-디코더, 인코더-디코더를 확장한 U-NET (Ronneberger et al., 2015)을 비롯해 ResNet (He et al., 2016), VGG (Simonyan and Zisserman, 2015) 등 다양한 아키텍처가 등장하였다. 또한, 학습과정에서 유용한 피처에 더 집중할 수 있는 SE 네트워크(squeeze-and-excitation network), CBAM (Convolutional Block Attention Module) 등 어텐션 메카니즘(attention mechanism)도 등장하였다(Hu et al., 2020; Woo et al., 2018).

이러한 아키텍처의 등장에도 딥러닝 학습 과정에서 어떤 요소들이 학습 과정에서 영향을 주는지에 대한 이해는 매우 중요하다. 따라서, 인공지능 모델을 분석하기 위해 설명가능한 인공지능 기법을 통해 모델 결과에 대한 이해도를 높이고, 인공지능 모델의 내부 구조 분석을 통해 모델의 학습과정에 영향을 미치는 요소들을 찾고 요소들의 조합을 통해 모델의 개선여부를 파악하고자 한다.

2. 방법 및 자료

2.1. 자료 확장에 따른 모델 성능의 민감도 분석

2.1.1. 자료

2.1.1.1. 관측자료

관측자료는 위성자료를 기반으로 식생지수, 해수면온도, 강수량 자료를 수집하고 이용하였다. 식생지수(Normalized Difference Vegetation Index, NDVI)는 미국 NOAA (National Oceanic and Atmospheric Administration)에서 자료를 제공하고 있으며, NIR(근적외 반사값)과 Red 채널의 반사값을 노말라이즈해서 -1부터 1사이의 값을 가지며 0보다 크면 식생의 상태가 좋은 것을 나타내고 0보다 작으면 물이나 눈과 같은 식물이 아닌 표면을 나타낸다(Figure 2.1). 자료는 1981년부터 서비스 하고 있으며 NOAA 극궤도 위성 AVHRR (Advanced Very High Resolution Radiometer), VIIRS (the Visible Infrared Imaging Radiometer Suite)센서 기반으로 만들어진 자료이며 NOAA-7,-9,-11,-14,-16,-17,-18,-19)에서 얻은 데이터를 사용하여 공간해상도는 0.05°의 해상도로 일자료를 제공하고 있기 때문에 얻어진 자료를 1.5° 일자료로 변환하였다. 데이터셋은 GSFC (NASA Goddard Space Flight Center)와 UMD (University of Maryland)의 지표면 기후 데이터셋의 하나로 지표면 반사값, 관측시간, 위,경도 자료등을 netCDF-4 포맷자료로 제공하고 있기 때문에, 이러한 내용을 바탕으로 전처리하였다.

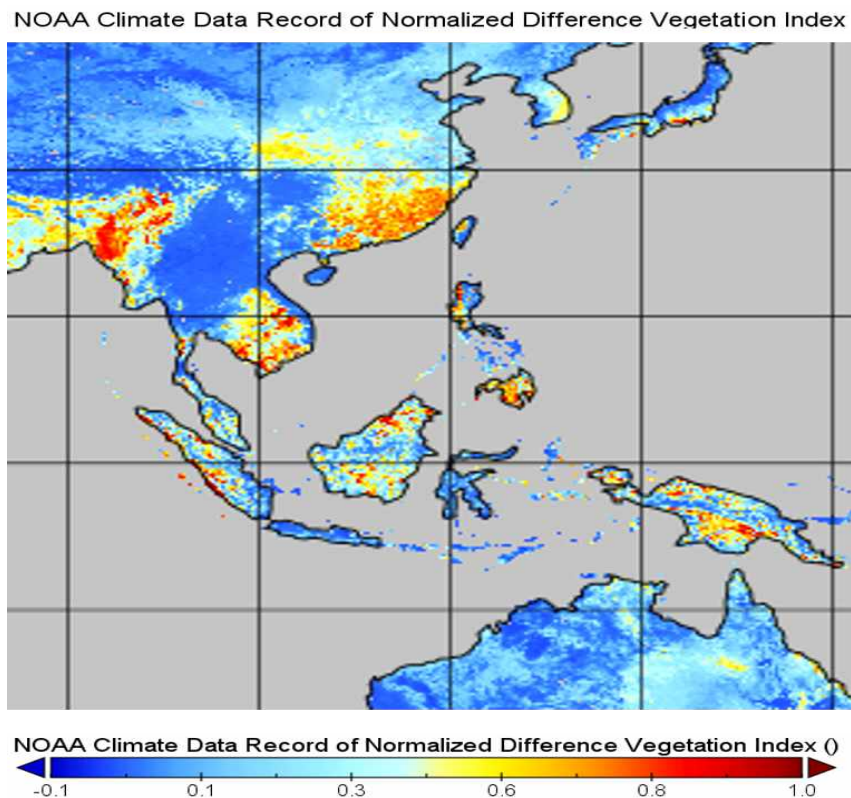


Figure 2.1. NOAA NDVI (Normalized Difference Vegetation Index) data form AVHRR satellite.

GPM IMERG (Daily Global Precipitation Measurement Integrated Multi-satellitE Retrievals for GPM)자료는 일별로 측정된 전 세계 강수량 데이터를 제공하는 product로 데이터는 GPM (Global Precipitation Measurement, 이하 GPM) 코어 위성과 협력 위성에서 수집한 자료를 결합하여 고해상도 강수량 분포와 지구의 강수 패턴 및 변동성 정보를 제공한다. GPM 위성은 강수의 양과 강수 패턴을 측정하기 위해 Ku, Ka 레이더센서와 마이크로파 이미저 센서를 사용하여 다양한 데이터를 생산한다. 이 연구에서 사용된 IMERG Final 데이터는 2000년 6월부터 자료를 수집하여 ECMWF와 같은 공간해상도인 1.5° 로 전처리하여 입력자료로 이용하였다(Figure 2.2).

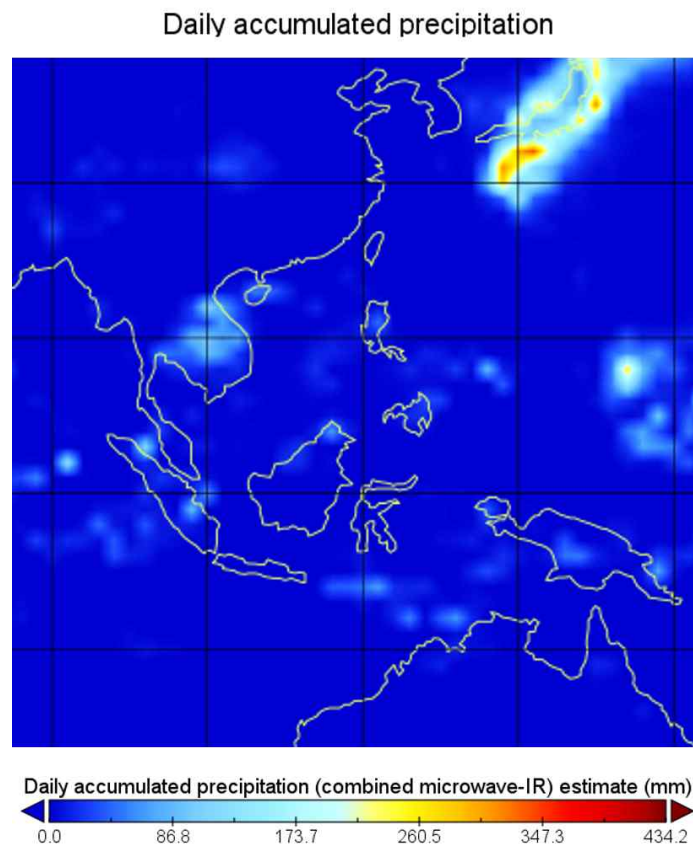


Figure 2.2. GPM IMERG Final data.

해수면 온도는 Optimum Interpolation Sea Surface Temperature (NOAA OISST) 자료를 수집하여 전처리하였다. 해수면온도 자료는 위성에서 산출한 해면 온도와 해양부이 및 선박관측 자료를 최적보간 기법으로 사용하여 전지구 일일 해면 온도를 산출하고 있다. 미국 국립기후데이터센터(NCDC)에서 1981부터 일자료를 산출하고 있으며 최근 Version 2의 데이터를 수집하였다. OISST자료는 “AVHRR-only” 와 “AVHRR+AMSR” 의 두 가지 버전이 있으며, 공간해상도는 0.25° × 0.25° 의 해상도로 자료포맷은 NetCDF4자료로 제공하고 있으며, AVHRR-only자료만 30년간의 장기간 기후자료를 제공하고 있다. (Reynolds et al, 2002). 사용할 수 있는 변수는 Sea Ice Concentration, Sea Surface Temperature, Sea Surface Temperature Anomalies들이고, 여기서 Sea Surface Temperature자료를 1998년부터 2017년자료를 시공간 보간하여 사용하였다

(Figure 2.3).

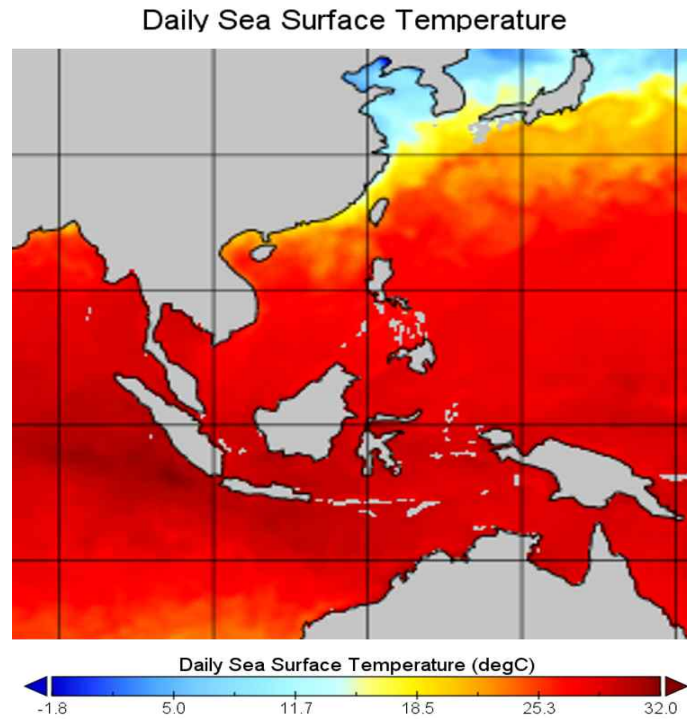


Figure 2.3. Sea surface temperature data from NOAA.

2.1.1.2 모델자료

계절내 기후자료는 ECMWF자료는 계절내에서 계절까지의 시간대에 예보 정확도와 이해를 높이기 위하여 자료를 생산하고 있으며, World Meteorological Organization (WMO)의 계절내 예측 프로젝트 기반하여 다양한 기후모델 자료(NCEP, BoM, JMA, ECMWF)를 제공하고 있다. 이 연구에서는 정확도 높은 ECMWF자료는 0-46일의 자료중에서 리드타임 3주자료를 1998년부터 2017년까지 모든 변수 자료를 수집하고 전처리 하였다. 변수는 MSLP, OLR, PREC, Q700, Q850, SST, T2M, TMAX, TMIN, U50, U850, V200, W850, Z200, Z500의 자료를 민감도 분석 학습자료로 맞춰 전처리하였다.

2.1.1.3 재분석자료

ERA5는 유럽중기예보센터(European Center for Medium-Range Weather Forecasts, ECMWF)에서 제공하는 대기 재분석 데이터셋의 5번 세대의 자료로 과거 관측 데이터를 일관되게 전처리하여 기후자료로 종합적이고 일관된 자료를 제공하고 있다(Figure 2.4). ERA5는 이전 버전인 ERA-Interim보다 고해상도 및 시간 해상도를 높여 제공하고 있으며, 물리 모델과 데이터 동화기술을 개선하여 제공하고 있다. ERA5 재분석자료는 1979년부터 현재까지 전지구적으로 자료를 제공하고 있으며, 연직프로파일은 137개로 지표면에서 80km까지의 포함하고 있으며, 온도, 습도, 풍속, 최고/최저 온도등 다양한 변수를 제공한다. 1998년부터 2017년까지 시간 자료를 일자료로 전처리하고 공간해상도도 1.5° 로 변환하여 사용하였다(Table 2.1).

Maximum temperature at 2 metres

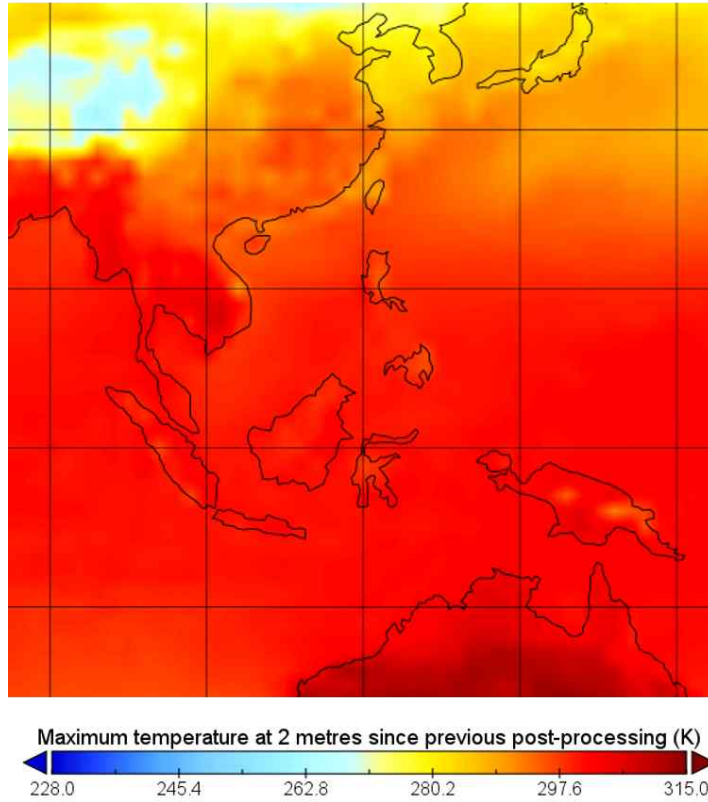


Figure 2.4. ERA5 Maximum temperature at 2 meters.

Table 2.1. Data description of ERA5.

Data Description	
Data type	Gridded
Projection	Regular latitude-longitude grid
Horizontal Coverage	Global
Horizontal resolution	Reanalysis: 0.25° x 0.25° (atmosphere), 0.5° x 0.5° (ocean waves) Mean, spread and members: 0.5° x 0.5° (atmosphere), 1° x 1° (ocean waves)
Temporal coverage	1959 to present
Temporal resolution	Hourly
File format	GRIB
Update frequency	Daily

ERA5는 관측자료인 NDVI, SST, ECMWF 모델 자료와 같은 기간 시/공간 해상도를 맞추어 타겟기간인 3주차에 맞춰 입력자료를 구성하였다. 타겟 변수는 강수량을 먼저 입력자료로 구성하고, 최대 온도값을 민감도 분석 입력자료로 전처리 하였다(Table 2.2).

Table 2.2. Overview of the characteristics of ERA5.

ERA5	
Period covered	1950–present
Production Period	Jan 2016–end 2017, then continued in near real-time
Model version	IFS Cycle 41r2
Assimilation system	IFS Cycle 41r2 4D-Var
Spatial resolution	31km globally, 62km for the Ensemble of Data Assimilation(EDA), 137 levels to 0.01hpa
Output frequency (temporal resolution)	Hourly analysis fields, 3-hourly for the Ensemble of Data Assimilations (EDA) Hourly forecast fields, 3-hourly for the Ensemble of Data Assimilations (EDA), up to 18 hours, with reduced frequency up to 10 days (not in initial release)
Uncertainty estimates	From a 10-member Ensemble of Data Assimilations (EDA) at 63 km resolution
Model input	Appropriate for climate (e.g. CMIP5 greenhouse gases, volcanic eruptions, SST and sea-ice cover)
Input observations	In addition, various newly reprocessed datasets and recent instruments that could not be ingested in ERA-Interim
Variational bias scheme	Also ozone, aircraft and surface pressure data
Satellite data	RTTOV-11, all-sky for various components
New parameters	ERA5 contains over 240 parameters on surface and single level alone, plus parameters on other level types. For specific parameters please compare the technical documentation linked below.
Parameters removed	A few parameters present in ERA-Interim are not available in ERA5. For availability of specific parameters please see the technical documentation linked below.
Handling of accumulated parameters	Accumulated from previous post-processing

2.1.2. 방법

2.1.2.1 딥러닝 모델

자료 입력에 따른 민감도 분석을 하기 위하여 딥러닝 모델은 U-Net의 개선버전인 Attention U-Net 모델(Oktay et al, 2018)을 이용하였다. U-Net 모델은 이미지 분석 및 예측에 있어서 신뢰할 수 있는 모델이며, 합성곱 신경망(CNN)의 대표적인 모델이다. U-Net 모델의 아키텍처는 인코더와 디코더 구조를 가지며, 인코더는 계층적 특징을 추출하고 디코더는 이를 사용하여 원래 이미지의 고해상도 세그멘테이션을 추출하기 위해서 사용된다. U-Net 모델의 특징 중의 하나는 인코더에서 추출한 피쳐의 특징을 디코더의 동일한 해상도의 층과 결합하여 해상도가 높은 이미지에서 산출 가능한 정보와 저해상도에서 산출 가능한 정보를 모두 사용하여 보다 정확한 세그멘테이션 맵을 산출 가능하게 해준다. 계절내 예측자료의 시계열 입력자료의 정확도 향상에 효과적으로 이용하기 위하여 U-net 모델이 이용되어왔다(Figure 2.5). Attention U-Net 모델은 U-Net 모델에서 어텐션 매커니즘을 추가하여 보다 정확도를 향상시킨 모델이다. 어텐션

매커니즘은 입력자료에서 중요한 부분에 가중치를 더 부여하여 모델이 좀더 중요한 부분에서 깊게 학습하도록 해주기 때문에 단독 U-Net모델을 이용한것보다 보다 정확도를 향상시킬 수 있기 때문에 본 연구에서 민감도 분석을 위하여 Attention U-Net모델을 사용하였다(Figure 2.6).

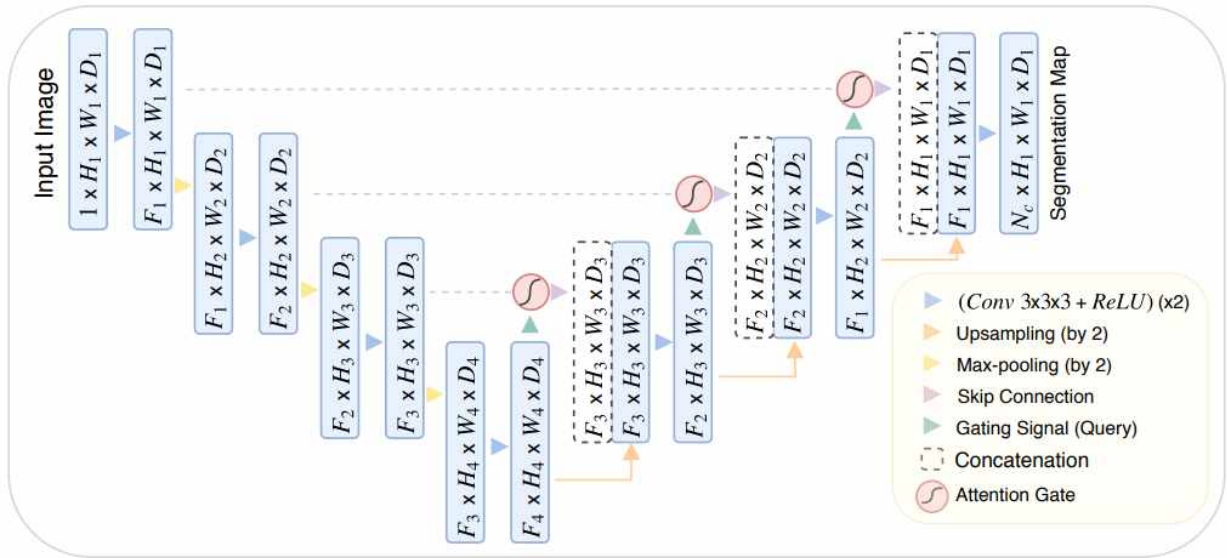


Figure 2.5. A block diagram of Attention U-Net architecture (Source: Oktay et al., 2018).

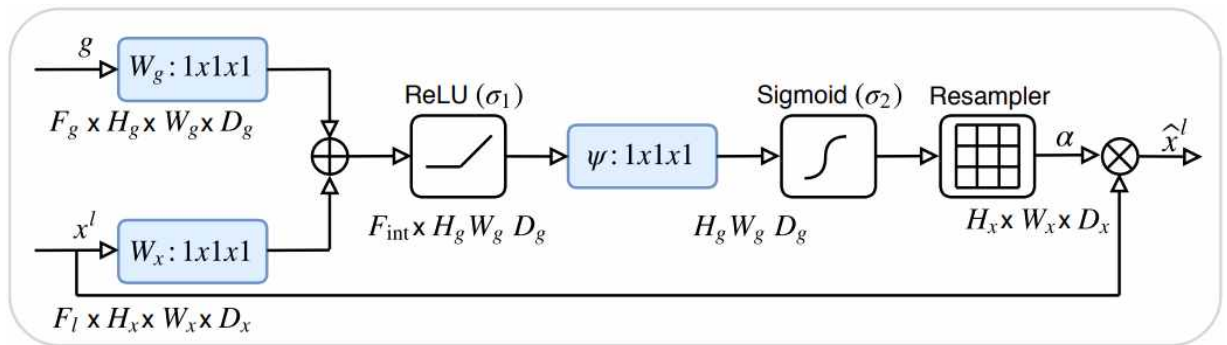


Figure 2.6. Schematic of the attention gate (Source: Oktay et al., 2018).

인코딩 과정에서는 Attention U-Net 민감도 입력자료인 모델자료와 관측자료에서 행렬의 주변 픽셀을 고려하여 범위를 넓혀가며 feature의 특징을 추출하고 3×3 합성곱을 수행할때마다 채널정보를 증가시키면서 레이어를 추가시켰다. 디코딩 과정에서는 인코딩에서 산출된 feature의 특징중에서 중요한 자료에 가중치를 더하고 upsampling을 하면서 최종적으로 온도 3 주치의 7일자료를 산출하였다. 하이퍼파라미터로 optimizer는 adam함수를 Loss는 MSE함수를 이용하였고, Epoch는 300, 배치 사이즈는 8로 고정하여 입력자료만 변화를 주고 학습하였다. 입력자료의 구성은 ECMWF 모델자료와 관측자료를 Attention U-Net의 학습자료로 이용하기

위하여 Figure 2.7과 같이 입력자료를 구성하였다. 모델자료 예측일인 2월 1일 기점으로 관측 자료는 2월 1일보다 일주일 전 자료를 이용하였고, 예측자료인 ECMWF모델자료는 리드타임 3주차의 7일자료를 이용하여 학습자료를 구성하였고, 타겟변수인 ERA5의 자료는 3주차 최고온도 7일자료를 타겟자료로 활용하였다.

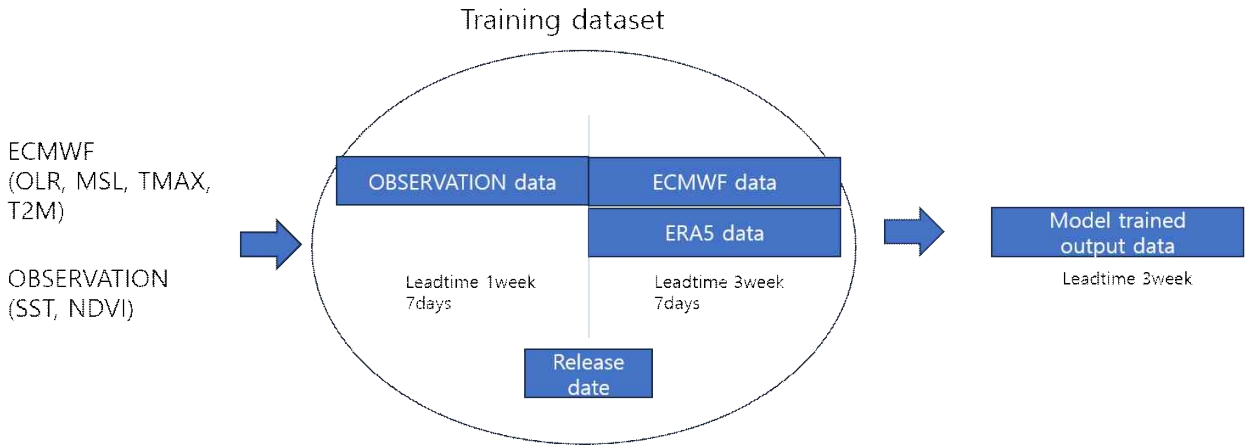


Figure 2.7. A diagram of input data from observation and ECMWF model data for Attention U-Net model.

민감도 분석을 위한 설계는 Attention U-Net모델을 최적화된 파라미터를 기반으로 고정하고 학습자료는 예측자료인 ECMWF와 관측자료인 NDVI, SST자료를 기준일 기준으로 관측자료는 과거 일주일 자료로 예측자료인 ECMWF자료는 리드타임 3주차 일주일 자료로 타겟변수인 강수량, 최고온도에 대해서 ERA5자료는 리드타임 3주차 일주일 자료로 구성하였다. 민감도 분석을 위하여 입력자료는 Filter, Wrapper, Embedded기법에 맞춰 입력자료에 따른 정확도 개선을 보고자 하였다(Figure 2.8).

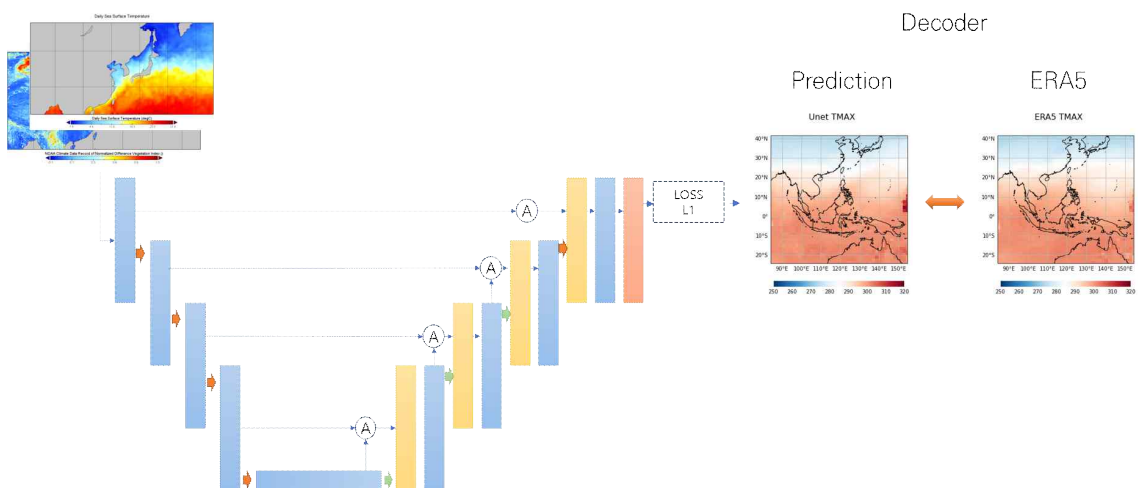


Figure 2.8. A diagram of trained datasets for sensitivity test.

2.1.2.2 Filter 기법

민감도 분석 기법중의 하나인 Filter 기법은 주로 신호 처리, 영상 처리, 그리고 기계 학습 분야에서 사용되는 기법중의 하나로 이미지 프로세싱에서의 필터인 이동 평균 필터는 시계열 데이터나 영상에서 잡음을 제거하거나 부드럽게 만들기 위해 사용되고 각 픽셀에 대해 주변 픽셀의 평균 값을 사용하여 중앙 픽셀 값을 대체한다. 가우시안 필터는 이미지에서 높은 주파수 성분을 감소시키기 위해 사용되는 필터로, 이미지를 부드럽게 만들거나 잡음을 감소시키는 역할을 하고, 신호 처리에서의 저패스 필터는 신호에서 낮은 주파수 성분만을 통과시키고 높은 주파수를 차단하는 필터방법이다. 기계 학습에서의 필터 민감도 분석 방법은 입력 데이터의 특징을 강조하거나 변형하여 정확도를 향상시키는데 사용된다. Attention U-Net의 바탕이 되는 합성곱 신경망(Convolutional Neural Network, CNN)에서 사용되는 컨볼루션 필터방법이 입력자료에서 특징을 추출하고 특성 중요도를 측정하여 중요한 특성만을 선택하여 정확도를 평가하고자 하였다. 따라서 ECMWF 모델 변수와 관측자료 변수에서 중요한 특정 패턴을 추출하여 정확도를 평가하고자 하였다. ECMWF의 MSLP, OLR, PREC, Q700, Q850, SST, T2M, TMAX, TMIN, U50, U850, V200, W850, Z200, Z500변수와 관측자료의 NDVI,SST을 ERA5의 최고온도와 의 상관성을 변수별로 구해서 상관성이 높은 변수만을 선택하여 학습자료로 구성하고 Attention U-Net모델로 정확도 성능을 비교분석하기 위하여 Filter기법을 사용하였다(Figure 2.9).

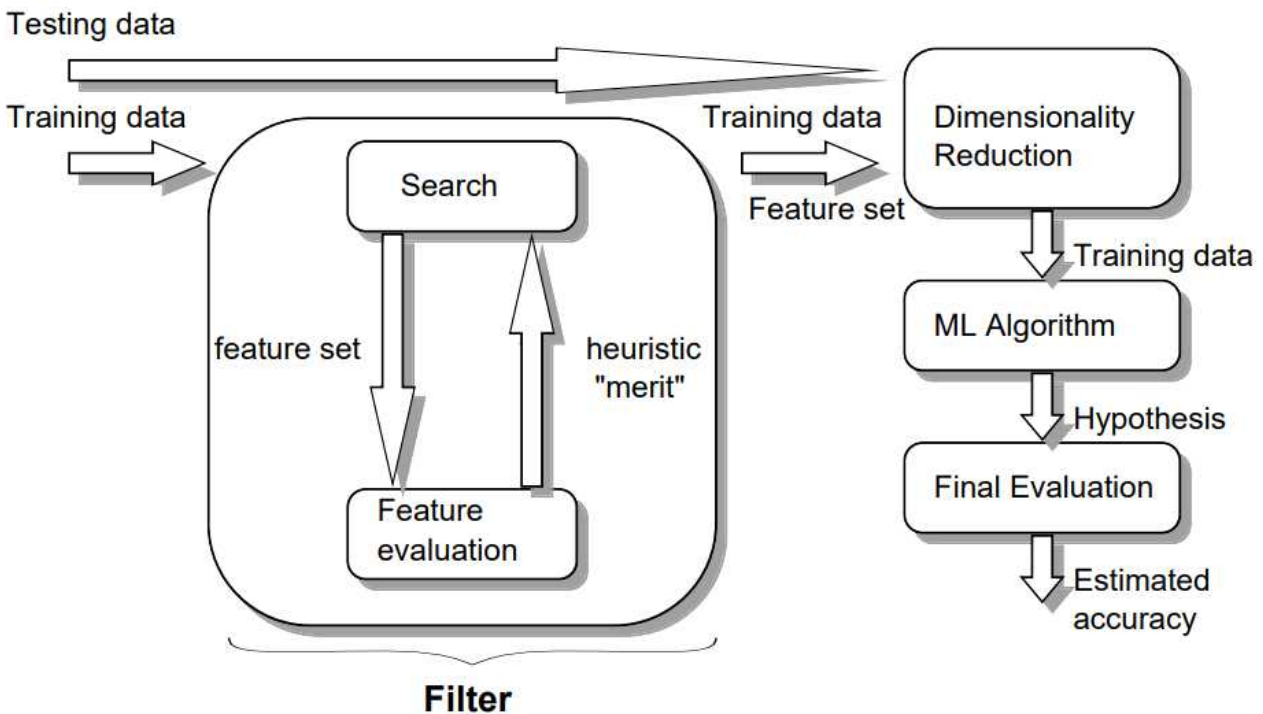


Figure 2.9. A diagram for filter feature selection scheme (Source: Hall, M., 1999).

모델/관측자료의 모든 변수와 재분석자료인 ERA5의 최고온도와의 상관성을 구한 결과 TMAX, TMIN, T2M, Z200, Z500, Q850, Q700, MSL, PRE, NDVI, SST의 순서대로 상관성이 높게 나와서 각각의 특성에 맞게 순서대로 입력자료를 만들어 Attention U-Net모델의 입력자료로 활용하였다(Figure 2.10; Figure 2.11).

Index	ERA	NDVI	OISST	MSL	OLR	TMAX	TMIN	SST	T2M	PRE	Q700	Q850	U50	U200	U850	V200	V850	W500	Z200	Z500
ERA	1	0.218385	-0.21648	0.55881	-0.439801	0.986976	0.982928	0.218893	0.98376	0.268856	0.585187	0.804289	-0.53766	-0.783689	-0.646431	-0.0699642	0.0800746	0.922617	0.883883	
NDVI	0.218385	1	-0.963888	0.04089	0.110463	0.289329	0.139159	0.541067	0.13214	0.212172	0.207443	0.210655	0.00715244	-0.18257	-0.0461398	-0.0223672	0.0245226	0.0206294	0.14221	0.12024
OISST	-0.21648	-0.963888	1	-0.0399993	-0.113319	-0.288169	-0.137126	-0.947898	-0.138542	-0.22328	-0.214438	-0.215867	-0.00653619	0.183582	0.045423	0.028477	0.0298188	-0.0288591	-0.142199	-0.128867
MSL	0.55881	0.04089	-0.0399993	1	-0.42977	0.556052	0.558822	0.0403719	0.560985	-0.0481733	0.162458	0.30113	-0.375526	-0.165375	-0.283291	-0.00764976	-0.0788668	0.0387837	0.72883	0.776851
OLR	-0.439801	0.110463	-0.113319	-0.42977	1	-0.436835	-0.453735	0.118354	-0.453188	0.217663	-0.185221	-0.259613	0.398272	0.261412	0.475857	0.0117747	0.0244165	-0.0677576	-0.474212	-0.485844
TMAX	0.986976	0.289329	-0.288169	0.556052	-0.436835	1	0.995393	0.211356	0.99564	0.273418	0.68839	0.826575	-0.532174	-0.695693	-0.629247	-0.121763	-0.0918367	0.0493481	0.919117	0.873173
TMIN	0.982928	0.139159	-0.137126	0.558822	-0.453735	0.995393	1	0.14014	0.998821	0.259274	0.599632	0.820012	-0.546592	-0.694866	-0.638718	-0.113379	-0.0924734	0.0471183	0.921716	0.877726
SST	0.218893	0.941887	-0.947898	0.04093719	0.118354	0.211356	0.14014	1	0.133768	0.216371	0.215399	0.21672	0.00596931	-0.18844	-0.0514781	-0.0188178	-0.0257135	0.0283344	0.143375	0.121265
T2M	0.98376	0.13214	-0.138542	0.560985	-0.453188	0.99564	0.998821	0.133768	1	0.259231	0.603292	0.823219	-0.542523	-0.694811	-0.636698	-0.114536	-0.0943542	0.0475312	0.922569	0.879807
PRE	0.268856	0.212172	-0.22328	-0.0481733	0.217663	0.273418	0.259274	0.216371	0.259231	1	0.384948	0.384561	-0.0496925	-0.27962	0.0179388	0.0183772	-0.0318893	-0.0112124	0.185893	0.138838
Q700	0.585187	0.207443	-0.214438	0.162458	-0.185221	0.68839	0.599632	0.215399	0.603292	0.384948	1	0.791531	-0.0788852	-0.483138	-0.243383	-0.185586	-0.243723	-0.195816	0.589562	0.449991
Q850	0.804289	0.218655	-0.21967	0.30113	-0.259613	0.826575	0.820812	0.21672	0.823219	0.384561	0.791531	1	-0.323374	-0.634446	-0.448598	-0.148698	-0.224471	-0.107214	0.712155	0.649554
U50	-0.53766	0.00715244	-0.00653619	-0.375526	0.398272	-0.532174	-0.546592	0.00596931	-0.542523	-0.0496925	-0.0788852	-0.323374	1	0.356892	0.532968	0.0277839	0.0179802	-0.10839	-0.548533	-0.525963
U200	-0.783689	-0.18257	0.183582	-0.165375	0.261412	-0.695693	-0.694866	-0.18844	-0.694811	-0.27962	-0.483138	-0.634446	0.356892	1	0.636695	-0.0713475	-0.0652446	-0.171282	-0.537238	-0.489717
U850	-0.646431	-0.0461398	0.045423	-0.283291	0.475857	-0.629247	-0.638718	-0.0514781	-0.636698	0.0179388	-0.243383	-0.448598	0.532968	0.636695	1	0.0112361	-0.0628749	-0.10745	-0.549781	-0.526513
V200	-0.0699642	-0.0223672	0.028477	-0.00764976	0.0117747	-0.121763	-0.113379	-0.0188178	-0.114536	0.0183772	-0.185586	-0.148698	0.0277839	-0.0713475	0.0112361	1	0.279649	0.295653	-0.0317644	0.8219516
V850	-0.0542759	0.0245226	0.0298188	-0.0788668	0.0244165	-0.0918367	-0.0924734	-0.0257135	-0.0943542	-0.0318893	-0.243723	-0.224471	0.0179802	-0.0652446	-0.0628749	0.279649	1	0.398816	-0.138416	-0.11412
W500	0.0800746	0.0206294	-0.0288591	0.0387837	-0.0677576	0.0493481	0.0471183	0.0283344	0.0475312	-0.0112124	-0.195816	-0.107214	0.10839	-0.171282	-0.10745	0.295653	0.398816	1	0.0445797	0.8741299
Z200	0.922617	0.14221	-0.142199	0.72883	-0.474212	0.919117	0.921716	0.143375	0.922969	0.185893	0.589562	0.712155	-0.548533	-0.537238	-0.549781	-0.0317644	-0.138416	0.0445797	1	0.984566
Z500	0.883883	0.12024	-0.128867	0.776851	-0.485844	0.873173	0.877726	0.121265	0.879807	0.138838	0.449991	0.649554	-0.525963	-0.489717	-0.526513	0.0219516	-0.11412	0.0441299	0.984566	1

Figure 2.10. Correlation heatmap of observation and ECMWF model variables.

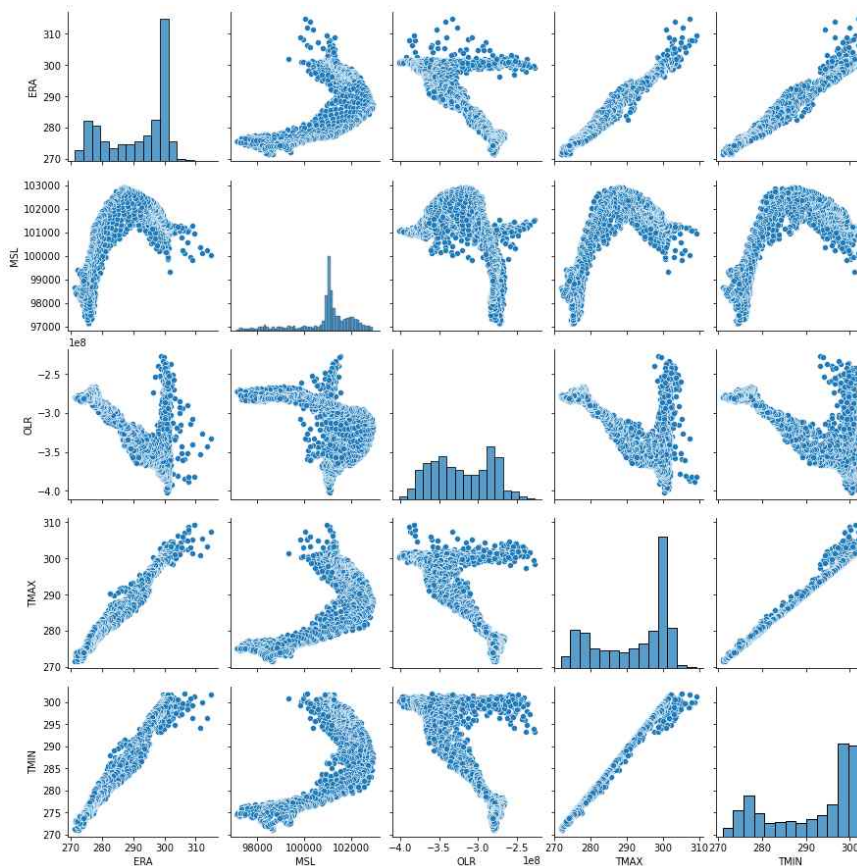


Figure 2.11. Scatterplot of correlation of variables.

2.1.2.3 Wrapper 기법

기계 학습에서 민감도 분석방법 중의 하나인 Wrapper기법은 단어에 주어진 의미와 같이 입력자료를 감싸거나 래핑하여 새로운 기능을 추가하거나 정확도 향상 민감도 분석을 나타내는 기법이다. 본 연구에서는 Attention U-Net모델에서 특성있는 변수의 특징을 알고자 이용하였다. 많이 이용되는 Wrapper 기법중에서 Feature Selection는 변수 특성 선택 알고리즘을 사용하여 가장 중요한 변수를 선택하고, 선택된 변수들로 Attention U-Net모델을 훈련하였다 (Figure 2.12). Wrapper방법중에는 앙상블 기법을 이용하여 다양한 모델을 결합하여 정확도를 평가하는 방법이나 여러 모델을 사용하여 각각의 예측을 다른 모델에 입력으로 사용하여 최종 예측을 생성하는 방식인데 본 연구에서는 딥러닝 모델의 Attention U-Net단독모델을 이용하였기 때문에 모델별 특성을 이용한 민감도 분석은 사용하지 않았다.

