INTEGRATING GRAPH NEURAL NETWORKS AND ATTENTION MECHANISMS FOR HIGH-ACCURACY PRECIPITATION PREDICTION

ZHANG TING

DOCTOR OF PHILOSOPHY

(COMPUTER SCIENCE)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

JULY 2024

INTEGRATING GRAPH NEURAL NETWORKS AND ATTENTION MECHANISMS FOR HIGH-ACCURACY PRECIPITATION PREDICTION

By

ZHANG TING

A thesis submitted to the
Department of Computer Science,
Faculty of Information and Communication Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Doctor Of Philosophy in Computer Science
July 2024

This thesis is submitted in partial fulfilment of the requirements for the degree of Computer Science at Universiti Tunku Abdul Rahman (UTAR). This thesisrepresents the work of the author, except where due acknowledgment has been made in the text.

ABSTRACT

INTEGRATING GRAPH NEURAL NETWORKS AND ATTENTION MECHANISMS FOR HIGH-ACCURACY PRECIPITATION PREDICTION

Zhang Ting

With the advent of big data and meteorological modernization, the role of meteorological data analysis in various fields, such as daily life and social production, has become increasingly prominent. Taking the Guangxi Meteorological Information Center as an example, the volume of meteorological observation data is enormous, posing challenges in data storage, processing, and real-time throughput. Traditional methods of meteorological data analysis are no longer sufficient to deal with the complexity and scale of big data, necessitating the adoption of new approaches to enhance the efficiency and accuracy of weather forecasting.

This thesis addresses these challenges by integrating innovative methodologies to organize and analyze meteorological data in Guangxi, focusing on big data processing, neural networks, and other advanced techniques. The research makes three significant contributions:

1. Development of GraphAT-NET: A novel forecasting model named

GraphAT-NET is proposed, which combines graph neural networks and channel attention mechanisms to enhance short-term precipitation forecasting. By extracting key information from TRAJGRU's feature maps and constructing a Graph structure with an Efficient Channel Attention (ECA) mechanism, GraphAT-NET achieves superior performance. Experiments using the moving MNIST dataset and real-world radar echo data show that GraphAT-NET reduces Mean Squared Error (MSE) to 1.23 and improves the Structural Similarity Index (SSIM) by an average of 16.26% compared to other models. This model demonstrates a substantial average enhancement of 65.4% in MSE and a 10.29% improvement in SSIM on real-world datasets, highlighting its effectiveness in predicting cumulonimbus cloud distribution.

2. Application of the Informer Model: The research explores the use of the Informer model for time series analysis of ground station data. The Informer model achieves a mean absolute error (MAE) of 0.0077, significantly outperforming traditional numerical weather prediction (NWP) techniques with an MAE of 0.02. This deep learning-based framework combines feature engineering and model optimization, enhancing the precision and reliability of short-term precipitation forecasting. The Informer model improves predictive accuracy by 74.2% over conventional methods and by 72.3% over other deep learning-based models, as measured by the root mean square error (RMSE).

3. Integration and Comparison of Methods: The study integrates GraphAT-NET and the Informer model to forecast rainfall using CAPPI data and ground station data, respectively. By comparing these methods, the results indicate that predictions based on ground station data using the Informer model are more accurate, outperforming radar reflectivity signal-based predictions by 5.16. This highlights the effectiveness of ground station data in precipitation forecasting.

In conclusion, this research significantly advances the field of meteorology by introducing innovative deep learning models that improve the accuracy and reliability of short-term precipitation forecasting. The findings have the potential to enhance meteorological services, aiding in better decision-making for weather-related activities and emergency planning.

In the future, we will focus on integrating multimodal data for rainfall prediction, combining radar, satellite imagery, ground sensors, and meteorological models. This comprehensive approach aims to enhance prediction accuracy and robustness. We also plan to develop a unified evaluation framework to assess multimodal models fairly, considering each modality's unique characteristics. This will help identify effective approaches, advancing rainfall prediction and improving meteorological services for better decision-making.

Keywords: artificial intelligence; big data analytics; deep learning; attention mechanism; precipitation prediction

Subject Area:T58.5-58.64 Information technology

ACKNOWLEDGEMENT

Firstly, I extend my deepest gratitude to my main supervisor, Prof. Liew Soung Yue, whose unwavering support has been essential throughout my doctoral journey at UTAR. Prof. Liew's guidance has not only been academically enriching but has also touched my life in numerous ways. As an international student new to Malaysia, Prof. Liew was instrumental in helping me acclimate to my new surroundings, showing genuine interest in my well-being, much like a good friend. In my academic pursuits, his seasoned expertise and profound knowledge were my beacon of light during moments of uncertainty. His contributions were essential for my progress.

Additionally, I am obliged to express my thanks to my co-supervisor, Dr. Hui-Fuang Ng, for his invaluable input and the strategic advice he has offered our research. His insights have often been the catalyst at pivotal junctures of our study.

In the same vein, I am immensely thankful to my co-supervisor, Prof. Qin Dong Hong, whose encouragement was the impetus for my academic sojourn at UTAR—a journey that has been both memorable and significant. His thoughtful concern for both my personal life and academic endeavors has been a source of comfort and inspiration, propelling me to persevere.

Lastly, I wish to acknowledge all those I have encountered at UTAR. Your patience and assistance amidst the challenges I have presented have not gone unnoticed, and I am truly grateful for your collective support.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS

CNNs	Convolutional Neural Networks
GCNs	Graph Convolutional Networks
GRU	Gated Recurrent Unit
MNIST	Modified National Institute of Standards And Technology Database
MSE	Mean Squared Error
SSIM	Structural Similarity Index
NWP	Numerical Weather Prediction
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
CAPPI	Constant Altitude Plan Position Indicating
GCAPPI	Guangxi Constant Altitude Plan Position Indicating
ECA	Efficient Channel Attention
LSTM	Long Short-Term Memory
GCRN	Graph Convolutional Recurrent Network
QPE	Quantitative Precipitation Estimation
ANNs	Artificial Neural Networks
CFS	Climate Forecast System
CEEMD	Complete Ensemble Empirical Mode Decomposition
IGA	Imperialist Competitive Algorithm

ARIMAX	Autoregressive Integrated Moving Average With Exogenous Variables
GPM	Global Precipitation Measurement
IOU	Intersection Over Union
LSTF	Long-Series Time Series Forecasting

CHAPTER 1

INTRODUCTION

Meteorological data analysis is a complex structure that is an essential part of modern society, influencing daily life and various sectors of social production. As China enters the modern era of big data, the importance of weather data has grown, encouraging weather departments to invest a lot in collecting, studying, and using this data. This includes developing advanced weather satellites, improved weather radars, and a network of weather stations. Utilizing big data analytics in the context of weather data processing not only elevates the data's practicality and significance but also acts as a vital impetus for the evolution of meteorological science.

In the world of big data, weather data analysis has moved beyond old methods, helping to find and take out hidden information from large data sets. This change in how we analyze data has made using weather data more efficient and has met the various needs of different industries and sectors, greatly increasing the business value of weather data.

Taking the Guangxi Weather Information Center as a case in point, the continuous advancement of weather modernization has resulted in a sharp rise in the volume of weather observation data, amounting to several terabytes annually. Since 2009, the systematic change of weather historical data into digital form has been happening, with a complete process of scanning and storing historical observation reports from both ground and upper-air observations. This has created a large collection of digital images.

Currently, the number of digital files has grown to between 4 to 5 million, with a data capacity of 3 terabytes. The historical data files from over 2500 automatic stations in the region, from their start, make up about 350 million files, using about 1.5 terabytes of data space. Also, the

basic data from Doppler weather radars in nine areas, from when they started, use about 3 terabytes. This data is stored on various server disks and other outside storage devices, with the amounts growing each year. Storing weather data is known for its large volume, many small files, complex data processing needs, and the need for high real-time data flow. The previous scattered and inefficient storage methods can no longer meet the department's strict internal requirements for data processing and getting data. The data, which comes in formats like .jpg, .txt, and compressed binary files, show a complexity, large volume, and diversity of types that are signs of big data. A scientific way of summarizing, organizing, and studying these non-weather data is necessary to get valuable information and to use the full power of big weather data.

The accuracy of short-term weather forecasts is a key part of weather work. Short-term forecasting usually includes predicting rain or severe stormy weather in a certain area in the near future and is very important in preventing weather disasters. On one hand, accurately predicting and giving early warnings about often dangerous stormy weather is still a challenge, even after many years of research, because these weather events change in very complex patterns. On the other hand, current weather business services need very high standards for the accuracy, spatio-temporal resolution, and timeliness of early warnings and short-term forecasts for severe stormy weather. Handling the large amount of data, which is diverse, multidimensional, and complex, and efficiently and accurately finding the patterns below has become a pressing issue for weather researchers. Finding meaningful information from the large amount of weather data is a big challenge faced by weather staff.

To further explain, the complexity of weather data requires a multifaceted approach to analysis. The combination of advanced computational techniques, such as machine learning and artificial intelligence, can offer deeper insights into the patterns and trends within weather data. These techniques can assist in predicting weather events, enhancing the reliability and precision of forecasts. Additionally, the continuous growth of weather data calls for the development of scalable storage solutions and efficient data retrieval systems to ensure timely access and analysis.

Moreover, interdisciplinary collaboration plays a crucial role in weather data analysis. Integrating knowledge from fields like climatology, physics, and computer science can lead to innovative methodologies for data analysis and predictive models. This collaborative approach can also facilitate the development of more advanced tools for data visualization and interpretation, making complex patterns in weather data more accessible to a wider audience.

In conclusion, the advancement of weather data analysis in the era of big data is a dynamic and evolving field. It requires a coordinated effort in terms of technological innovation, interdisciplinary collaboration, and the development of robust data management systems. By addressing these challenges, weather departments can not only improve the accuracy of weather forecasts but also contribute to a broader understanding of atmospheric phenomena, ultimately benefiting society as a whole.

1.1 Problem Statement

1.1.1 Research Status of Radar Satellite Meteorological Data Prediction

In recent years, the Long Short-Term Memory (LSTM) model, a method driven by data, has gained a lot of attention for predicting how much water flows away after it rains, both for single events and for larger areas over time. Researchers including Souto and others have shown that deep neural networks that loop back on themselves, including LSTM and models with gates that control the flow of information, work well for predicting how much water flows away every day. Using a method that looks at the main parts of the data can help improve the model's performance by choosing the right variables to look at. This is especially useful for finding the most important factors that lead to water flowing away, which improves how accurately the model can predict.