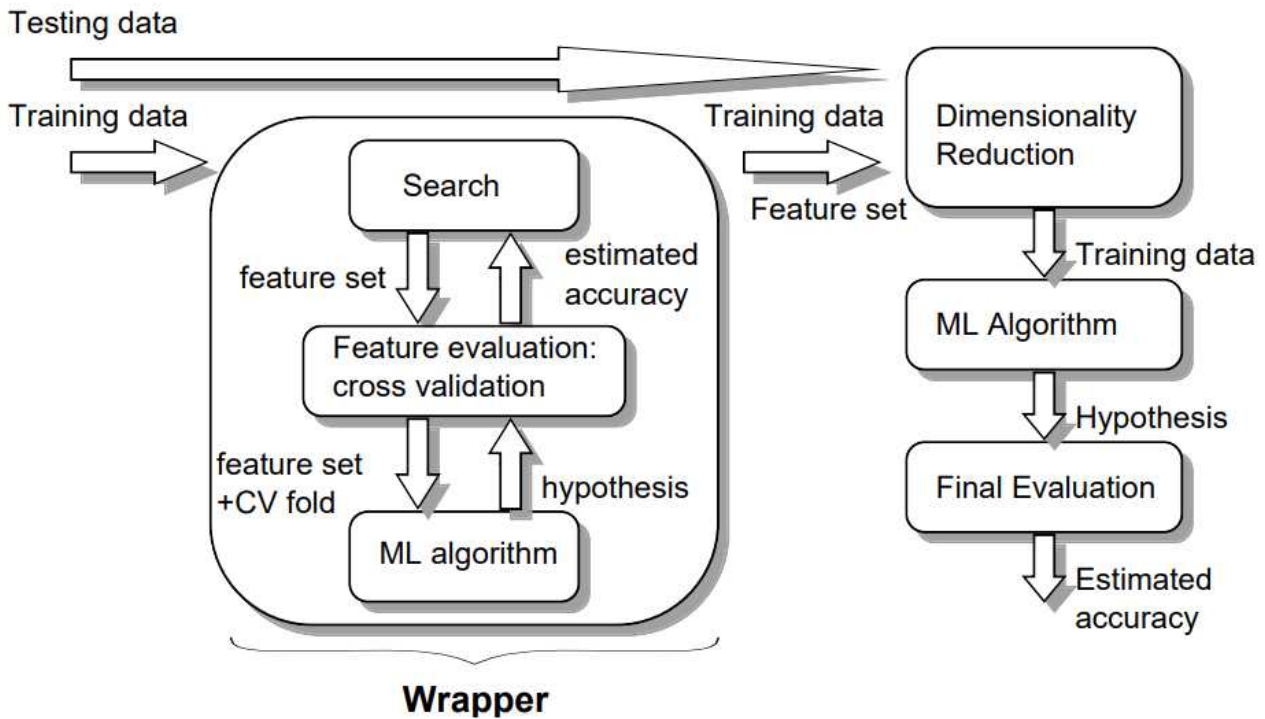


Figure 2.12. A diagram for wrapper feature selection scheme (Source: Hall, M., 1999).

2.1.2.4 Embedded 기법

Embedded 기법은 Filter와 Wrapper의 장점들을 결합한 방법으로 각각의 변수를 직접 학습하며, Attention U-Net모델의 정확도에 기여하는 변수를 선택하여 학습자료의 입력변수로 선택하여 분석하였다. 본 연구에서는 회귀식을 통한 회귀계수의 변수가 높은 순서대로 각각의 학습결과를 비교분석하였다(Figure 2.13).



Figure 2.13. A diagram for embedded method for feature selection (Source: Trabelsi et al., 2021).

2.1.2.5 분석 및 검증

민감도 분석방법은 다양한 입력자료에 따른 변수 선택에 따른 정확도를 분석하기 위하여 Attention U-Net 모델을 고정하고 Filter, Wrapper, Embedded 방법에 따른 변수선택을 하여 학습자료를 구성하고 정확도를 비교 검증하였다. 다양한 입력자료들의 학습시간을 단축하기 위하여 파라미터를 최적화할 때 배치파일을 최대한 크게 잡아 학습하였다.

예측의 정확도를 평가하기 위하여 입력자료는 과거 관측자료와 리드타임 3주차 모델자료를 구성하여 ERA5 최고온도에 자료를 매치시키고 정확도를 평가하였다. 정확도를 평가하기 위하여 ACC를 통계방법을 이용하였다(Eq. [2.1]).

$$ACC(t) = \frac{\frac{1}{n} \sum_{y=1}^n (S'_{yt} - \bar{S}_t)(O_{yt} - \bar{O}_t)}{\sqrt{\frac{1}{n} \sum_{y=1}^n (S'_{yt} - \bar{S}_t)^2} \sqrt{\frac{1}{n} \sum_{y=1}^n (O_{yt} - \bar{O}_t)^2}} \quad [2.1]$$

여기서 S 는 Attention U-Net 모델에서 산출된 Filter, Wrapper, Embedded 기법에 활용된 변수들의 의해 학습된 최고온도자료이며, O 는 ERA5의 최고온도자료이다.

2.2. 훈련기법, 초매개변수 조정 및 불확실성 분석을 통한 아키텍처 개선

2.2.1. 확장 U-Net 모델 계절내 기후예측 훈련자료

기본 U-Net은 2015년 Ronneberger et al. (2015)에 의해 제안되었으며(Figure 2.14), 22년 연구의 기본 모델이다. U-Net은 이미지의 전반적인 특징 정보를 얻기 위한 수축 경로

(Contracting Path)의 네트워크와 정확한 지역화를 위한 팽창 경로(Expanding Path)의 네트워크를 U자형의 대칭 형태로 갖는데, 기후 분야에서 활용도가 높은 모델이다 (Weyn et al., 2021). 23년 연구에서는 기본 U-Net에 Attention 및 Residual block을 장착하여 확장한 U-Net 모델들을 바탕으로 계절내 예측 목표 기후 변수 (예, 최고 및 최저기온, 강수)의 예측성을 개선하고자 했다. 23년 연구에서 선정한 확장 U-Net을 Table 2.3에 제시하였다.

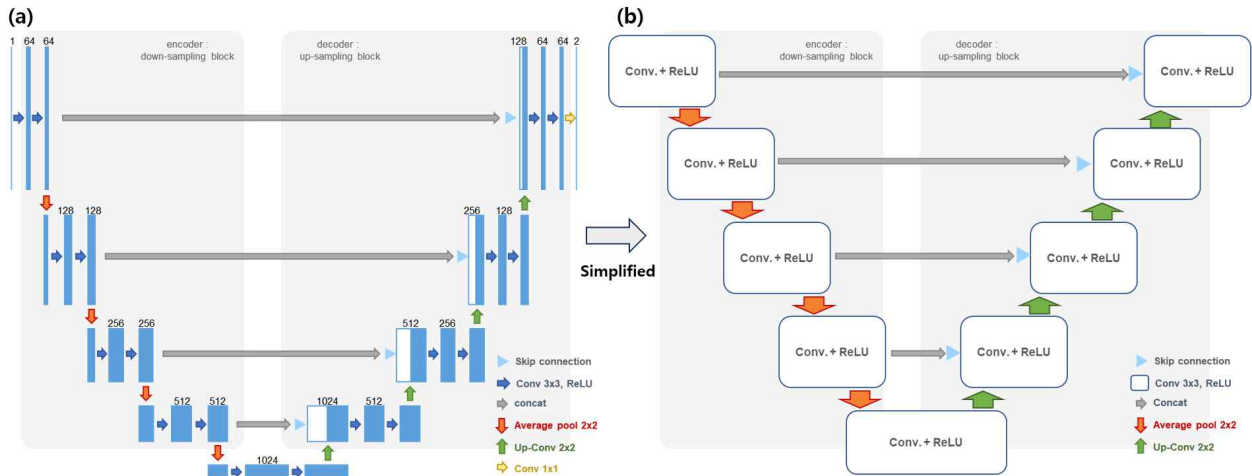


Figure 2.14. The U-Net which is used in the 2022 study. (a) is the original U-Net (Source: Ronneberger et al., 2015), and (b) is the U-Net schematized briefly in the study.

Table 2.3. The U-Net models applied in the study.

Applied Model	Key features
(1) U-Net	Encoder-decoder, pixel-level semantic segmentation, skip connections
(2) Attention U-Net	Focus more on important features, focus less on unimportant ones (convolutional block attention modules, CBAM)
(3) Residual U-Net	Shortcut connections, easier to optimize residual compared to original, improve gradient flow, avoid performance degradation in very deep networks
(4) Attention based-on Residual U-Net	Simultaneously embeds the attention mechanism and residual units into the U-Net structure for further performance improvement

(2) Attention U-Net

앞서 언급한 바와 같이 U-Net 모델은 image segmentation을 목적으로 제안된 fully-convolutional 기반 모델 형태가 U자형으로 context 포착과 세밀한 localization을 위해서 대칭으로 구성되었다. 이에 반해 합성곱 신경망(Convolutional Neural Network, CNN)은 인간의 시신경을 통해서 모습이 보이는 것에서 착안하여 만든 딥러닝 모델이다 (Oktay et al., 2018). 우리의 눈은 전체를 보지만 일부를 집중해서 보는 것과 같이 겹치는 부분이 있는데, 이처럼

CNN은 이미지 처리에 특히 특화되어 있으며 convolutional layer, pooling layer가 있다. U-Net, U-Net++, VB-Net 등이 다양하게 사용되다가, 기존 U-Net의 학습 정확도를 개선하기 위해 Attention Gate(이하, AG) 장치가 장착된 Attention U-Net을 제안하였다(Figure 2.15).

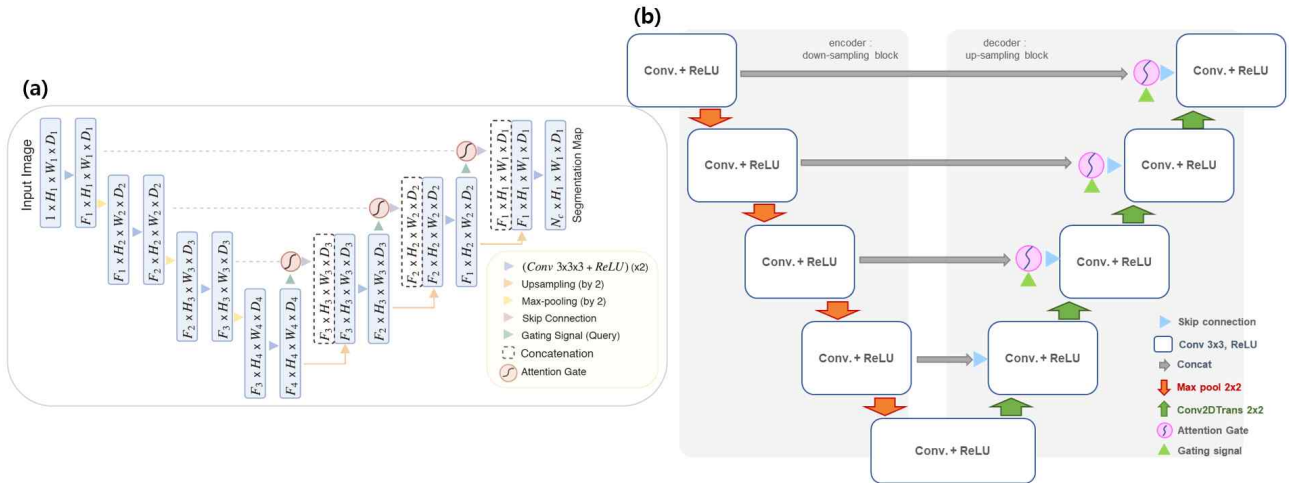


Figure 2.15. Attention U-Net: (a) is the original Attention U-Net (Source: Oktay et al., 2018), and (b) is the Attention U-Net schematized briefly in the study.

Attention 메카니즘은 2016년에 computer vision에서 사용되었다(Bahdanau et al., 2016). Neural Network에서 prediction을 만드는 것이 무엇인지 이해하기 위해서 사용되었다. Attention 메카니즘은 입력에 따른 다음 결과를 예측하는 작업에서 특정 정보에 더 집중해야 하기에 encoder에서의 어떤 값에 집중해야 하는지를 찾는 방법이다. 기존의 seq2seq는 크게 정보를 압축하는 Encoder 부분과 이를 통해 새로운 Data를 생성해 내는 Decoder 부분으로 나뉘어져 있다. 근데 seq2seq는 인코더의 최종 결과에만 의존하기 때문에 결과적으로 모든 시점에서의 출력이 동일한 정보에 기반하여 예측된다고 볼 수 있다. 그렇기 때문에 현재 시점에서의 상태를 제대로 반영하지 못한다고 볼 수도 있다. 많은 시계열 분석에 LSTM를 적용할 때 잘 작동하는 경우가 많지만, LSTM (Long Short Term Memory)의 Encoder에서 Decoder로 들어가는 과정에서, 전달되는 정보에서 소실되는 시간 정보들이 생기게 되었다. 기존 순환신경망(Recurrent Neural Network, RNN)의 문제점은 long-range dependencies, gradient vanishing 및 gradient explosion, local/global minima에 도달하는데 길고 많은 training steps이 필요하다는 것이다. 이와 같은 RNN의 문제점들에 대해서 Attention 메카니즘의 해결 방법 중 첫 번째는 sequence의 모든 부분에서 connections을 찾기에 기존의 long-range dependencies 문제점을 해결하고, 두 번째는 모든 sequence가 동시에 trained 되면서 적은 layers만 추가되기에 더 이상 gradient vanishing 과 gradient explosion은 문제가 되지 않는다고 주장하는 것이다. 세 번째는 RNN보다 train 하는데 적은 step을 요구하는 것이다. 따라서, Attention 메카니즘에서는 RNN에서 제기된 다양한 문제점들을 해결하기 위해 Encoder의 모든 시점에서의 출력들과 현재의 상태를 비교한다. 그리고 여러 시점에서의 데이터들을 모두 활용하여 정보 손실을 최소화한다는 특징을 갖고 있다. 이를 통해 가장 유사한 값에 대해 가중치를 주는 것이다. 그리고 이 정보를 참조하여 예측 작

업의 성능을 높일 수 있다. 즉, Attention을 적용하면 Encoder에서의 각 은닉층에 대한 정보들을 확인한다. Attention의 목표는 현재 상태와 가장 연관이 깊은 값에 더욱 집중하는 것이다. 그리고, 관련성의 척도로 유사도를 사용한다. 유사도를 구하는 방법에 따라서 Attention의 종류가 달라진다. 가장 자주 쓰이는 방법은 dot-product이다. 참고로 Dot-product는 내적이다. 내적은 두 벡터를 곱하는 두 방법 중 하나로 그 결과가 Scalar 값이 된다는 특징이 있다. 유사도를 계산한 후에는 softmax 함수를 통해 정규화를 수행한다. 유사도는 실수 전체 범위로 나오지만 확률은 0~1 사이의 실수 범위이기 때문이다. softmax 결과가 높을수록 결과적으로 해당 은닉 상태에 대한 가중치가 커지고, 결과적으로 그 은닉 상태에 집중하는 효과를 준다. dot-product로 얻은 Attention score들을 softmax 함수에 넣어서 Attention Distribution을 얻을 수 있다. 계산된 Attention Distribution 값과 Encoder의 Hidden state와 가중합을 구해서, Attention Value값을 구하여 Decoder의 Hidden state 차원 밑에 그대로 붙이고(Concatenate), 이를 output 계산의 input으로 넣으면 Attention 메카니즘이 마무리 된다. Query에 대해 어떤 Key가 유사한지 비교하고, 유사도를 반영하여 Value들을 합성한 것이 Attention Value를 설명하며 compare 함수로는 dot-product (inner product)가 많이 사용되며 Aggregation은 weighted sum을 많이 사용한다.

다시 말하면, Attention 메카니즘은 Key와 Query간에 dot-product를 통해 그에 따른 유사도를 가중치처럼 사용하여 Key에 맞는 value들의 조합으로 Aggregation (가중합)을 통하여 Attention value를 만들어 주는 구조가 Attention 메카니즘이다. 결국 Query와 비슷하면 비슷할수록 높은 가중치를 주어 출력을 주는 것이다. Attention Gate (AG)를 다시 설명하면 CNN 아키텍처와 쉽게 통합할 수 있고 (with U-Net), 최소한의 계산량 증가와 함께 모델의 민감도와 예측 정확도를 향상시킬 수 있다. 이것은 U-Net도 Convolutional layer에 ReLU 함수가 결합된 layer가 한 쌍(pair)으로 CNN에서 파생되었으므로 CNN과 잘 맞을 것이며, dimension이 줄어들기 때문에 비용 (예, 계산 속도 등)이 감소한다는 것은 즉, 계산량이 작은 상태에서도 정확도를 높일 수 있으니 “효율적이다” 라고 하는 것이다. 이것은 layer가 깊어지면 모델 매개변수(parameter)가 과도하게 중복되고 유사한 저수준의 피쳐(feature)가 반복적으로 추출되므로 계산 리소스도 많이 들기 때문이다. 이것을 개선하려고 시도한 것이 AG이다. Hard attention과 Soft attention이 있는데, Hard attention은 외재적으로 하이라이트할 영역을 지정해서 강조하며 미분은 불가능하고 강화학습을 필요로 하는 반면, Soft attention은 학습 도중에 내재적으로 task에 유리한 정보를 강조하고 불필요한 정보를 덜 강조하는 효과를 주는 것이 특징이다.

위에서 언급했듯이 통상적인 Attention은 dot product로 attention을 계산하지만 원래 연구(Oktay et al., 2018)은 상위레벨 피쳐와 하위레벨 피쳐의 크기를 동일하게 만든 후 단순 덧셈(attention zoomed Figure 2.16에서 동그라미 안의 덧셈을 설명한 것임)으로 element-wise 하게 강조될 부분은 강조되도록 유도한다. 그런 다음 ReLU를 거쳐 high activation만 남긴 후 sigmoid로 weight화하여 최종적으로 Upsampling 후 피쳐에 attention weight로써 곱해진다(Figure 2.16).

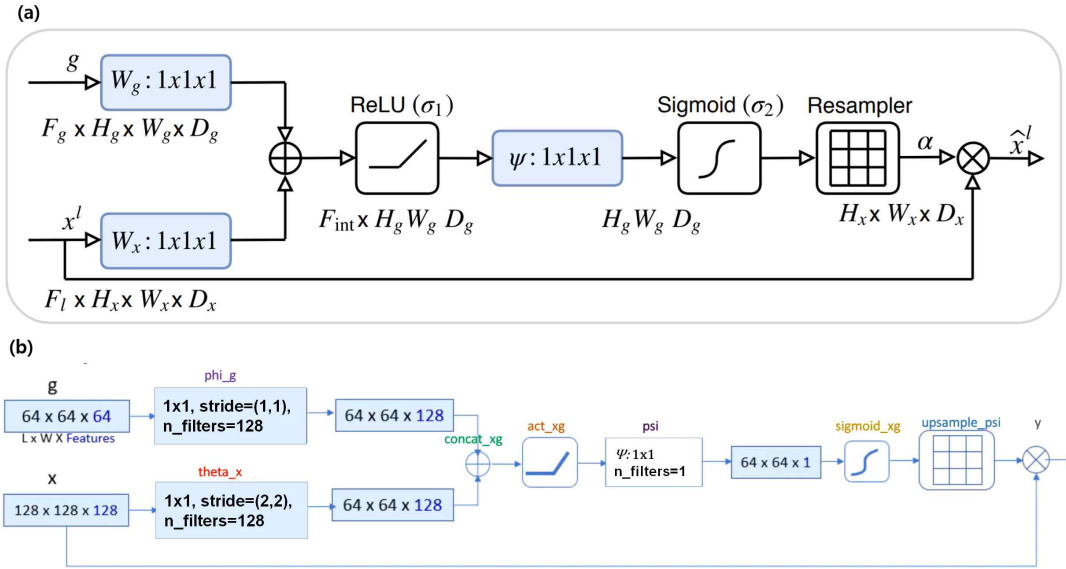


Figure 2.16. Attention Gate: (a) is the Attention Gate presented in the original paper (Oktay et al., 2018), and (b) is the Attention Gate expressed through coding in the study.

gating vector (g_i)는 각각의 픽셀 i 에 초점을 맞출 지역을 결정하기 위해 사용된다. gating vector는 저수준(low-level)의 feature 반응들을 제거하기 위해서 문맥 정보를 포함한다. 이는 이미지 분류에서의 AG들을 사용한다. 그리고 추가적인 attention을 사용하여 gating coefficient를 얻는다(Eq. [2.2] 및 Eq. [2.3]).

$$q_{at}^l = \Psi^T(\sigma_1(W_x^T x_i^l + W_g^T g_i + b_g)) + b_\Psi \quad [2.2]$$

$$a_i^l(x_{i,c}) = \sigma_2(q_{at}^l(x_i^l, g_i; \Theta_{at})), \quad a_2(x_{i,c}) = \frac{1}{1 + \exp(-x_{i,c})} \quad [2.3]$$

선형변환(Linear transformation)은 channel-wise input tensor에 1x1x1 conv를 사용하여 계산된다. 일반적인 모델에서는 softmax 활성화 함수, Eq. [2.2]는 주의 계수(σ_2)를 정규화하는 데 사용된다. 그러나 softmax 활성화 함수를 순차적으로 사용하면 출력에서 sparse (희소) 활성화가 생성된다. 이러한 이유로 원래 연구(Oktay et al., 2018)에서는 sigmoid 활성화 함수를 사용하였는데, 이것은 실험적으로 AG 매개변수에 대한 더 나은 훈련 수렴을 가져온다. 이것은 실험적으로 AG 매개변수에 대한 더 나은 훈련 수렴을 가져오기 때문에 grid-attention technique를 제안한다. Gating 신호는 모든 이미지 픽셀에 대한 전역 단일 벡터가 아닌, 이미지 공간 정보에 맞춰진 단일 grid 신호이다. 더 중요한 것은 각 skip connection에 대한 게이팅 신호는 Figure 2.16처럼 여러 이미징 크기에서 정보를 집계하여 쿼리 신호의 그리드 해상도를 높이고 더 나은 성능을 달성한다는 것이다. 그리고 AG 파라미터들은 hard-attention에서 사용되는 샘플링 기반의 업데이트하는 방법 없이도 표준 역전파 업데이트로 훈련될 수 있다. 따라서,

Oktay et al. (2018)에서 제안된 AG들은 표준 U-Net에 통합되어 두드러진 특징을 강조하고, AG는 정방향 패스와 역방향 패스에서 모든 뉴런의 활성화를 필터링한다. 배경 영역에서 발생하는 gradient는 역방향 패스 동안 가중치가 낮다. 이것은 주어진 작업과 관련된 공간 영역을 기반으로 얇은 레이어의 모델 매개변수를 업데이트할 수 있도록 해주는 것이다. 또한, 저수준의 feature map, 즉 첫 번째 skip-connection은 AG 기능에 사용되지 않는다. 즉, 고차원 공간에서 입력 데이터를 나타내지 않는다. deep supervision을 사용하여 중간 feature map이 각 이미지 스케일에서 의미론적으로 구별되도록 한다. skip connection의 작은 부분집합에서 조밀한 예측이 재구성되는 것을 방지하는 역할을 하게 된다.

Attention 세션을 정리하면, 1차원으로만 적용 가능한데, time-series 별로 혹은 feature 별로만 적용 가능했다. 이 때문에, Dual-Stage Attention 아이디어가 나오기도 했다(Liu et al., 2019; Qin et al., 2017). 또한, Attention은 길이 다시 말하면 time-sequence가 길어질수록, 내적 연산에 따른 속도 저하가 굉장히 크다고 알려져 있다. Beam-Search 알고리즘 등을 사용하면 속도 저하 문제를 조금 타파할 수는 있겠지만, 분류 문제에서 선택 가능하고, 회귀 문제에서는 선택 불가능하다. 예를 들면, Encoder와 Decoder의 time-series가 무척 긴 경우 (예를 들면, 약 200개 이상씩 될 때), 매 step마다 찾아 나가서 결정하다 보니 느려지는 경향이 있다고 한다. 그래서, Attention 시계열 데이터 예측에 대한 연구가 비교적 최근부터 시작되었다. Attention은 정해진 Time-Sequence에서만 집중할 수 있다. 하지만 현실의 문제는 다르다, 실제 데이터가 100일 범위로만 결정되어질 수 있을 것이라 생각했지만, 특정 어떤 데이터는 200일 전 데이터가 Attention이 더 잘 될 수도 있기 때문이라고 Qin et al. (2017)은 언급하였다. 이런 데이터 구성의 경우 Attention이 잘 동작하지 않는다. 또한, 대부분의 문제 해결 방식이 CNN을 이용하고 Attention을 진행하기 때문에, CNN을 적용하는 부분이 Dataset과 적합한 모델 아키텍처 구조인지도 판단해 봐야 한다고도 언급하였다. 그리고, Natural Language Processing (NLP) 외에는 연구의 부족, 메모리 병목, 특정 지점에서의 한계 (제한) 등이 거론되며, 실제로 SOTA (State of the Art)를 달성한 것으로 보이는 몇몇 제시된 논문은 (Shih et al., 2019), 그 데이터의 카테고리가 매우 제한적이며 실제 사례에 적용된 레퍼런스를 찾아볼 수 없다. Shih et al. (2019) 에 따르면 Attention을 이용하여 Time-Series Forecasting을 하는 것은 아직 시기상조라고 한다. Attention은 어떤 한 시점을 기준으로 유한한 단어를 학습시키기 위해 최적화 되어있고, 현실 세계에서 주가와 같은 무한한 실수 및 정수 예측 문제에서는 활용할 수 없다는 것이다. 대부분 %와 같은 상대적인 유한한 값으로 치환하여 사용한다 던지 하는 형태로 활용만 하고 안정성이 떨어진다고 한다. 또한, Attention 기반의 Time-Series Forecasting은 현재로서는 불안정하고 아직 예측성이 부족하다고 언급하였다. 그래서, seq2seq나 Encoder-Decoder 기반의 BI-LSTM 형태로 구성하는 것이 현재로서는 가장 합리적인 Time-Series Forecasting 방법이라고 제안하였다.

(3) Residual U-Net

딥러닝 학습은 레이어를 깊게 쌓을수록 성능이 좋아질 여지가 생기지만 문제는 레이어를 깊게 쌓을수록 기울기 소실(gradient vanishing)과 기울기 폭주(gradient exploding)이 쉽게 나타

나서 모델을 학습시키기 쉽지 않다. 이렇게 레이어가 깊어졌음에도 성능은 오히려 하락하는 문제로 앞서 언급한 문제인데 이것을 degradation이라고 한다. 쉽게 말하면, 기존의 네트워크는 입력 x 를 받고 여러 layer들을 거쳐 $H(x)$ 를 출력하는데 모델의 레이어를 더 깊게 쌓았더니 성능이 나빠졌다는 것은 다시 말해 차라리 레이어를 안 쌓는 것이 더 낫다는 것을 의미하기 때문이다. 그래서 이러한 degradation 문제를 해결하기 위해서 Residual Learning이라는 개념이 등장했다 (He et al., 2016; Maji et al. 2022; Singh et al., 2021; Zhou et al., 2022).

Residual Learning은 Figure 2.18b처럼 convolutional layer의 input에 해당하는 x 값을 convolutional layer의 output에 다시 더해줌으로써 convolutional layer가 input 값과 output 사이의 차이, 즉 잔차를 학습하는 데만 집중할 수 있도록 하는 기법을 의미한다. Residual Learning을 위한 input 값을 output 값에 더하는 과정을 shortcut connection 또는 skip connection으로 표현하며 이렇게 residual learning을 하도록 구성된 하나의 단위를 residual block이라고 한다. 이때 input 값과 output 값을 더하려면 데이터의 shape 및 dimension 차원도 같아야 할 것이다. 3×3 크기의 kernel을 사용하기 때문에 패딩(padding) 값을 주어 크기를 유지할 수 있지만, input 값의 채널 수와 output 값의 채널 수가 다르면 문제가 된다. 이런 문제를 해결하기 위해서 input 값의 채널 수와 output 값의 채널 수가 같을 때는 input 값을 output 값에 그대로 더하는 일명 identity shortcut connection을 수행하되 채널 수가 서로 다를 때는 shortcut connection 부분에서도 별도로 1×1 크기의 kernel을 가진 convolutional layer를 사용하여 채널 수를 조정하는 projection short connection을 수행하여 데이터의 shape를 맞춘다(He et al., 2016).

그렇다면 성능이 나빠지는 시점 (layer)부터 출력 $H(x)$ 가 입력 x 가 되는 identity mapping을 하면 적어도 성능이 유지되지 않을까 라는 제안이 있었지만 neural network은 상당히 복잡한 형태의 비선형 함수이기 때문에 자기 자신으로 돌아가는 방향으로 학습하는 것이 상당히 어렵다. 이것을 ‘잔차 학습’이라고 하는 Residual Learning은 $H(x)$ 가 아닌 출력과 입력의 차이인 $H(x)-x$ 를 얻도록 목표를 수정함으로써 자기 자신으로 가는 identity mapping을 보장해 준다. 이것이 잔차 학습이 잘 맞는 이유이다. 그래서, Residual Learning을 통해 여차하면 자기 자신 그대로 매핑할 수 있으므로 층을 깊게 쌓아서 성능이 하락할 일은 없게 되는 것이다. 또한 identity mapping의 미분값은 1이고 이를 단순히 더함으로써 backward시 gradient 소멸 문제를 해결할 수 있게 되는 것이다(Figure 2.18b).

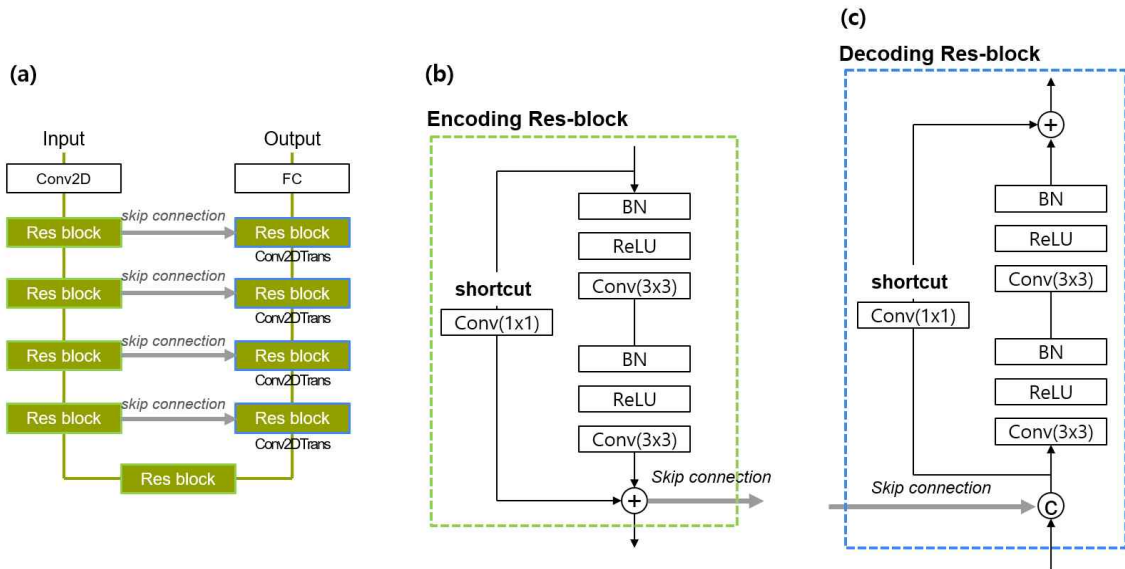


Figure 2.17. Residual block: (a) represents a general Residual U-Net, and (b) and (c) show residual blocks in encoding and decoding, respectively.

이것은 정보의 손실을 최소화해서 low-level feature를 인식할 수 있도록 하는 장치로, identity mapping으로 최초 입력된 정보 ($F(x_1)$)를 마지막에 더해주는 것이다 (Figure 2.18b). 다시 설명하면, Encoder block과 Decoder block을 연결하며, 각 layer를 그림에서 보이는 Residual unit으로 연결한다. Image를 통과시킬 때 감소 관점에서 shape-aware information을 유지할 수 있는 장점이 있으며, 이것은 계산 cost 측면에서도 이점이 있다. shape-aware information은 fully-convolutional network 관점에서 multi-layer를 쌓아 감에 있어 초기 layer의 filter는 이미지의 shape, edge와 같은 location feature를 추출하며, final layer의 filter는 이미지의 형태적 정보보다는 semantic feature (의미가 있는 것만 추출)를 추출하는 경향이 있다. 그러므로 layer를 쌓을 때 Residual unit (identity mapping)으로 연결하면 앞 단락에서 설명한 shape-aware information, 즉 $F(x_1)$ 을 final layer에도 최대한 유지시키게 된다.

Residual Learning에서 하나 더 주목할 만한 부분은 배치 정규화(batch normalization)를 사용한다는 것인데(Figure 2.17b 및 Figure 2.17c), 본 연구에서는 배치 정규화를 사용하지 않았다. He et al. (2016)에서 convolutional layer 직후에 batch normalization layer를 두고, 그 뒤에 ReLU 함수를 배치해서 사용했는데, 배치 정규화는 gradient vanishing과 gradient exploding 문제를 해결하고 과적합(overfitting)을 줄이는데 효과적이라고 알려져 있다. 배치 정규화 레이어는 미니 배치 단위로 평균과 분산을 계산하여 정규화를 수행하는데, 모델 학습 과정에서 scale 값과 shift 값을 학습해서 정규화에 함께 사용한다. 즉, 배치 정규화 레이어는 미니 배치 단위로 평균과 분산을 통해 표준화를 한 뒤, 그 값에 학습된 scale 값을 곱하고 학습된 shift 값을 더하는 방식으로 정규화를 수행하는 레이어이다.

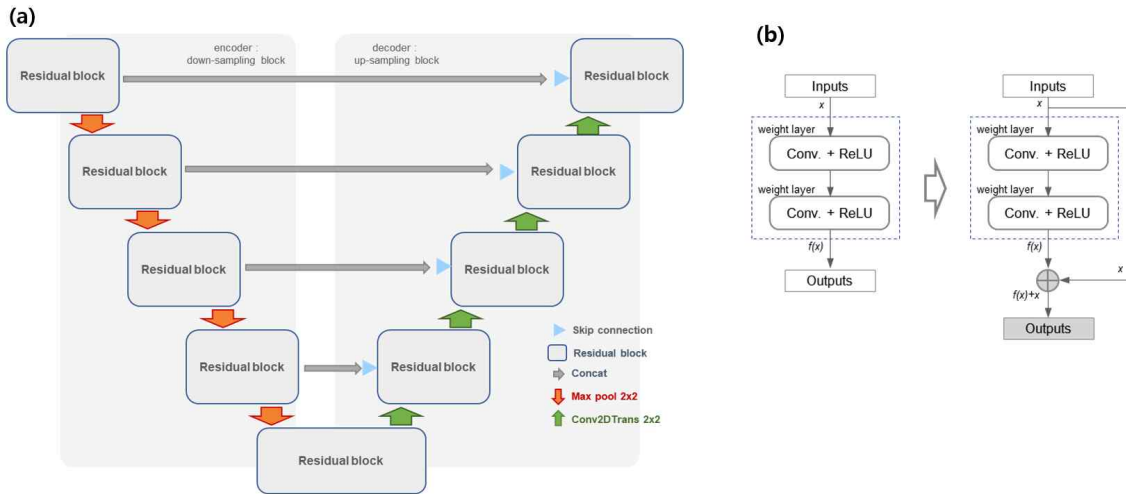


Figure 2.18. Residual U-Net: (a) is the Residual U-Net briefly expressed in the study, and (b) shows the residual block mounted on the Residual U-Net of the study.

Residual Learning에서 하나 더 주목할 만한 부분은 배치 정규화(batch normalization)를 사용한다는 것인데, 본 연구에서는 배치 정규화를 사용하지 않았다. He et al. (2016)에서 convolutional layer 직후에 batch normalization layer를 두고, 그 뒤에 ReLU 함수를 배치해서 사용했는데, 배치 정규화는 gradient vanishing과 gradient exploding 문제를 해결하고 overfitting을 줄이는데 효과적이라고 알려져 있다. 배치 정규화 레이어는 미니 배치 단위로 평균과 분산을 계산하여 정규화를 수행하는데, 모델 학습 과정에서 scale 값과 shift 값을 학습해서 정규화에 함께 사용한다. 즉, 배치 정규화 레이어는 미니 배치 단위로 평균과 분산을 통해 표준화를 한 뒤, 그 값에 학습된 scale 값을 곱하고 학습된 shift 값을 더하는 방식으로 정규화를 수행하는 레이어이다.

Figure 2.18는 본 연구에서 구축한 Residual U-Net으로 기존 U-Net에서 Encoder 및 Decoder 부분에서 layer마다 residual unit with identity mapping을 적용하는 것이다. 기존의 U-Net은 인코더의 얇은 구조로 인해 특징 압축 과정에서 손실이 발생하고, 이는 객체의 클래스 분류에 필요한 context 정보 부족을 초래하고 segmentation 정확도를 감소시키는 문제가 있다. 이를 개선하기 위해 Residual Learning을 활용한 Encoder-Decoder 구조의 모델을 제안하였다. Residual Learning은 특징 손실과 기울기 소실 문제를 방지하는데 효과적입니다. 또한, 인코더에서 down-sampling 연산을 줄여 feature map에 포함된 공간 정보의 손실을 개선하였다. 다시 말하면, Residual Learning 기반의 U-Net 아키텍처는 기존 U-Net 아키텍처 내에 위에서 설명한 residual block 연결을 통합하는 U-Net의 변형 아키텍처라고 할 수 있다. 이러한 residual block 연결은 Vanishing Gradient 문제를 완화하고 네트워크의 전반적인 성능을 향상시키는 데 도움이 될 것이다. 이러한 방법은 기존 U-Net 방법에 비해 segmentation 결과가 약 12% 향상되었다는 연구 결과가 있다(Chen et al., 2021; Shin et al., 2021a, 2021b). 그래서 Residual U-Net이 제안된 목적은 의료 분야에서 낮은 의료 이미지의 특징 분포를 Encoding하기 위해 Residual U-Net이 도입되었다. 기존 U-Net은 Encoder 부분에서 convolutional layer 수

가 적어 추출되는 특징이 부족하다. 추출된 특징은 객체의 범주를 분류하는 데 필수적이며, 부족할 경우 분할 정확도를 저하시키는 문제를 초래한다. 따라서, 이 문제를 개선하기 위해 본 연구에서도 Encoder에 Residual Learning을 활용한 Residual U-Net을 제안하였다(Figure 2.18). 앞서 Residual Learning에 대해 설명한 것과 같이 특징 추출 능력을 개선하고, 연속적인 convolutional으로 발생하는 특징 손실과 기울기 소실 문제 방지에 효과적이어서 여러 분야에서 Residual U-Net이 활용되고 있다.

(4) Attention based-on Residual U-Net

최근 앞서 설명한 가중치 계산 및 전달 장치를 다양하게 결합한 하이브리드 U-Net 모델은 Residual Net과 U-Net을 기반으로 하며, Attention 메커니즘을 추가하여 성능을 향상하기 위해 제안되었는데, Attention 메커니즘을 사용하여 모델이 이미지의 중요한 부분에 더 집중할 수 있도록 유도하여 불필요한 정보를 감소시켜 학습 정확도를 높여 정확하게 이미지를 분할할 수 있다는 장점이 있다. Residual 연결은 네트워크의 깊이를 증가시키면서도 gradient vanishing 문제를 완화시켜 학습을 용이하게 하며, 이것은 네트워크가 더 깊어져도 성능을 향상시킬 수 있는 장점이다. 또한, U-Net 구조의 유연성인데, U-Net은 입력과 출력의 크기가 다른 영상에 대한 분할에 유용한 구조를 제공하기 때문에 Encoder와 Decoder 구조를 결합하여 고해상도 이미지에도 좋은 성능을 발휘할 수 있다는 장점이 있다. 그래서, 이와 같은 모델은 의료 영상 분할, 자연어 처리, 음성 인식 등 다양한 분야에서 활용될 수 있다. 그러나, 모델 구현이 다소 어렵고 상대적으로 복잡하여 학습 시간이 오래 걸려 높은 계산 비용을 유발할 수 있다. 특히, 큰 규모의 데이터셋과 이미지에 적용할 때 학습 및 추론 시간이 오래 걸릴 수 있다. 앞서 언급한 바와 같이 깊고 복잡한 모델 구조 때문에 작은 Dataset에서도 과적합의 위험을 증가시킬 수 있으며 충분한 데이터가 없는 경우 모델이 새로운 데이터에 대해 일반화하기 어려울 수 있다는 단점이 있다. 복잡한 모델은 많은 초매개변수를 포함하고 있어, 이를 조정하는 것이 어려울 수 있어서 적절한 초매개변수 튜닝 없이는 최적의 성능을 이끌어내기 어려울 수 있다는 단점이 제기되고 있다. 본 연구에서는 Attention based-on Residual U-Net이다(Figure 2.19).