However, LSTM models have some limits when it comes to linking two points in a sequence of events over time, and radar images of rain are complex. Because of this, Shi and others suggested a new kind of network called ConvLSTM [1]. This network can change 2D radar images into 3D versions, which helps with predicting when and where it will rain. ConvLSTM combines the good parts of traditional LSTM with

a structure that can see patterns, making it good for analyzing data that changes over time and in space. The ability of ConvLSTM to handle how data is related in space and time has opened up new ways to study things, especially in recognizing actions in videos, where understanding movement and changes over time is very important.

In addition, the TrajGRU model is a new kind of network that does better than ConvLSTM [2]. It uses changes in position to predict what will happen in the future. By using several layers of networks that can remember things and a part that knows about position, TrajGRU did better than other models like SWIRLS and ConvLSTM. This new way of handling data about movement could change how we analyze how things move in many areas, from understanding weather to building robots. However, both ConvLSTM and TrajGRU have a lot of parts that make them complicated, which makes it hard to train them on computers at the same time. This need for complexity means we have to develop better ways to train them and hardware that can handle the big demands of these calculations.

Convolutional Neural Networks (CNNs) have been used a lot in research about predicting rain. Models like U-Net [3], SmaAt-UNet [4], and SE-ResNet [5] are well-known. CNNs are good at getting detailed information by using special tools called convolutional kernels, but they are not as good at combining how things change over time with more complex meanings for better learning of features. This is a big limit when it comes to predicting weather, where how things change over time is just as important as where they are.

To fix this limit, researchers have started using Graph Convolutional Networks (GCN) to make the representation of features better [6]. GCN is good at using information from nodes and their neighbors in graph data, which is a promising way to do things. Kipf and others introduced a way to learn that can handle a lot of data at once, showing that it works better than other methods by efficiently encoding the structure of graphs and the features of nodes. This could change how we understand

and predict complex systems, where how things are related to each other is just as important as the things themselves.

Wu and others introduced a new model called Graph Convolutional Recurrent Network (GCRN) with several ways to see patterns, which can capture complex features of where rain falls [7]. By extending how space is related beyond the main node and neighboring nodes, this model did better than traditional graph models like Quantitative Precipitation Estimation (QPE) while needing fewer parts to work [8]. The ability of GCRN to capture complex relationships in where rain falls could lead to more accurate and reliable predictions of rain, which is very important for many things like farming, planning cities, and managing disasters.

These new developments show the potential of GCN in making the learning of feature representation better for predicting rain, showing a change towards more complete and effective ways of making models. Adding GCN into models that predict rain could give a more detailed understanding of how weather variables interact with each other, leading to more exact forecasts.

Research in this area is still new, and there is a need for new ideas and a lot of work to make it better and turn it into useful technology. Looking at research from inside the country and around the world shows that there are important scientific questions that need attention:

Existing networks are good at handling data that has a regular structure, like one or two dimensions that follow a pattern. However, real-world data often has irregularities, like the graph structures seen in radar images of rain. These data structures with unique surroundings for each node challenge the traditional CNN and Recurrent Neural Network (RNN) methods, which means there is a need for research on how to handle data that has a graph structure. Developing algorithms that can effectively process such irregular data structures is very important for moving the field forward.

Spatiotemporal modeling, like what is seen in predicting rain in the near

future and predicting videos, has big challenges because of the complex nature of spatio-temporal sequences. LSTM and CNN methods find it hard to combine how things change over time with more complex meanings for more precise representation of features. Finding ways to build relationships between features, turning traditional image features into graph structures, and setting up how features are related for better extraction of features is very important for making spatiotemporal modeling more accurate. This will allow for more solid predictions in environments that change, where changes over time are just as important as patterns in space.

1.1.2 Research Status of Surface Meteorological Data Forecasting

Traditional meteorological forecasting methods typically involve two main approaches. The first is the traditional meteorological statistical method, where data collected from the atmosphere is carefully analyzed by experienced meteorologists to make predictions about future weather conditions. However, this method, which relies heavily on the expertise of these professionals, often does not make full use of all available meteorological data, which can lead to increased errors in prediction.

The second approach is the numerical weather prediction method, a scientific endeavor that is deeply rooted in mathematical and physical principles. This method is designed to simulate and forecast the changes in atmospheric conditions. With a history that covers many decades, it has been shaped by the collective work of numerous scientists and researchers, rather than being the work of a single individual. It has developed alongside improvements in computational power and meteorological theories [9]. This method involves converting the state of the atmosphere into partial differential equations, using well-known physical laws to predict future changes. Various computational techniques, such as the complete prediction method, model output statistics method, Kalman filtering method, and regression analysis method, are used for this purpose. While the numerical weather prediction method has proven to be a very useful tool, it has limitations

in capturing the full complexity of atmospheric processes with complete accuracy, which has led to ongoing research and gradual improvements in this area.

Recently, the combination of machine learning algorithms, especially neural networks like the Backpropagation Neural Network (BPNN) and Support Vector Machines (SVMs), has greatly improved the accuracy of meteorological forecasts. This progress has resulted in a significant breakthrough in temperature prediction within the field of meteorology and related areas, marking the beginning of a new era of enhanced forecasting capabilities.

The latest advancements in weather prediction have seen the strategic use of various neural network models. Genetic algorithms have been used to optimize Artificial Neural Networks (ANNs) for historical precipitation forecasts, demonstrating their stability and adaptability. The combination of Empirical Mode Decomposition (EMD) with the Least Squares Support Vector Machine (LS-SVM) model has significantly improved the accuracy of temperature predictions. Neural network models such as BPNN, Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), residual autoencoders, and LSTM with Attention have been successfully used for temperature forecasting, extracting features from meteorological data, and improving predictive accuracy without overfitting [10]. These innovations have shown the strength of neural networks in improving predictive accuracy, especially in dealing with challenges like data loss that is inherent in meteorological data processing.

Furthermore, the rapid advancement of computer technology has pushed deep learning into a new era of artificial intelligence. Its application in areas such as natural language processing and computer vision has not only faced significant technical challenges but has also driven technical innovation. This has created new possibilities for technological progress in meteorological forecasting. By using deep learning technology in an integrated way, new methods have emerged for enhancing meteorological forecasting techniques. Using deep learning methods to

reveal hidden patterns in data related to temperature and precipitation has led to significant improvements in computational efficiency, predictive accuracy, and the ability to issue alerts for severe weather conditions, even when dealing with large volumes of data. This study explores the use of deep learning techniques on ground observation and radar image data to increase predictive accuracy. For radar-based predictions, the development of the GraphAT Net model combines predictive models with attention mechanisms, Gated Recurrent Units (GRU), and Graph Convolutional Networks (GCN). Comparative analysis shows that this model performs better than others in terms of efficiency and accuracy in meteorological forecasting, providing valuable insights to improve the precision and effectiveness of meteorological forecasting for professionals in the industry.

Meteorological data collected from various ground stations make up a complex, high-dimensional set of information. The emergence of deep learning methodologies has introduced new ways to forecast precipitation using observations from ground stations. This review looks deeply into existing weather forecasting literature, focusing on the Transformer and Informer models for their ability to predict different meteorological elements. Techniques for reducing noise were applied to handle extensive multidimensional meteorological time series data, with an emphasis on variables related to precipitation to enhance how data is represented within the network. Results from both models demonstrate their effectiveness in forecasting precipitation, highlighting the strategic advantage of combining meteorology with artificial intelligence, especially using Transformer and Informer models for short-term precipitation predictions.

Additionally, the study explores the merging of these two rainfall forecasting techniques to improve the accuracy and reliability of predictions. It examines the combination of rainfall prediction methods using different data from radar cloud distributions and ground station data to create more accurate models for predicting rainfall. Through experimentation, the study aims to measure the accuracy of rainfall predictions made from ground station information compared to those

based on radar echo maps, with the ultimate goal of refining and enhancing the precision of rainfall forecasting.

1.1.3 Research Status of Ground Meteorological Data Forecasting

In the continuous development of weather forecasting technologies, scholars have made significant achievements by using various neural network models. These models have not only shown a strong ability to predict important meteorological parameters such as annual rainfall, temperature changes, and wind speeds, but they have also opened the path for more advanced forecasting methods. The novel approaches employed in these research endeavors encompass the enhancement of connection weights via genetic algorithms. This technique facilitates an efficient exploration within the solution space and bolsters the model's capacity to generalize from the training data. Moreover, the use of the data breakdown method has been important for breaking down complex weather signals into simpler, more manageable parts, thus improving the predictive accuracy of neural networks.

The use of Backpropagation Neural Networks (BPNN), Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTM), residual autoencoder models, and the combination of attention mechanisms within LSTM models are among the techniques that have been explored. These methods have proven to be especially effective in capturing the complex patterns and changes over time present in meteorological data. Attention mechanisms, in particular, have been a useful feature because they allow the model to focus on the most relevant parts of the input data, thereby improving the process of getting key data and overall forecast accuracy.

These studies, when considered together, demonstrate the effectiveness of neural networks in taking out valuable information from meteorological data and improving prediction accuracy. The ability of these models to gain knowledge from historical weather patterns and make accurate forecasts is proof of the power of deep learning in the field of meteorology. While traditional neural network models have

shown remarkable performance, there are still challenges in dealing with data loss issues that are inherent in handling weather data, given its time-related nature and complexity. The research presented here provides a compelling basis for the ongoing development and application of neural network approaches in weather forecasting, showing their possibility for more precise predictive results.

Meteorological data collected from various ground stations provide rich insights across many dimensions. The arrival of deep learning methodologies has introduced new ways to predict precipitation using observations from ground stations. This review goes deep into meteorological forecasting literature, focusing on Transformer and Informer models for their wide-ranging forecasting capabilities. These models have been especially successful in dealing with the large and complex data sets typically found in meteorological studies. By using methods to reduce errors, large amounts of multidimensional meteorological time series data are processed, with a special focus on key variables related to precipitation for better representation of data within the networks. The results from these models demonstrate their effectiveness in predicting precipitation, highlighting the strong combination of meteorology and artificial intelligence, especially through the use of Transformer and Informer models in short-term precipitation forecasts.

Transformer and Informer models, in particular, have shown to be good at handling the time-related nature of meteorological data. Their ability to capture dependencies over time and represent changes over time makes them well-suited for tasks such as predicting precipitation. The use of these models together with techniques to reduce errors allows for a more accurate portrayal of the underlying weather patterns, leading to more dependable forecasts.