본 연구에서 Attention based-on Residual U-Net은 앞서 설명한 Attention U-Net과 Residual U-Net의 조합으로, residual block 연결과 attention 메커니즘을 모두 통합한 것이다. 이 아키텍처는 신경망의 전반적인 성능을 향상시키고 입력 이미지의 잡음과 변형에 대해 신경망 아키텍처를 더욱 견고하게 만드는 데 도움이 될 수 있을 것이다. 이 Residual-Attention 학습 메커니즘은 Residual-learning과 Attention-block을 기반으로 하며 두 가지로 나뉜다. Residual-Attention이 잘 맞을 수밖에 없는 이유는 바로 Skip connection이 존재하기 때문에 Main path에서 Optimal depth이후의 Weight와 Bias가 전부 0에 수렴하도록 학습된다면 Optimal depth에서의 Output이 바로 output (예, Classification 혹은 regression)으로 넘어갈 수 있기 때문이다. Self-attention은 입력 시퀀스의 다른 위치 간의 의존성을 모델링하는 방법으로, 주어진 위치와 다른 위치 간의 상대적인 중요성에 대한 가중치를 계산한다. 즉, 입력 레이어를 다시 참조하는 residual function과 특징에 집중하는 attention을 사용하여 정확도 향상이 가능하다는 것이다. 이것은 이미지 분할 및 의료 이미지 분석과 같은 작업에서 특징을 더욱 잘 고려하고

효과적으로 학습할 수 있도록 도와주기 때문에 의료 분야에서 상당히 많이 활용되고 있는 이 유이다(Le et al., 2023; Maji et al., 2022; Mu et al., 2023; Zhang et al., 2020).

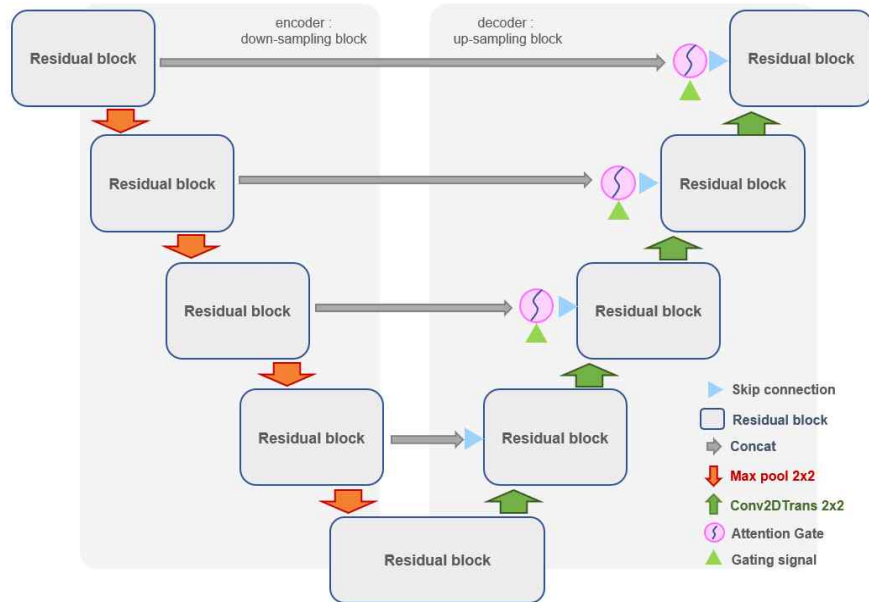


Figure 2.19. Attention gate based-on Residual U-Net of the study.

2.2.2. 그리드 탐색 알고리즘

초매개변수(hyperparameter)는 모델을 생성할 때 사용자가 직접 설정하는 변수이고, 파라미터(parameter)는 모델의 학습 과정에서 생성되는 변수이다. 신경망 모델에서 에포크(epoch, 학습 회수), 배치 사이즈(batch size, 한 번에 학습하는 양), 학습률(learning rate), 활성화 함수(activation function) 등이 초매개변수에 속한다(Table 2.4). 그리드 탐색(Grid Search)은 초매개변수를 일정한 간격으로 변경하며 최적의 파라미터를 찾아가는 기법이 그리드를 생성한다고 하여 붙여진 것이다. Figure 2.20처럼 가로축이 초매개변수이고 세로축이 목표 함수일 때, 목표 함수 값이 최대가 되는 초매개변수를 찾는 문제를 풀어야 한다고 가정할 때, 그리드 탐색은 특정 범위 내에서 초매개변수를 일정 값만큼 일일이 변경, 즉 순차적으로 모두 입력한 뒤 가장 높은 성능을 보이는 출력값을 비교하는 방식으로 동작한다. 즉, 가장 높은 성능을 가진 값을 찾기 위해 가능한 모든 초매개변수 조합을 검색하는 기법이기에 때문에 계산 비용이 많이 들기도 하지만 모델에 가장 적합한 초매개변수를 찾는 데에는 가장 체계적이고 신뢰할 만한 접근 방식으로 알려져 있다.

그리드 탐색은 2가지 한계점이 있는데, 첫째는 최적의 해를 찾을 수 있다는 보장이 없다는 것이다. Figure 2.20에서 초매개변수가 0 근처에서 목표 함수가 최댓값을 갖는다. 그러나, 목표 함수 값을 탐색하는 파란색 점선이 최적의 해를 그냥 지나치는 것을 볼 수 있다. 이처럼 그리드 탐색은 최적의 해를 찾지 못하는 경우가 생길 수 있다.

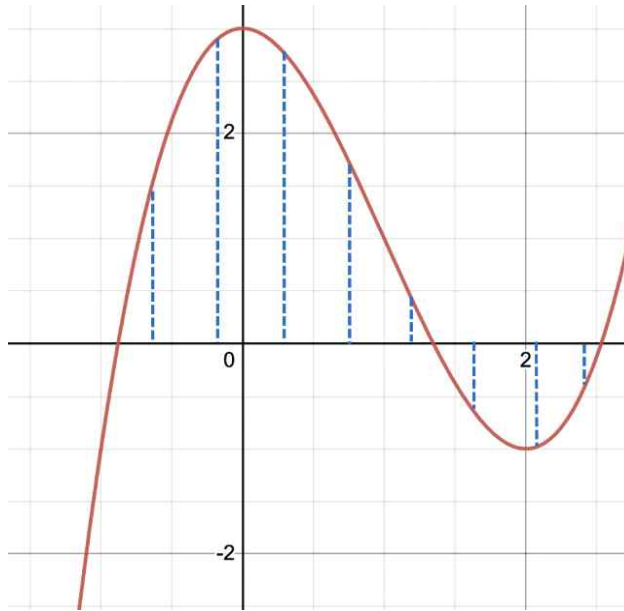


Figure 2.20. Grid Search: expressing the method of finding optimal hyperparameters as a function.

둘째는 동작 방식이 비효율적이다. 지정한 모든 구간을 탐색해야 하기 때문에 굳이 탐색하지 않아도 되는 초매개변수 값에 대해서도 탐색하는 경우가 생긴다. 예를 들어, Figure 2.20에서 아래로 블록한 구간은 사실상 출력이 음수이기 때문에 최댓값을 구하는 문제에서는 탐색할 필요가 없는 구간임에도 불구하고 그리드 탐색에서는 이러한 구간까지 탐색하기 때문에 탐색 과정 (시간 등)이 비효율적이라고 할 수 있기 때문이다.

Table 2.4. Types of hyperparameters in neural network models.

Hyper-parameter	Explanation
epoch	one cycle where the entire dataset is passed forward and backward through the neural network for training
batch size	the number of data samples that are processed and used to update the weights of the model during one iteration. For example, if the batch size is 32, the model processes and updates its weights using 32 data samples at a time
learning rate	how much the model's weights are updated during training, affecting the speed and stability of learning
activation function	the output of a neuron and enables learning complex patterns from data by introducing nonlinearity (e.g., softmax, Tanh, ReLU, leaky ReLU etc.)
optimizer	adjusts the model's parameters during training to minimize the loss function (e.g., Stochastic Gradient Descent: SGD, Adam, RMSprop, Adagrad, Momentum etc.)

Table 2.5은 본 연구에서 설정한 초매개변수 조합으로 에포크는 50에서 500까지인데, 랜덤 탐색(random search) 알고리즘처럼 범위(range)를 설정한 것이 아니라, 이전 값의 배수로 증가 시킨 것이다. 즉, 50 다음에는 100과 같이 설정하였다. 배치 사이즈는 64에서 시작하여 128, 256, 1024, 2048로 4의 배수로 설정하였다. 또한, 학습률은 0.001과 0.0001, 0.00001로 설정했는데, 중간에 0.005를 삽입하지 않았는데, 추후 상세한 간격으로도 테스트가 필요할 것으로 판단된다. 또한, 손실함수를 최소화하면서 적용 모델의 매개변수 갱신을 위한 optimizer 및 활성화 함수는 본 연구에서 여러 범위 및 종류로 설정하지 않고 활성화 함수는 가장 많이 적용되는 ReLU로 설정하고, optimizer는 최근 가장 성능이 좋다고 알려진 Adam으로만 설정하였는데, 역시 추후 활성화 함수와 optimizer에 대한 그리드 탐색 실험도 필요할 것으로 판단된다.

Table 2.5. Information of hyper-parameters and their ranges set in the study.

Hyper-parameter	Setting of the study
epoch	50, 100, 250, 500
batch size	64, 128, 256, 1024, 2048
learning rate	1e-03, 1e-04, 1e-05

2.2.3. 계절내 기후 예측 훈련자료

본 연구에서는 22년 연구에서 사용한 훈련자료를 사용하였는데, European Centre for Medium-Range Weather Forecasts (이하 ECMWF)의 S2S 예측 자료(Vitart and Robertson, 2018; de Andrade et al., 2019; ECMWF, 2020)와 날씨 예보 현업 연구 대상 모델인 Global Seasonal Forecast System의 버전 5 (이하 GloSea5)의 S2S (Scaife et al., 2014; MacLachlan et al., 2015) 뿐만 아니라, Meteorological Service of Canada (이하 ECCC), NOAA의 National Centers for Environmental Prediction (이하 NCEP), China Meteorological Administration (이하 CMA) 및 영국(United Kingdom)의 Met Office (이하 UKMO) 등 6개 개별 기후모델을 이용하였다(Table 2.6).

Table 2.6. S2S (Sub-seasonal to Seasonal) climate models used in this study, number of ensembles and period of hindcast of each models.

Climate model	Hindcast ensemble size		Hindcast length
GloSea5	3	mean	1991-2010
UKMO	7	mean	1993-2015
CMA	4	mean	1994-2014
ECMWF	11	mean	1998-2017
NCEP	4	mean	1999-2010
ECCC	4	mean	1998-2014
MME	-	Averaged	1995-2014

개별 기후모델의 기후 변수는 TMAX 및 TMIN (Maximum and minimum air temperature

at 2m), T2M (2 meter air temperature), MSLP (Mean Sea Level Pressure), PREC (Precipitation), SST (Sea surface temperature), OLR (Top net thermal radiation), U50, U200, U850, V200, V850(U and V component of wind at 50, 200 and 850hPa), W500 (Vertical velocity at 500hPa), Q850 (Specific humidity at 850hPa), Z200 및 Z500 (Geopotential height at 200 and 500hPa) 등 약 16개인데, 이 중 UKMO는 MSLP을, KMA는 SST, OLR, Q700, Q850을 이용할 수 없다. S2S 예측 기후 변수의 시간 해상도는 TMAX 및 TMIN (6시간 간격의 시간 자료)을 제외하고 일별이며 (daily), 공간 해상도는 1.5도 (적도 부근에서 약 100km)의 전지구 규모로 위도 121개 및 경도 240의 격자 배열을 갖는다. 또한 앙상블 수, 과거 재현 예측 (이하 Hindcast) 자료의 생산 방법, 선행예측시간(lead-time), Hindcast의 기간, 예측 자료의 생산 주기 등이 모두 다르다.

S2S 예측 자료의 앙상블은 기후모델의 기후 변수가 증가하면 그 앙상블도 증가되는데, 이것은 합성곱 신경망(Convolutional Neural Network, 이하 CNN) 네트워크의 속도 및 컴퓨팅 계산의 문제가 될 수 있어 개별 기후모델의 기후 변수의 앙상블은 control과 perturbation에 상관없이 평균하였다. 예를 들면, ECMWF 기후모델의 경우 control에 대한 17개 기후 변수와 perturbed에 대한 17개 기후 변수는 Hindcast에 대해 11개 앙상블로 총 187개 (11 ensembles × 17 variables)로 구성될 뿐만 아니라, 이것이 전체 도메인 격자 (121x240) 뿐만 아니라 기간 및 주기 (20년 × 46일 × 52주)로 구성되기 때문에, read/open 및 속도에 어려움이 발생하였다. 따라서 입출력 및 속도 문제 해결을 위해 (control 및 perturbation에 상관없이) 17개 변수별로 해당 앙상블 11개를 평균하였다. 마찬가지로 GloSea5, UKMO, CMA, NCEP, ECCC 기후 모델도 각각의 기후 변수별로 앙상블이 평균이 되도록 처리하였다.

S2S의 MME (Multiple Model Ensemble) 예측 자료의 경우는 본 연구에서 6개 개별 기후모델의 S2S 예측 자료를 모두 결합하여 MME를 생산하였다. 6개 개별 기후모델의 Hindcast 기간 전체가 포함되는 범위는 1999년에서 2010년인데, 12년에 불과하므로 이 기간을 MME 가능 기간으로 설정하기에는 너무 부족하였다. 6개 개별 기후모델의 Hindcast의 모든 기간이 완벽하게 포함되지 않더라도 Hindcast 기간을 최대 20년에 맞추기 위해, NCEP 및 GloSea5 기후모델의 앞뒤로 4년을 포함하지 않더라도 4개 개별 기후모델이 모두 배치될 수 있는 기간으로 1995년에서 2014년을 선정하였다. MME 시작일은 ECMWF와 ECCC의 시작일 중 목요일을 기준으로 설정하였다. 예를 들어, 2018년 1월 4일은 목요일로 ECMWF와 ECCC의 시작일인데, 시작일이 매일, 즉 일별인 CMA와 NCEP은 1월 4일과 결합이 될 수 있지만 GloSea5와 UKMO는 1월 1일이 시작일로 앞에 3일 패스하고 1월 4일과 결합하게 된다. 선행예측시간 (lead-time)도 각각 다른데, MME의 선행예측시간은 30일로 설정하였다. 또한, S2S 예측 자료들은 학습에 활용하기 쉽게 다차원 배열 (예, 시간, 선행예측시간, 위도, 경도, 변수로 구성된 5차원 배열) 데이터 구조로 구성하였다.

2.2.4. 연구 지역 및 평가 방법

전체 훈련 자료에 대한 훈련 기간과 테스트 기간은 개별 기후모델의 Hindcast 기간 및 MME 기간에 따라 각각 달라지는데 Hindcast 기간 전체에 대해 80%는 훈련 기간, 20%는 테스트

트 기간으로 계산된다. 예를 들면, MME의 Hindcast 기간이 1995년부터 2014년이므로 1995년부터 2010년까지 16년 (80%)은 훈련 기간으로, 2011년부터 2014년까지 4년 (20%)은 테스트 기간이 되며 교차 검증을 수행하였다. 연구 지역, 도메인은 CNN 네트워크 규모 등을 고려할 때 교차 검증 (k-fold cross validation)은 시간이 많이 걸리는데, 본 연구에서는 이런 점을 고려하여 4-fold (4겹) 교차 검증을 수행했으며, 훈련자료 구축은 전지구 규모이지만, 이번 연구에서 속도 및 컴퓨팅 계산을 고려하여 딥러닝 훈련 모델에 입력될 때의 입력 및 출력 도메인은 경도 52에서 160, 위도 -50에서 58의 1.5도로 y축 (경도)으로 72개, x축 (위도)로 72의 격자를 가진 동아시아 지역으로 설정하였다. 딥러닝 훈련 모델은 케라스(Keras) 및 텐서플로우 (TensorFlow) 프레임워크를 사용하였으며, CNN의 최적화를 위한 손실함수(loss function)는 평균절대오차 (Mean Square Error, MSE)로 계산되었으며, 측정함수(metrics)로는 일 최고기온과 강수에 대한 오차와 정확도 평가를 위해 코사인 유사도 및 평균평방근오차(Root Mean Square Error, RMSE)를 계산하여 학습하였다. 예를 들면, 예측 일 강수량이 생성되면 예측 시점의 관측 강수량과의 편차를 나타내는 손실함수가 계산된다. 일 최고 및 최저기온 역시 동일한 과정으로 예측된다. 학습 및 예측 결과의 공간적 예측 정확성 및 변동성은 편차상관계수(Anomaly Correlation Coefficient, ACC)를 계산하여 평가하였다 (Eq. [2.4]).

$$ACC_j = \frac{\sum_{i=1}^N \Delta x_{i,j} \Delta y_{i,j}}{\sqrt{\sum_{i=1}^N \Delta x_{i,j}^2} \sqrt{\sum_{i=1}^N \Delta y_{i,j}^2}} \quad [2.4]$$

여기서, j 는 예측 선행기간이고 i 는 격자이다. $\Delta x_{i,j}$ 와 $\Delta y_{i,j}$ 은 격자 i 에서 선행 예측기간 j 의 평균 아노말리(anomaly)이다. N 은 연구지역 전체 격자 개수이다.

또한, 딥러닝 훈련 모델을 전혀 적용하지 않은 S2S 예측 자료의 ACC와 비교하여, 4개 확장 U-Net 모델에 따른 학습 및 예측 결과의 ACC의 개선 정도 (예, 양 혹은 음의 값)를 쉽게 이해하기 위해 평가 지수 (ACC)에 대한 개선 비율(improvement ratio)을 기술 점수(skill score)로 정의하고 평가하였다 (Eq. [2.5]).

$$skill\ score = \frac{ACC_{pred} - ACC_{raw}}{ACC_{raw}} \quad [2.5]$$

여기서, ACC_{pred} 는 훈련 모델에 의해 후보정된 목표변수의 예측 성능을 평가한 지수 (훈련 후 평가된 ACC)이고, ACC_{raw} 는 훈련 모델에 입력하기 전, 즉 학습되지 않은 목표변수의 원시자료 예측 성능을 평가한 지수(훈련 전 평가된 ACC)이다.

2.3. 준지도학습 기법 적용 및 모델의 민감도 분석

2.3.1. 사용된 자료 및 전처리 방법

본 연구에서는 딥러닝 모델에 사용할 입력 변수로 널리 활용되고 있는 MJO 인덱스인 RMM 계산에 사용되는 Outgoing Longwave Radiation (OLR), 850hPa 및 200hPa의 상하층 동서 방향 바람장(U-component winds at 850 and 200 hPa; U850, U200)과 VPM 인덱스에 사용되는 Velocity Potential (VP)을 사용하였다. U850 및 U200 자료는 센터에 구축된 ECMWF ERA5 재분석자료(Hersbach et al., 2020)로부터 계산된 1.5° 공간해상도의 일일 평균 자료를 사용하였다. OLR은 센터에 구축된 NOAA Interpolated OLR 자료를 활용하였다 (Liebmann and Smith, 1996). NOAA OLR 자료가 2.5° 공간해상도로 제공되고 있으므로 사용한 나머지 변수들을 선형 보간 방법으로 평균하여 같은 해상도로 맞추어 활용하였다. VP는 200hPa 동서 및 남북 바람장을 사용하여 계산하였다 (Dawson, 2016). 목적변수는 총 9개의 클래스(class)로 8개의 MJO 위상과 Inactive MJO (Amplitude 1 이하)를 포함한다. 자료 전처리는 기본적으로 Wheeler and Hendon, 2004에서 제시된 방법을 따른다. MJO 이외의 파동 영향을 제거하기 위해 수행된 세 가지의 장기 계절 및 경년 시그널 제거 전처리 절차들은 Figure 2.21과 같다. Climatology mean, first three harmonics가 제거되며, 이전 120일 평균 제거 절차들이 적용된다. 전처리 된 변수들을 기반으로 EOF 분석이 수행되고, PC1 및 PC2가 각각 RMM1 및 RMM2로 사용된다. RMM1 및 RMM2의 원점(origin)으로부터의 거리(즉, amplitude; 강도)가 1 이상인 경우 active MJO 라고 정의하였다. 자료 누수(Data leakage) 문제를 피하고자 모든 전처리 및 정규화 계산은 훈련 기간에 대해서 수행하였으며 훈련 기간에서 산출된 결과를 검증 및 테스트 자료에 적용하여 구축하였다. Figure 2.22는 최종적으로 구축된 각 MJO 위상의 OLR에 대한 아노말리 합성장을 보여준다. 각 위상의 이미지 오른쪽 하단은 위상별로 가용한 이미지 자료의 개수를 보여준다.

딥러닝 모델을 학습하기 위해서 훈련(Training) 자료와 훈련 중에 모델의 과적합(over-fitting)을 방지하기 위해 검증(Validation) 자료를 사용하였고, 학습된 최종 모델을 평가하기 위해서 테스트(Test) 자료가 사용된다. 테스트 자료는 모델을 훈련하는 데 사용되지 않은 완전히 독립적으로 구분된 자료셋이다. 본 연구에서는 자료 분리 방법에 따른 모델 성능을 테스트하기 위해 시간적 분리(Temporal Split) 및 무작위 분리(Random Split) 방법을 사용하였다 (Figure 2.23). 시간적 분리는 전체 시계열 자료를 훈련, 검증, 테스트 자료를 각각 1975-2005 (31년), 2006-2013 (8년), 2014-2022 (9년)으로 나누었고, 무작위 분리 방법은 훈련과 검증 기간에 대해서 랜덤하게 각각 67% 및 33%로 나누어 활용하였다.

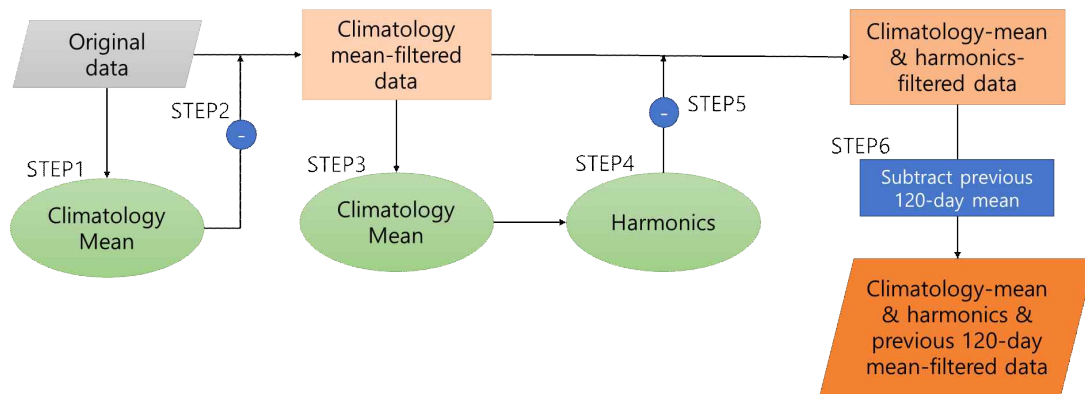


Figure 2.21. Steps of data preprocessing for filtered anomaly data.

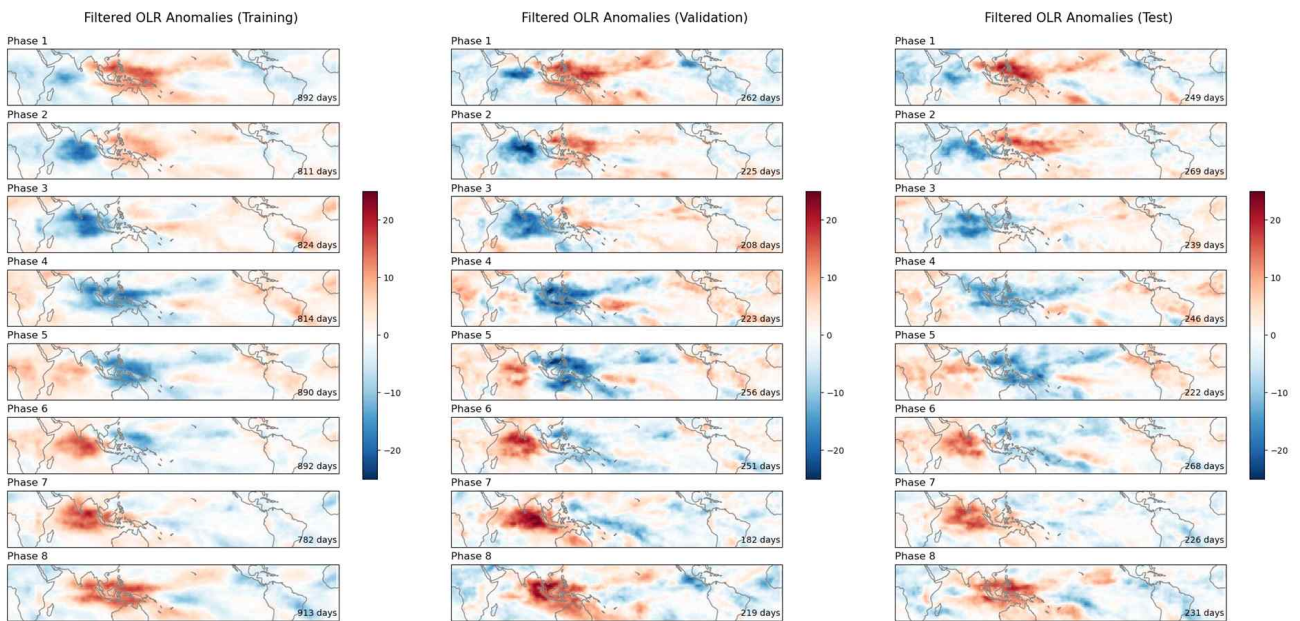


Figure 2.22. Composite images of the filtered OLR anomaly (W/m^2) for each MJO phase for training (left), validation (center), and test (right) datasets, respectively. The number of days available for each phase are given in the bottom right corner. Negative (Positive) anomalies indicate more (less) rain.

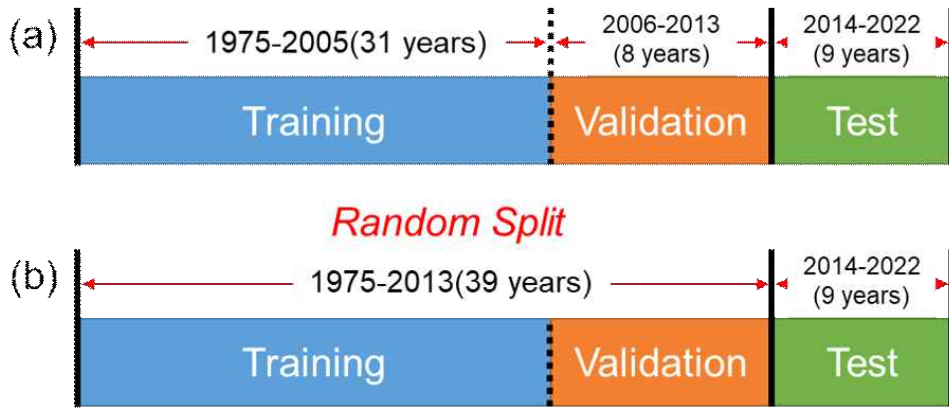


Figure 2.23. Data split strategy. (a) Temporal split, (b) Random split method.

2.3.2. 준지도학습 기준 모델 구축 방법

준지도학습 기준 모델을 선정하기 위해서 다양한 이미지 기반 모델을 평가하였다. Table 2.7은 다양한 변수 조합, 자료 분리 방법, 모델 종류 및 모델의 Fully Connected Layer (FC 층)에 따른 모델의 민감도 실험을 위한 다양한 조건들을 보여준다. 기타 설정으로는 Adam 최적화 함수를 사용하였고 (Kingma and Ba, 2014), 초기 학습률(learning rate)은 $5e-3$ 으로 설정하였으며, 배치 크기(batch size)는 32, 에포크(epoch)는 50으로 설정하였고, 과적합 방지를 위한 조기 종료(Early Stopping)는 10 epoch로 설정하였고, 사용된 모델들은 파이토치(Pytorch)의 토치비전(Torchvision)에서 제공되는 사전 학습(pretrained)된 모델을 사용하였다. 사전 학습된 모델이란 ImageNet 자료셋에 대해 사전 학습된 모델을 말한다. 사전 학습되지 않은 모델을 테스트해 보았지만 사전 학습된 모델이 더 나은 성능을 보여주었기 때문에 이를 활용하였다.

Table 2.7. List of experiments for model sensitivity analysis on input variables, data split methods, model architectures.

Experiments	Conditions	
Input variables	RMM index	1) U850, U200, OLR
	VPM index	2) U850, U200, VP
	UniEOF	OLR only
Data split method	Temporal split	
	Random split	
Image-based backbone AI models	VGG16	
	Resnet18, Resnet50	
	Wide Resnet 50_1, Wide Resnet 101_2	
Variations of fully connected layers	NoADD (no layer added)	
	Layer ver. 1, 2, 3, 4	

Figure 2.24는 본 연구에서 수정 활용한 VGG16 모델의 구조를 보여준다. VGG16은 Simonyan and Zisserman (2015)에 의해 개발된 모델로 단순히 모델 구조의 깊이(depth)를 증가시켰을 때 성능이 어떻게 증가하는지 실험하기 위해 제안된 방법이다. 이러한 접근법은 여러 이미지 분류 문제의 정확도를 높일 수 있었다. 하지만 층이 깊어질수록 gradient vanishing, exploding gradient, 과적합 등의 문제가 발생하게 되어 오히려 깊이가 깊어질수록 성능 저하를 보였다. 또한 VGG 모델은 학습해야 할 파라미터의 수가 많아 학습에 많은 시간이 소요되는 문제가 있다. VGG 모델의 한계점을 개선하기 위해서 He et al. (2016)는 ResNets (Residual Nets) 모델을 제안하였다. Figure 2.25는 ResNet 모델의 Residual 학습 구조를 보여준다. 모델은 skip connection이라 부르는 층의 입력(x)을 출력에 연결하여 잔차(Residual)를 학습하는 개념을 도입하였다. 이러한 접근법은 정확도를 획기적으로 올릴 수 있었고 학습 시간도 현저하게 줄었다. Figure 2.26은 본 연구에서 수정 활용한 ResNet 모델 구조 및 FC 층의 다양한 변형 버전을 보여준다. 층의 개수와 층 내의 노드의 개수를 달리하여 네 가지의 FC 층을 테스트하였다.

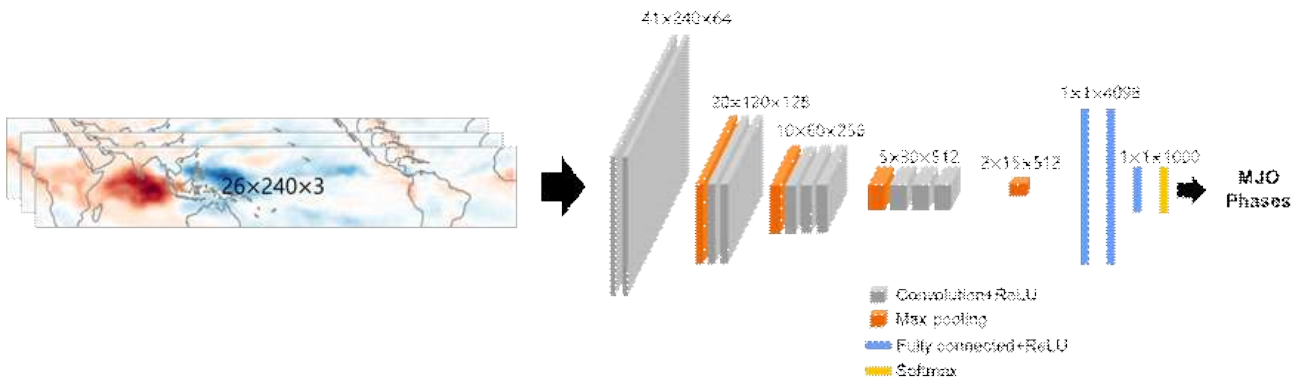


Figure 2.24. Diagram of modified VGG16 model used in this study.

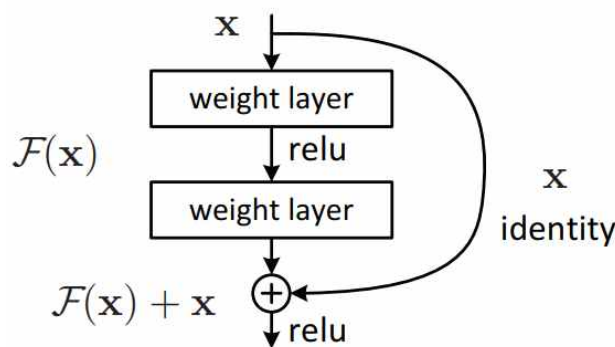


Figure 2.25. Residual learning structure of ResNets model (Source: He et al., 2016).

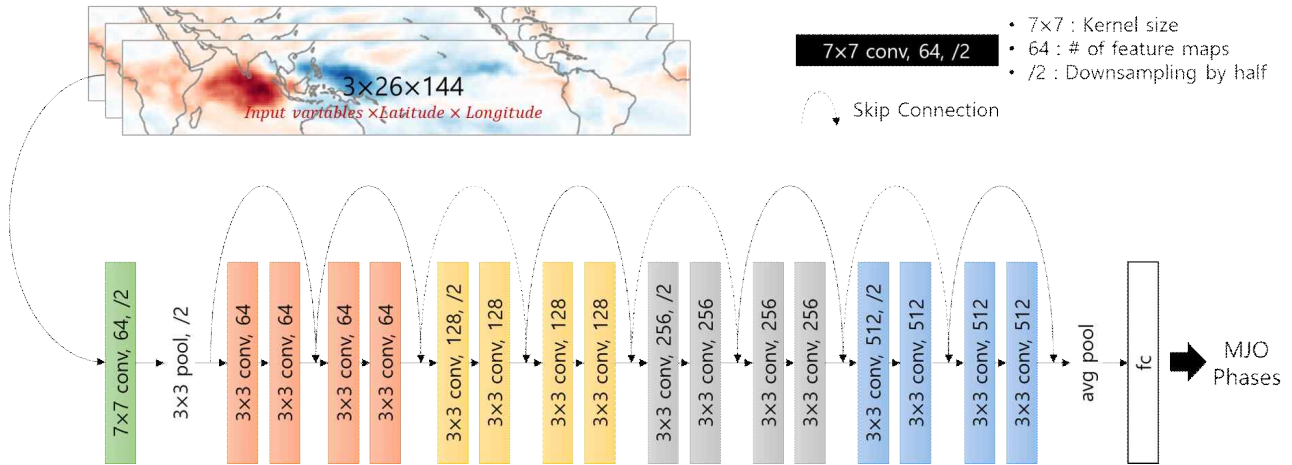


Figure 2.26. Diagram of modified Resnet18 model and different versions of fully connected layers used in this study.

2.3.3. 준지도학습 기반 MJO 위상 분류 모델 구축 및 민감도 테스트 방법

Figure 2.27은 Mean Teacher 준지도학습 방법의 모식도를 보여준다 (Tarvainen and Valpola, 2018). Student 모델은 지도학습, Teacher 모델은 비지도학습 방식으로 학습된다. Student 모델의 출력값은 기존의 라벨 자료와 비교하여 Classification cost (Supervised loss; 지도학습 손실)를 계산한다. Teacher 모델은 지수 이동 평균(EMA; Exponential Moving Average) (Eq. [2.6])을 통해 Student 모델 파라미터의 일정 부분이 넘어가는 방식으로 모델의 파라미터가 업데이트된다(즉, Backpropagation이 수행되지 않음). Teacher 모델은 비라벨링 자료에 대해서 추론(inference)하고 이를 Student 모델과 비교하여 Consistency loss를 계산한다. 전체적인 모델은 Classification cost와 Consistency cost를 함께 고려하여 학습된다.

$$\theta'_t = \alpha \theta'_{t-1} + (1 - \alpha) \theta_t \quad [2.6]$$

Figure 2.28은 Mean Teacher 알고리즘 내에 기존의 자료 증강 방법을 보여준다. 본 연구에서는 MJO의 수평 이동하는 특성에 따라 수평 반전이나 값 자체를 흐리게(blur)하거나 무작위 섭동을 주는 자료 증강 방법들을 테스트하였다.

Table 2.8는 준지도학습 알고리즘의 여러 파라미터 옵션들에 따른 실험 조건들을 보여준다. Batch size 및 Labeled batch size는 각각 전체 자료에 대한 배치 크기와 라벨링된 일부 자료에 대한 배치 크기를 조정한다. Consistency loss weight는 전체 cost 함수의 consistency cost 항에 주는 가중치이다. Consistency ramp-up은 준지도학습 모델을 처음 시작할 때 초기 몇 에폭은 라벨링된 자료로 학습한다. 이는 초기부터 비라벨링 자료의 정보가 사용되면 오히려 학습을 크게 저하시킬 수 있기 때문이다. 본 연구에서는 Mean Teacher 준지도학습 알고리즘을 적용한 결과에 집중한다. Pseudo-labeling 알고리즘을 테스트했으나 크게 성능 향상을 보이지 않았다.

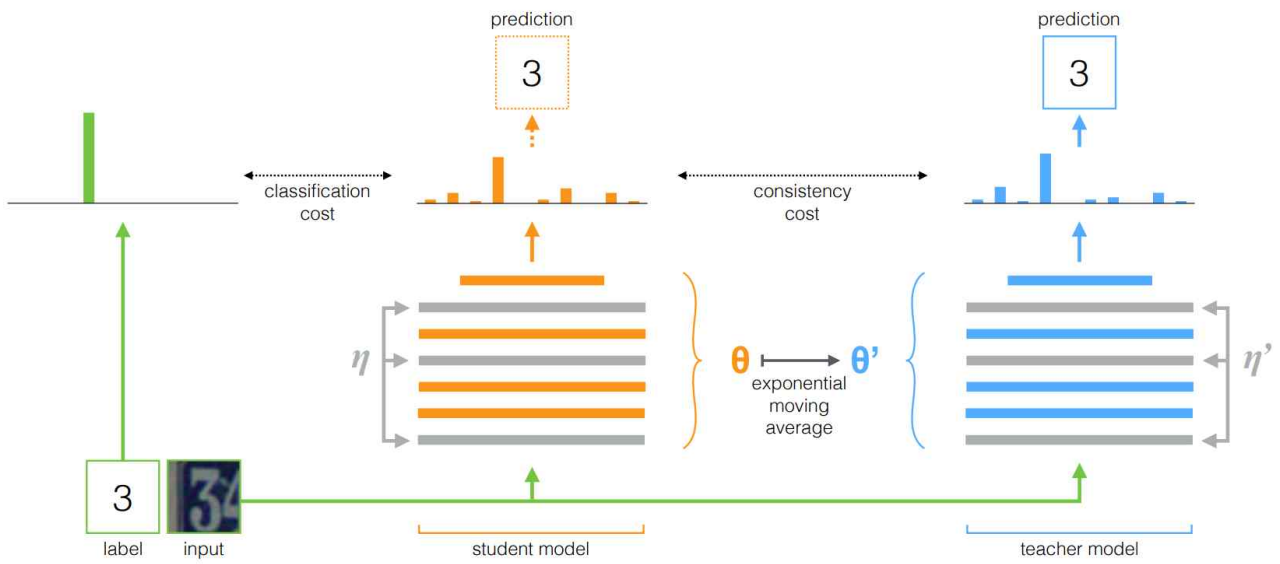


Figure 2.27. Diagram of Mean Teacher algorithm (Source: Tarvainen and Valpola, 2018)

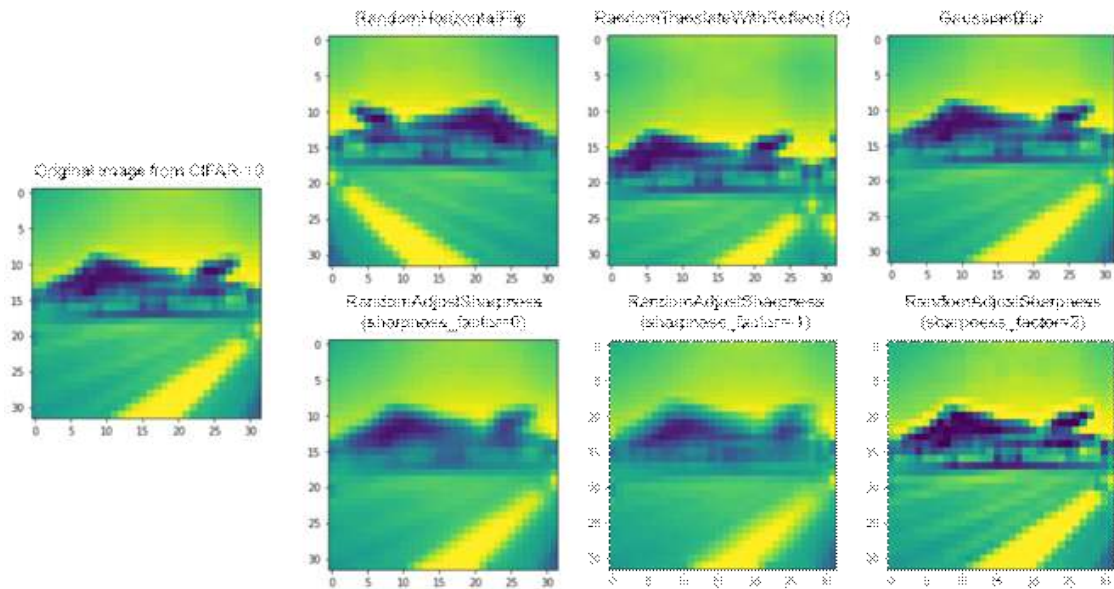


Figure 2.28. Data augmentation methods used in Mean Teacher algorithm.

Table 2.8. List of experiments for sensitivity analysis of SSL-based AI model on model learning strategy, loss weights, and data augmentation methods.