Future research will focus on combining these two precipitation forecasting techniques to enhance the exactness and reliability of rainfall predictions. There will be an exploration into using various kinds of information provided by rainfall forecasting methods based on radar

cloud distributions and ground station data to create an improved rainfall forecasting model. Through testing, comparisons will be made between the accuracy of rainfall predictions based on ground station observations and radar echo maps, aiming to make the best of rainfall information predictions. This combination of different data sources and methodologies is expected to result in a more complete understanding of weather patterns and lead to more accurate and timely forecasts.

1.2 Objectives

In the pursuit of increasing the accuracy of weather forecasts, especially for short-term rainfall predictions, analyzing radar echo signals and data from ground stations is extremely important. Our goal is to create a strong model that can understand these different sources of data to predict when it will rain next. By using deep neural network algorithms, we want to train a model that can predict rain with high accuracy and also adjust to the small changes in local weather patterns.

The model will be carefully trained on a wide range of data, including both radar echo signals and observations from ground stations. Using both of these approaches together helps us fully understand the conditions in the atmosphere that cause rain. Also, by doing numerical analysis, we will compare and talk about the characteristics and how accurate predictions are from these two different types of data. Comparing these analyses is very important for understanding what each data source does well and where it might fall short, and this helps us make our model better at predicting.

(1) Establishment of a Deep Learning Model: Our first job is to build a deep learning model that is made for predicting rain in the short term in the Guangxi area. We will make this model better by using Graph Convolutional Network (GCN) methods, which are good at getting features from data that is arranged like a graph. Using GCN will let the model sort nodes, classify graphs, predict links, and get information about the graph. These abilities are necessary for understanding the complicated relationships in weather data. Even though GCN has a lot

of potential, there is not much research on using it for predicting rain, which means there is a big chance for more study and new ideas. Combining GCN with radar data to predict rain could change the way we do things, giving us a better way to understand how rain falls.

(2) Algorithm Design and Implementation: We will plan and create algorithms that are made for predicting rain in the short term in the Guangxi area. This includes studying and making features that are based on what the graph data looks like. Features that are based on the meaning are very important because they give information that helps the model make sense of the data and be more accurate. We will copy the neural network algorithms that are popular now and do experiments to see which ones work the best. Adding attention mechanisms and other advanced parts will keep happening, with the goal of making the model better at getting features. A key goal is to change traditional picture features into graph structure information, creating connections between features that help get more precise features and, as a result, more accurate predictions.

(3) Model Optimization: Making our model better at predicting rain in the short term in the Guangxi area is what we want to do most. Once the model is made, we need to make it the best it can be. Making loss functions, like errors in classification, better is a hard task. Common practice of reducing errors, even though it is usual, can often cause the model to be too fitted to the data, which can make it not work well with new data. To prevent this, we will study how to make deep models better, exploring different strategies like changing how fast the model learns, the size of batches, which method we use to change the model, making the model less likely to overfit, adjusting features and labels, starting conditions, and using methods like Shake-shake regularization. All these efforts are aimed at making the model learn better and predict more accurately, making sure it can change with new data and give trustworthy forecasts.

1.3 Organization of Thesis

The doctoral thesis is organized into seven comprehensive chapters, the structure and content of each chapter are outlined as follows:

Chapter 1: Introduction

This chapter sets the stage for the research by providing a concise overview of the background and significance of the study. It explores the importance of meteorological predictions, particularly in the context of ground-based and radar satellite meteorological predictions. The chapter highlights the achievements and applications of these predictions in various fields, such as agriculture, urban planning, and disaster management. It then introduces the research questions and objectives, laying the foundation for the subsequent chapters. The introduction also emphasizes the gap in the current literature and the potential contributions of this research to the field.

Chapter 2: Literature Review

In this chapter, a thorough comparison is made between traditional forecasting methods and the emerging deep learning approaches in the domain of short-term rainfall prediction. It provides a historical perspective on the evolution of deep learning, especially its application in rainfall prediction. The chapter also introduces the concept of the Attention Mechanism and its significance in enhancing the performance of deep learning models. The literature review serves as a critical analysis of existing research, identifying areas of improvement and potential avenues for innovation.

Chapter 3: Theory and Data Introduction

This chapter introduces the relevant technical theories, data information, mathematical definitions, and fundamental knowledge of the models used in the study. It covers the essential concepts in the field of rainfall prediction, such as the correlation between radar reflectivity and rainfall intensity, the Z-R relationship, and the relationship between ground

observation data and rainfall prediction. Additionally, it provides a detailed description of the geography and climate characteristics of Guangxi, which is crucial for understanding the regional context of the study. The chapter also discusses the methodology for data preprocessing and the construction of the dataset, which is vital for the accuracy and reliability of the model.

Chapter 4: Short-term Rainfall Prediction Based on Radar Observation Data

This chapter focuses on the integration of the temporal dimension with advanced semantic information found in images to aid the model in learning more accurate feature representations. The study employs Graph Convolutional Networks (GCN) to address the challenge of deep generalization. It outlines the construction of the GCN structure based on the prediction model of the trajgru neural network, the use of the ECA channel attention mechanism to reassign weights to graph information, and the development of a predictive model named GraphAT-NET. The chapter presents comparative experiments that confirm the accuracy and effectiveness of this model in rainfall prediction, demonstrating its superiority over traditional methods.

Chapter 5: Prediction of Rainfall Based on Ground Station Observations Using Deep Learning Approaches

In this chapter, the application of Transformer and Informer models is demonstrated as valuable solutions for short-term rainfall predictions. It discusses the use of extensive, high-dimensional meteorological time series data collected from multiple ground stations. The chapter explores how these models can process and analyze large volumes of data to provide accurate and timely rainfall predictions. The discussion includes the advantages of these models in handling complex patterns and relationships in meteorological data.

Chapter 6: Utilization of Z-R Relationship for Rainfall Prediction

This chapter describes the correspondence between radar reflectivity

factor (Z) and rainfall intensity (R). It details the process of calculating the predicted rainfall distribution map based on the Z-R relationship obtained from radar echo data. The study compares the predicted rainfall intensity derived from radar data and ground station data with actual rainfall intensity, providing insights into the accuracy and reliability of radar-based rainfall predictions.

Chapter 7: Conclusion and Future Prospects

The final chapter summarizes the primary research work conducted in the thesis. It elucidates any deficiencies or limitations encountered during the study and provides a clear direction for future research. This chapter reflects on the contributions of the research to the field of meteorological predictions and suggests potential areas for further exploration and development. It also highlights the practical implications of the findings and their potential impact on related fields.

CHAPTER 2

LITERATURE REVIEW

Precipitation forecasting is an essential component of meteorological science, playing a key role in reducing the risks of natural disasters, reducing socio-economic impacts, and addressing many challenges associated with weather-related events. The pursuit of enhancing the precision and timeliness of rainfall predictions has been a central theme in the field of meteorology. This extensive literature review focuses on examining the notable progress in precipitation prediction, highlighting the creative use of cutting-edge technologies in the manipulation and interpretation of meteorological data.

Meteorological data, meticulously collected from ground-based weather stations, serve as the foundation for understanding and predicting weather patterns. These datasets include a broad spectrum of parameters, such as air pressure, temperature, humidity, wind direction, wind speed, snow depth, sunshine duration, cloud thickness, air quality indices, and precipitation measurements. The in-depth exploration and rigorous analysis of this diverse meteorological data are crucial for facilitating real-time weather forecasting and increasing the reliability of future predictions. The application of both traditional and cutting-edge deep learning methodologies, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have markedly elevated the accuracy of forecasting precipitation events [11]. These technologies have demonstrated remarkable efficacy in processing meteorological data in an efficient and effective manner.

Recent breakthroughs in deep learning models, particularly the Transformer architecture and its derivatives, have had a major impact on

the field of meteorological data processing. Transformers have showcased an exceptional ability to handle multi-temporal data and to capture the complex spatio-temporal dynamics natural in meteorological phenomena. The introduction of innovative attention mechanisms, such as the a-NDTT (adaptive-neighbor-distance-temperature-time) mechanism, has further enhanced the capabilities of Transformer-based models in solving multi-element time series classification problems [12]. These advanced technologies offer novel approaches for augmenting the precision of precipitation forecasts and refining meteorological predictions across a spectrum of application domains.

Moreover, the integration of these advanced technologies with traditional meteorological forecasting methods has opened new frontiers in the field. The synergy between machine learning algorithms and empirical models has led to the development of hybrid systems that leverage the strengths of both approaches. This integration has not only improved the accuracy of predictions but also increased the robustness of forecasting systems against uncertainties in data and environmental conditions.

The significance of precipitation forecasting extends beyond academic research; it has profound meanings for various sectors such as agriculture, urban planning, disaster management, and public health. Accurate and timely rainfall predictions can inform critical decision-making processes, leading to better resource allocation, infrastructure planning, and emergency response strategies. Therefore, the continuous improvement of precipitation forecasting models is not just scientifically valuable but also socially and economically beneficial.

This literature review delves into the significance of precipitation forecasting, the diversity of meteorological data collection, and the transformative impact of advanced technologies in meteorological data processing. By examining these key aspects, this review aims to provide a comprehensive understanding of the current trends and future directions in precipitation forecasting research. It seeks to contribute to the advancement of meteorological science and the development of

more reliable weather prediction systems, ultimately enhancing our ability to anticipate and respond to weather-related events.

2.1 Development of Short-term Precipitation Prediction

Uncertainty assessment is a fundamental aspect of short-term flood forecasting, particularly when it involves the integration of advanced models like the short-range global Numerical Weather Prediction (NWP) model. The high spatial resolution of 0.5 degrees in the NWP model is crucial for capturing the fine-scale features of weather systems that can lead to flooding. By coupling this with a distributed rainfall runoff model, a more holistic and detailed approach to flood prediction is achieved. The Thu Bon River basin in Central Vietnam, covering an extensive area of 3,150 km, serves as the focal point for this study [13]. The geographical spread of this basin offers a rich context for exploring the complexities of flood prediction modeling, presenting both unique challenges and opportunities.

The study delves into the complexities of 2-m surface temperature and precipitation extremes, extending its scope beyond Vietnam to include the broader continents of North and South America. This broadened perspective allows for a more comprehensive understanding of the climatic factors that influence precipitation patterns and their potential for causing floods. Additionally, the research examines sea surface temperature extremes in regions that are critical to global climate patterns, such as the Nino-3.4 and the Atlantic hurricane main development regions. The Climate Forecast System (CFS) global climate model, which spans from 1982 to 2010 [14], is utilized to provide a robust dataset for this analysis. The nearly three-decade-long dataset enables a detailed examination of weather patterns and their implications for short-term flood predictions.