Test	1	2	3	4
Batch Size	32	16	16	32
Labeled Batch Size	16	8	8	16
Consistency loss weight	100	3	3	100
Consistency ramp-up	5	5	5	10
DA method	Random vertical flip	Random vertical flip	No DA	Random vertical flip

2.3.4. MJO 입력장 예측을 위한 시공간 이미지 예측 인공지능 모델 개발 방법

본 연구를 위해서는 필터링된 OLR, 동서 방향 상하층 바람장의 2차원 아노말리 이미지를 활용하였다. 추가로 주변 배경 정보가 제거된 아노말리 정보뿐만 아니라 배경에 대한 정보를 함께 학습하기 위해서 원자료를 입력 변수로 활용하여 테스트하였으나 사용 전후 유의미한 차이가 없었다. 훈련, 검증, 테스트 기간은 각각 1979~2005, 2006~2013, 2014~2022년이다. 7, 30, 60, 120일의 사전 시계열 자료로 다음날(t+1일) 및 7일(t+1~7일) 연속 예측 모델링을 테스트하였다. 추가로 일일 훈련 자료를 주(weekly) 평균하여 4, 8, 12주 전 자료로 1, 4주 예측 모델링 테스트를 수행하였으나 전반적인 패턴을 잘 예측하나 일일 자료에 비해 공간적인 상세함이 매우 떨어졌다. 입력장 예측을 위해서 테스트된 모델은 첫째로 Simpler yet Better Video Prediction (SimVP; Tan et al., 2022) 모델로 시공간 예측 분야의 현 최신 모델(2023.12. 기준)이다. SimVP 모델의 gated SpatioTemporal Attention (gSTA) 모듈을 사용하였다(Figure 2.29). gSTA는 시공간적인 패턴을 효과적으로 학습하는 어텐션(Attention) 모듈이다. Attention이란 처음 적용된 자연어처리 분야에서는 입력 시퀀스의 각 위치가 시퀀스의 다른 모든 입력과 관련성을 계산하도록 하여 서로 멀리 떨어져 있더라도 그 관계성이 계산되어 학습되는 방식이다. 서로 높은 관계를 갖는 단어가 더 높은 값을 가진다. 본 연구에서는 입력 자료의 각 그리드가 다른 모든 그리드와 관련성을 살펴보고 높은 관계를 가지는 부분을 찾는 것으로 이해할 수 있다. 또 다른 하나는 PredRNN v2 모델로 Predictive Recurrent Neural Network인 순환신경망의 한 종류이다(Figure 2.30; Wang et al., 2022). 이는 시공간적인 정보를 동시에 학습함으로써 미래의 이미지를 예측하는 모델이다. 기존의 순환신경망은 단방향으로 학습되는 반면 PredRNN 모델은 수직 수평으로 학습함으로써 시공간적인 정보를 동시에 추출하고 학습한다. SimVP 모델은 PredRNN에 비해서 학습 속도가 현저히 빠른 장점을 가져 여러 가지 실험을 수행하는데 이점을 가진다.

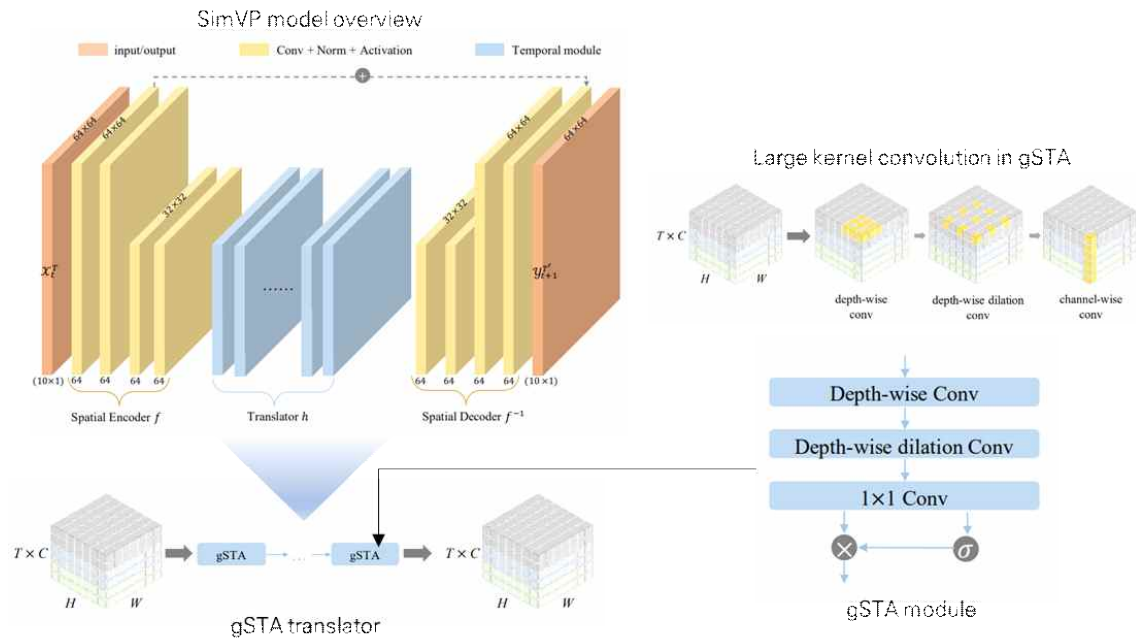


Figure 2.29. Diagrams of SimVP model architecture and its gSTA module (Source: Tan et al., 2022).

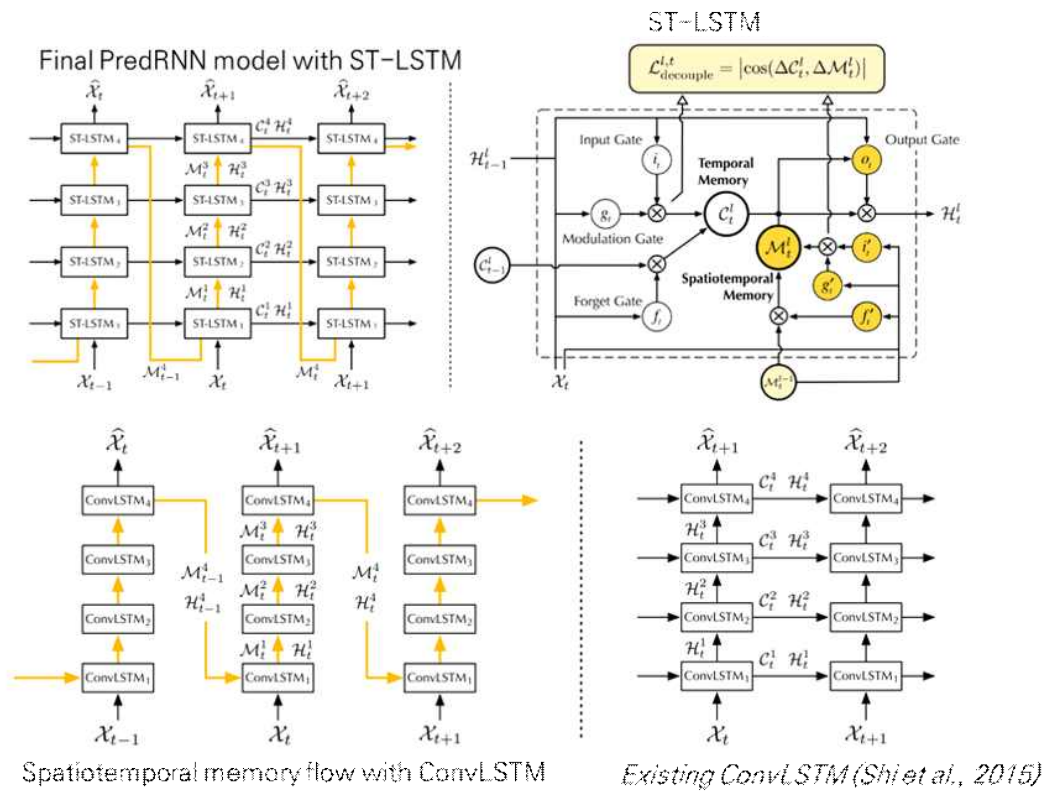


Figure 2.30. Diagrams of PredRNN version 2 (Source: Wang et al., 2022).

2.4. 설명 가능한 인공지능(XAI), 모델 구조 분석 및 모델 개선

2.4.1. XAI 기법 적용을 통한 모델 분석

2.4.1.1. 설명가능한 인공지능 (XAI)

딥러닝 모델은 모델의 해석 가능성에 따라 화이트 박스(white-box), 블랙 박스(black box), 그레이 박스(gray-box)로 구분된다. 화이트 박스는 모델의 내부 로직을 이해하고 결과가 나오는 과정에 대해 쉽게 이해할 수 있다. 블랙박스는 모델의 내부 로직이 복잡하여 결과를 결정하는 과정을 이해할 수 없다. 그레이 박스는 화이트 박스와 블랙박스 중간 모델로 모델의 내부 동작 과정을 부분적으로 분석하고 의사 결정 과정을 이해할 수 있다. 딥러닝 모델의 정확도와 설명 및 해석 가능성은 반비례로 정확도가 높을수록 모델의 설명 및 해석 가능성은 낮아진다. 화이트 박스모델은 의사결정나무(decision trees), 선형/로지스틱 회귀분석(linear/logistic regression), K-means 알고리즘, k-Nearest Neighbors 등이 있으며, 블랙 박스 모델은 앙상블 방법(random forest), 커널 기반 방법, 뉴럴 네트워크(MultiLayer Perceptron, MLP), 순환신경망(Recurrent Neural Network, RNN), 합성곱 신경망(Convolutional Neural Network, CNN), 딥러닝 모델, 트랜스포머(Transformer) 등이 있다. 그레이 박스 모델은 베이저안 네트워크(bayesian network), 퍼지 시스템(fuzzy system) 등이 있다(Ali et al., 2023).

설명 가능한 인공지능(eXplainable Artificial Intelligence, 이하 XAI)는 블랙박스 모델의 의사결정 과정을 이해하기 위해 고안된 방법이다. XAI 기법은 영역(scope), 방법론(methodology), 사용법(usage)에 따라 구분된다(van der Velden et al, 2022). 영역(scope)은 XAI 기법으로 어느 영역에 초점을 두는지에 지역(local)과 전역(global)로 나뉜다. 방법론(methodology)은 입력 자료, 모델 파라미터 등 어디에 초점을 두는지에 따라 역전파(back propagation)와 섭동(perturbation)으로 나뉜다. 역전파 방법은 모델의 아웃풋 예측 레이어에서 입력 레이어까지 역전파 과정에서의 그레이언트(gradients)에 기반을 둔 방법이며, 섭동 방법은 입력자료의 수정을 통해 예측결과가 변화하는 과정을 분석하는 방법이다. 사용법은 모델의 복잡도를 훈련하기 내재적(intrinsic)과 임의의 모델의 학습이 끝난 후에 모델을 분석하는 사후 추론(Post-Hoc)으로 나뉜다.

본 연구에서는 섭동 기반의 occlusion 기법을 이용하여 MJO phase 분류 모델에 적용하였다(Figure 2.31). occlusion 기법은 입력자료의 윈도우(window) 영역을 스트라이드(stride) 간격만큼 이동하면서 윈도우 영역의 자료를 0으로 처리하고 타겟 클래스의 확률값을 원본 입력자료의 결과값과의 차이를 이용하여 확률 변화의 민감도 히트맵(sensitivity heatmap of probability changes)을 작성한다(Ordookhanians et al., 2019). Occlusion은 입력자료의 변화를 통해 XAI 분석을 하는 기법으로 역전파가 필요 없으며 XAI 기법 적용이 어려운 모델에도 적용이 가능한 장점이 있다. Captum 라이브러리(<https://captum.ai/>)를 이용하여 occlusion 기법을 구현하였다. Captum 라이브러리는 PyTorch 라이브러리로 개발된 딥러닝 모델에 적용 가능한 XAI 기법들을 제공한다. Occlusion 기법 외에도 Feature Permutation, Guided GradCam, Shapely Value 등 다양한 XAI 기법들을 제공한다. Tensorflow로 개발된 딥러닝 모델은 iNNvestigate 라이브러리(Alber et al., 2019) 등을 이용하면 XAI 분석이 가능하다.



Figure 2.31. Perturbation-based occlusion in XAI.

2.4.1.2. 손실지형 (Loss surface)

딥러닝 모델 훈련은 매우 복잡한 N차원의 손실지형에서 전역 최소화 지점을 찾는 과정이다. 딥러닝 모델의 손실지형을 묘사할 수 있다면, 손실지형 분석을 통해 모델 수행 작업의 이해도 증진, 모델의 overfitting과 underfitting을 판단하는데 도움을 받을 수 있다(Li et al., 2018). 또한, 손실지형 분석을 통해 단순히 네트워크 깊이를 늘리는 것만으로는 좋은 성능을 얻지 못하고 오히려 결과를 저해할 수 있는 것을 알 수 있으며, 잔여 연결(residual connection)을 사용하면 최적화 과정이 크게 간소화되어 심층 네트워크를 훈련할 수 있다(Ma et al., 2022).

손실지형 분석을 위해 Li et al. (2018)이 공개한 소스코드 (<https://github.com/tomgoldstein/loss-landscape>)를 이용하였다. 손실지형 분석에는 딥러닝 훈련 과정에서 에포크당 모델의 가중치, 바이어스 등의 자료가 필요하다.

2.4.2. 딥러닝 모델 내부 구조 분석 및 개선

2.4.2.1. 실험설계 및 평가방법

딥러닝 모델의 파라미터 분석, 피쳐 시각화 분석, 어텐션 메카니즘 기법을 이용한 모델 구조 변경 등을 통해 딥러닝 모델 내부 구조 분석과 개선을 하고자 한다. 피쳐 붕괴(feature collapse) 분석을 통해 모델의 성능 저하를 일으키는 구간을 분석하고 모델의 구조 변경을 통해 예측성 향상 가능성을 보고자 한다. 이를 위해 이성규(2023)이 제안한 딥러닝 기반 1개월 T2M 예측모델을 이용하고자 한다. 실험은 파라미터 조정을 통한 모델의 예측성 분석과 모델 내부 구조 변경에 따른 예측성 분석으로 나누어 설계하였다. 모델 학습에 사용된 데이터셋은 1965년~2022년 기간의 총 58년의 ERA5 재분석자료 중 T2M 변수를 이용하였으며 구축하였다(Figure 2.32). 실험 데이터셋은 학습(training)/검증(validation), 테스트(test) 데이터로 구성되어 있으며, 학습/검증 자료는 기본 데이터셋은 1986~2015년 총 30년 기간의 데이터를 사용하였다. 데이터 확장에 따른 모델의 정확도 분석을 위해 1965년까지 데이터 기간을 확장하고 구축하였다. 학습/검증 데이터는 해당 기간의 자료를 섞은 후에 7 대 3의 비율로 나뉘어 구축하였다.

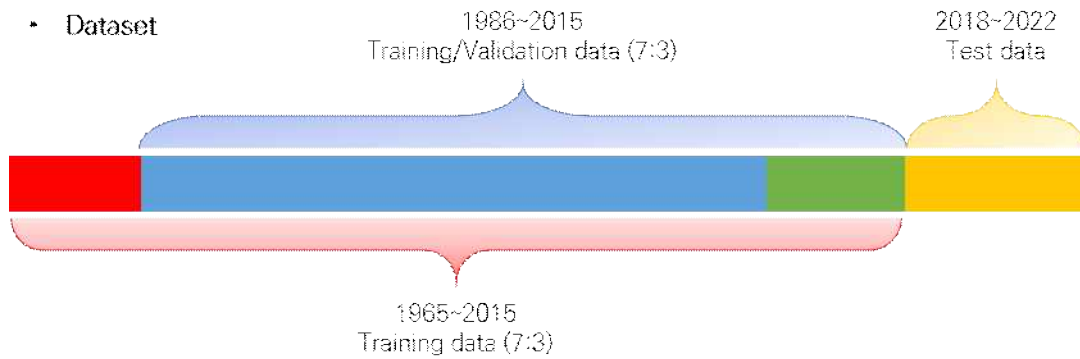


Figure 2.32. Training dataset.

딥러닝 기반 1개월 T2M 예측모델은 U-NET 아키텍처를 기반으로 구축되었으며, 피쳐의 특징 추출을 위해 일반적으로 사용되는 2개의 2차원 합성곱 레이어(2D convolutional layer), 활성화 레이어(activation layer), 배치 정규화(batch normalization)의 결합구조가 아닌 2개의 2차원 합성곱 레이어와 어텐션 레이어로 구성된 CCA 블록(conv2d-conv2d-attention)을 이용하였다 (Figure 2.33). 또한, 입력자료는 ERA5 재분석자료와 ERA5 재분석자료를 이용하여 생산한 15년 climatology를 이용하였다. 모델은 tensorflow를 이용하여 구현하였다(이성규, 2023).

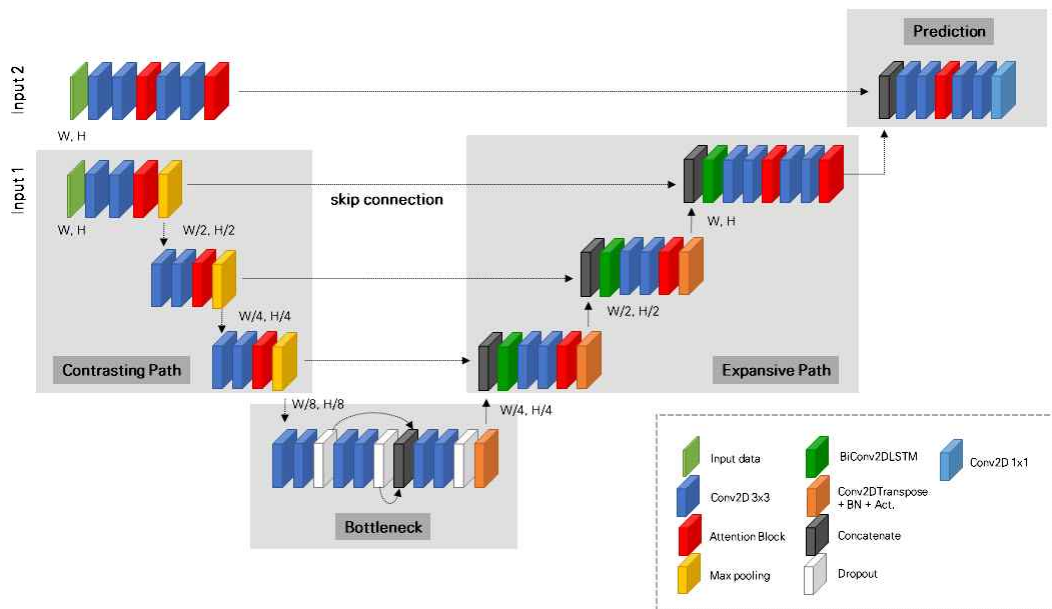


Figure 2.33. Architecture of deep learning-based 1-month T2M prediction model.

실험을 위해 기존 딥러닝 기반 1개월 기온예측 모델의 구조를 어텐션 메커니즘, 학습 데이터셋, 레이어 필터 수 확장 등을 고려하여 설계하였다. 어텐션 메커니즘을 중심으로 SE, CBAM, 그리고 학습데이터 확장, 레이어 필터 수 확장을 반영하였다. 모델은 SE, SE+BIGDATA,

SE+BIGDATA+FLT2, CBAM, CBAM+BIGDATA, CBAM+BIGDATA+FLT2이다.

모델은 과거 14일 자료를 이용하여 1일 예측하는 모델로 구성하였다. 활성화 함수는 Relu, 옵티마이저는 ADAM, 학습률은 $1e-4$, 손실함수는 MSE (mean square error), 커널 사이즈는 3×3 , 레이어 풀링은 MaxPooling, 드롭아웃률은 0.5를 이용하였다.

모델의 평가는 3주와 4주의 예측자료에서 남한지역의 5개 격자를 평균하여 계산한 확률 예측값과 ASOS(Automatic Synoptic Observing System) 관측자료로 계산한 3분위값(above normal, near normal, below normal)을 HSS(Heidke Skill Score)를 이용하여 평가하였다. 딥러닝 기반 1개월 기온예측 모델을 이용하여 rolling prediction method로 ECMWF S2S 예보자료의 1-2주 자료를 입력자료로 이용하여 3주와 4주를 재예측(reforecast)하였다. 재예측은 forecast, hindcast 자료 모두에 적용하였다. 학습자료에서 2016년과 2017년은 ASOS 관측자료로 계산한 3분위 자료가 없어서 제외하였다(Figure 2.32).

딥러닝 기반 1개월 T2M 예측모델은 딥러닝 모델 내부 구조 분석 및 개선 연구에 활용하고자 하며, Li et al.(2018)이 제안한 손실지형 분석에는 Tensorflow로 작성된 모델을 pytorch로 재구축하고 이용하였다. Pytorch에서는 Bi-Directional ConvLSTM2D 레이어를 제공하지 않아서 별도로 구현하여 이용하였다.

2.4.2.2. 모델 파라미터

딥러닝 모델은 파라미터(parameter)와 하이퍼파라미터(hyperparameter)로 구성된다. 파라미터는 모델 학습에 사용된 데이터로부터 모델 설정을 위해 추정된 값을 말하며, 하이퍼파라미터는 딥러닝 모델링을 위해 사용자가 설정해주는 값을 말한다. 파라미터는 모델의 weight, bias 등이 있으며, 하이퍼파라미터는 배치크기(batch size), 옵티마이저(optimizer), 학습률(learning rate), 딥러닝 모델의 레이어 설정값(e.g., 레이어의 필터 크기 등)이 있다. 배치크기는 하드웨어의 한계로 인해 학습자료를 한 번에 학습할 수 없어 배치크기만큼 한 번에 학습 가능한 데이터의 샘플 수를 의미한다. 배치크기는 손실률이 최소화 되는 전역 최솟값(global minima)를 찾는 딥러닝 문제에서 배치크기가 적을수록 손실률이 최소화되는 지점을 찾는 데 유리하지만 학습시간이 늘어나는 문제가 있다.

Figure 2.34는 딥러닝 기반 T2M 예측모델(Figure 2.3)의 예측 레이어(prediction layer)의 마지막 2개 레이어의 활성화맵이다. 가장 오른쪽 아래에 있는 활성화맵은 최종 예측결과이다. 활성화맵 시각화를 통해 모델의 내부구조를 보면 유용한 피처가 많을수록 예측결과에 영향을 줄 수 있음을 파악하였다. 따라서, 하이퍼파라미터 중 배치크기, 레이어 필터 조정을 통해 모델의 개선여부를 분석하고자 한다. 또한, 학습자료 늘려서 학습데이터의 증가에 따른 모델의 개선여부를 분석하고자 한다. 학습자료의 테스트 기간은 2018~2022년으로 같은 기간의 자료를 이용하여 모델을 평가하였다(Figure 2.32). 2016년과 2017년은 테스트 기간에서 제외하였다.

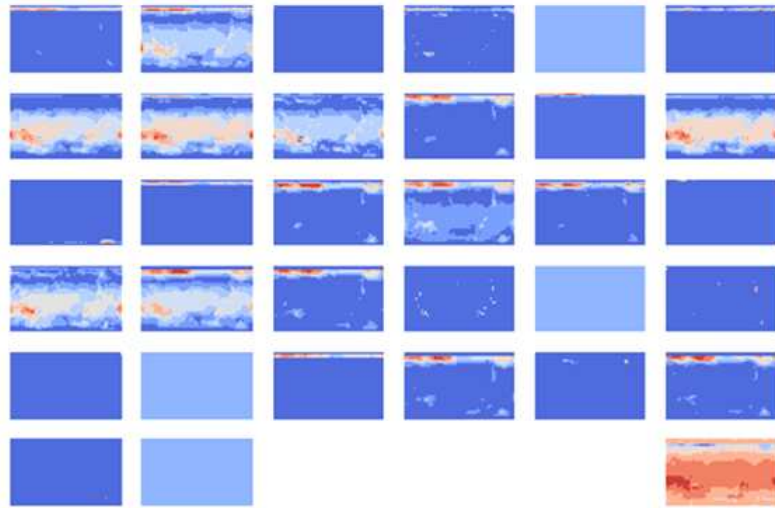


Figure 2.34. Visualization of activation map.

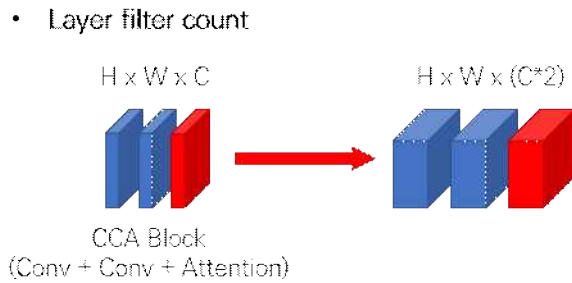


Figure 2.35. Example of scaling the number of layer filters.

2.4.2.3. 어텐션 메카니즘

CNN 기반의 딥러닝 모델은 합성곱 레이어(convolutional layer), 활성화 레이어(activation layer), 배치 정규화(batch normalization), 드롭아웃(dropout) 등 N차원 행렬자료를 입력자료로 받아서 유용한 피처를 추출하는 과정을 통해 분류(classification), 탐지(detection), 예측(prediction) 등 목적에 맞게 개발이 가능하다. 합성곱 레이어는 필터(filter)를 이용하여 이전 입력자료에서 피처를 추출하는 역할을 한다. 어텐션 메커니즘은 피처의 정보 중에 학습에 유용한 정보에 가중치(weight)를 부여하는 방법을 통해 좀 더 유용한 정보에 집중하는 기법이다. 딥러닝 모델에서 주로 사용되는 어텐션 메커니즘은 공간(spatial) 어텐션, 채널(channel) 어텐션 등이 있다. 본 연구에서는 대표적인 채널 어텐션 기법인 SE 네트워크(squeeze-and-excitation network)와 채널 어텐션과 공간 어텐션을 결합한 CBAM (Convolutional Block Attention Module)을 이용하여 딥러닝 모델의 구조를 변경하고 모델의 예측성능을 분석하였다(Figure 2.36 및 2.37).

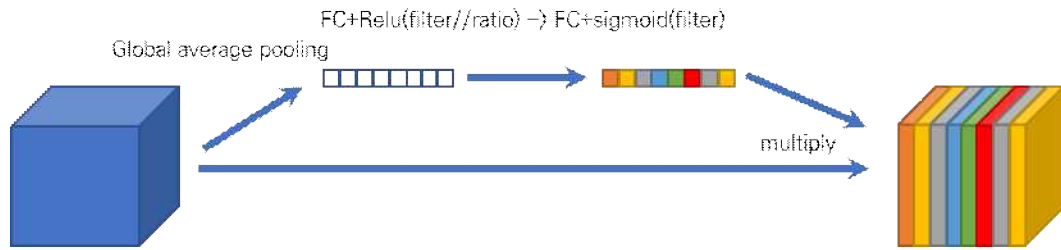


Figure 2.36. The overview of SE block (adapted from Hu et al. (2020))

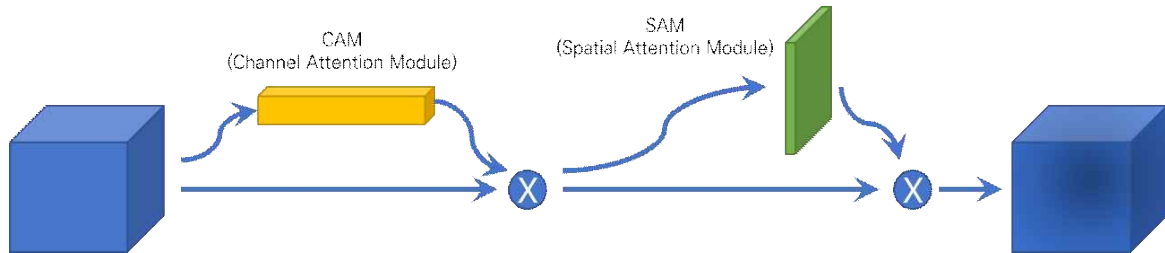


Figure 2.37. The overview of CBAM (adapted from Woo et al. (2018)).

2.4.2.4. 피쳐붕괴 분석

합성곱 신경망(Convolutional neural network, CNN) 구조의 딥러닝 모델은 필터(filter)를 통과하는 과정에서 뉴런(neuron)이 죽거나(dead) 포화(saturation)하는 현상에 의해 피쳐붕괴(feature collapse) 현상이 발생한다(Figure 2.38). 모델의 예측성을 높이기 위해서는 입력자료가 convolutional layer, activation layer 등을 통과 후 죽거나 포화되는 뉴런이 적을수록 모델의 예측성능을 높일 수 있다. 즉, 딥러닝 모델이 학습과정에서 유용한 정보를 생성하는데 방해가 되며 결과적으로는 예측성능을 떨어지는 요인이 된다.

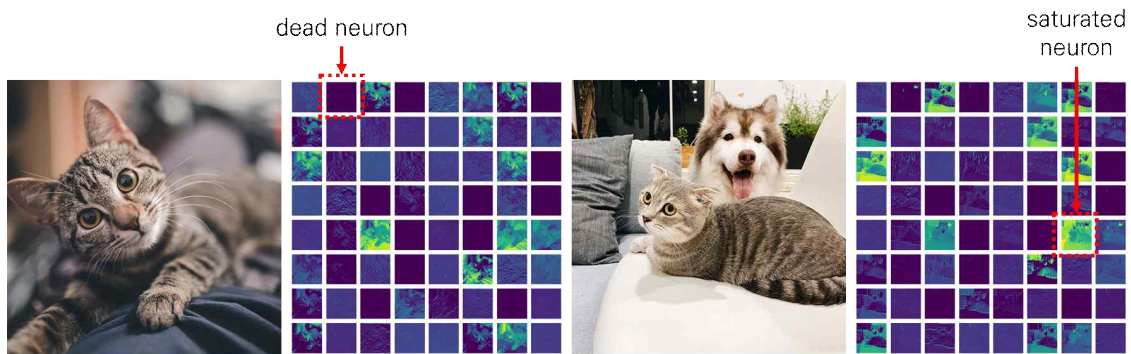


Figure 2.38. Visualization of activation map (input images obtained from <https://github.com/ismailuddin/gradcam-tensorflow-2>).

피쳐붕괴 현상을 Activation layer의 시각화를 통해 확인하는 것은 비효율적인 방법으로,

본 연구에서는 Woo et al.(2023)이 CNN 레이어의 성능 분석에 사용한 feature pair-wise cosine distance (Eq. [2.7])를 이용하여 딥러닝 모델의 주요 구간의 피쳐붕괴 분석을 하였다. cosine distance는 자연어처리 분야에서 문서 분류에서도 사용된다.

$$\frac{1}{C^2} \sum_i^C \sum_j^C \frac{1 - \cos(X_{F_i}, Y_{F_j})}{2} \quad [2.7]$$

여기서, C 는 채널(channel) 수, F_i 와 F_j 는 convolutional layer, activation layer 등을 통과한 i 와 j 번째 피쳐(feature), i 와 j 는 전체 채널의 순서를 의미한다. 코사인 거리가 길수록 붕괴현상이 적고 짧을수록 붕괴현상이 심한 것을 알 수 있다.

3. 결과

3.1. 자료 확장에 따른 모델 성능의 민감도 분석

3.1.1. 강수량 자료 학습결과

자료확장에 따른 모델 성능의 정확도를 빠르게 평가하기 위하여 정확도 개선이 어려운 강수량에 대해서 Attention U-Net모델의 입력자료로 ECMWF의 강수량자료와 관측자료인 GPM 강수량자료를 3D입력자료(20년*7일,위도,경도,변수)로 구성하고, 이 중에서 15년치는 학습자료로 나머지 5년치는 테스트 자료로 훈련셋을 구성하였다. Attention U-Net모델에서 convLSTM2D함수를 이용하여 합성곱으로 레이어를 구성하고 activation은 relu 함수를 사용하여 모델을 구성하였다. 손실함수는 MSE (Mean Square Error) 함수로, optimizer 함수는 Adam 함수로 이용하고 batch_normalization을 적용하지 않고 최종적으로 리드타임 3주차 7일 강수량자료가 나오도록 ERA5자료를 두고 학습을 진행하였다. Figure 3.1에서 보듯이 두 개의 변수만으로 학습한 결과는 극치가 평균화되어 잘 표현되지 못하는 것을 볼 수 있다. 강수량의 경우에는 Attention U-Net모델에서 강수 특성이 리드타임 3주차까지 전달되지 않는 것을 볼 수 있다.

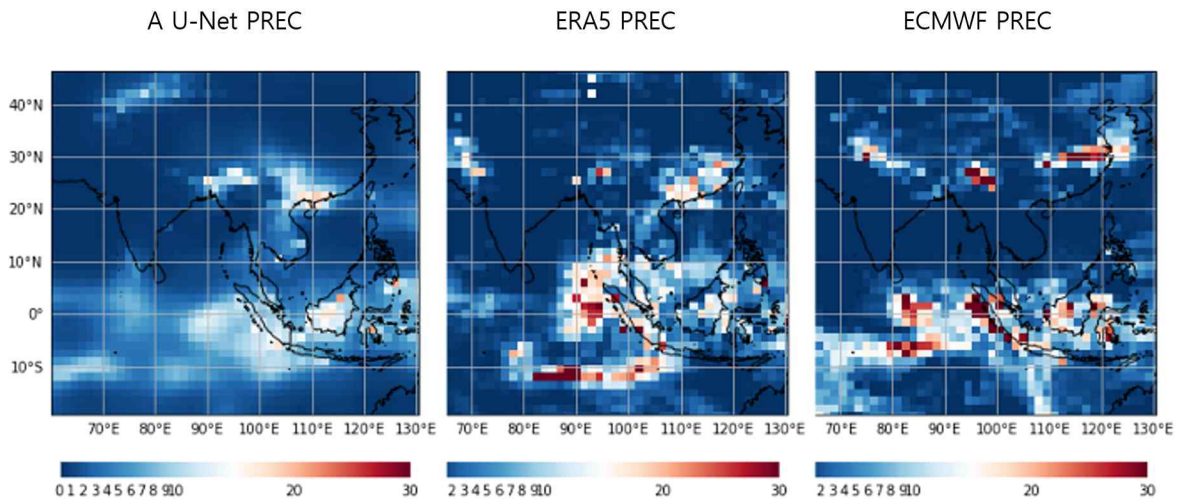


Figure 3.1. Comparison of the result of Attention U-Net and ECMWF model.

3.1.2. Filter 기법에 따른 민감도 분석 결과

관측자료인 NDVI, SST와 ECMWF의 모든 변수를 ERA5와의 상관성을 조사하였다. 각각의 변수별과 ERA5의 최고온도에 상관성 중에서 가장 상관성이 높은 변수를 중심으로 입력자료를 구성하였다(Table 3.1). 각각의 변수는 TMAX, TMIN, T2M, Z200, Z500, Q850, Q700, MSL, PRE, NDVI (Obs), SST (Obs)의 순서대로 상관성이 높았다. 상관성이 높은 변수를 선택하고 각각의 변수중에서 상관성이 높은 순서대로 Attention U-Net모델에 입력자료로 구성하여 학습을 진행하였다.

Table 3.1. Index and variables of subsets for filter method.

Filter 입력변수	
A	TMAX(ECMWF)
B	TMAX,TMIN
C	TMAX,TMIN,T2M
D	TMAX,TMIN,T2M,Z200
E	TMAX,TMIN,T2M,Z200,Z500
F	TMAX,TMIN,T2M,Z200,Z500,Q850
G	TMAX,TMIN,T2M,Z200,Z500,Q850,Q700
H	TMAX,TMIN,T2M,Z200,Z500,Q850,Q700,MSL
I	TMAX,TMIN,T2M,Z200,Z500,Q850,Q700,MSLPRE
J	TMAX,TMIN,T2M,Z200,Z500,Q850,Q700,MSLPRE,NDVI(OBS)
K	TMAX,TMIN,T2M,Z200,Z500,Q850,Q700,MSLPRE,NDVI(OBS),SST(OBS)

Filter 기법 분석결과 Table 3.1에서 보듯이 A의 카테고리인 ECMWF 계절내 모델의 TMAX를 단독으로 사용하여 ERA5와의 리드타임 3주차 ACC값이 다른 모든 상관성이 높은 변수별로 학습한 최고온도와 ERA5의 최고온도가 ACC보다 높았다. ECMWF의 TMAX값의 ACC값이 0.31로 ACC 0.2대의 상관성 높은 변수들의 학습결과보다 정확도가 높았는데 ECMWF모델 자체의 예측성능이 좋은 것으로 판단된다(Figure 3.2).

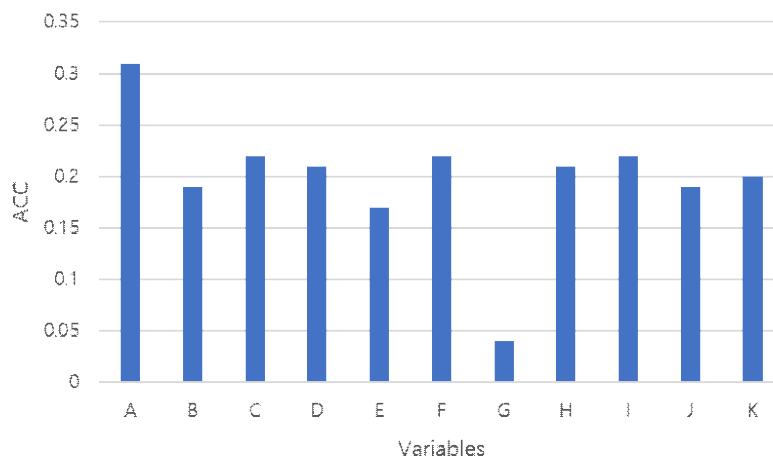
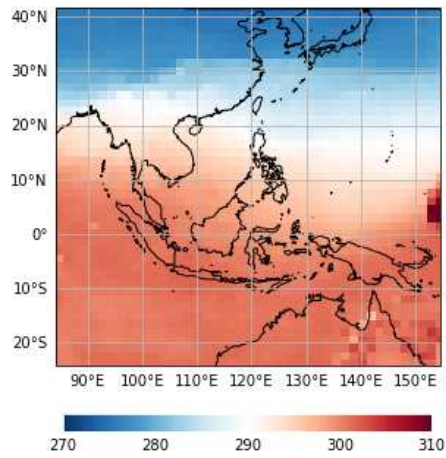


Figure 3.2. Trained results obtained from variables based on filter method.

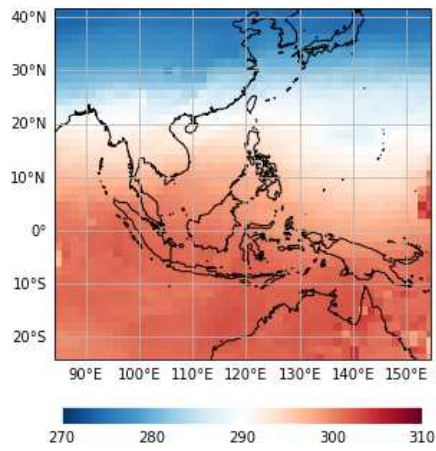
3.1.3. Wrapper 기법에 따른 민감도 분석 결과

Wrapper기법의 결과를 분석하기 위하여 관측자료 NDVI, SST을 기반으로 ECMWF모델 변수들로 이용하여 Attention U-Net모델을 산출한 최고온도자료와 ERA5 최고온도 및 ECMWF단독 변수 최고온도 자료를 가시화하여 산출하였다(Figure 3.3).

A-Unet TMAX



ERA5 TMAX



ECMWF TMAX

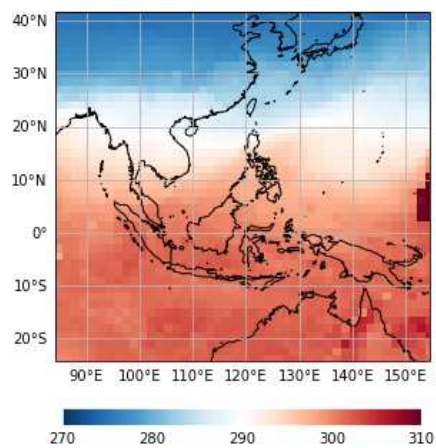


Figure 3.3. Visualization of the result of Attention U-Net by wrapper method.

위성자료인 NDVI (OBS), SST (OBS)와 ECMWF 모델의 OLR, MSL, TMAX변수의 조합으로 입력자료를 구성하고 Attention U-Net모델의 학습한 최고온도를 ERA5의 최고온도와 비교한 결과가 ECMWF모델의 최고온도 단독으로 ERA5와 비교한 결과보다 ACC가 높았다는 것을 볼 수 있다(Figure 3.4). 이 외에 관측자료 NDVI(OBS), SST(OBS)와 ECMWF모델의 OLR, MSL, TMAX, T2M, SST와의 모든 조합의 민감도 분석에서 단독의 ECMWF모델의 최고온도보다 ACC가 높은 구성을 보여주지 못한다. 여전히 3주차에 대한 딥러닝 모델의 학습의 어려움을 보여준다.

	입력자료
A	TMAX(ECMWF)
B	NDVI(obs),OLR,MSL
C	NDVI(obs),OLR,MSL,TMAX
D	NDVI(obs),SST(obs),OLR,MSL,TMAX
E	NDVI(obs),SST(obs),OLR,MSL,TMAX,T2M
F	NDVI(obs),SST(obs),OLR,MSL,TMAX,T2M,SST
G(Model)	OLR,MSL,TMAX,T2M,SST
H(Obsevation)	NDVI,SST

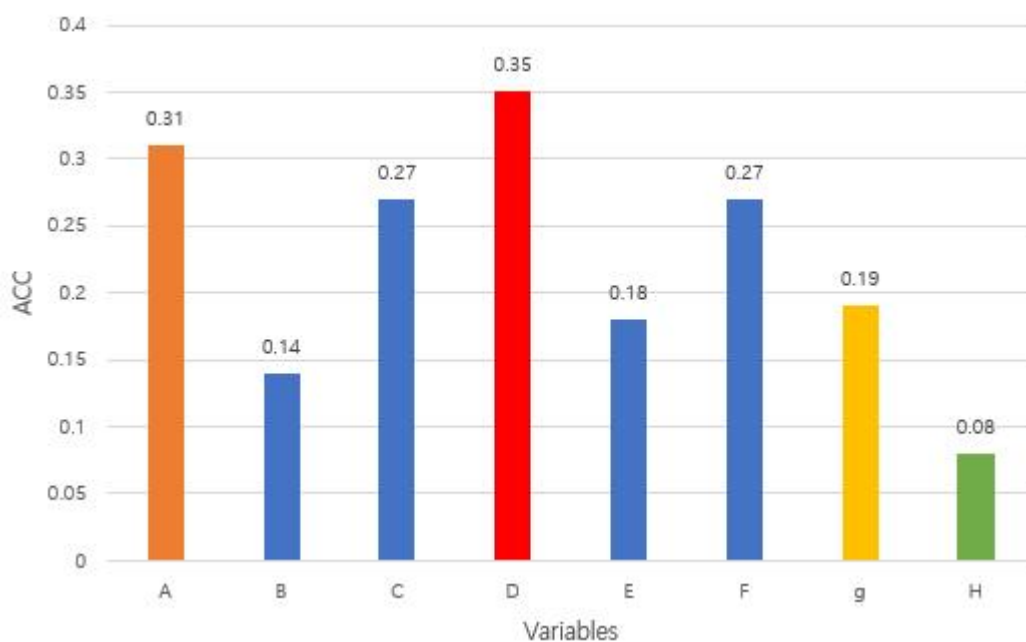


Figure 3.4. Comparison of the result of Attention U-Net by Wrapper method.

NDVI(OBS), SST(OBS), OLR, MSL, TMAX, T2M, SST 변수의 조합으로 학습자료를 구성하여 학습한 결과 카테고리 D인 NDVI(OBS), SST(OBS), OLR, MSL, TMAX의 입력변수일 때 가장 ACC가 높다는 것을 확인하였다. 더불어 Q700, Q850, U50, U200, U850, V200, W500, Z200, Z500의 변수를 덧붙여 학습결과를 비교하고자 하였다. 따라서 최종입력자료를 NDVI(OBS), SST(OBS), OLR, MSL, TMAX, T2M, SST, Q700, Q850, U50, U200, U850, V200, W500, Z200, Z500 입력자료를 구성하고 모든 변수에 대한 학습을 진행하였다(Table 3.2).

Table 3.2. Index and variables of subsets for wrapper method.

Wrapper 입력변수	
A	TMAX(ECMWF)
B	NDVI(OBS),OLR,MSL
C	NDVI(OBS),OLR,MSL,TMAX
D	NDVI(OBS),SST(OBS),OLR,MSL,TMAX
E	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M
F	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,TMAX,T2M,SST
G	OLR,MSL,TMAX,SST
H	NDVI(OBS),SST(OBS)
I	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700
J	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850
K	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50
M	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200
N	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200 U850
O	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200 U850,V200
P	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200 U850,V200,W500
Q	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200 U850,V200,W500,Z200
R	NDVI(OBS),SST(OBS),OLR,MSL,TMAX,T2M,SST,Q700,Q850,U50,U200 U850,V200,W500,Z200,Z500

모든 변수의 각각의 A에서 R까지 카테고리에서 ECMWF 단독 최고온도보다 ACC가 높은 경우는 카테고리 D인 NDVI(OBS), SST(OBS), OLR, MSL, TMAX 변수의 조합으로만 학습결과가 높았다(Figure 3.5).

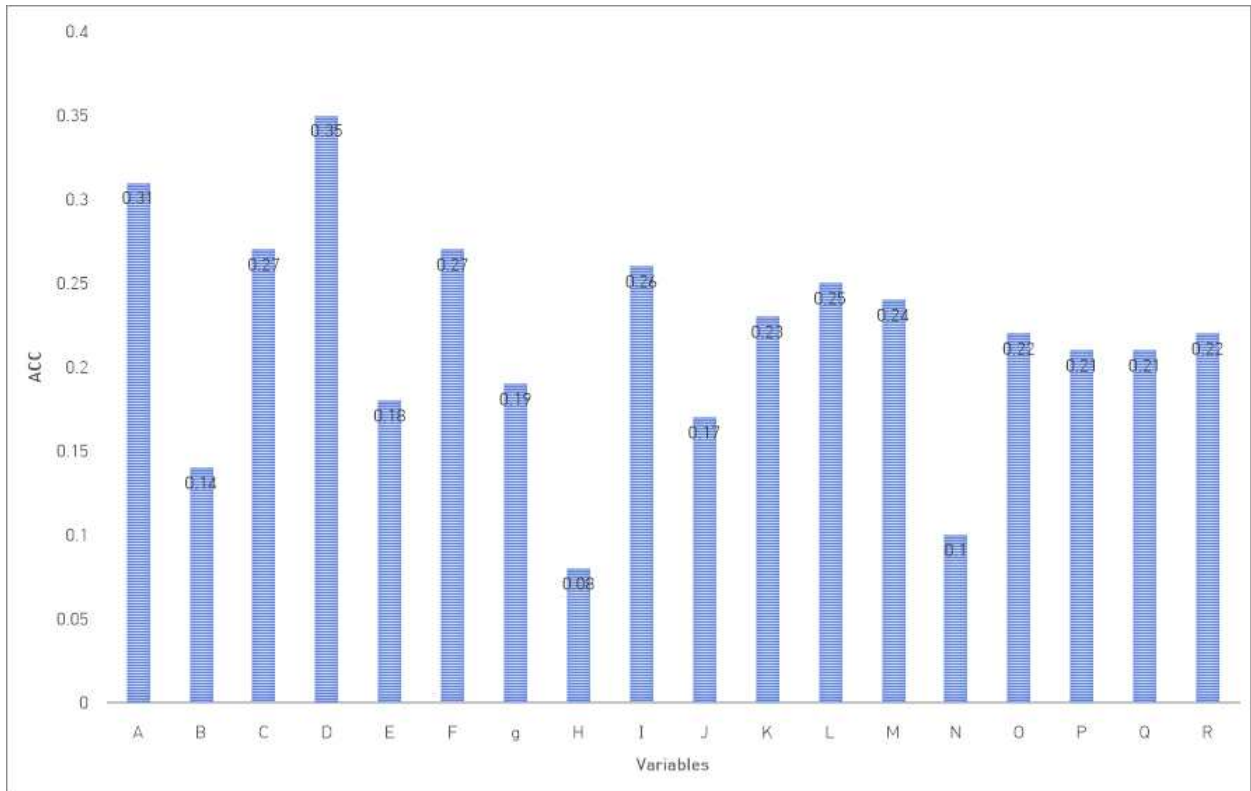


Figure 3.5. Comparison of the result of all variables by Wrapper method.

3.1.4. Embedded 기법에 따른 민감도 분석 결과

Embedded 기법은 ECMWF의 모든 변수를 이용하여 리드타임 3주차 7일에 대한 ERA5의 최고온도에 대해서 회귀식을 이용하여 회귀계수를 산출하였다. 회귀계수중에서 가장 계수가 높은 순서대로 변수를 추출하고 선택하였다. 관측자료인 NDVI와 SST는 관측자료가 없는 지역의 NaN값으로 본 연구에서는 변수에서 제외하였다(Figure 3.6).

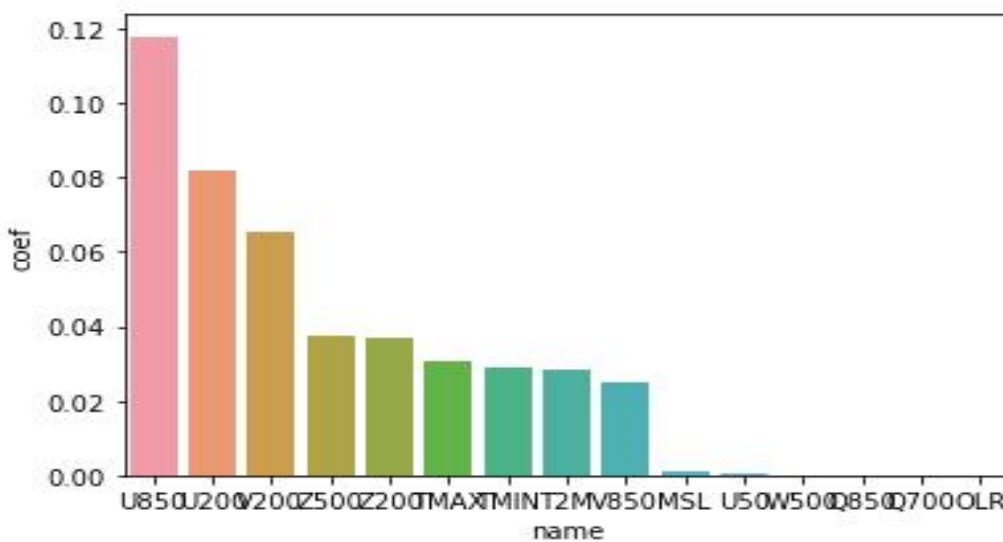


Figure 3.6. Correlation coefficient of the variables from ECMWF model by embedded method.

회귀계수 중요도에서 보면 변수중에서 U850, U200, V200, Z500, Z200, TMAX, TMIN, T2M, V850의 변수들의 중요도가 높아서 선택을 하였고 중요 변수들의 모든 조합에 따른 학습 결과를 산출하여 비교하였다(Table 3.3).

Table 3.3. Index and variables of subsets for embedded method.

Embedded 입력변수	
A	TMAX(ECMWF)
B	U850,U200
C	U850,U200,V200
D	U850,U200,V200,Z500
E	U850,U200,V200,Z500,Z200
F	U850,U200,V200,Z500,Z200,TMAX
G	U850,U200,V200,Z500,Z200,TMAX,TMIN
H	U850,U200,V200,Z500,Z200,TMAX,TMIN,T2M
I	U850,U200,V200,Z500,Z200,TMAX,TMIN,T2M,V850

비교분석 결과 Filter기법과 같이 ECMWF의 모델 단독 최고온도 ACC값이 가장 좋음을 알 수 있다. 이의 결과는 변수선택과 더불어 딥러닝 모델의 기후분야의 지속적인 발전이 필요한 것으로 판단된다(Figure 3.7).

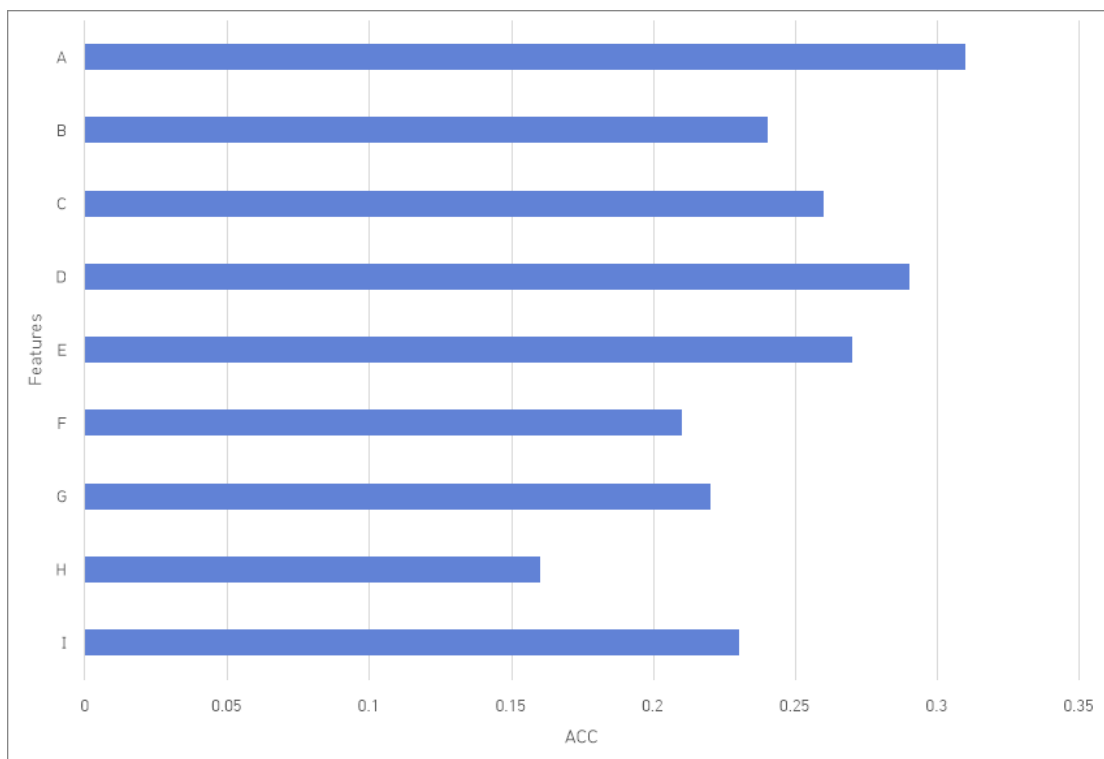


Figure 3.7. Comparison of the result of all variables by Embedded method.

3.2. 훈련기법, 초매개변수 조정 및 불확실성 분석을 통한 아키텍처 개선

3.2.1. 확장 U-Net 모델 구축

기존 U-Net은 22년 연구에서 사용한 기준 모델인데, 이것을 바탕으로 2.2.1 세션에서 언급한 Attention Gate (AG)를 가중치 및 특징 정보 전달 방법으로 추가하여 Attention U-Net을 코딩하였다. 특히, 기존 U-Net에서 UpSampling2D를 Conv2DTranspose로 변경하였다. Upsampling2D는 이미지의 간단한 확대만 수행하지만, Conv2DTranspose는 커널을 학습하는 일반적인 합성곱 연산(convolution operation), 즉 conv2d와 동일한 연산을 수행하기 때문이다. 즉, 확대만 하고 정보가 없는 Upsampling2D 대신 연산하고 가중치 정보를 유지하고 있는 Conv2DTranspose로 수정했다. 또한, 기존 U-Net의 AveragePooling2D 대신 MaxPooling2D로 수정하였다. AveragePooling2D는 입력 데이터의 평균값을 구하고 MaxPooling2D는 입력 데이터의 최대값을 구하는데, AveragePooling2D는 MaxPooling2D보다 더 많은 정보를 유지하지만, MaxPooling2D는 더욱 강력한 특징을 추출할 수 있어서, MaxPooling2D가 더 많이 사용되어서 AveragePooling2D 대신 MaxPooling2D로 수정하였다. Residual U-Net은 기존 U-Net의 합성곱 레이어 (convolutional layer)와 bottle neck의 합성곱 레이어를 모두 residual block으로 수정하였다. 그 효과는 skip-connection이 Figure 2.2 및 Figure 2.5와 같이 bottle neck에서 한번 더 skip-connection 되는 것으로, 입력값을 그대로 출력값에 더해주는 방식이기 때문에 합성곱 레이어로 축소되어 오다가 bottle neck에서 입력값을 그대로 받아서 다시 팽창으로 전달하기 때문에 학습 및 예측 성능이 기존 U-Net보다 우수할 것으로 예상되었는데 결과는 비교적 맞았다. Attention based-on Residual U-Net은 기존 U-Net의 합성곱 레이어로 대신 기 작성된 Residual U-Net의 residual block으로 수정하고 decoding, 즉 팽창 부분에 Attention Gate를 추가, 즉 Attention U-Net과 Residual U-Net을 결합하여 구축을 완료하였다.

3.2.2. 그리드 탐색: 초매개변수 조합 학습 성능

초매개변수 (hyperparameter)는 사용자가 모델 학습을 위해 세부적으로 설정할 수 있는 변수로 쉽게 학습 규율이라고도 이해할 수 있다. 학습 성능을 결정하는 초매개변수 선정 및 반복 학습으로부터 모델 성능을 높이는 과정을 초매개변수 최적화(hyperparameter optimization) 과정인데, 모델에서 가장 적합한 초매개변수를 찾기 위해 초매개변수의 다양한 조합을 순차적으로 입력한 후, 가장 높은 성능을 보이는 초매개변수들을 찾는 탐색 방법이라고 2.2.2 세션에서 언급하였다. 그리드 탐색(Grid Search)은 초매개변수의 범위와 간격을 정해 놓고 각 범위의 경우에 대해 모두 대입해 최적의 경우의 수를 찾기 때문에 초매개변수의 수가 추가됨에 따라 계산량이 기하 급수적으로 증가하는 단점이 있지만, 주어진 조건 (예, 일정한 값들을 제시)에 대해서는 놓치지 않고 평가하고 구현이 매우 쉽다는 장점이 있다. 그래서, 본 연구에서는 배열과 반복 구문을 활용한 그리드 탐색을 초매개변수 최적화 방법으로 사용하였다. 이때 레이어

깊이 및 필터 수(filter number), 즉 커널 크기(kernel size)도 아키텍처의 학습 및 예측 성능에 중요한 영향을 미치는데, 본 연구에서 레이어 깊이는 기본 깊이로 설정하였으며 초기 필터 수는 32와 64, 2가지 경우로 설정했는데 결과는 32로 결정하였다. 적용한 초매개변수 조합은 1회 학습할 때의 크기 batch size의 배열은 [64, 136, 256, 512, 1024, 2048], 학습 횟수 epoch의 배열은 [50, 100, 250, 500], 학습률(learning rate)의 배열은 [1e-3, 1e-4, 1e-5], 합성곱 레이어에서 filter number (즉, kernel size)의 배열은 [32, 64]로 숫자 조합에 대해 for loop 구문 활용하여 반복 학습하도록 코딩하였다. 그리드 탐색에서 초매개변수의 전체 조합을 ECWMF의 계절내 최고기온 및 강수 훈련 자료에 적용하여 학습률, 배치사이즈 및 적용된 확장 U-Net 모델에 따라 학습 및 검증 손실함수의 변화를 각각 비교하였다 (Figure 3.8 to Figure 3.11).

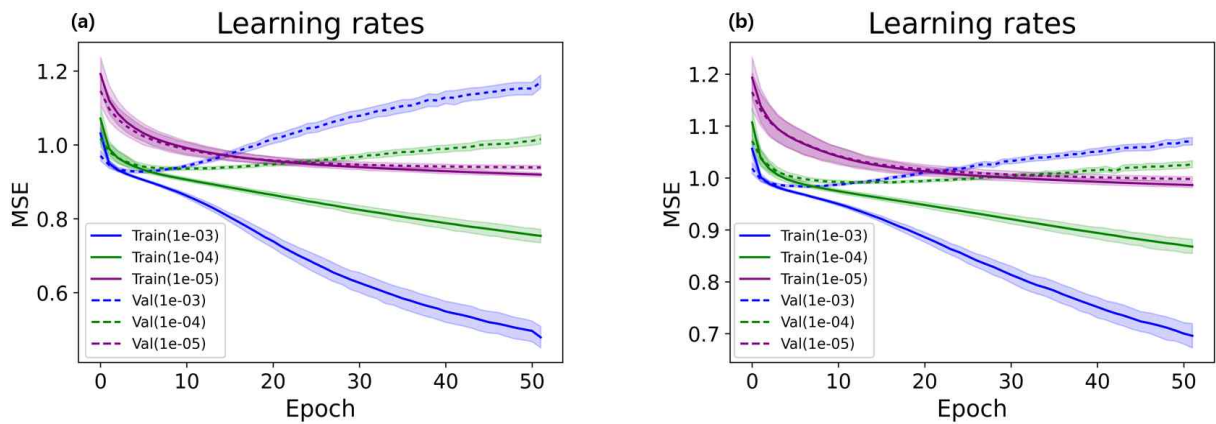


Figure 3.8. Comparison of loss function changes to learning rate according to hyperparameter combinations in the study.

Figure 3.8a와 Figure 3.8b는 각각 ECMWF의 최고기온과 강수에 대해 확장 U-Net 4개 (기존 U-Net, Residual U-Net, Attention U-Net, Attention based-on U-Net)에서 초매개변수 조합별 학습 성능을 학습률에 따라 손실함수의 변화를 비교한 것이다. 최고기온 및 강수에서 모든 확장 U-Net 조합과 배치사이즈 조합에 대해 파란색 실선과 점선은 학습률 1e-03 (0.001)에서의 훈련과 검증 손실함수의 변화, 초록색과 보라색은 각각 학습률 1e-04 (0.0001)와 1e-5 (0.00001)의 훈련 및 검증 손실함수의 변화를 비교한 것이다. 학습률 1e-03에서 훈련과 검증에 대한 손실함수는 에포크 5 이후에 급격히 차이가 나고, 학습률 1e-04의 훈련과 검증에 대한 손실함수는 학습률 1e-03의 에포크보다 큰 10 이후에 차이가 증가한다. 반면, 학습률 1e-5의 훈련과 검증 손실함수는 에포크 40이 될 때까지도 학습률 1e-03과 1e-04에서 보였던 큰 차이를 보이지 않고 일정하게 유지하는 것으로 나타났는데, 이것은 학습률 1e-05가 입력된 정보에 대해 안정적으로 훈련 및 검증을 수행한다고 추정할 수 있다. 또한, 학습률에 따른 손실함수 변화는 최고기온과 강수에서 유사하였다. 손실함수의 최저값은 최고기온 및 강수 모두에서 학습률 1e-03에서 나타났다.

Figure 3.9a와 Figure 3.9b는 각각 ECMWF의 최고기온과 강수에 대해 확장 U-Net 4개에서 초매개변수 조합별 학습 성능을 배치사이즈에 따른 손실함수의 변화를 비교한 것이다. 모든 확

장 U-Net 조합과 학습률 조합에 대해 파란색 실선과 점선은 배치사이즈 64에서의 훈련과 검증 손실함수의 변화, 초록색은 배치사이즈 128, 분홍색은 배치사이즈 256, 보라색은 배치사이즈 512, 연두색과 하늘색은 각각 배치사이즈 1024와 2048의 훈련 및 검증 손실함수의 변화를 비교한 것이다. 훈련과 검증에 대한 손실함수의 변화는 학습률에서의 패턴과 유사하였지만 배치사이즈가 가장 크게 증가한 배치사이즈 2048에서도 에포크 15 이후에는 훈련과 검증에 대한 손실함수 차이가 증가하였다. 이것은 배치사이즈 증가가 입력된 정보에 대해 안정적인 훈련과 검증에 영향을 미치지 않는다고 추정할 수 있다. 또한 배치사이즈에 따라 손실함수 변화는 최고기온과 강수에서 유사하였다. 손실함수의 최저값은 두 기후변수 모두에서 배치사이즈 64에서 나타났다.

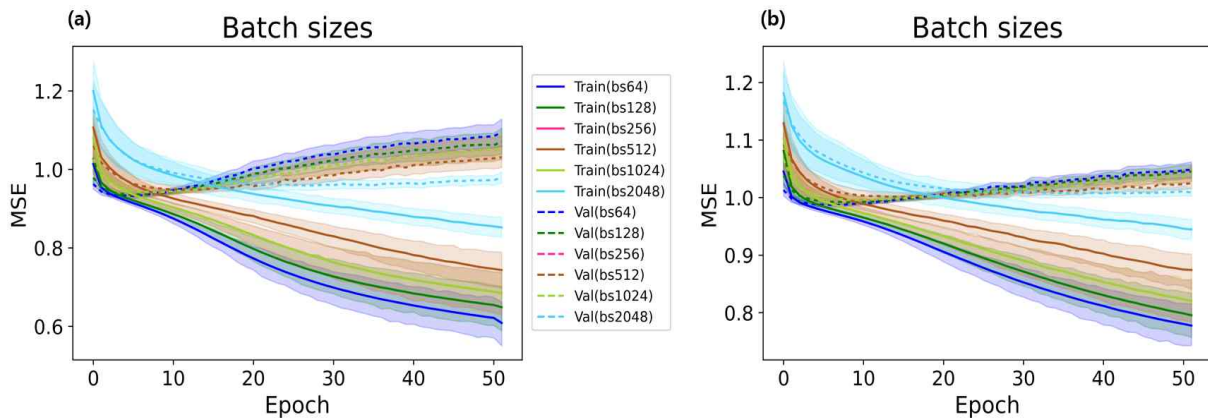


Figure 3.9. Comparison of loss function changes to batch size according to hyperparameter combinations in the study.

Figure 3.10a와 Figure 3.10b는 각각 ECMWF의 최고기온과 강수에 대해 확장 U-Net 4개에서 초매개변수 조합별 학습 성능을 확장 U-Net 모델 4개, 기존 U-Net, Attention U-Net, Residual U-Net, Attention based-on Residual U-Net에 따른 손실함수의 변화를 비교한 것이다. 모든 학습률과 배치사이즈 조합에 대해 파란색 실선과 점선은 기존 U-Net에서의 훈련과 검증 손실함수의 변화, 하늘색은 Attention U-Net, 보라색과 연두색은 각각 Residual U-Net과 Attention based-on Residual U-Net의 훈련 및 검증 손실함수의 변화를 비교한 것이다.

최고기온에서 기존 U-Net과 Attention U-Net의 학습률과 배치사이즈 전체 조합에 따른 손실함수 변화가 유사하였고, Residual U-Net과 Attention based-on Residual U-Net의 손실함수가 유사하게 나타나서 최고기온의 경우 신경망 모델이 초매개변수 조합에 따라 패턴이 두 가지로 분리되어 나타나는 경향을 보였다.

강수에서도 패턴이 분리되는 경향이 나타났는데, Attention based-on Residual U-Net만 다른 3개 확장 U-Net의 손실함수와 다소 다른 경향을 보였다. 그러나, 두 기후변수 모두에서 훈련과 검증의 손실함수가 벌어지는 경향은 학습률이나 배치사이즈에서 나타난 것과 유사하여 적용 신경망 모델의 종류가 입력된 정보에 대해 안정적인 훈련과 검증에 영향을 미치지 않는다고 판단된다. 그러나, 학습률과 배치사이즈 조합에 따라 신경망 모델의 훈련 및 검증 손실

함수 변화가 달라지는 것은 적용 신경망 모델의 초매개변수가 달라야 한다는 것을 증명하는 것이라고 판단된다. 최고기온에서 손실함수의 최저값이 나타나는 확장 U-Net은 뚜렷하지 않은 반면, 강수에서는 Attention U-Net에서 훈련과 검증 손실함수의 최저값이 나타났다.

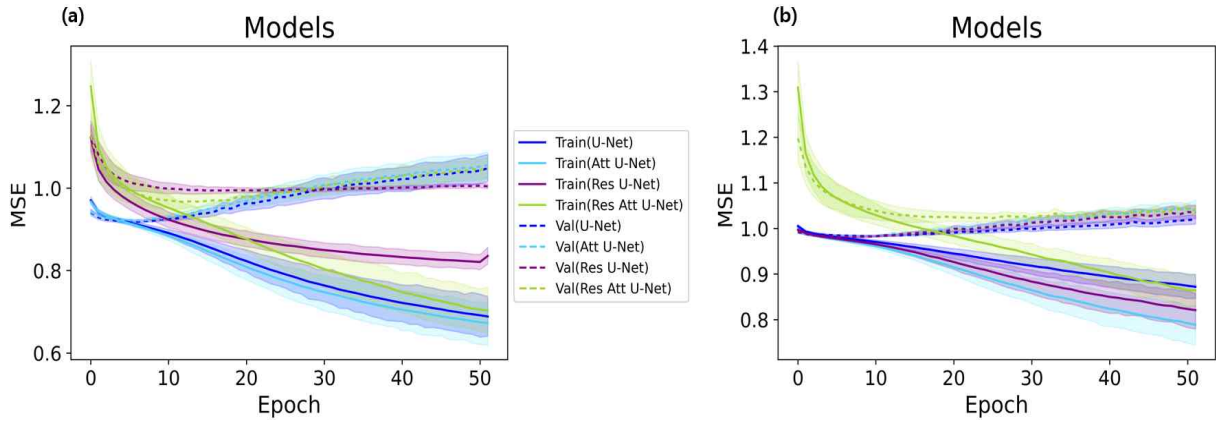


Figure 3.10. Comparison of loss function changes for the four extended U-Net models applied in this study in hyperparameter combinations.

전반적으로, 학습률 및 배치사이즈에 따른 손실함수 변화는 최고기온과 강수에서 유사하였지만 확장 U-Net 모델의 변화에는 다소 다른 패턴이 나타남으로써 적용 신경망 모델의 초매개변수가 달라야 한다는 것을 추정할 수 있다. 또한, 입력된 정보에 대한 훈련 및 검증 손실함수 변화에 영향을 미치는 초매개변수는 학습률이 가장 민감하고 배치사이즈나 에포크는 크지 않은 것으로 추정된다. 이것은 배치사이즈와 학습률이 상호 보완관계 (trade-off), 즉 한 쪽이 증가하면 다른 한 쪽은 감소 해야 하는 관계임을 확인하였다. 즉, 배치사이즈가 64에서 128로 증가하면 학습률은 $1e-04$ 에서 $1e-3$ 으로 둘의 관계가 역관계가 된다. 두 기후 변수에서 학습률의 손실함수의 최저값은 학습률 $1e-03$ 에서 나타났고, 배치사이즈는 64에서 손실함수의 최저값이 나타났다. 손실함수 최저값이 나타난 확장 U-Net은 최고기온의 경우는 뚜렷하지 않은 반면, 강수에서는 Attention U-Net에서 손실함수의 최저값이 나타남으로써 Attention 장치가 입력 정보의 특징을 추출하고 피처에 가중치를 전달 과정에서 학습 및 예측 성능에 영향을 미쳤다고 판단된다. Figure 3.11는 전체 초매개변수 조합의 손실함수에서 가장 낮은 값들만을 추출한 후, 확장 U-Net의 종류와 배치사이즈에 따라 낮은 손실함수를 보이는 학습률의 변화를 산포도 (scatter graph)로 비교한 것이다. 하늘색에서 초록색 범위는 학습률 $1e-03$ (0.001), 주황색에서 빨간색 범위는 학습률 $1e-04$ (0.0001), 보라색에서 노란색 범위는 학습률 $1e-05$ (0.00001)에서의 손실함수의 최저값을 나타낸다.

최고기온의 경우, 배치사이즈 512 이상부터 손실함수가 증가하는 것으로 나타났는데, 앞서 언급한바와 같이 손실함수 최저값은 기존 U-Net이나 Attention U-Net 및 Attention based-on Residual U-Net에서 모두 나타나서 모델간 차이가 크지 않았다 (Figure 3.11a). 확장 U-Net 모델별 손실함수 차이는 학습률 0.001보다 0.0001에서 민감했는데 (data not shown), 낮은 손실함수를 보인 학습률은 Attention based-on Residual U-Net의 0.001이었다. 이것은 가중치 및 공간

특성 장치가 동시에 장착되었을 경우 낮은 학습률이 학습에 유해할 수 있으므로 추정된다. 그러나, Attention과 Residual 장치가 모두 결합된 Attention based-on Residual U-Net에서 대부분의 조합에 대한 손실함수 변화가 안정적이며 균일하다고 판단하여, 최고기온의 경우 Attention based-on Residual U-Net에서 배치사이즈 64 및 학습률 $1e-03$ (0.001)을 선정하였다.

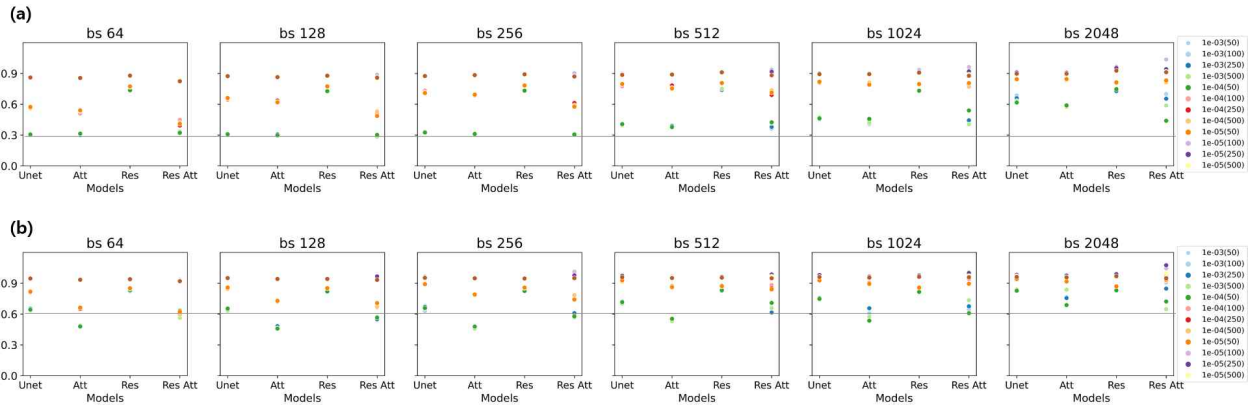


Figure 3.11. Comparison of changes in learning rate in the extended U-Net applied according to the batch size by extracting a low loss function: the x-axis shows the four extended U-Nets, the y-axis is the loss function. (a) is the S2S maximum temperature of ECMWF and (b) is the S2S precipitation of ECMWF.

강수의 경우, 배치사이즈 512 이상부터 손실함수가 증가하여 배치사이즈 증가에 대한 낮은 손실함수 패턴은 최고기온과 유사한 경향을 보였다 (Figure 3.11b). 그러나, 앞서 전체 하이퍼파라미터에 대한 학습 성능 평가의 Figure 3.10들에서 언급한 바와 같이, 낮은 손실함수간 차이는 모델 변화에 따라 뚜렷하게 나타났는데, Attention U-Net의 모든 배치사이즈 조합에서 최저의 손실함수를 균일하게 보였다. 따라서, 강수는 Attention U-Net에서 배치사이즈 64 및 학습률 $1e-03$ (0.001)을 선정하였다. 이러한 경향은 나머지 계절내 기후모델 (KMA, UKMO, NCEP, MME 등)에서 유사하게 나타났다.

3.2.3. 선정된 최적 초매개변수 조합의 예측 성능 평가

그리드 탐색 기반 최적화 알고리즘 테스트 결과로부터 도출된 배치사이즈 64와 학습률 $1e-03$ (0.001)의 조합에 대해 확장 U-Net 4개의 예측 평가 지수(예, Anomaly Correlation Coefficient 혹은 spatial Pattern Correlation Coefficient)와 기술 점수의 민감도 분석을 수행하였다 (Figure 3.12 및 Figure 3.13).

Figure 3.12는 ECMWF 최고기온에 대해 테스트 기간 동안 적용된 확장 U-Net의 예측 성능을 ACC로 평가한 것이다. 전체 그래프에서 검정색은 학습되지 않은 원시 ECMWF 최고기온의 ACC이다. 즉, 관측 최고기온(ERA5)과 원시 ECMWF 최고기온의 ACC이다. 파란색 실선은 기존 U-Net, 주황색 점선 Attention U-Net, 초록색 점선은 Residual U-Net, 빨간색 점선은 Attention based-on Residual U-Net으로부터 예측된 ECMWF 최고기온과 관측 최고기온으로부터 계산된 ACC이다. 파란색 영역 및 파란색 선 확대 그래프 (Figure 3.12b)는 2주에서 3주, 노

란색 영역 및 노란색 선 확대 그래프는 4주에서 5주까지 ACC이다 (Figure 3.12c). 가운데 및 우측 확대 그래프 안에 있는 작은 막대 그래프는 보라색 영역 기간 (Figure 3.12b, week3: 14-21일)에 대한 원시 ACC와 확장 U-Net ACC 간의 차이, 즉 Eq. [2.4]으로 계산할 수 있는 skill score (기술 점수)를 비교한 것이다. 빨간색 막대 바 (Res Att U-Net), 즉 Attention based-on Residual U-Net의 ACC가 우수 (약 4% 범위)한 것으로 나타났다. 따라서, ECMWF 최고기온의 경우 Attention based-on Residual U-Net의 학습 및 예측 성능이 우수할 것으로 예상된다. 2주에서 3주까지 (Figure 3.12b), 낮은 손실함수를 보인 Attention based-on Residual U-Net이 15일에서 21일 사이에 ACC가 원시 ACC 보다 높은 것을 알 수 있다. 그러나, 4주 이후에는 4개 확장 U-Net의 ACC가 원시 ACC보다 높아지는데 이것은 선례 연구에서도 나타났던 경향으로 기후 모델에서 나타나는 경향으로 추후 분석이 더 필요할 것으로 판단된다. 또한, 4주 이후에서는 Residual U-Net의 ACC가 다른 확장 U-Net 보다 높게 나타나는 것도 추후 분석이 요구된다.

정리하면, 최고기온의 경우 Attention과 Residual 두 개가 모두 장착된 Attention based-on Residual U-Net의 ACC가 다른 U-Net보다 향상되었다. 특히, 앞과 뒤의 차이를 활용하는 Residual 장치가 기온 예측 개선에 영향을 준 것으로 추정되는데, 기온 변수의 특성이 연속성, 즉 전날의 기온 특성이 다음 날의 기온 특성으로 연결되는 목표 변수의 특성 때문이 아닌가 추측된다.

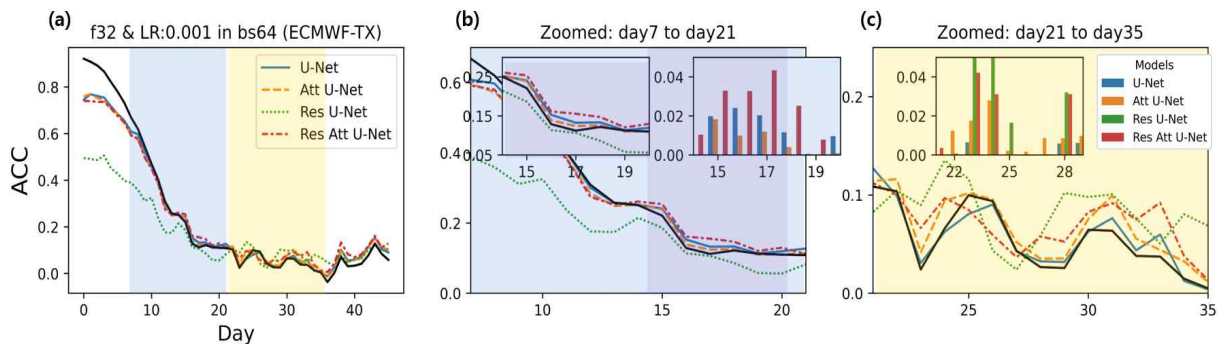


Figure 3.12. Comparison of ACC of ECMWF S2S daily maximum temperature tested in the four extended U-Net models selected hyperparameter combinations in the study: (a) ACC for the entire 46-day forecast frequency of ECMWF, and (b) is zoomed that ACC from 7th to 21th day out of the total 46-day period, and (c) is zoomed which the ACC from the 21st to the 35th day.

Figure 3.13은 ECMWF 강수에 대해 테스트 기간 동안 적용된 확장 U-Net의 예측 성능을 ACC로 평가한 것이다. 역시 전체 그래프에서 검정색은 학습되지 않은 원시 ECMWF 최고기온의 ACC이며, 파란색 실선은 기존 U-Net, 주황색 점선 Attention U-Net, 초록색 점선은 Residual U-Net, 빨간색 점선은 Attention based-on Residual U-Net으로부터 예측된 ECMWF 강수와 관측 강수로부터 계산된 ACC이다. 마찬가지로, 파란색 영역 및 파란색 선 확대 그래프는 2주에서 3주 (Figure 3.13b), 노란색 영역 및 노란색 선 확대 그래프는 4주에서 5주까지 ACC이다 (Figure 3.13c). 가운데 및 우측 확대 그래프 안에 있는 작은 막대 그래프는 보라색

영역 기간(week3: 14-21일)에 대한 원시 ACC와 확장 U-Net ACC 간의 차이 (skill score, 기술 점수)를 비교한 것이다. 보라색 영역 기간에 대해 확대된 막대 그래프를 보면 (Figure 3.13b), 주황색 막대 바 (Att U-Net), 즉 Attention U-Net의 ACC (비율로 보면, 약 4-8% 이상)가 우수한 것으로 나타났다. 따라서, ECMWF 강수는 Attention U-Net의 학습 및 예측 성능이 우수할 것으로 예상된다. 요약하면, 2주에서 3주까지, 낮은 손실함수를 보인 Attention U-Net에서 ACC가 원시 ACC 보다 높을 뿐만 아니라 skill score도 높았다 (Figure 3.13b).

강수에서 Attention Gate가 장착된 Attention U-Net에서 다른 U-Net보다 ACC가 향상된 것은 불연속적이고 지역적 특성을 가지는 강수 변수의 특성에 현재 상태 혹은 특정 정보와 가장 연관이 깊은 값(예, 유사도)에 집중하는 Attention의 가중치 계산 및 전달 기법이 강수 예측 개선에 반영된 것으로 추측된다.

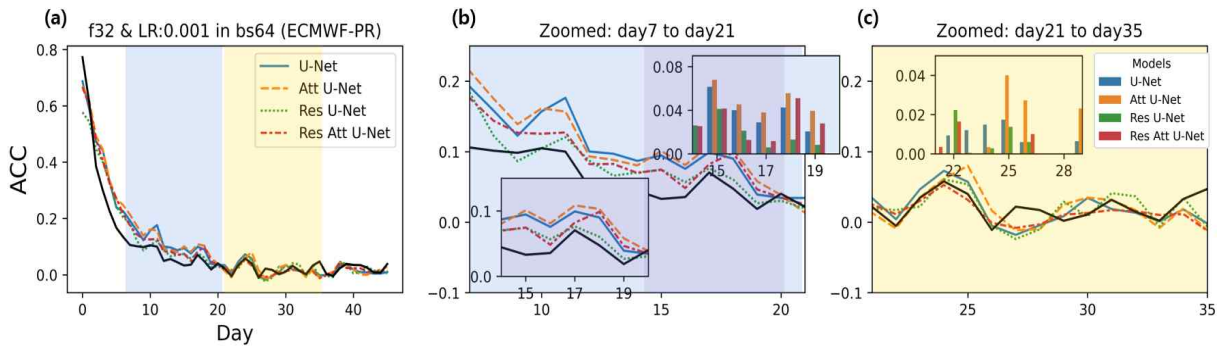


Figure 3.13. Comparison of ACC of ECMWF S2S daily total precipitation tested in the four extended U-Net models with selected hyperparameter combinations in the study: (a) ACC for the entire 46-day forecast frequency of ECMWF, and (b) is zoomed that ACC from 7th to 21th day out of the total 46-day period, and (c) is zoomed which the ACC from the 21st to the 35th day.