The study further explores the complex relationship between weather conditions and the abundance of mosquitoes, which are a significant vector for diseases such as West Nile Virus (WNV). The research demonstrates that weather conditions can provide robust short-term

predictions of regional daily mosquito abundance, with significant implications for public health during the summer months. This is especially pertinent when considering local habitat variations or the impact of mosquito control efforts on disease transmission dynamics [15]. Understanding this relationship can aid in the development of more effective public health strategies to mitigate the risks of mosquito-borne diseases.

Advancements in machine learning algorithms have been essential in predicting inflow into the Soyang River Dam in South Korea. This prediction model is developed using an extensive dataset spanning 40 years of weather and dam inflow data, highlighting the potential of these algorithms in power load forecasting [16]. The development of such predictive models is not only technologically impressive but also practically significant, offering insights into water resource management and energy production. These models can help in optimizing dam operations, ensuring efficient water use, and contributing to energy planning and management.

Innovative methods are proposed to further enhance precipitation prediction accuracy. SCENT, inspired by optical flow methods, and MLSTM-AM are introduced as new approaches to improve monthly precipitation forecasting [17]-[19]. These methodologies represent a significant leap forward in the field, providing more subtle and precise predictions. The introduction of these methods could potentially transform the way precipitation is predicted, leading to more accurate and reliable forecasts.

A novel combination model is introduced, leveraging Complete Ensemble Empirical Mode Decomposition (CEEMD), T-S FNN optimized by the Imperialist Competitive Algorithm (IGA), and Markov error correction. This integrated approach aims to enhance ultra-short-term wind power prediction precision [20]. The development of such models is crucial for the renewable energy sector, enabling more efficient energy production and distribution. The integration of these advanced techniques could lead to more reliable and efficient wind

power generation, contributing to the global shift towards sustainable energy sources.

The application of these predictive models extends to water quality management as well. Models are designed for short-term dissolved oxygen (DO) prediction at the Han River junction in Anyangcheon, addressing potential water quality accidents [21]. The ability to predict DO levels is essential for maintaining aquatic ecosystems and ensuring public health. Accurate DO prediction can help in the early detection of water quality issues, allowing for timely interventions to prevent environmental and health risks.

Furthermore, the development of a rainfall prediction system using the Waterfall Model Development Life Cycle Software, coupled with backpropagation artificial neural networks, is aimed at improving prediction accuracy [22]. This system represents a fusion of software engineering principles and artificial intelligence, offering a robust framework for rainfall prediction. The integration of these technologies can lead to more accurate rainfall forecasts, which are crucial for various applications such as agriculture, urban planning, and disaster preparedness.

The quest for the best ANN model to predict groundwater level fluctuations at Manvi is a testament to the ongoing advancements in artificial intelligence. The study reveals significant features linked to bias changes, as outlined through multivariate empirical orthogonal function analysis [23]. This analysis provides a deeper understanding of the complex interactions between groundwater levels and environmental factors. Accurate prediction of groundwater levels is essential for sustainable water resource management and can help in mitigating the impacts of climate change on water availability.

Sub-seasonal rainfall prediction over the South China Sea and its surroundings during the spring-summer transitional season is scrutinized, with the aim of enhancing model performance [24]. This region, with its unique climatic patterns, presents a challenging yet critical area for accurate rainfall prediction. Improved predictions in this region can have

significant implications for agriculture, fisheries, and climate change adaptation strategies.

Soil loss prediction due to erosion is examined using a support vector machine model, focusing on advanced AI models for daily rainfall prediction in Hoa Binh province, Vietnam [25]. This study underscores the importance of accurate rainfall prediction in managing soil erosion and its environmental impacts. Effective management of soil erosion is crucial for maintaining soil fertility, water quality, and overall ecosystem health.

The development of Autoregressive Integrated Moving Average with eXogenous variables (ARIMAX) models to forecast autumn rainfall in the South West Division of Western Australia incorporates Global Precipitation Measurement (GPM) rainfall, satellite soil moisture products, and statistical downscaling through machine learning to model changes in rainfall patterns across different agro-climatic zones due to climate change [26]-[28]. This comprehensive approach not only predicts rainfall but also considers the broader implications of climate change on agricultural practices and water resource management. Understanding these changes is vital for developing adaptive strategies to ensure food security and sustainable water use in the face of a changing climate.

2.2 Development of Radar Applied to Rainfall Precipitation

Rainfall prediction is a complex task that requires the advanced integration of various deep learning models to effectively capture the spatial and temporal dynamics of weather systems. The current landscape of rainfall forecasting is dominated by three primary deep network models, each bringing different capabilities and addressing specific aspects of the prediction problem.

Convolutional Neural Networks (CNNs): The first model, CNN, distinguishes itself by treating input grid weather elements similar to visual data. Through the application of image filters, CNNs engage in

feature learning to capture spatial patterns, which is crucial for understanding the distribution of weather elements across geographical regions [29]. CNNs excel at identifying and interpreting complex spatial relationships within the data, which is invaluable for predicting rainfall. However, their effectiveness is somewhat limited when handling sequential data, as they are mainly designed for fixed-length datasets. This limitation can hinder their ability to fully account for the temporal evolution of weather systems, which are naturally dynamic and time-dependent.