Figure 3.14는 ECMWF 최고기온과 강수에 대해 테스트 기간 동안 적용된 확장 U-Net의 예측 성능을 PCC로 평가하여 요약한 것이다. 막대 바 그래프에서 초록색은 (S2S) 학습되지 않은 원시 ECMWF 최고기온과 강수의 PCC이며, 파란색 막대 바는 기존 U-Net, 회색 막대 바는 Attention U-Net, 하늘색 막대 바는 Attention based-on Residual U-Net, 남색 막대 바는 Residual U-Net으로부터 예측된 ECMWF 최고기온과 강수와 관측 강수로부터 계산된 PCC를 주별로 요약하여 비교하였다. 선례 연구에서 이미 언급한 바와 같이, 기온의 1주는 신경망 모델로 후보정이 어렵다는 것을 ACC 및 PCC 비교에서 보여주었다. 그러나 2주 이후에는 신경망 모델로 후보정하여 S2S의 예측 성능을 높일 수 있고 Attention U-Net (회색 막대 바)이 확장 U-Net 모델들에서 우수한 것으로 나타났다 (Figure 3.14a). 강수는 선례 연구에서도 이미 밝혀진 바와 같이 신경망 모델 기반 후보정을 통해 S2S 강수 예측을 개선할 수 있음을 일관되게 보여주고 있었다. 확장 U-Net 모델들 중에서도 특히 Attention (회색 막대 바) 및 Residual (하늘색 막대 바)이 추가되면 훨씬 더 개선할 수 있으므로 보여주었다 (Figure 3.14b).

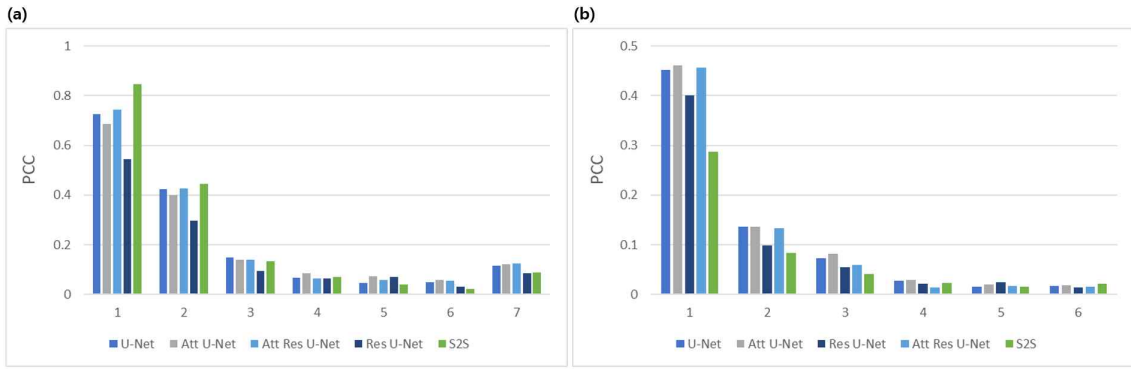


Figure 3.14. Comparison of weekly PCCs for ECMWF S2S daily maximum temperature (a) and daily total precipitation (b) tested on the four extended U-Net models with selected hyperparameter combinations in the study.

Figure 3.15와 Figure 3.16은 선정된 초매개변수 조합 (배치사이즈 64 및 학습률 $1e-03$, 에포크 100)의 확장 U-Net 4개에서 테스트 기간 동안 예측된 ECMWF의 최고기온과 강수 중에서 각 주 (week)에 따라 7일씩 맵 (map)으로 출력한 것이다.

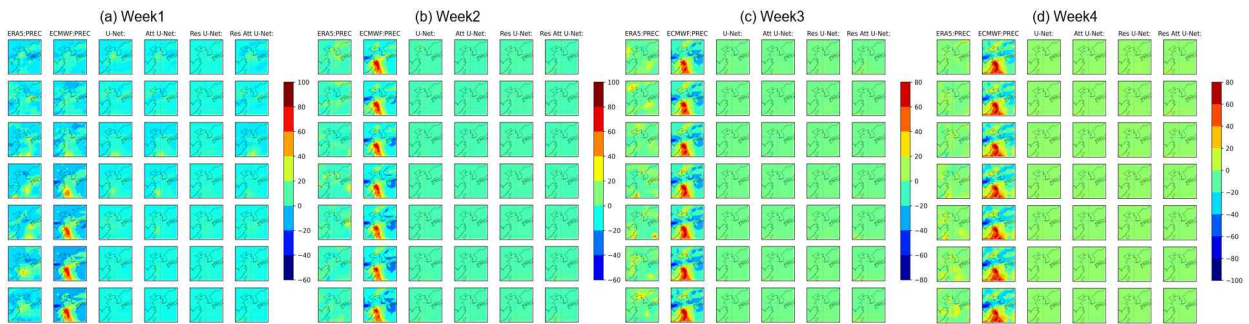


Figure 3.15. Maximum temperature maps selected for 7 days within each week from the 46-day prediction frequency of ECMWF S2S maximum temperature predicted during the test period in the four expanded U-Nets with selected hyperparameter combinations. (a) to (d) are from week 1 to week 4.

Figure 3.15의 b와 c에서 마지막 패널은 (Attention based-on Residual U-Net) 첫 번째 패널 (ERA5, 즉 관측) 보다 두 번째 패널 (ECMWF, 라벨)의 맵들과 유사한 것을 알 수 있었다. 이것은 최고기온 훈련에서는 라벨, 즉 자기 상관의 영향이 크게 미치는 것은 아닌가 추측되며 (Chung et. al., 2023), ACC나 PCC 가 낮은 이유가 아닌가 추정된다. 다시 말하면, 관측 정보보다는 ECMWF 입력 정보 (목표 변수 이외 다른 기후 변수가 함께 입력된 훈련 자료)에 너무 큰 영향을 받는 것은 아닌가 생각된다. 그러나, 강수 표면 맵 (Figure 3.16)에서는 4개 확장 U-Net 에서 예측된 강수 표면 맵 (세 번째 패널에서 마지막 패널까지)이 모두 첫 번째 패널과 유사하여 모델간 구별은 어렵지만 모델이 라벨이 아닌 관측을 모사하는 것이 최고기온과 다름을 추정할 수 있어서 이것이 ACC나 PCC 평가에 영향을 미친 것으로 판단된다. 이것은 Chung et. al. (2023)에서 강수의 경우 상향장파 (OLR)와 850hPa의 비습도 (Q850)가 강수 훈련 및 예측에 일관되게 영향을 주는 역할이었기 때문이라고 했는데 이것이 재분석 자료와도 연관이 있기 때

문인 것은 아닌가 추정된다.

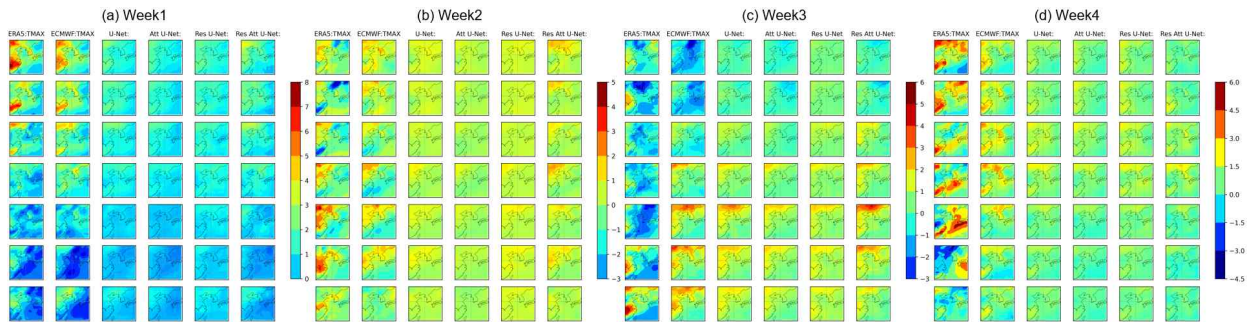


Figure 3.16. Precipitation maps selected for 7 days within each week from the 46-day prediction frequency of ECMWF S2S total precipitation predicted during the test period in the four expanded U-Nets with selected hyperparameter combinations. (a) to (d) are from week 1 to week 4.

3.3. 준지도학습 기법 적용 및 모델의 민감도 분석

3.3.1. 다양한 변수 조합, 자료 분리 방법, 모델 구조에 따른 모델 민감도 테스트 결과

Figure 3.17은 다양한 변수 조합, 자료 분리 방법, 모델 구조에 따른 테스트(Test) 자료에 대한 모델의 정확도를 보여준다. VGG16 모델 및 Wide Resnet 모델들은 대체로 낮거나 일관되지 않은 정확도를 보여준다. 앞선 설명에 따라 VGG는 Resnet 계열의 모델에 비해서 낮은 성능을 보여줄 것은 쉽게 예상할 수 있었고 Resnet에 비해서 훨씬 더 긴 훈련 시간이 소요됐다. Resnet18 모델이 전반적으로 더 깊거나(Resnet50) 넓은(WideResnet) 모델들에 비해서 더 나은 성능을 보여주었는데 이는 제한적인 자료에 모델이 복잡해질수록 훈련 자료에만 과적합 되는 문제가 생기기 때문에 오히려 더 낮은 성능을 보일 수 있다. RMM 변수 조합(U820, U200, OLR 사용)에 비해 VPM 변수 조합(U850, U200, VP 사용)이 전반적으로 조금 더 나은 결과를 보여준다. 이는 VP가 상층의 발산을 더 잘 보여주기 때문에 더 직접적으로 MJO와 관련될 수 있으며 (Ventrice et al., 2013), 따라서 모델은 더 명확한 대류의 특성을 기반으로 더 나은 성능을 보인 것으로 판단된다. 무작위 분리(Random Split)가 시간적 분리(Temporal Split)에 비해 조금 더 나은 성능을 보여준다. 이는 무작위 분리를 통해서 모델을 학습할 때 좀 더 일반화 성능이 향상된 것으로 판단된다. 모델 학습과는 독립적으로 구성된 테스트 자료에 대해서 큰 성능 차이가 나지 않았다는 것은 테스트 기간의 자료 분포에 대해서도 좋은 성능 결과를 내며 일반화 성능이 향상되었다는 것을 의미한다.

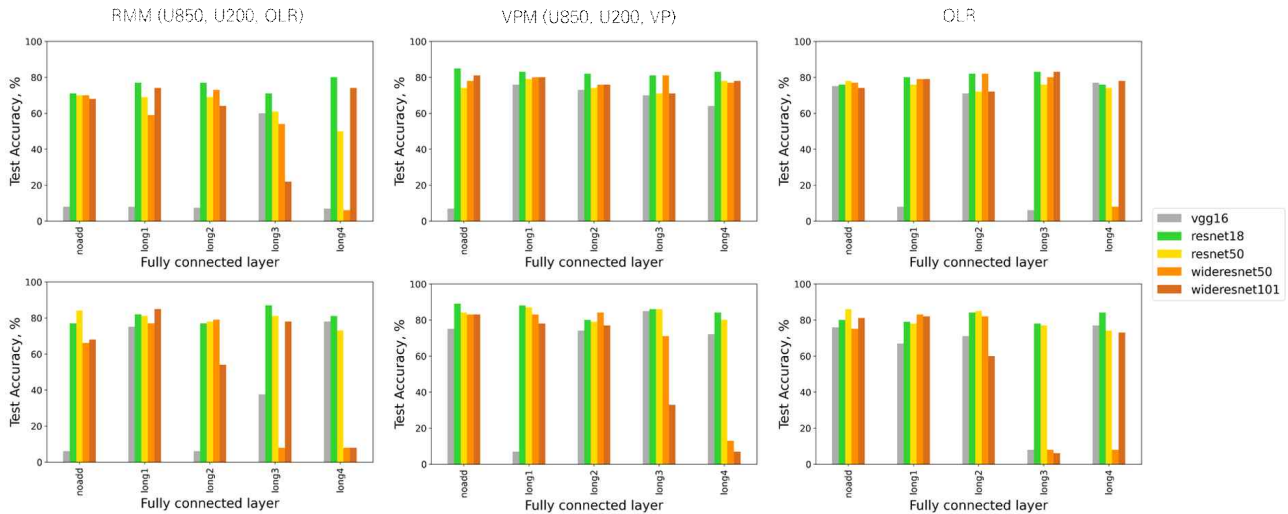


Figure 3.17. Summary of model skill for MJO phase classification with the test data. Colored bars of each panel show the test accuracy (%) of the models. Left panels are the results with RMM input variables (U850, U200, OLR), center ones with VPM variables (U850, U200, VP), and the right ones with OLR only. The upper row shows the results with temporal split data and the lower ones with random split data.

3.3.2. 준지도학습 기반 MJO 위상 분류 모델 민감도 테스트 결과

Table 3.4은 자료 증강 방법에 따른 준지도학습 기반 MJO 위상 분류 정확도 결과를 보여 준다. 수평 이동하는 MJO의 특성에 따라 수평적인 자료 변환 방법은 오히려 정확도를 낮추는 한편 수직 반전은 성능 향상에 기여하는 것(83.15%)으로 나타났다. 자료 증강을 적용하지 않았을 때(No DA) 오히려 증강을 적용한 것(수평 반전, 가우시안 흐림, 무작위 변환)보다 조금 더 나은 정확도를 보였다. 이 실험을 통해 자료 증강 방법은 목적으로 하는 변수의 특성에 맞게 적용되어야 함을 시사한다.

Table 3.5는 RMM 및 VPM 인덱스 변수 조합에 대한 세 가지 라벨 개수에 따른 각 실험 별(cf. Table 2.2) 준지도학습 기반 MJO 위상 분류 모델의 테스트 자료에 대한 정확도를 보여 준다. 전반적으로 적은 라벨링 자료를 사용한 경우(900 labeled samples) 낮은 정확도를 보이며 Student 모델과 Teacher 모델 간의 차이가 커 준지도학습의 효과가 두드러진다. 4500개의 라벨링 자료 결과는 9000개에 비해 적은 양에도 불구하고 종종 더 높은 정확도를 보이는 결과를 보인다. 실험 간에는 배치 크기, 손실함수 가중치, 자료 강화 등의 모델 내 파라미터에 따른 차이들을 나타낸다. Test 1과 2는 모델 학습 방법의 차이(배치 크기 차이), Test 2와 3은 자료 강화 효과, Test 1과 4는 손실함수 영향의 효과를 나타낸다. 각 정확도의 위의 값은 Student 모델의 정확도, 아래는 Teacher 모델(EMA model)의 정확도이며, 이는 알고리즘 내에서의 준지도학습의 효과를 일정 부분 반영한다 (Student 모델도 Teacher 모델의 결과를 반영하여 학습됨). Test 1과 2의 결과에서 Test 1은 4500개의 라벨링 자료를 사용한 결과가 9000개에 비해 더 높은 정확도를 보이고, Test 2는 더 많은 라벨링 자료를 사용할수록 더 높은 정확도를 보이는 경향이 있다. Test 2와 3의 비교에서는 자료 강화를 적용하는 것이 전반적인 성능 향상에 기여하는 것을 보여준다. Test 1과 4는 라벨링 자료의 양이 적을수록 준지도학습 초기에 몇 Epoch는

라벨링 자료로 학습하는 것이 전체적인 성능 향상을 보였다. Figure 3.18은 지도학습 결과(cf. 본문 3.3.1)와의 비교를 보여준다. 앞선 지도학습(Supervised learning; 라벨링된 모든 자료를 활용하여 학습)과 비교하여 4500 및 9000개의 라벨링 자료를 가지고도 비슷하거나 더 높은 정확도를 보였다. 이 실험을 통해 적은 자료를 가지고도 준지도학습 환경에서 모델을 학습하면 지도학습과 비슷한 성능을 낼 수 있음을 알 수 있었다. 이는 향후 가용한 자료가 제한적이거나 부정확한 경우 일부의 자료만 가지고도 비슷한 성능을 내는 모델을 개발하는 것이 가능함을 시사한다.

Table 3.4. Accuracy of SSL-based model in the test data for MJO phase classification for different data augmentation methods.

Data Augmentation Method	Accuracy(%)
No DA	79.37
Horizontal Flip	77.05
Gaussian Blur	77.96
Vertical Flip	83.15
Gaussian Blur + Vertical Flip	80.92
Random Perturbation (-0.01, 0.01)	76.78
Random Perturbation (-0.1, 0.1)	78.01
Random Perturbation (-0.1, 0.1) + Vertical Flip	79.92

Table 3.5. Summary of SSL-based model skill of the experiments (cf. Table 2.2) for MJO phase classification with the test data for MJO index and different labeled samples.

MJO index	Labeled samples	Test1	Test2	Test3	Test4
RMM	900	78.20	70.32	53.30	69.56
		78.97	77.38	74.67	78.97
	4500	87.28	81.52	72.12	87.03
		87.31	86.39	83.32	87.06
	8100	83.29	88.00	85.63	82.01
		83.26	89.89	86.67	82.68
VPM	900	80.91	69.13	73.91	80.49
		80.91	79.15	77.87	80.79
	4500	87.46	86.67	82.25	87.28
		87.06	89.50	85.39	87.64
	8100	82.77	91.02	89.47	83.81
		82.71	91.45	89.71	83.74

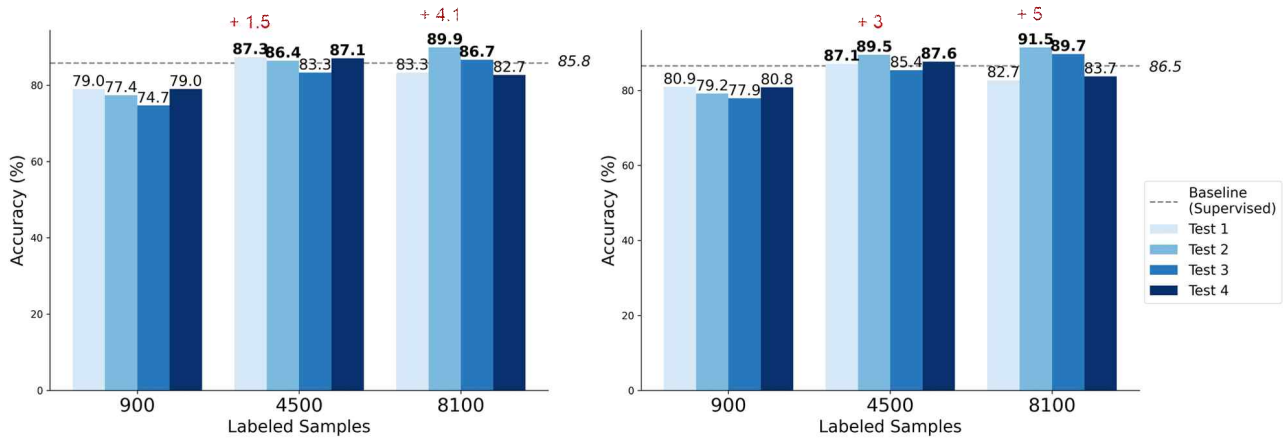


Figure 3.18. Bar charts of the SSL-based model skill of the experiments in the test data with comparison to supervised learning results (baseline) shown as horizontal dashed gray lines. Colored bars indicate the experiments (cf. Table 2.2).

Figure 3.19와 3.20는 준지도학습 기반 MJO 위상 분류 모델의 월별 및 계절별 테스트 자료에 대한 결과를 보여준다. 실험 결과는 실험 2(cf. Table 2.2)의 결과이며 Bar 그림의 y축 및 Heat map의 값은 오분류된 MJO 발생 사례의 수를 나타내었다. 전반적으로 RMM 보다 VPM 자료 기반 결과가 더 나은 결과를 보였다. VPM의 경우 전반적으로 여름철에 에러가 좀 더 증가하는 경향을 보였다. 이는 MJO는 겨울철에 많이 발생하는 현상으로 여름철 정확도가 낮을 것으로 예상된다. RMM의 겨울철 성능의 저하는 상대적으로 낮은 모델 성능에 기인하는 것으로 판단된다. Active MJO에 비해 Inactive MJO(파란색 bar)가 조금 더 에러가 높은 경향을 보였다. 모델의 성능은 연도별로 다르게 나타나지만 RMM의 경우 특정 연도에 치우쳐 나타나는 경향은 없고 VPM의 경우 2017년도 여름철에 성능 저하가 두드러지는 편이다.

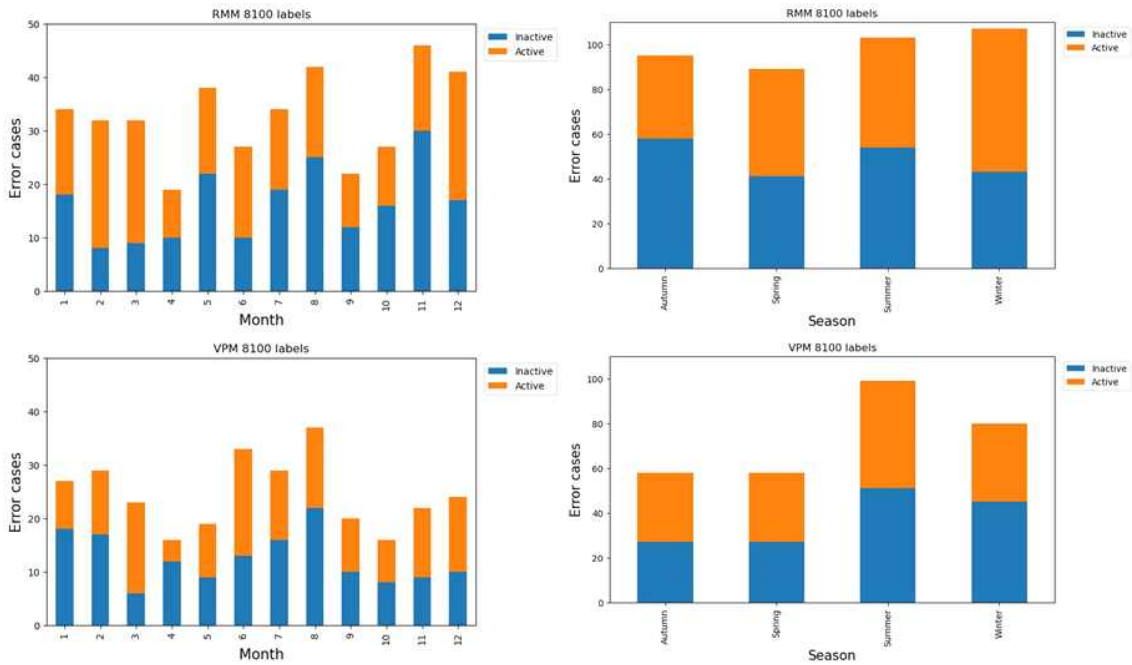


Figure 3.19. Monthly and seasonal SSL-based model performances with the test data for MJO phase classification grouped by active (orange) and inactive (blue) MJO cases. Error cases on the y-axis indicate MJO events misclassified by the model.

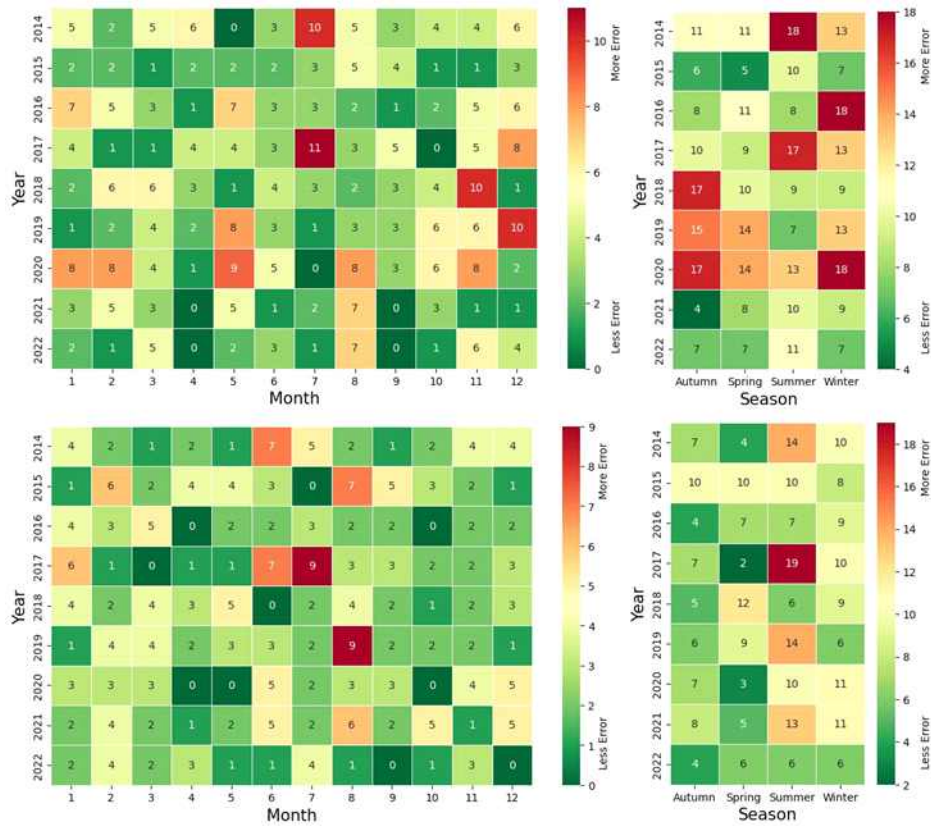


Figure 3.20. Monthly and seasonal SSL-based model performance heat maps with the test data for MJO phase classification. The number of errors in each grid shows the MJO events misclassified by the model.

3.3.3. MJO 입력장 예측을 위한 시공간 이미지 예측 인공지능 모델 개발 결과

Figure 3.21과 3.22은 여러 이전 시계열 자료로 다음날(t+1일) 및 7일(t+1~7일) 연속 예측 모델링 테스트 결과를 보여준다. 다음 날 예측이 7일 연속 예측보다 더 나은 결과를 보였다. 7일 연속 예측은 시간이 갈수록 공간적인 패턴이 무너지며 절대적인 값이 실제에 비해 낮은 결과를 산출했다. PredRNN 모델이 SimVP 모델보다 실제와 더 가까운 절댓값을 산출하는 것을 확인했다.

Figure 3.23은 샘플링 방법에 따른 모델 결과를 보여준다. SimVP gSTA 모델과 120일 이전 시계열 자료를 활용한 결과이다. 왼쪽 상단은 모든 자료(모든 계절 및 active/inactive MJO 발생 케이스 포함)를 사용한 결과, 오른쪽 상단은 Active MJO만 활용한 결과, 왼쪽 하단은 여름철을 제외한 결과, 오른쪽 하단은 여름철과 Inactive MJO 케이스를 제외한 결과이다. 모든 자료를 활용했을 때 좀 더 뚜렷한 값을 보이지만, 전반적으로 서로 큰 공간적인 차이는 없어 보였다. 향후 개선 방향으로 예측 성능을 향상시켜줄 수 있는 해수면 온도(Sea Surface Temperature), 비습(specific humidity) 등의 입력 변수를 추가하고자 한다 (Martin et al., 2022, Kang et al., 2022). 비습은 MJO 발달 전면에 하층 수증기량의 증가(MJO의 동진 전파에 유리한 조건)와 관련되며, 해수면 온도는 MJO 진행 방향의 동쪽은 SST 상승 서쪽으로는 SST 하강이

나타나 MJO 예측에 활용할 수 있을 것으로 생각된다.

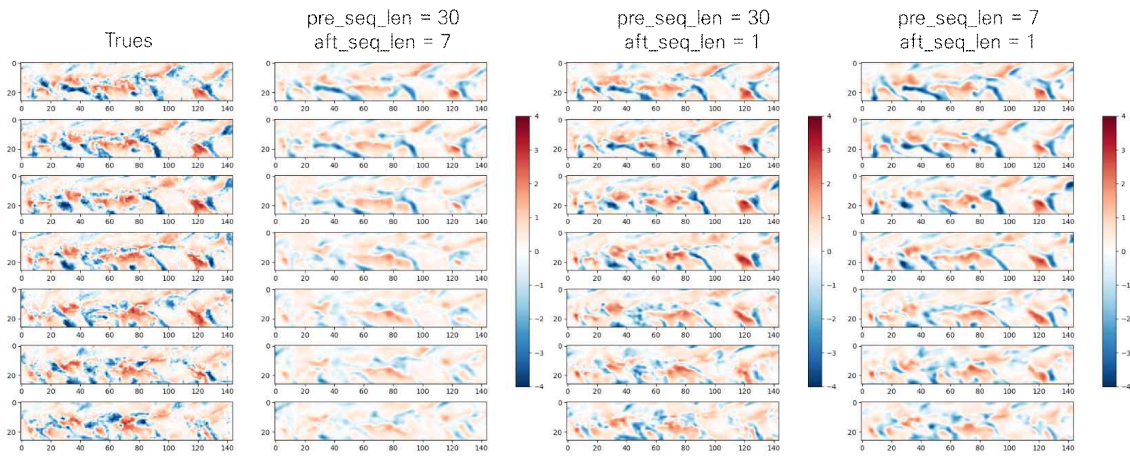


Figure 3.21. Predictions of OLR from PredRNN v2 model for pre-sequence length of 7 or 30 days and after-sequence length of 1 or consecutive 7 days with reference (Trues) from NOAA interpolated OLR data (left column).

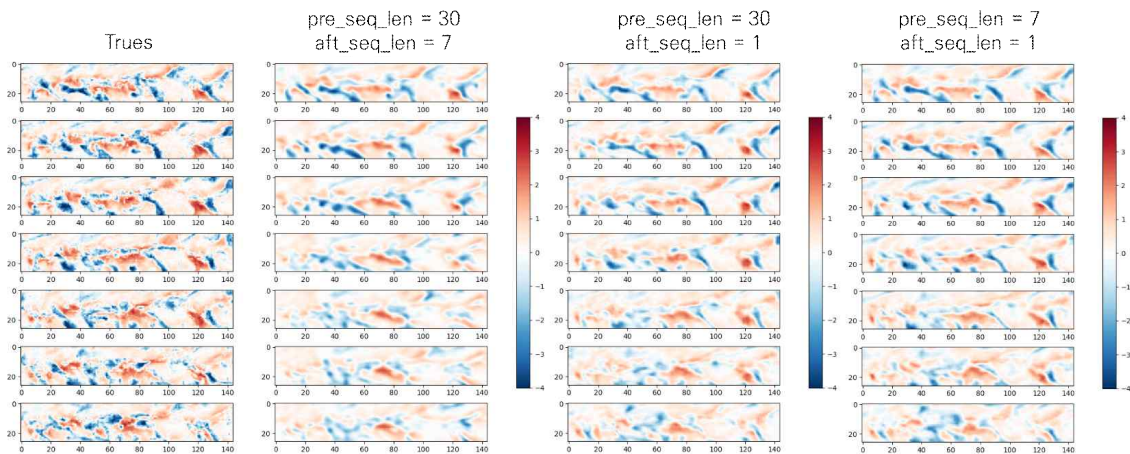


Figure 3.22. Predictions of OLR from SimVP gSTA model for pre-sequence length of 7 or 30 days and after-sequence length of 1 or consecutive 7 days with reference (Trues) from NOAA interpolated OLR data (left column).

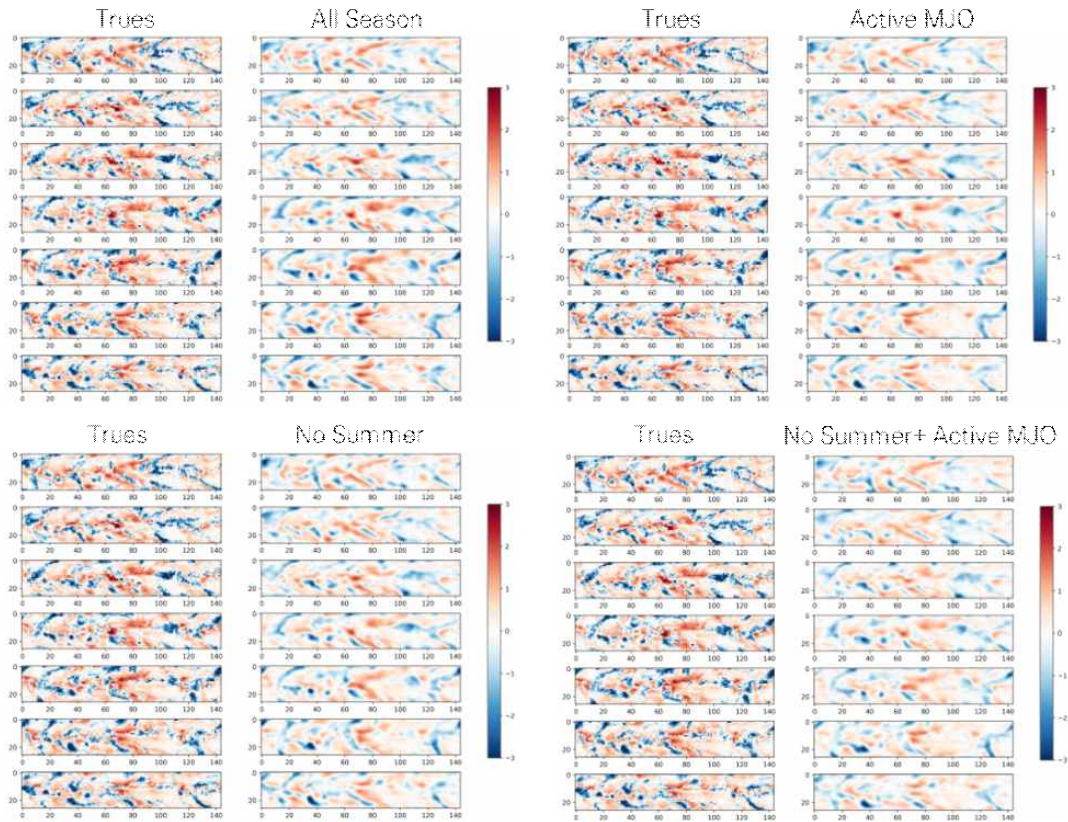


Figure 3.23. Predictions of OLR for different sampling strategies from SimVP gSTA model with pre-sequence length of 120 days and after-sequence length of 1 day with reference from NOAA interpolated OLR data (left columns).

3.4. 설명 가능한 인공지능(XAI), 모델 구조 분석 및 모델 개선

3.4.1. XAI 기법 적용을 통한 모델 분석

3.4.1.1. XAI 기법 적용을 통한 모델 분석

앞서 개발된 MJO phase 분류 모델에 XAI 기법을 적용하고 분석하였다. MJO phase 분류 모델은 준지도학습(semi-supervised learning)이 적용되기 전 모델과 적용 후 모델을 대상으로 XAI 비교 분석하였다. Figure 3.24는 MJO phase 분류 모델 학습에 사용된 훈련자료를 phase 1-8까지 클래스별로 합성한 자료(composite data)이다. Figure 3.25은 준지도학습이 적용되기 전 모델 중 resnet18 아키텍처로 학습한 모델로 occlusion 분석한 결과이다. 윈도우 크기는 5x5이며, 스트라이드(stride)는 3x3이다. Figure 3.26은 준지도기법을 이용하여 학습한 모델을 이용하여 occlusion 분석한 그림이다. Figure 3.26a는 윈도우 10x10과 스트라이드 7x7이고, Figure 3.26 b는 윈도우 5x5와 스트라이드 3x3이다. Occlusion 분석에서 윈도우와 스트라이드 크기는 확률 변화의 민감도 히트맵의 공간해상도에 영향을 준다. 준지도 학습 전과 후의 occlusion 분석 결과, 준지도 학습 전과 후 큰 차이를 보이지는 않았다.

Composite data

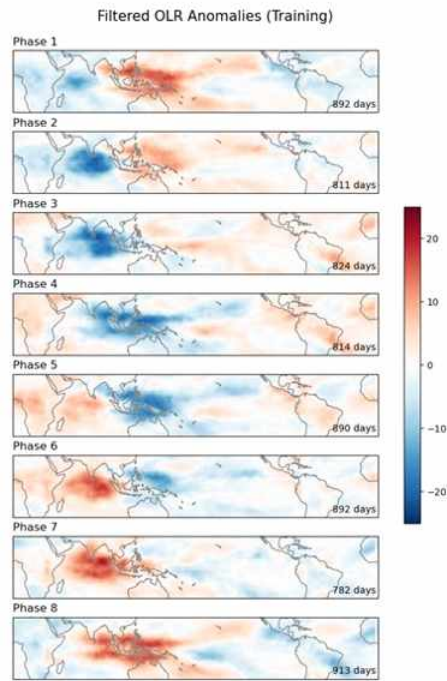


Figure 3.24. Composite data of the filtered OLR anomalies (training data).

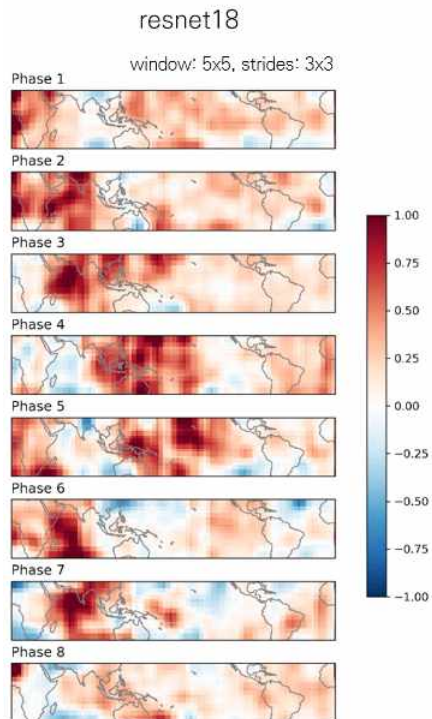


Figure 3.25. Occlusion analysis using results predicted from test data with the resnet18 model (window 10x10 and strides 7x7).

Semi-Supervised Learning

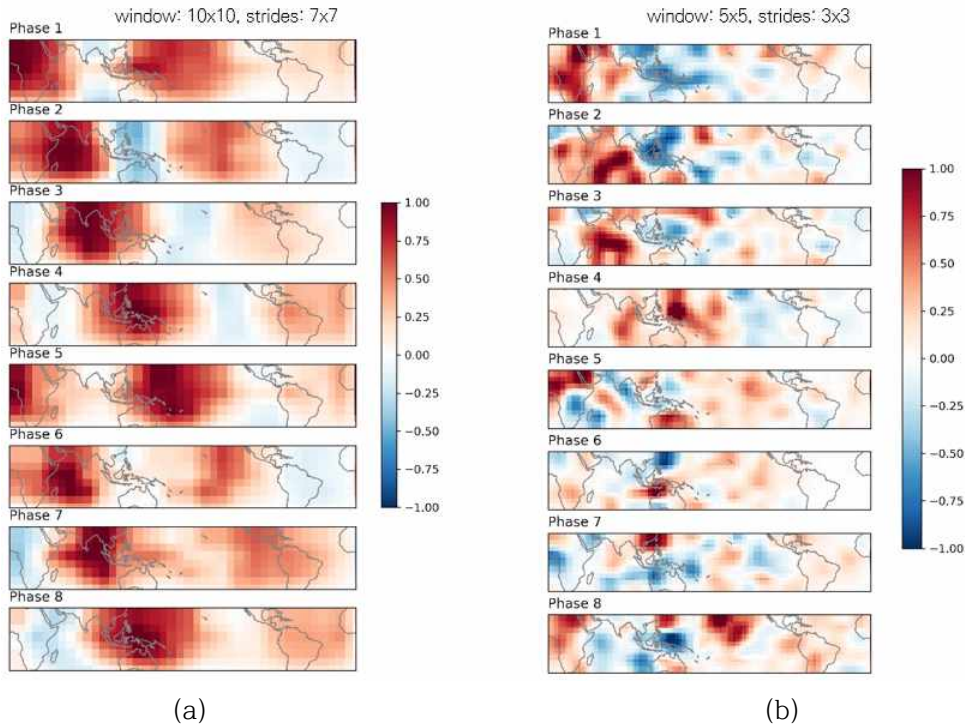


Figure 3.26. Occlusion analysis using results predicted by test data with the resnet18 model and the deep learning model after applying semi-supervised learning: (a) XAI result of the resnet 18 model with window 10x10 and strides 7x7, (b) window 10x10 and strides 7x7, (c) window 5x5 and strides 3x3.

3.4.1.2. 손실지형 결과 및 분석

딥러닝 모델의 학습은 손실값이 전역 최솟값(global minima)이 되는 지점을 찾는 과정이다. 모델의 손실지형은 지역 최솟값(local minima)에 빠지기 쉬운 구조인지를 판단하는데 유용하다. 또한, 손실지형은 딥러닝 모델의 과적합(overfitting)과 과소적합(underfitting)을 판단하는데도 도움이 된다. Li et al. (2018)이 제안한 손실지형 시각화 방법을 통해 딥러닝 기반 1개월 기온예측 모델의 손실지형을 시각화하고 분석하였다. 또한, Densenet121, Resnet56 등 기존 모델과의 비교하였다. Figure 3.27, 3.28, 3.29은 딥러닝 모델 설계에 대표적으로 많이 사용하는 모델이다. Figure 3.28와 3.29은 Resnet (residual neural network) 모델에서 입력자료가 레이어를 통과하지 않고 레이어를 통과한 자료와 결합하는 identity mapping(또는 shortcut connection, skip connection)에 따른 손실지형을 나타낸다. Densenet은 이전 레이어의 피처를 이후 모든 레이어에 연결하는 구조이다. Resnet 모델에 비해 지형이 단조로워서 전역 최솟값을 찾는 데 유리한 장점이 있다(Figure 3.27). Resnet은 Densenet에 비해 지형이 다소 복잡하고 특히 identity mapping이 빠진 모델의 경우에는 손실지형이 더 복잡하여 학습과정에서 지역 최솟값에 빠질 가능성이 매우 높다. Densenet이 Resnet 등과의 성능 비교에서 좋은 성능을 보였다(Huang et al. 2018). 딥러닝 기반 1개월 기온예측 모델은 다른 모델과 다르게 완만한 손실지형을 보였다(Figure 3.30). 이는 모델이 전역최저점을 찾기 쉬운 구조로 모델이 심층학습하는데

유리한 것을 의미한다.

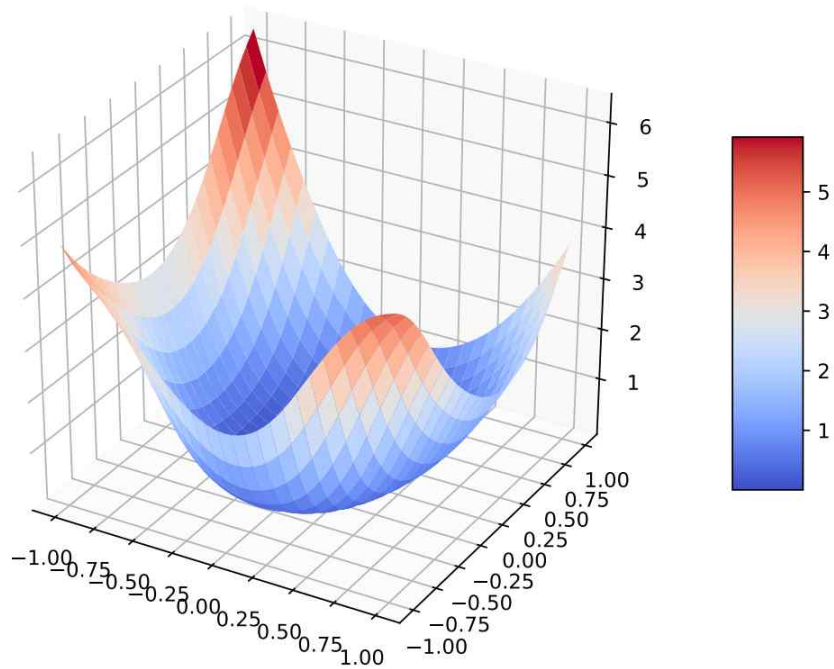


Figure 3.27. Visualization of loss surface of Densenet121 model with 0.1 learning rate, 64 batch size and SGD (Stochastic Gradient Descent) optimizer.

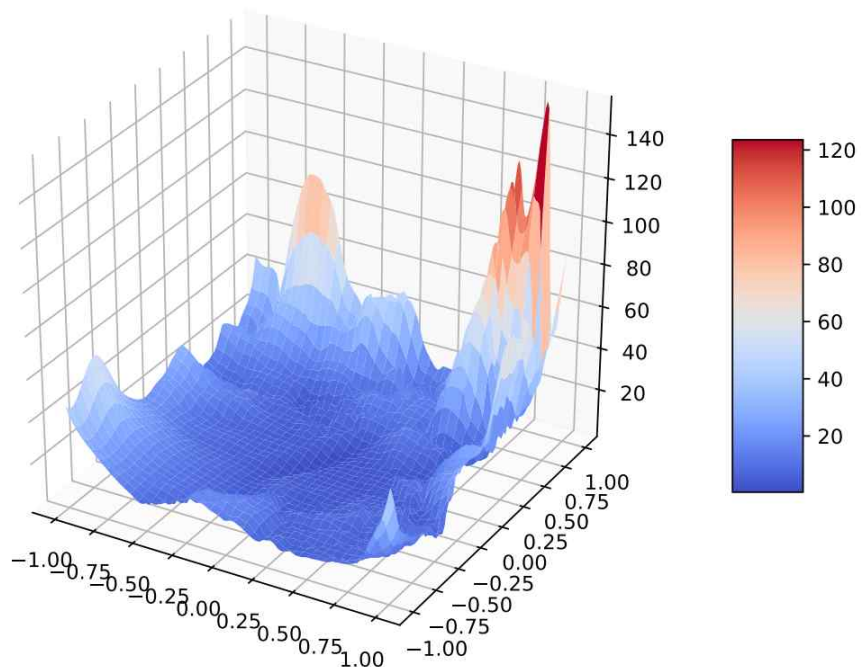


Figure 3.28. Visualization of loss surface of Resnet56_noshort with 0.1 learning rate, 64 batch size and SGD optimizer.

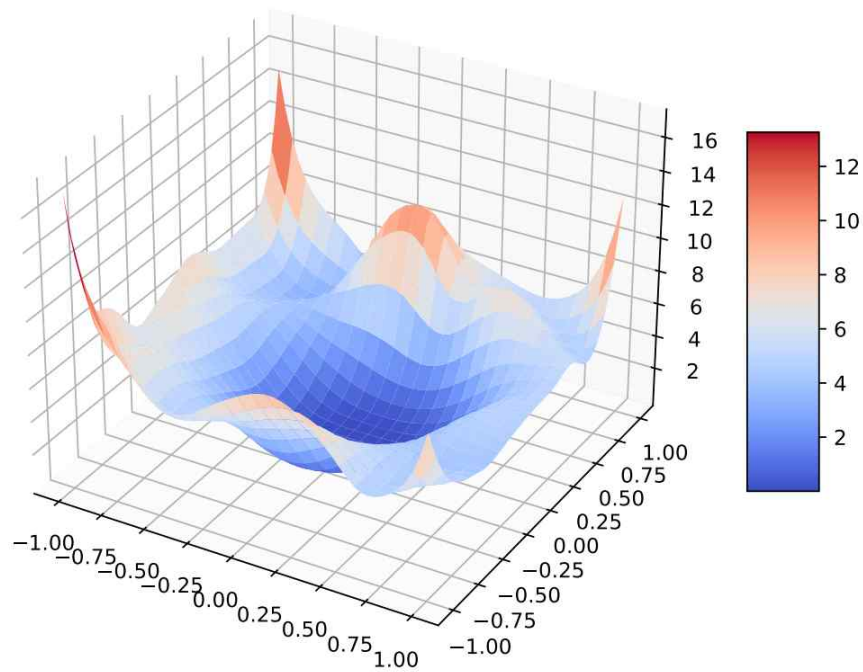


Figure 3.29. Visualization of loss surface of Resnet56 model with 0.1 learning rate, 128 batch size and SGD optimizer.

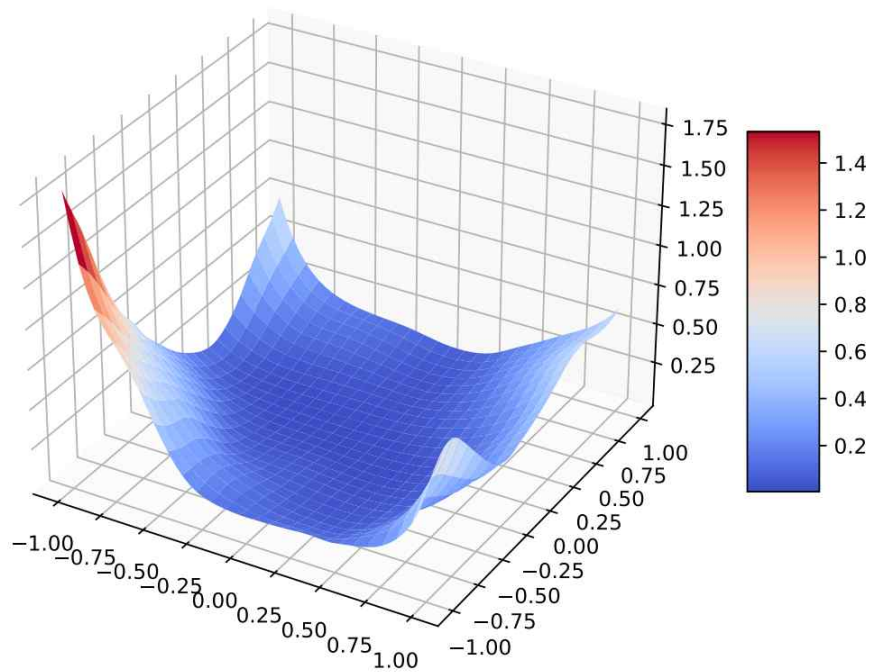


Figure 3.30. Visualization of loss surface of 1-month T2M prediction model with 0.001 learning rate, 16 batch size and ADAM (adaptive moment estimation) optimizer.

3.4.2. 딥러닝 모델 내부 구조 분석 및 개선 결과

1개월 기온예측모델을 기초로 어텐션 메카니즘, 데이터확장, 레이어 필터 수 증가 등을 고려하여 ECMWF S2S 예측자료와 HSS 기법을 통해 예측성을 분석하였다. 실험에 사용한 딥러닝 모델은 어텐션 메카니즘을 중심으로 구분이 되며, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, D-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2로 구성하였다. SE와 CBAM은 어텐션 메카니즘, BIGDATA는 데이터 학습자료 기간 확장, FLT2는 레이어 필터 수 2배를 의미한다. 실험은 2018년~2022년 기간의 ECMWF S2S 예측자료의 1-2주 자료를 입력자료로 이용하여 예측 1주(F1, W+3), 예측 2주(F2, W+4)를 예측하였다. ECMWF S2S 예측자료와 딥러닝 예측자료를 KMA ASOS 자료(관측값)와 3분위(above normal, near normal, below normal) 확률예측을 수행하고 비교하였다.

최근 연구에서는 모델의 파라미터수가 기하급수적으로 늘어나면서 그리드 탐색(grid search) 알고리즘 등과 같은 기법을 이용한 최적 하이퍼파라미터를 찾는 연구는 줄어드는 추세이다. 그러나 그리드 탐색 알고리즘 등은 여전히 최적 하이퍼파라미터를 찾는 중요한 기법이다. 본 연구에서는 앞선 연구에서 수행한 그리드 탐색 알고리즘 대신 딥러닝 모델의 하이퍼파라미터 중 배치크기만을 고려하여 앞서 설계된 모델 중 CBAM 어텐션 메카니즘이 적용된 모델을 이용하여 학습하고 예측성을 분석하였다. 실험에 사용한 모델은 딥러닝 기반 1개월 기온예측 모델이다. 실험 결과, 배치 크기가 증가할수록 validation loss가 증가하는 것을 보였다 (Figure 3.31). 배치크기가 증가함에 따라 학습에 소요되는 시간은 줄어들지만 예측성은 낮아지는 것을 보였다. 이 실험의 결과를 토대로 앞서 설계된 6개월 모델의 학습에는 배치크기 2를 적용하였다.



Figure 3.31. Plot of model training results by batch size.

학습데이터셋의 테스트기간 중 2018-2020년 총 3년에 대해 ACC 평가를 수행하였다. 계절별로는 JJA가 다른 계절에 비해 낮은 ACC를 보였다(Figure 3.32). 연도별 ACC는 큰 차이를 보이지 않았다(Figure 3.33). 그러나 배치 크기가 커질수록 예측성은 낮아지는 것으로 나타났다.

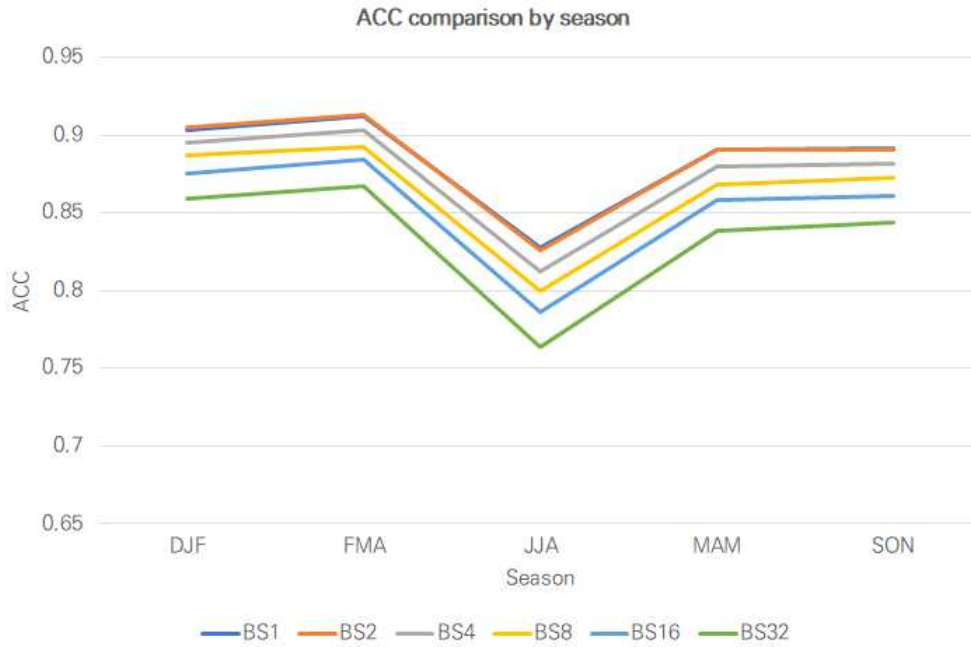


Figure 3.32. Plot of the comparison of the ACC by season.

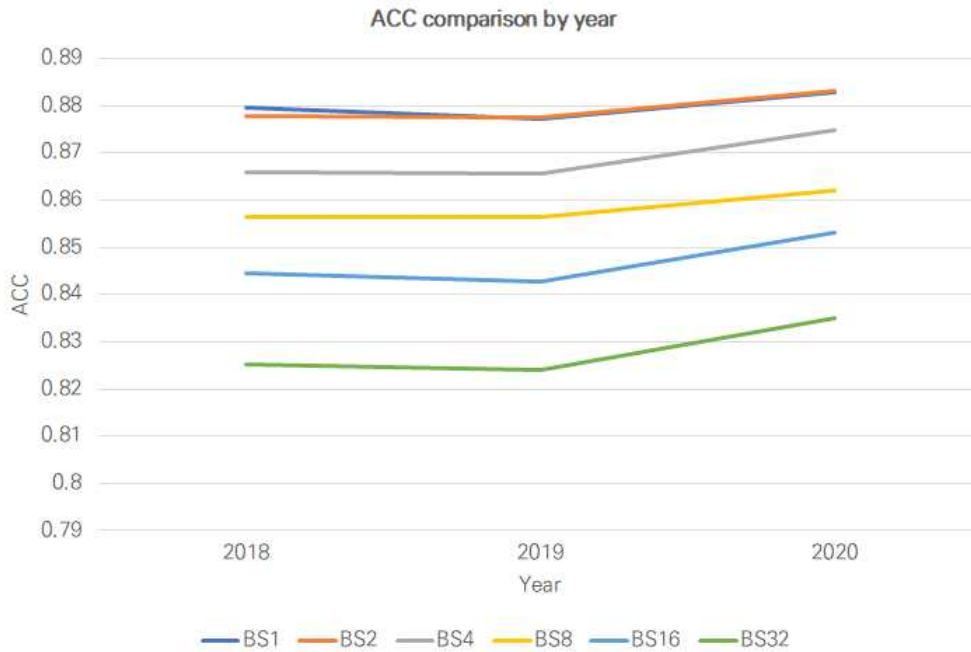


Figure 3.33. Plot of the comparison of the ACC by year.

전체 평가기간의 HSS 분석 결과, ECMWF는 F1과 F2에서 각 0.18, 0.08을 보였다. 딥러닝 모델은 F1에서 DL-SE 모델을 제외한 나머지 모델들이 ECMWF보다 좋은 성능을 보였다. F2에서는 DL-SE+BIGDATA 모델을 제외한 나머지 모델들이 ECMWF보다 낮은 성능을 보였다(Figure 3.34). 연도별 HSS 분석결과, F1에서 2018년, 2019년, 2021년은 DL-SE 모델을 제외한 모든 모델들이 높은 예측성을 보였으며, 2020년은 모든 딥러닝 모델이 좋은 성능을 보였다. 2022년은 모든 딥러닝 모델이 ECMWF보다 낮은 성능을 보였다(Figure 3.35). F2에서 2018년은 모든 딥러닝 모델이 높은 예측성을 보였으며, 2019년은 DL-SE, 2020년은 모든 딥러닝 모델이 낮은 성능을 보였다. 2021년은 DL-SE+BIGDATA와 DL-CBAM 모델이 ECMWF보다 높은 예측성을 보였으며, 2022년은 모든 딥러닝 모델이 낮은 예측성을 보였다(Figure 3.36).

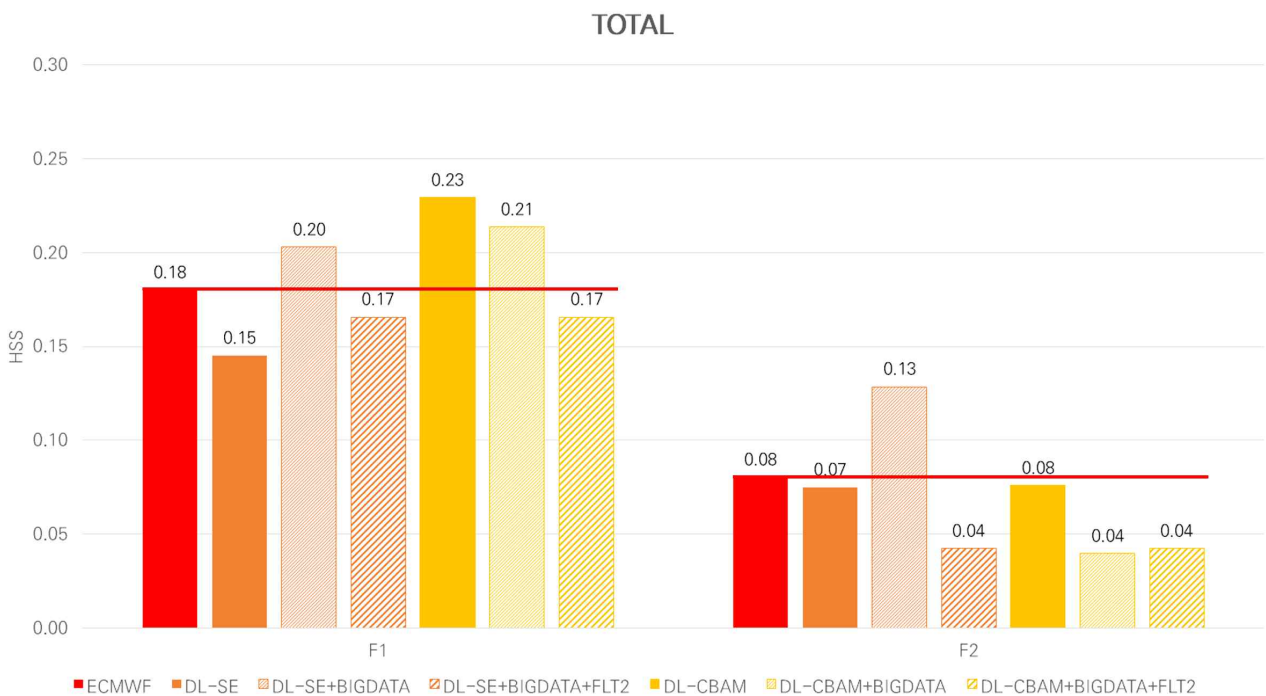


Figure 3.34. Summary of the HSS assessment of lead times F1(W+3) and F2(W+4) for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2.

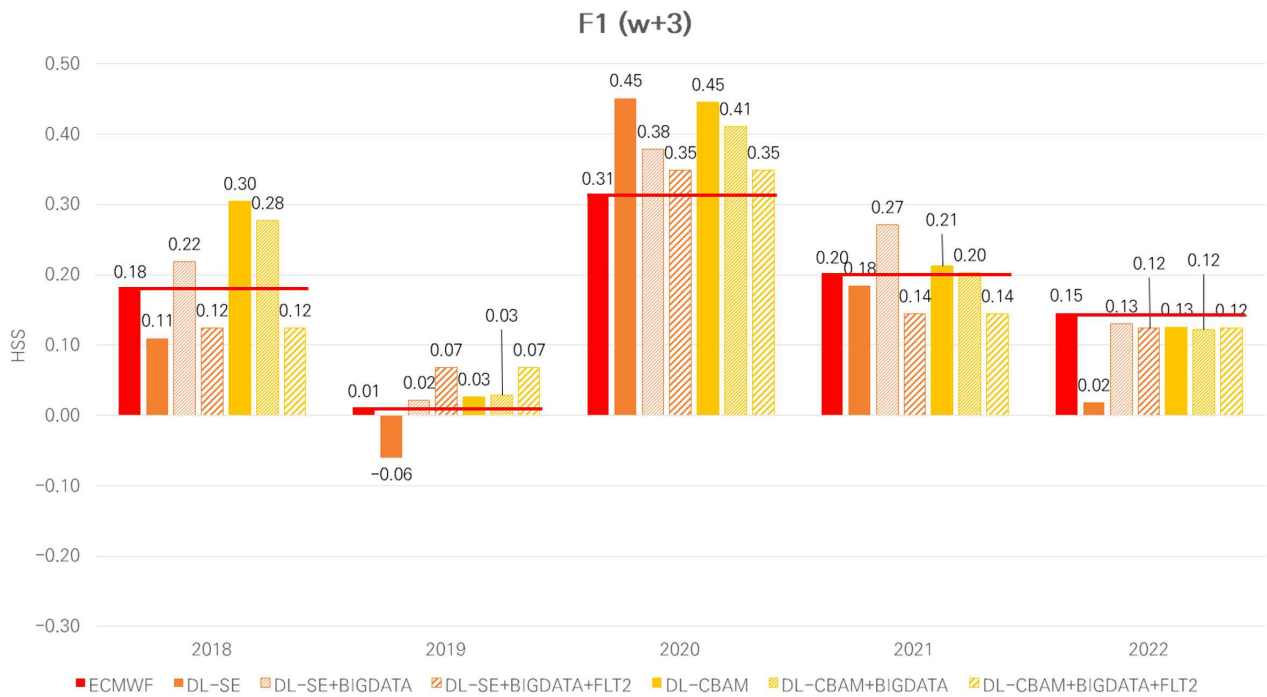


Figure 3.35. Summary of the HSS assessment of lead times F1(W+3) by year for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2.

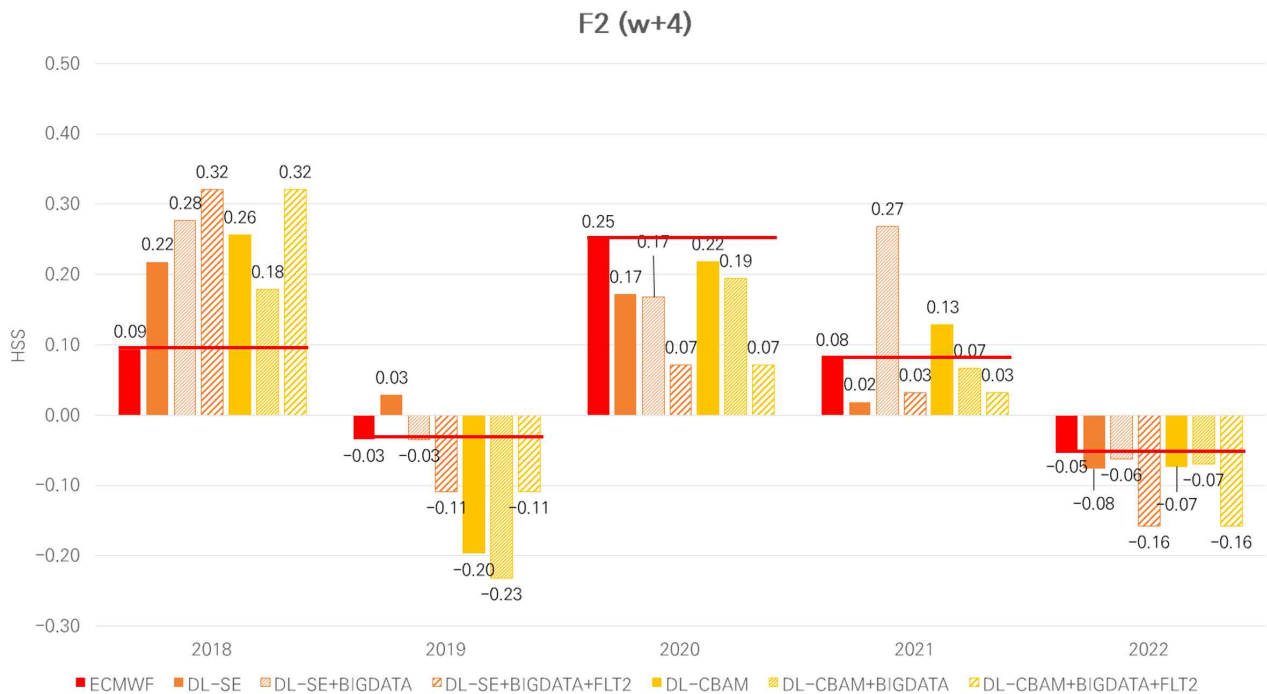


Figure 3.36. Summary of the HSS assessment of lead times F2(W+4) by year for 6 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2.

딥러닝 모델의 학습데이터셋, 어텐션 메커니즘, 레이어 필터 수 증가에 따른 모델별 피쳐 붕괴(feature collapse) 분석을 통해 모델의 정확도에 영향을 미치는 구간을 분석하였다. 모델의 분석에 앞서, 모델의 에포크 증가에 따른 모델의 피쳐붕괴 분석을 통해 학습 손실률의 감소와 피쳐 붕괴 간 상관관계에 대해 분석하였다. 분석 결과, 딥러닝 모델의 손실률 감소에 따라 모델의 피쳐 붕괴는 낮아지는 것으로 나타났다(Figure 3.37 및 3.38). 이는 피쳐 붕괴는 모델의 예측성에 영향을 미치는 것을 의미한다.

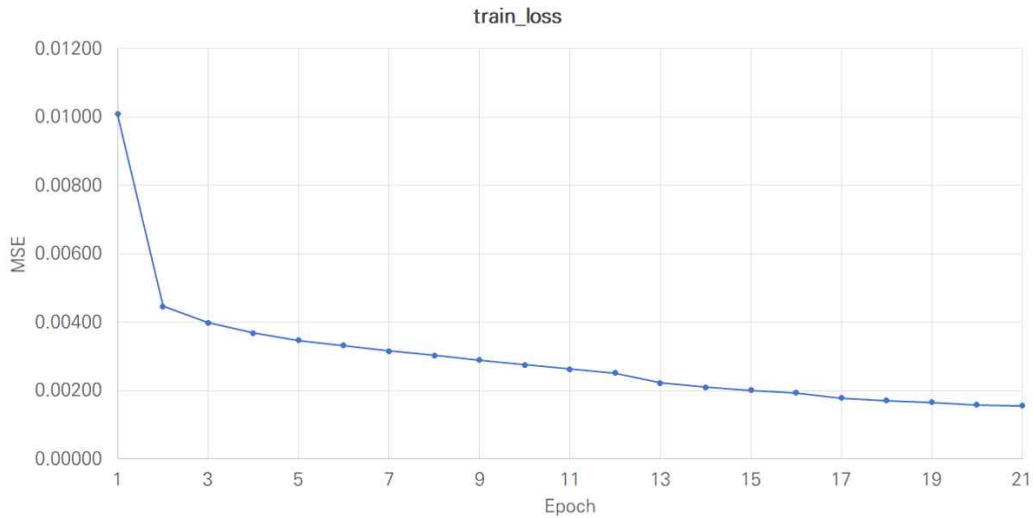


Figure 3.37. Line chart of training loss of deep learning model.

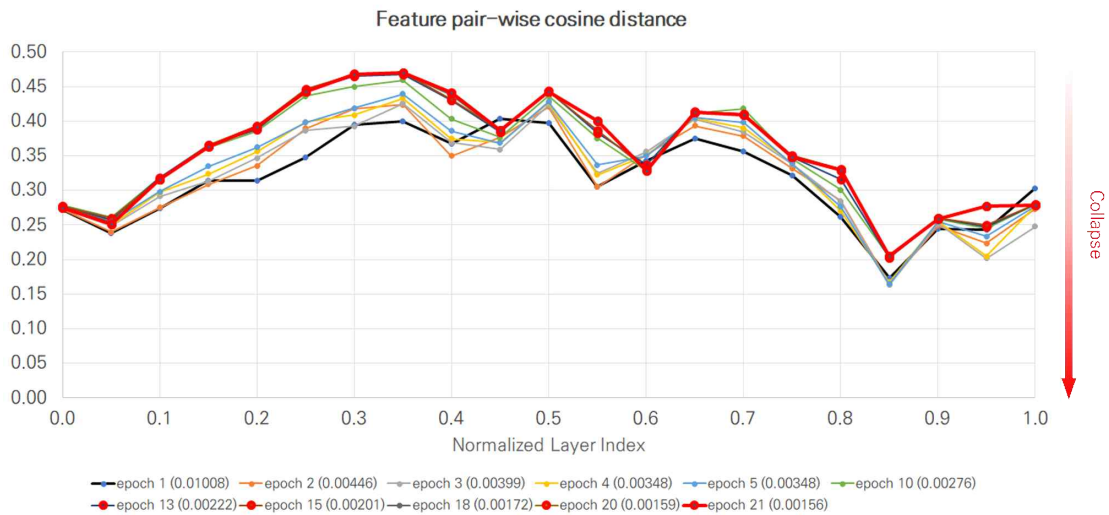


Figure 3.38. Feature pair-wise cosine distance analysis of deep learning model.

앞서 실험에 사용한 6개 모델의 주요 구간(2차원 합성곱 레이어, 활성화 레이어 등)에 대해 피쳐 붕괴 분석을 수행하였다(Figure 3.39 및 3.40). 피쳐 붕괴 분석에는 피쳐 코사인 거리

(feature cosine distance)가 사용되며, 거리가 가까울수록 피쳐 붕괴가 심화되는 것을 의미한다. 분석 결과, 어텐션 메카니즘 SE와 CBAM 모두 U-NET 아키텍처의 bottleneck 구간에서 일어나는 것으로 나타났다. 그래서 bottleneck 구간을 contrasting path와 expansive path에서 사용한 CCA(conv2d+conv2d+attention block) 블록을 이용하였다(Figure 3.41). CCA 블록은 FLT2가 적용된 두 모델(DL-SE+BIGDATA+FLT2, DL-CBAM-BIGDATA+FLT2)에 적용하고 학습 후 피쳐붕괴 분석을 하였다. Bottleneck 개선 전과 후의 피쳐 코사인 거리가 크게 개선되는 것을 보였다(Figure 3.42 및 3.43).

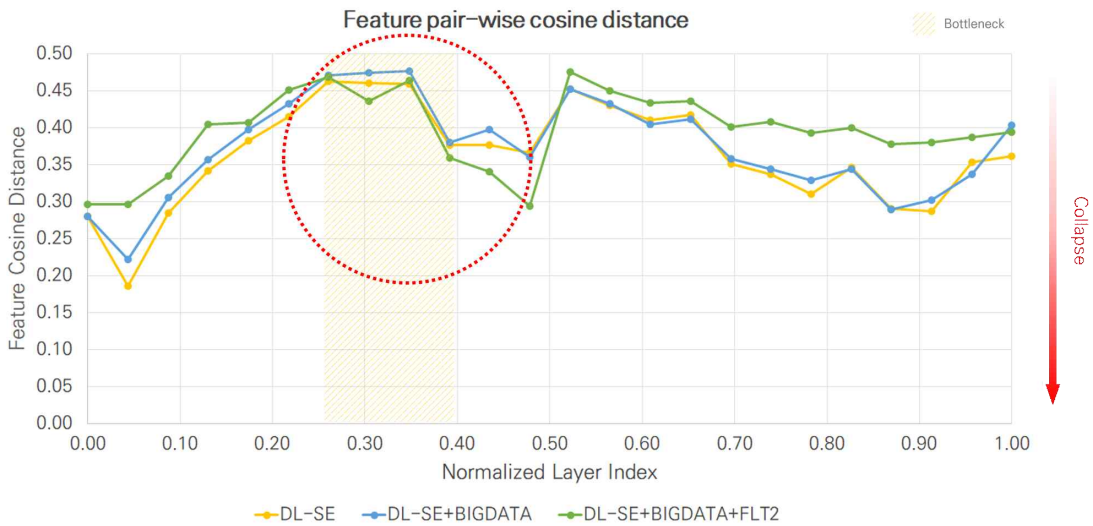


Figure 3.39. The results of the feature pair-wise cosine distance analysis of the deep learning models with the SE block.

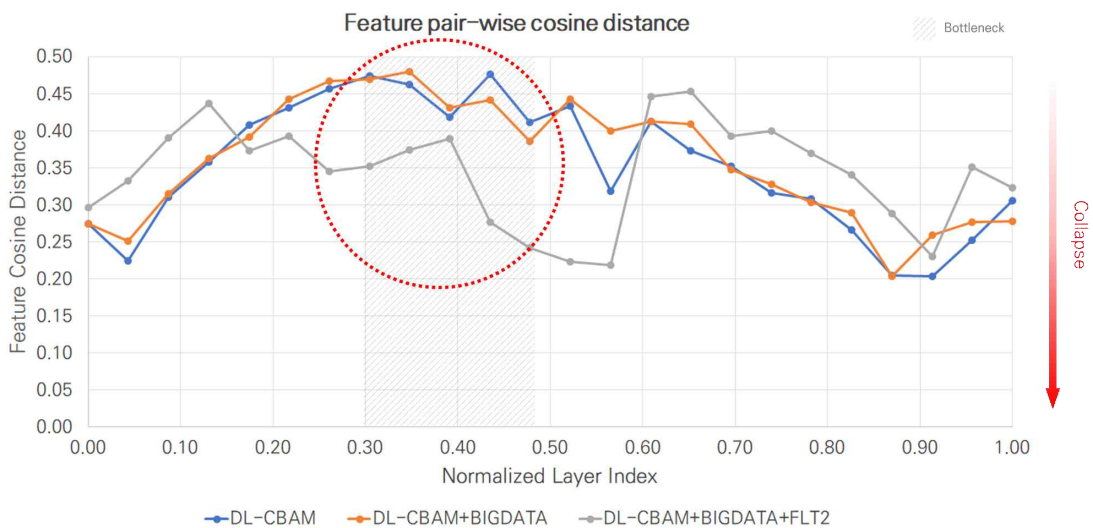


Figure 3.40. The results of the feature pair-wise cosine distance analysis of the deep learning models with the CBAM block.

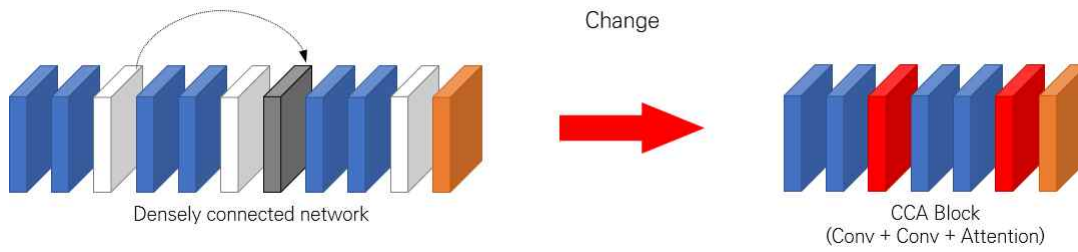


Figure 3.41. Design of CCA block.

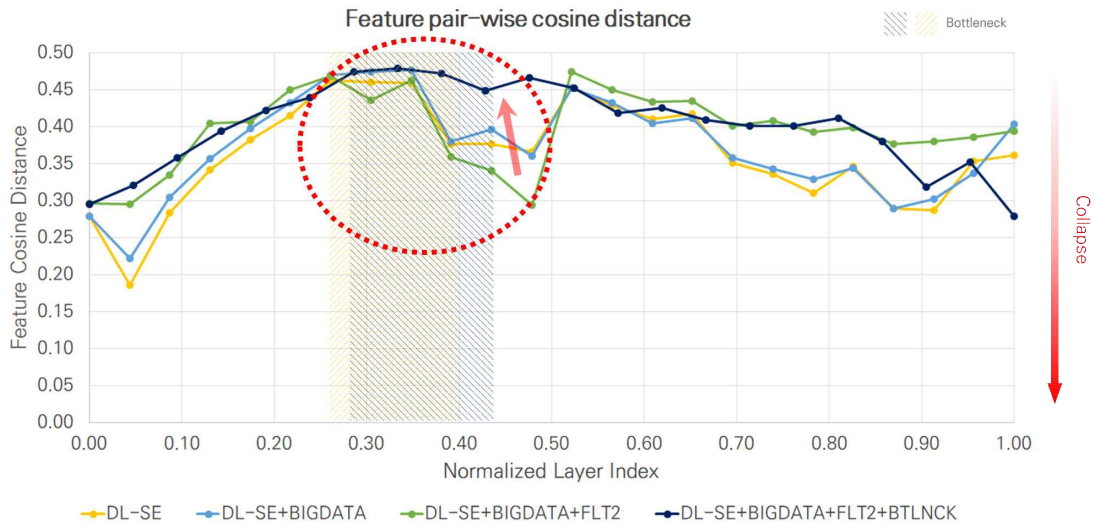


Figure 3.42. The results of the feature pair-wise cosine distance analysis of the deep learning models with the SE block after changing the bottleneck block.

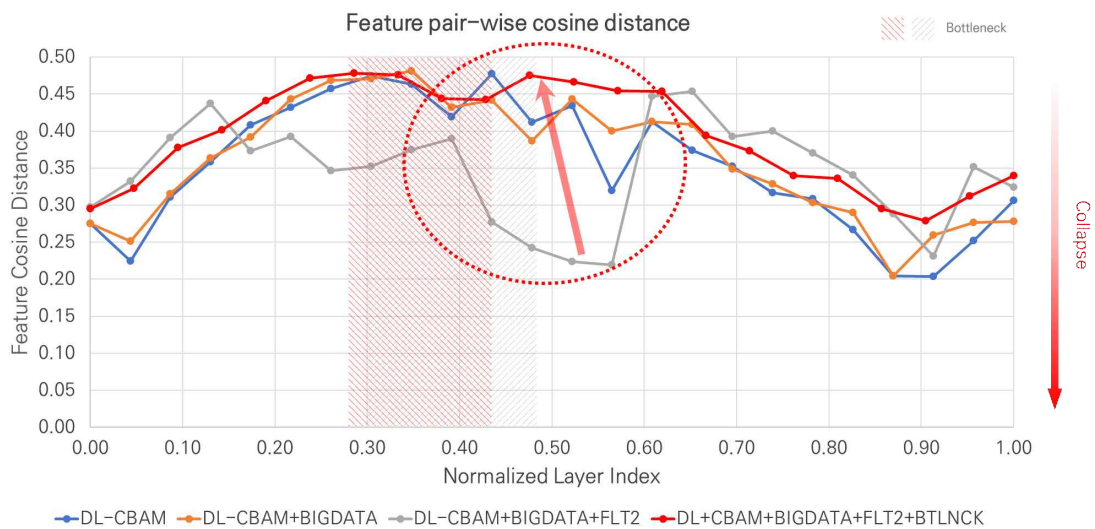


Figure 3.43. The results of the feature pair-wise cosine distance analysis of the deep learning models with the CBAM block after changing the bottleneck block.

Bottleneck이 개선된 모델을 이용하여 HSS 평가를 수행하였다. 전체 기간에 대해서는 F1에서 DL-SE+BIGDATA+FLT2는 0.17에서 0.25, DL-CBAM+BIGDATA+FLT2는 0.17에서 0.20, F2에서 DL-SE+BIGDATA-FLT2는 0.04에서 0.08, DL-CBAM+BIGDATA+FLT2는 0.04에서 0.05로 개선되었다(Figure 3.44). 예측 1주(F1, W+3)에서 2021년도 DL-CBAM+BIGDATA+FLT2 모델을 제외하고는 모두 개선되었으며, 2022년 DL-SE+BIGDATA+FLT2모델은 0.12에서 0.33으로 2배 이상 개선되었다(Figure 3.45). 예측 2주(F2, W+4)에서는 2020년 DL-SE+BIGDATA+FLT2, 2021년 DL-SE+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2, 2022년 DL-SE+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2 모델에서는 개선되었지만 그 외에서는 기존 모델보다 예측성이 떨어지는 경향을 보였다(Figure 3.46). 예측 2주는 ECMWF S2S 모델의 1-2주를 입력자료로 이용하여 14일을 예측한 결과 중 8~14일의 기간을 사용한다. 피쳐 붕괴 현상을 줄임으로써 예측 1주의 예측성을 높일 수 있지만 예측 기간이 길어질수록 모델의 예측성이 떨어지는 현상이 나타났다. 장기간의 예측성을 높일 수 있는 모델의 설계가 필요할 것으로 판단된다. 2021년 2월 8일 기준의 ECMWF S2S 예보자료를 기초로 8개의 모델의 예측결과는 Figure A1과 같다.

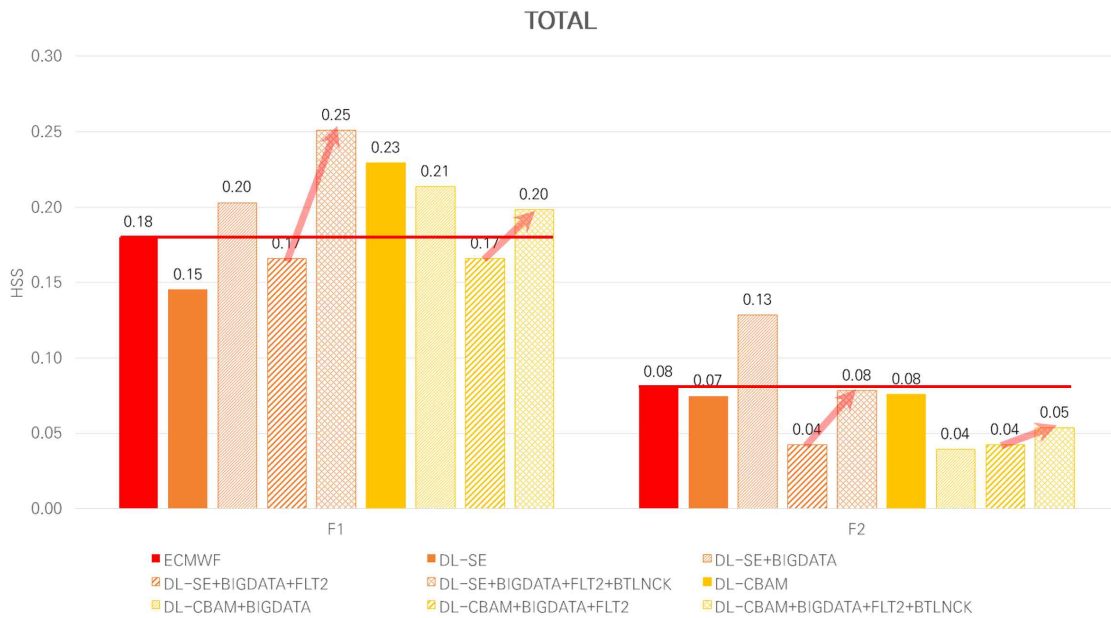


Figure 3.44. Summary of the HSS assessment of lead times F1(W+1) and F2(W+4) for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.

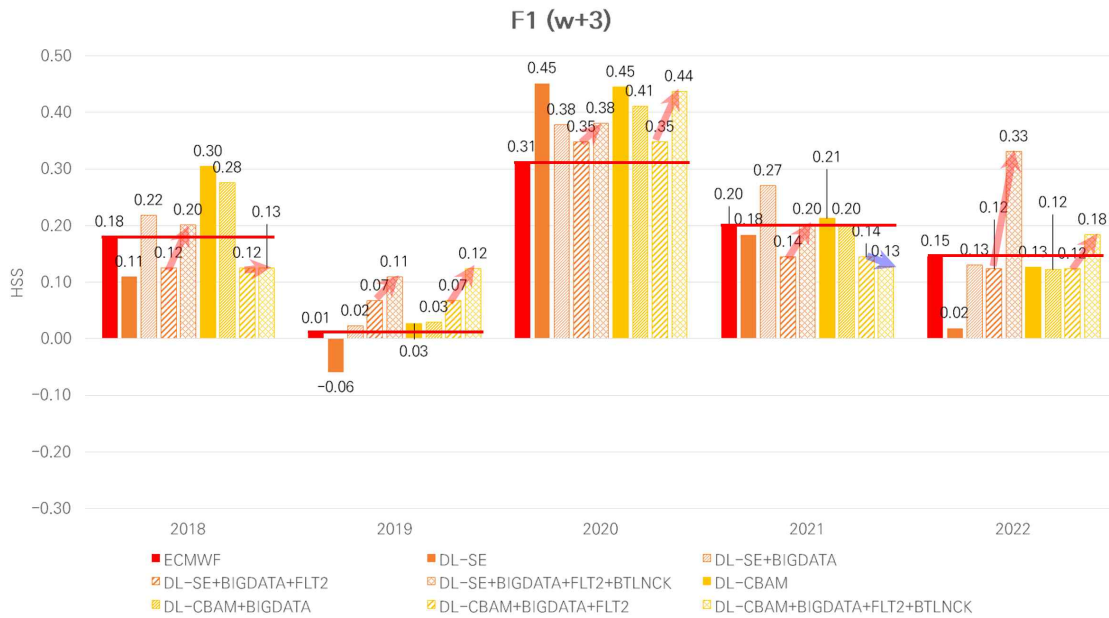


Figure 3.45. Summary of the HSS assessment of lead times F1(W+1) by year for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.

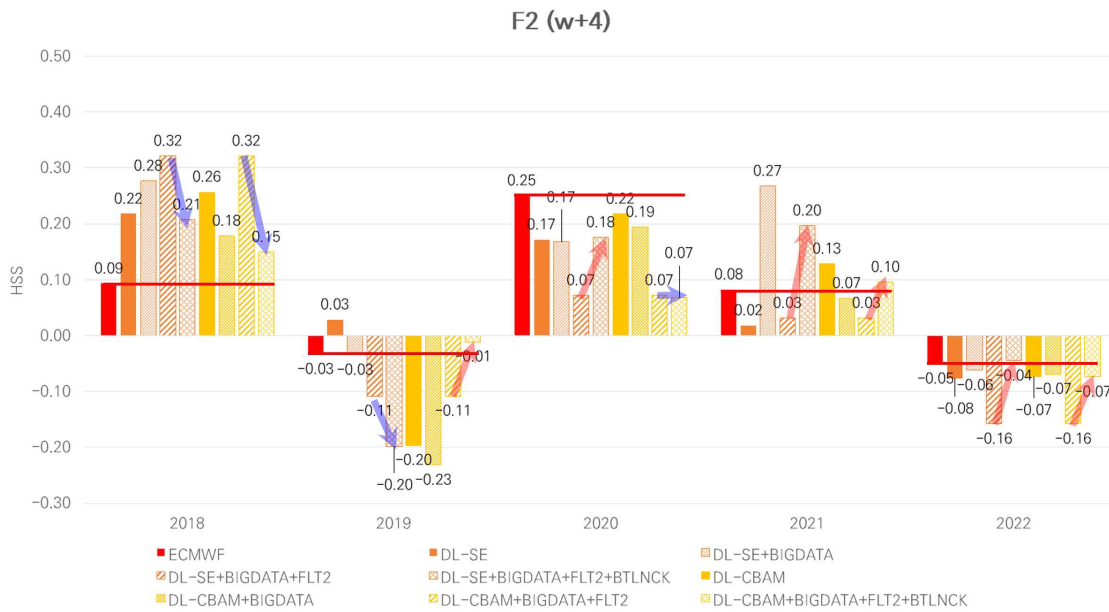


Figure 3.46. Summary of the HSS assessment of lead times F2(W+2) by year for 8 models: (from left)ECMWF, DL-SE, DL-SE+BIGDATA, DL-SE+BIGDATA+FLT2, DL-SE+BIGDATA+FLT2+BTLNCK, DL-CBAM, DL-CBAM+BIGDATA, DL-CBAM+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2+BTLNCK.

4. 요약 및 결론

동아시아 태평양 지역의 리드타임 3주에 대한 계절내 예측자료(최고온도) 정확도를 평가하기 위하여 모델자료 이외에 관측자료를 추가하고 Filter, Wrapper, Embedded 기법에 기반으로 민감도 분석을 실시하기 위하여였다. 딥러닝 모델은 U-Net모델에서 어텐션 매커니즘을 추가하여 보다 정확도를 향상시킨 모델인 Attention U-Net모델을 구성하였다. Attention U-Net모델의 어텐션 매커니즘은 입력자료에서 중요한 부분에 가중치를 더 부여하여 모델이 좀더 중요한 부분에서 깊게 학습하도록 해주기 때문에 파라미터를 최적화하여 모델을 고정하였다.

Filter 기법에서는 학습자료와 타겟자료와의 각각의 변수에 대한 상관성을 조사하여 상관성 높은 변수만 선택하여 학습한 결과 모든 조건하에서 EMCWF 모델의 최고온도 단독으로 사용한 결과가 학습결과보다 높게 나타났다. Wrapper기법에서는 모든 변수를 하나씩 채워서 학습한 결과 NDVI (OBS), SST (OBS)와 ECMWF 모델의 OLR, MSL, TMAX변수의 카테고리에서 학습한 최고온도의 ERA5최고온도 비교결과가 ECMWF 단독 모델 최고온도 결과보다 높게 나타났다. Embedded 기법은 EMCWF모델의 모든 변수를 이용하여 ERA5의 최고온도에 맞춰 회귀식으로 학습한 결과의 가장 높은 회귀계수를 가진 변수만을 선택하여 결과를 산출하였다. 분석결과 Filter 기법과 마찬가지로 ECMWF의 단독 최고온도 자료가 가장 정확도가 높다는 것을 알 수가 있었다. 이러한 민감도 분석결과를 볼 때 변수 선택만으로 정확도를 개선하기가 아직 어렵다는 것을 확인할 수 있었다. 향후 리드타임 3-4주 기간에 맞는 개선된 딥러닝 모델의 발전이 필요할 것으로 판단된다.

가중치 계산 및 전달 방법을 다양하게 변형할 수 있는 장치, 예를 들면, Attention 혹은 Residual을 기존 U-Net에 추가함으로써 다양한 U-Net 기반의 아키텍처를 구축하고 각각 아키텍처의 초매개변수를 최적화함으로써 S2S 예측 기후변수(예, 최고기온 및 강수)의 후보정 개선에 적합한 딥러닝 모델과 초매개변수 조합을 탐색하였다. 가중치 전달 및 계산 방법에 따라 Attention과 Residual 장치를 기존 U-Net에 각각 추가하여 Attention U-Net과 Residual U-Net을 구축하고, 이 두개의 장치를 모두 결합한 Attention based-on Residual U-Net을 준비하였다. 먼저, Attention의 목표는 현재 상태와 가장 연관이 깊은 값에 더욱 집중하는 것으로 입력에 따른 다음 결과를 예측하는 작업에서 특정 정보에 더 집중해야 하기 때문에 decoder에서의 어떤 값에 집중해야 하는지를 찾는 방법으로 그 관련성의 척도로 유사도를 사용한다. 그래서, 첫 번째 Attention U-Net은 decoder 부분에서 Attention 장치를 합성곱 레이어(convolutional layer)에 추가하였다. 두 번째는 Residual learning, 즉 잔차 학습은 레이어를 깊게 쌓을수록 기울기 소실(gradient vanishing)과 기울기 폭주(gradient exploding)이 쉽게 나타나서 딥러닝 학습이 쉽지 않다는 degradation 문제를 해결하기 위해서 제안되었는데, 본 연구에서 기존 합성곱 레이어의 input에 해당하는 값을 합성곱 레이어의 output에 다시 더해줌으로써 합성곱 레이어가 input값과 output 사이의 차이, 즉 잔차(residual)를 학습하는 데만 집중할 수 있도록 하는 기법이다. 그래서, 기존 U-Net의 전체 합성곱 레이어를 모두 residual block layer로 변경하였다. 또한, 원래 Residual block에서는 합성곱 레이어 직후에 배치 정규화를 두는데 본 연구에서는 배치 정규화를 사용하지 않았다. 배치 사이즈를 두는 목적은 과적합(overfitting)을 줄이기 위함인데, 본 연구에서는 오히려 배치 정규화로 인해 과소적합(underfitting)이 발생하는 것으로 추정되었기

때문에 배치 정규화를 제외하였다. 마지막 확장 U-Net은 이 두 장치를 그대로 결합하였다. 이렇게 준비된 4개 확장 U-Net의 초매개변수를 최적화하기 위해, 특정 범위 내에서 초매개변수를 일정 값만큼 일일이 변경하여 가장 높은 성능을 보이는 출력값을 순차적으로 모두 비교하기 때문에 계산 비용이 많이 든다는 단점이 있지만 모델 프리 기반으로 가장 적합한 초매개변수를 찾는 데에는 체계적이고 구축이 용이한 그리드 탐색(Grid Search) 알고리즘을 최적화 기법으로 선정하였다. 본 연구에서 설정한 초매개변수 조합은 에포크는 50, 100, 250, 500으로 4개로 설정하였고, 배치 사이즈는 64에서 시작하여 128, 256, 1024, 2048로 4의 배수로 5개로 설정하였다. 또한, 학습률은 0.001과 0.0001, 0.00001로 3가지로 설정하였다. 아키텍처에서 filter number, 즉 kernel size도 중요한 요소인데 본 연구에서는 32을 기준으로 설정하였다. 훈련 자료는 22년에 구축한 계절내 기후 예측자료를 활용하였다. 최적화한 결과, 대부분의 계절내 기후 모델에서 유사한 경향으로 나타났다. 최고기온과 강수에서, 배치사이즈 64 및 학습률 0.001, 에포크 100 조합에서 훈련 기간 동안 손실함수(Mean Square Error, MSE)가 감소하였다. 이렇게 선정된 조합으로 준비된 기존 U-Net과 확장 U-Net 3개에서 테스트 기간 동안 최고기온과 강수의 ACC (Anomaly Correlation Coefficient)를 비교한 결과, Attention 혹은 Residual이 추가된 U-Net에서 ACC는 향상되었는데, 최고기온의 경우, Attention과 Residual 두 개가 모두 장착된 Attention based-on Residual U-Net의 ACC가 다른 U-Net보다 향상되었다. 특히, 앞과 뒤의 차이를 활용하는 Residual 장치가 기온 예측 개선에 영향을 준 것으로 추정되는데, 기온 변수의 특성이 연속성, 즉 전날의 기온 특성이 다음 날의 기온 특성으로 연결되는 목표 변수의 특성 때문이 아닌가 추측된다. 강수의 경우는 Attention만 장착된 Attention U-Net에서 다른 U-Net보다 ACC가 향상되었는데, Attention은 현재 상태 혹은 특정 정보와 가장 연관이 깊은 값 (예, 유사도)에 집중하는 기법으로 불연속적이고 지역적 특성을 가지는 강수 변수의 특성에 Attention의 가중치 계산 및 전달 기법이 강수 예측 개선에 반영된 것으로 추측된다. 요약하면, Attention 및 Residual 장치를 기존 U-Net에 추가하여 아키텍처를 개선하고 그 아키텍처의 학습 성능을 높일 수 있도록 적합한 초매개변수를 탐색 및 선정함으로써 S2S 예측 최고기온과 강수의 예측 성능을 개선할 수 있었다. 그러나, 기온의 미흡한 1주 예측 성능과 강수의 2-3주 예측 성능을 크게 끌어올리지 못한 점 등은 여전히 남아 있는 문제이다. 따라서, 기 구축된 전처리 및 초매개변수 최적화 기법과 다양한 아키텍처들을 기반으로 후속 연구에서는 예측 성능이 높은 앙상블 멤버들을 탐색하고 이들을 믹싱할 수 있는 앙상블 기법 개발 통해 S2S 기온 및 강수 예측 자료의 후보정에 적합한 기법을 제시할 수 있을 것으로 기대한다.

본 연구는 준지도학습 기법을 적용하여 MJO 위상을 분류하는 이미지 기반 인공지능 모델을 개발하였다. 우선 준지도학습 기법을 적용하기 전 지도학습 기반 MJO 위상을 분류하는 이미지 기반 인공지능 모델 개발을 위해 MJO 인덱스 종류에 따른 입력 변수 조합, 인공지능 모델 구조, 훈련 자료 분리 방법 등을 테스트하여 MJO 위상을 분류하는 최적의 인공지능 모델을 개발하였다. 이후 준지도학습 환경에서 MJO 위상 분류 인공지능 모델을 개발하였다. MJO 인덱스 종류에 따른 입력 변수 조합, 라벨링된 자료의 개수 차이, 모델 학습 방법 등에 따라 준지도학습 기반 모델의 민감도를 분석하였고 지도학습 기반의 모델과 비교 평가하였다.

준지도학습의 적용은 MJO 위상 분류 문제에 유용한 것으로 나타났다. 라벨링된 자료의 양을 줄인 후에도 MJO 위상 분류에 중요한 기후 패턴이나 특징들을 효과적으로 학습하고 테스트 평가에서 더 많은 라벨링을 사용하거나 모든 자료를 라벨링해 사용하는 지도학습의 결과

에 비해서 비슷하거나 더 높은 정확도를 보였다. VPM 인덱스 기반 결과에서 여름철 기간에 여러가 증가하는 경향을 보였는데 이는 MJO 인덱스의 종류에 따라서 MJO 위상 분류에 여러 기 후 변동성을 학습하는데 차이가 있는 것을 확인하였고 딥러닝 기반 MJO 관련 연구를 수행할 때 이러한 요소들을 고려하는 것이 필요하다는 것을 시사한다. 준지도학습 알고리즘 내 여러 자료 증강 방법에 따른 민감도 실험에서 수평 이동하는 MJO의 특성에 따라 수평적으로 이미지를 반전하는 자료 증강 방법은 오히려 정확도를 낮추는 결과를 보였고 한편 수직 반전 방법은 성능 향상에 도움이 됐다. 입력 자료의 이미지를 변형하는 다양한 증강 방법들은 공간적인 특징이 중요한 기후 자료에 적용할 때 목적변수의 특성에 맞게 적절히 적용되면 가용한 자료가 제한적일 때 유용하게 활용될 수 있을 것이다.

추가적으로 필터링된 MJO 입력장 아노말리 이미지를 시계열 자료로 활용하여 입력장을 예측하는 시공간 예측 인공지능 모델링을 수행하였다. MJO의 발생 주기에 따라 최소 7일부터 최대 120일의 이전 시계열 자료를 기반으로 학습하고 그다음 날 및 연속 7일 예측을 수행했다. 바로 다음 날의 예측은 비교군과 비슷한 결과를 산출했지만, 시간이 갈수록 공간적인 패턴이 무너지는 것을 확인하였다. Inactive MJO 발생 사례를 제거하거나 여름철을 제거하는 등의 다양한 샘플링 방법을 통해서 테스트를 수행하였지만, 큰 예측 차이를 보이지 않았다. 실험을 통해 현재 활용한 변수가 이미지의 해당 날짜의 시점 정보이기 때문에 향후 SST나 비습 등의 MJO 발달 전후의 상태를 반영하는 변수를 추가하여 예측 성능을 향상시킬 수 있을 것으로 기대된다.

본 연구는 MJO 위상을 분류하는 데 있어 더 적은 이미지 자료만을 활용하는 준지도학습 기술의 적용 가능성을 보여준다. 이는 자료의 양이 제한적인 기후 분야의 다양한 연구에 적합한 기술이 될 수 있다. 본 연구에서 제시하는 모델은 상대적으로 단순하고 독립적인 실행 및 적용으로 더 다양한 기후 인자나 다른 기후 연구에 다양하게 활용될 수 있으며 더 넓은 연구 지역으로의 확장이나 다양한 시간 규모에 적용될 수 있을 것이다.

MJO 위상 분류 모델에 설명가능한 인공지능 기법 중 occlusion 기법을 적용하여 준지도학습 전과 후 모델의 변화를 분석하였다. occlusion 기법은 확률 변화의 민감도 히트맵을 제공해주며, 준지도학습 전과 후의 히트맵 분석 결과 큰 차이를 보이지 않았다. 이는 준지도학습을 적용하더라도 모델이 레이블된 자료로 학습하는 지도학습과 같은 성능을 보일 수 있음을 의미한다. 준지도학습을 통해 모델이 좀 더 디테일하게 입력자료를 해석하고 예측이 가능하다.

본 연구에서는 딥러닝 기반 1개월 기온예측 모델을 이용하여 손실지형 분석, 모델 내부 구조 분석 및 개선을 수행하였다. 손실지형 분석을 통해 딥러닝 모델의 과적합과 과소적합을 판단할 수 있으며, 모델의 구조가 전역 최솟값을 찾는데 유리한 지형인지 판단이 가능하다. 기온 예측 모델은 완만한 손실지형을 보임으로써 전역최저점을 찾는데 쉬운 구조임을 보였다.

딥러닝 모델 내부 구조 분석 및 개선을 통해 1개월 기온예측 모델의 예측성 향상을 도모하고자 하였다. 기초 모델을 토대로 어텐션 메카니즘, 데이터확장, 레이어 필수 수 조정 등을 조합하여 실험 모델을 설계하였다. 평가는 KMA ASOS 관측자료로 계산한 남한지역의 3분위 확률값을 ECMWF S2S 예측자료, 딥러닝 모델의 예측자료와 각각 HSS 기법을 이용하여 분석하고 비교하였다. 비교 결과, 2018년~2022년 전체 평가기간의 예측 1주에서 DL-SE 모델을 제외한

DL-SE+BIGDATA, DL-CBAM, DL-CBAM+BIGDATA 모델이 비교대상인 ECWMF 보다 높은 예측성을 보였으며, F2에서는 DL-SE+BIGDATA가 ECMWF보다 높은 예측성을 보였다. 딥러닝 모델의 예측성이 떨어지는 현상을 분석하기 위해 피쳐 붐비 분석을 이용하여 모델의 주요구간을 분석하였다. 분석 결과, 1개월 기온예측 모델에 적용된 U-NET 아키텍처의 bottleneck 구간에서 피쳐 붐비 현상이 일어나는 것으로 나타났다. 그래서 bottleneck 구간을 contrasting path, expansive path 파트에서 사용 중인 CCA(conv2d+conv2d+attention block) 블록으로 개선하고 모델의 예측성을 기존 모델과 비교평가하였다. 전체 기간에 대해서는 F1에서 DL-SE+BIGDATA+FLT2는 0.17에서 0.25, DL-CBAM+BIGDATA+FLT2는 0.17에서 0.20, F2에서 DL-SE+BIGDATA+FLT2는 0.04에서 0.08, DL-CBAM+BIGDATA+FLT2는 0.04에서 0.05로 개선되었다. 예측 1주(F1, W+3)에서 2021년도 DL-CBAM+BIGDATA+FLT2 모델을 제외하고는 모두 개선되었으며, 2022년 DL-SE+BIGDATA+FLT2모델은 0.12에서 0.33으로 2배 이상 개선되었다. 예측 2주(F2, W+4)에서는 2020년 DL-SE+BIGDATA+FLT2, 2021년 DL-SE+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2, 2022년 DL-SE+BIGDATA+FLT2, DL-CBAM+BIGDATA+FLT2 모델에서는 개선되었지만 그 외에서는 기존 모델보다 예측성이 떨어지는 경향을 보였다. 이는 모델의 예측기간이 길어질수록 예측성이 낮아지는 현상으로 향후 장기간의 예측성을 높일 수 있는 모델의 설계가 필요할 것으로 판단된다.

사사

본 연구는 APCC의 지원을 통해 수행되었습니다.

References

- 이성규, 2023: AI를 이용한 기후예측 기술 개발, 2023년도 제4차 APCC 기후예측워크숍.
- Alber, M., S. Lapuschkin, P. Seegerer, M. Hägele, K.T. Schütt, G. Montavon, W. Samek, K.R. Müller, S. Dähne, P.-J. Kindermans, 2019: iNNvestigate neural networks!, *Journal of Machine Learning Research*, **20**(93), 1-8.
- Ali, S., T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. D. Ser, N. Díaz-Rodríguez, F. Herrera, 2023: Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, **99**, 101805. <https://doi.org/10.1016/j.inffus.2023.101805>.
- Bahdanau, D., K. Cho, and Y. Bengio, 2016: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473, <https://doi.org/10.48550/arXiv.1409.0473>.
- Bi, K., L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, 2023: Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, **619**, 533-538 (2023). <https://doi.org/10.1038/s41586-023-06185-3>.
- Bonev, B., T. Kurth, C. Hundt, J. Pathak, M. Baust, K. Kashinath, A. Anandkumar, 2023: Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere. <https://doi.org/10.48550/arXiv.2306.03838>.
- Cai, S., Y. Tian, H. Lui, H. Zeng, Y. Wu, and G. Chen, 2020: Dense-UNet: a novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network. *Quantitative Imaging in Medicine and Surgery*, **106**, 1275-1285.
- Chae, S., J. Shin, S. Kwon, S. Lee, S. Kang, and D. Lee, 2021: PM10 and PM2.5 real-time prediction models using an interpolated convolutional neural network. *Scientific Reports*, **11**, 11952, <https://doi.org/10.1038/s41598-021-91253-9>.
- Chae, S., S. Jang, S. Lee, and D. Lee, 2020: Complex System Analysis of Korean Peninsula Earthquake Data. *Scientific Reports*, **10**, 2686, <https://doi.org/10.1038/s41598-020-59619-7>.
- Chen, T., S. Son, A. Park, and S.-J. Baek, 2021: ROI Detection of Spectral Signal using Deep Learning Model Combining ResNet and Unet. *Journal of Korea Academia-Industrial Cooperation Society*, **22**, 9-16, <https://doi.org/10.5762/kais.2021.22.12.9>.
- Chung, U., J. Rhee, M. Kim, S.-J. Sohn, 2023: Investigating data preprocessing algorithms of a deep learning postprocessing model for improvement of sub-seasonal to seasonal climate predictions. *Korean Journal of Agricultural and Forest Meteorology*, **25**, 80-98, <https://doi.org/10.5532/KJAFM.2023.25.2.80>.

- Dawson, A., 2016: Windspharm: A high-level library for global wind field computations using spherical harmonics. *Journal of Open Research Software*, **4**(1), <https://doi.org/10.5334/jors.129>.
- de Andrade, F. M., C. A. S. Coelho, and I. F. A. Cavalcanti, 2019: Global precipitation hindcast quality assessment of the Subseasonal to Seasonal (S2S) prediction project models. *Climate Dynamics*, **52**, 5451–5475, <https://doi.org/10.1007/s00382-018-4457-z>.
- ECMWF, 2020: Retrieved from document. <https://doi.org/https://www.ecmwf.int/>.
- ECMWF, 2023: ECMWF unveils alpha version of new ML model, Accessed 14 December 2023, <https://ecmwf.int/en/about/media-centre/aifs-blog/2023/ECMWF-unveils-alpha-version-of-new-ML-model>.
- EU, 2019: Ethics guidelines for trustworthy AI. Accessed on 12 December 2023, <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>.
- Hall, M. 1999: *Correlation-based Feature Selection for Machine Learning*, PhD thesis. University of Waikato
- He, K., X. Zhang, S. Ren, and J. Sun, 2016: Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition*, 2016, 770–778, <https://doi.org/10.1109/CVPR.2016.90>.
- Hersbach, H., and Coauthors, 2020: The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730), 1999–2049, <https://doi.org/10.1002/qj.3803>.
- Hinton, G. E., S. Osindero, and Y.-W. Teh, 2006: A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554, <https://doi.org/10.1162/neco.2006.18.7.1527>.
- Hu, J., L. Shen, S. Albanie, G. Sun, E.H., Wu, 2020: Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**(8), 2011–2023. <https://doi.org/10.1109/TPAMI.2019.2913372>
- Huang, G., Z. Liu, L. Van Der Maaten, 2017: Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*, pp.4700–4708.
- Ju, H.-J., J.-Y. CHAE, E.-J. LEE, Y.-T. KIM, and J.-H. Park, 2022: Impacts of Seasonal and Interannual Variabilities of Sea Surface Temperature on its Short-term Deep-learning Prediction Model Around the Southern Coast of Korea. *The Sea. Journal of the Korean Society of oceanography*, **27**, 49–70. <https://doi.org/10.7850/JKSO.2022.27.2.049>.
- Kang, D., D. Kim, S. Rushley, and E. Maloney, 2022: Seasonal locking of the MJO’s southward detour of the Maritime Continent: The role of the Australian monsoon. *Journal*

of Climate, **35**(24), 8153–8168, <https://doi.org/10.1175/JCLI-D-22-0234.1>.

- Kiladis, G. N., Dias, J., Straub, K. H., Wheeler, M. C., Tulich, S. N., Kikuchi, K., Weickmann, K. M., and Ventrice, M. J., 2014: A comparison of OLR and circulation-based indices for tracking the MJO. *Monthly Weather Review*, **142**(5), 1697–1715, <https://doi.org/10.1175/MWR-D-13-00301.1>.
- Kingma, D. P., and Ba, J., 2014: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, <https://doi.org/10.48550/arXiv.1412.6980>.
- Laine, S., and T. Aila, 2016: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242, <https://doi.org/10.48550/arXiv.1610.02242>.
- Lam R., A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia, 2023: Learning skillful medium-range global weather forecasting. *Science*. <https://doi.org/10.1126/science.adi2336>.
- Le, P. T., B.-T. Pham, C.-C. Chang, Y.-C. Hsu, T.-C. Tai, Y.-H. Li, and J.-C. Wang, 2023: Anti-Aliasing Attention U-net Model for Skin Lesion Segmentation. *Diagnostics*, **13**, <https://doi.org/10.3390/diagnostics13081460>.
- Lee, D. H., 2013: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, *ICML*, **3**(2), 896.
- Li, Hao, Z. Xu, G. Taylor, C. Studer, T. Goldstein, 2018: Visualizing the Loss Landscape of Neural Nets. *Neural Information Processing Systems (NIPS) 2018*. arXiv:1712.09913 <https://arxiv.org/abs/1712.09913>
- Liebmann, B., and C. A. Smith, 1996: Description of a complete (interpolated) outgoing longwave radiation dataset. *Bulletin of the American Meteorological Society*, **77**(6), 1275–1277.
- Liu, Y., C. Gong, L. Yang, and Y. Chen, 2019: DSTP-RNN: a dual-stage two-phase attention-based recurrent neural networks for long-term and multivariate time series prediction. arXiv:1904.07464. <https://doi.org/10.48550/arXiv.1904.07464>.
- Ma, X., C. Qin, H. You, H. Ran, and Y. Fu, 2022: Rethinking network design and local geometry in point cloud: a simple residual MLP framework. *International Conference on Learning Representations (ICLR) 2022*, arXiv:2202.07123
- Maclachlan, C., and A. Arribas, K. A. Peterson, A. Maidens, D. Fereday, A. A. Scaife, M. Gordon, M. Vellinga, A. Williams, R. E. Coner, J. Camp, P. Xavier, and G. Madec, 2015:

- Global Seasonal Forecast System version 5 (GloSea5): A high-resolution seasonal forecast system. *Quarterly Journal of the Royal Meteorological Society*, **141**, 1072–1084, <https://doi.org/10.1002/qj.2396>.
- Maji, D., P. Sigedjar, and M. Singh, 2022: Attention Res-UNet with Guided Decoder for semantic segmentation of brain tumors. *Biomedical Signal Processing and Control*, **71**, 103077, <https://doi.org/10.1016/j.bspc.2021.103077>.
- Martin, Z. K., E. A. Barnes, and E. Maloney, 2022: Using Simple, Explainable Neural Networks to Predict the Madden-Julian Oscillation. *Journal of Advances in Modeling Earth Systems*, **14**(5), <https://doi.org/10.1029/2021MS002774>.
- Miyato, T., S. I. Maeda, M. Koyama, and S. Ishii, 2018: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, **41**(8), 1979–1993, <https://doi.org/10.1109/TPAMI.2018.2858821>.
- Mu, N., Z. Lyu, M. Rezaeitaleshmahalleh, J. Tang, and J. Jiang, 2023: An attention residual u-net with differential preprocessing and geometric postprocessing: Learning how to segment vasculature including intracranial aneurysms. *Medical Image Analysis*, **84**, 102697, <https://doi.org/10.1016/j.media.2022.102697>.
- Nguyen, T., J. Brandstetter, A. Kappor, J. K. Gupta, and A. Grover, 2023: ClimaX: A foundation model for weather and climate. <https://doi.org/10.48550/arXiv.2301.10343>
- NIA, 2018: EU의 인공지능 新 규제메카니즘: ‘설명가능 인공지능(XAI)’ , NIA Special Report 2018-3, 2018.3.15.
- Okatay, O., J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y Hammerla, B. Kainz, B. Glocker, and D. Rueckert, 2023: Attention U-Net: Learning Where to Look for the Pancreas. <https://doi.org/10.48550/arXiv.1804.03999>
- Oktay, O., J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y Hammerla, B. Kainz, B. Glocker, and D. Rueckert, 2018: Attention U-Net: Learning Where to Look for the Pancreas. [arXiv:1804.03999](https://doi.org/10.48550/arXiv.1804.03999), <https://doi.org/10.48550/arXiv.1804.03999>.
- Ordookhanians, A., X. Li, and S. Nakandala, 2019: Demonstration of Krypton: Optimized CNN Inference for Occlusion-based Deep CNN Explanations. *PVLDB*, **12**(1), 1894–1897. <https://doi.org/10.14778/3352063.3352093>
- Ouali, Y., C. Hudelot, and M. Tami, 2020: An overview of deep semi-supervised learning. arXiv preprint [arXiv:2006.05278](https://doi.org/10.48550/arXiv.2006.05278), <https://doi.org/10.48550/arXiv.2006.05278>.

- Pham, H., Z. Dai, Q. Xie, and Q. V. Le, 2021: Meta pseudo labels. In Proceedings of *the IEEE/CVF conference on computer vision and pattern recognition*, 11557–11568.
- Qin, Y., D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, 2017: A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. arXiv:1704.02971, <https://doi.org/10.48550/arXiv.1704.02971>.
- Reed, S., K. Zolna, Em. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, 2022: A Generalist Agent. arXiv:2205.06175, <https://doi.org/10.48550/arXiv.2205.06175>.
- Reynolds, R. W., N. A. Rayner, T. M. Smith, D. C. Stokes, and W. Wang, 2002: An improved in situ and satellite SST analysis for climate. *Journal of Climate*, **15**, 1609–1625, [https://doi.org/10.1175/1520-0442\(2002\)015<1609:AIISAS>2.0.CO;2](https://doi.org/10.1175/1520-0442(2002)015<1609:AIISAS>2.0.CO;2).
- Ronneberger, O., P. Fischer, and T. Brox, 2015: U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.0459, <https://doi.org/10.48550/arXiv.1505.04597>.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986: Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, 318–362.
- Scaife, A. A., A. Arribas, E. Blockley, A. Brookshaw, R. T. Clark, N. Dunstone, R. Eade, D. Fereday, C. K. Folland, M. Gordon, L. Hermanson, J. R. Knight, D. J. Lea, C. MacLachlan, A. Maidens, M. Martin, A. K. Peterson, D. Smith, M. Vellinga, E. Wallace, J. Waters, and A. Williams, 2014: Skillful long-range prediction of European and North American winters. *Geophysical Research Letters*, **41**, 2514–2519, <https://doi.org/10.1002/2014GL059637>.
- Shi, X., Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, 2015: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28.
- Shih, S. Y., F. K. Sun, and H. yi Lee, 2019: Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, **108**, 1421–1441, <https://doi.org/10.1007/s10994-019-05815-0>.
- Shin, S., S. H. Lee, and H. H. Han, 2021a: A Study on Residual U-Net for Semantic Segmentation based on Deep Learning. *Journal of Digital Convergence*, **19**, 251–257. <https://doi.org/10.14400/JDC.2021.19.6.251>.
- Shin, S., S. H. Lee, and H. H. Han, 2021b: Atrous Residual U-Net for Semantic Segmentation in Street Scenes based on Deep Learning. *Journal of Convergence for Information Technology*, **11**, 45–52, <https://doi.org/10.22156/CS4SMB.2021.11.10.045>.

- Simonyan, K., and Zisserman, A., 2014: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, <https://doi.org/10.48550/arXiv.1409.1556>.
- Singh, M., B. Kumar, S. Rao, S. S. Gill, R. Chattopadhyay, R. S. Nanjundiah, and D. Niyogi, 2021: Deep learning for improved global precipitation in numerical weather prediction systems. arXiv:2106.12045, <https://doi.org/10.48850/arXiv.2106.12045>.
- Tan, C., Z. Gao, S. Li, and S. Z. Li, 2022: Simvp: Towards simple yet powerful spatiotemporal predictive learning. arXiv preprint arXiv:2211.12509, <https://doi.org/10.48550/arXiv.2211.12509>.
- Tarvainen, A., and H. Valpola, 2017: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* 30.
- Trabelsi, I., B. Zeddini, M. Zolghadri, M. Barkallah and M. Haddar, 2021: Obsolescence Prediction based on Joint Feature Selection and Machine Learning Techniques. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence – Volumn 2: ICAART*, ISBN 978-989-758-484-8, 787-794, <https://doi.org/10.5220/0010241407870794>.
- van der Velden, B. H.M., H. J. Kuijf, K. G.A Gilhuijs, and M. A. Viergever, 2022: Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, **79**, 102470. <https://doi.org/10.1016/j.media.2022.102470>.
- Ventrice, M. J., M. C. Wheeler, H. H. Hendon, C. J. Schreck, C. D. Thorncroft, and G. N. Kiladis, 2013: A modified multivariate Madden-Julian oscillation index using velocity potential. *Monthly Weather Review*, **141**(12), 4197-4210, <https://doi.org/10.1175/MWR-D-12-00327.1>.
- Vitart, F., and A. W. Robertson, 2018: The sub-seasonal to seasonal prediction project (S2S) and the prediction of extreme events. *npj Climate and Atmospheric Science*, **1**, 3. <https://doi.org/10.1038/s41612-018-0013-0>.
- Wang, Y., H. Wu, J. Zhang, Z. Gao, J. Wang, S. Y. Philip, and M. Long, 2022: Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**(2), 2208-2225, <https://doi.org/10.48550/arXiv.2103.09504>.
- Weyn, J. A., D. R. Durran, R. Caruana, and N. Cresswell-Clay, 2021: Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, **13**, <https://doi.org/10.1029/2021MS002502>.
- Wheeler, M. C., and H. H. Hendon, 2004: An all-season real-time multivariate MJO index:

Development of an index for monitoring and prediction. *Monthly weather review*, **132**(8), 1917–1932, [https://doi.org/10.1175/1520-0493\(2004\)132<1917:AARMMI>2.0.CO;2](https://doi.org/10.1175/1520-0493(2004)132<1917:AARMMI>2.0.CO;2).

Woo, S., J. Park, J. Lee, and I.S. Kweon, 2018: CBAM: Convolutional block attention module. In: Computer Vision – ECCV 2018. *Lecture Notes in Computer Science*, **11211**. 32–19. https://doi.org/10.1007/978-3-030-01234-2_1

Woo, S., S. Debnath, R. Hu, X. Chen, Z. Liu, I.S. Kweon, and S. Xie, 2023: ConvNeXt V2: Co-designing and Scalling ConvNets with Masked Autoencoders, arxiv:2301.00808.

Zhang, J., X. Lv, H. Zhang, and B. Liu, 2020: AResU-Net: Attention Residual U-Net for Brain Tumor Segmentation. *Symmetry*, **12**, <https://doi.org/10.3390/sym12050721>.

Zhou, C., H. Li, C. Yu, J. Xia, and P. Zhang, 2022: A station-data-based model residual machine learning method for fine-grained meteorological grid prediction. *Applied Mathematics and Mechanics*, **43**, 155–166, <https://doi.org/10.1007/s10483-022-2822-9>.

부록 A

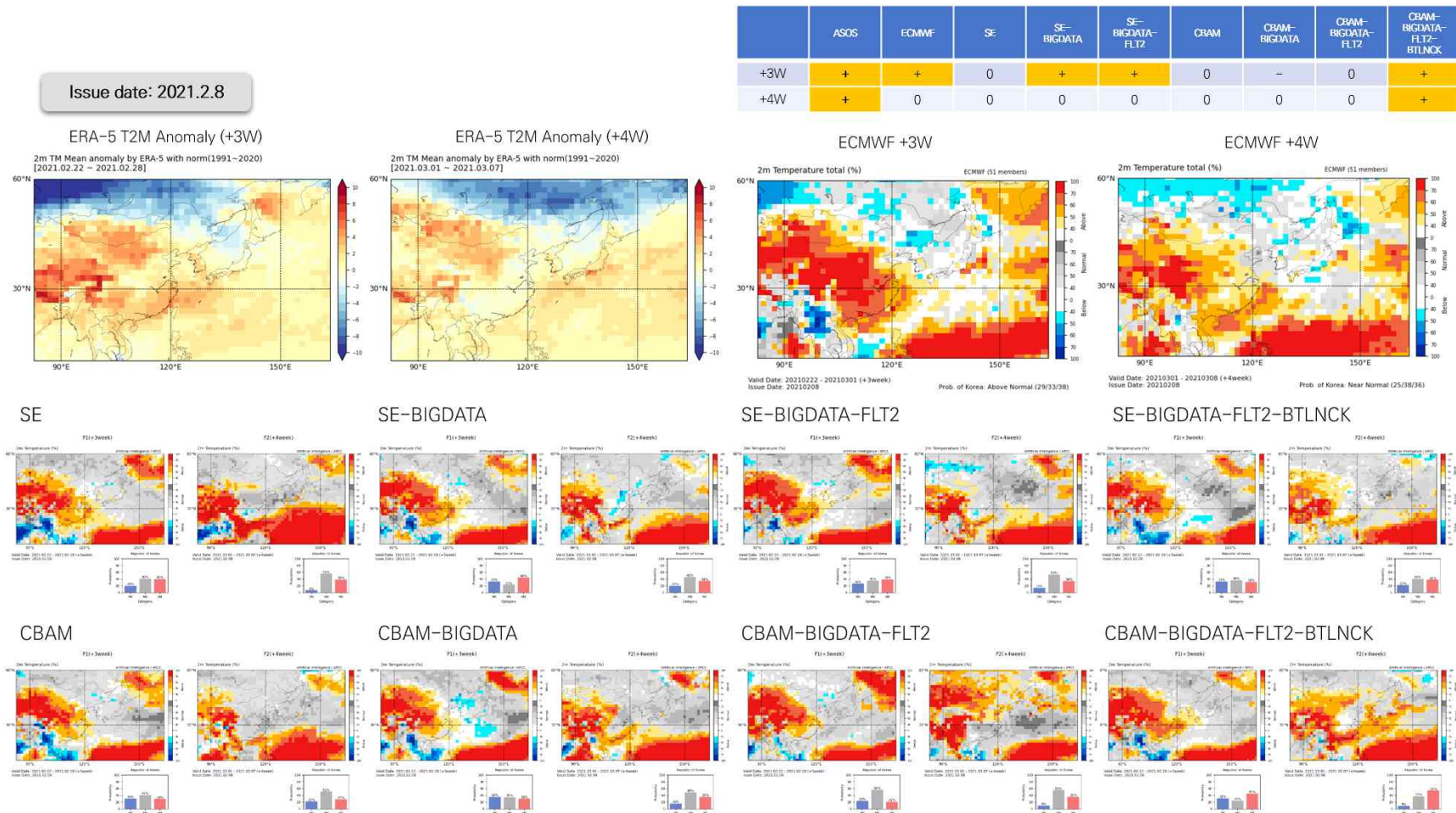


Figure A1. Comparison of probability forecasts from KMA ASOS, ECMWF S2S model, and deep learning models on February 8, 2021.