Recurrent Neural Networks (RNNs) are a type of neural network that is good at handling sequences of data. They are often used in natural language processing because they can remember information from previous steps and use it to help understand and process the next steps in a sequence. This framework enables RNNs to excel in comprehending temporal relationships within data, making them particularly well-suited for tasks that require an understanding of time-based dependencies [30]. Despite their strength in temporal analysis, RNNs do not inherently incorporate the spatial attributes of grid data, which can limit their learning capacity when it comes to spatially diverse meteorological phenomena. This shortcoming can be particularly limiting in the context of rainfall prediction, where both spatial and temporal dimensions are crucial for accurate forecasting.

Hybrid Models: The third methodology involves integrating convolutional and recurrent neural networks in diverse configurations. This hybrid approach, as validated in prior research [31], shows a heightened efficacy in capturing the multi-dimensional nature of rainfall data. By combining the spatial acuity of CNNs with the temporal sensitivity of RNNs, a more complete approach is realized, enabling a comprehensive grasp of the spatial and temporal characteristics that are essential for precise rainfall forecasting.

Long Short-Term Memory (LSTM) Networks: Although the LSTM structure is a potent instrument in deep learning, it encounters a

significant challenge owing to its large number of parameters. This complexity can render it less optimal for scenarios that require parallel computing during training. In response to this limitation, Cho et al. introduced the Gated Recurrent Unit (GRU) [32], a derivative of LSTM that simplifies the model architecture by consolidating the forget gate and input gate into a singular "update gate," and merging the cell state and hidden state. This innovation not only reduces the parameter overhead but also enhances the model's inference capabilities and its ability to generalize on modest datasets.

TrajGRU and GA-ConvGRU Models: Shi X et al. [33] emphasized the importance of modifying the loop structure itself to enhance performance in solving problems. Their TrajGRU model, which adjusts the structure of loop connections and explores different numbers of links to optimize connection efficiency, outperformed ConvLSTM on the improved HKO-7 dataset and Moving MNIST dataset. Additionally, L. Tian et al. [34] highlighted the limitations of ConvGRU, which uses mean squared error as a loss function, leading to subpar extrapolated images. To address this issue, they introduced GA-ConvGRU, an opposing model composed of a generator and identifier, producing more realistic inferences. However, Xie P et al. [35] argued that GA-ConvGRU has inherent drawbacks due to the easy coordination of the generator and discriminator during training, resulting in instability. Their solution, EBGAN-Forecaster, surpasses existing models. Furthermore, Yu T et al. [36] identified that conventional methods add independent parallel storage units outside the inner loop unit, lacking interconnectivity. They proposed ATMConvGRU, an axial attention memory module enhancing spatial-temporal feature correlation. Finally, Zhang et al. [37] introduced the M-ConvGRU model, merging input and previous output data into ConvGRU neurons, employing convolution-based gate preprocessing to capture contextual relations effectively. Demonstrating superior performance to ConvLSTM for echo predictions exceeding an hour.

CNN Applications in Rainfall Prediction: Besides examining the application of RNN in forecasting rainfall, scholars have also delved

into the use of CNN in this domain. Various models like U-Net, Smatunet, and Seresunet have been devised for rainfall prediction. In the Weather4cast competition, the model based on RNN secured the first-place position, while the U-Net-based model achieved second place. It has been observed that U-Net-based models perform better in spatial transfer learning when more parameters and additional weather variables are included as input. A novel U-Net model, integrating a numerical model with a deep learning model, has been proposed. This model takes numerical data from the NWP system as input and refines the data through U-Net to enhance the accuracy of final predictions. Trebing K et al. [38] introduced SmaAt-UNet, which incorporates an attention module and depthwise separable convolution within the efficient U-Net architecture. Experimental results have demonstrated that SmaAt-UNet delivers comparable prediction performance to other U-Net models while utilizing only a quarter of the trainable parameters. Furthermore, Song K et al. [39] proposed SE-ResNet to distinguish moving/deformed rain regions from random noise regions. They employed input-output cross-entropy as a loss function to eliminate noise in radar images. The regressor combines FCN and integrates the attention mechanism with the Intersection over Union (IOU) regression loss function. Experimental results have indicated that the algorithm outperforms RNN in rain region detection.

Graph Convolutional Networks (GCNs): GCNs have the capability to learn from data with arbitrary graph structures and have been successfully utilized in various tasks. In the case of graph data, GCN can integrate information from itself and neighboring nodes, thereby establishing high correlations between connected nodes. Kipf T N et al. [40] introduced a scalable semi-supervised learning approach for graph structures that scales linearly with the number of graph edges, enabling the learning of hidden layer representations that encode local graph structures and node features. Extensive experiments have demonstrated that this method outperforms related approaches significantly. Wu Y et al. [41] proposed GCRN, incorporating a multi-convolution mechanism to accommodate the varying spatial correlations in actual precipitation

data. By extending the central node and adjacent rain gauges, GCRN captures more complex spatial features of precipitation. When compared with another graph recurrent architecture, GCRN achieves superior performance with fewer parameters, surpassing the QPE models. Despite the numerous advantages of GCN, there is limited literature applying it to rainfall prediction currently, underscoring its research significance.

Spatio-Temporal Modeling: Spatio-temporal modeling poses challenges as it involves predicting spatio-temporal sequences, where both input and output are spatio-temporal sequences. Unlike typical time series prediction problems, spatio-temporal sequences require consideration of the interdependence of data samples over time. The proximity of two samples in a time series indicates a higher degree of relevance, rendering common classifiers unsuitable as they assume sample independence. Current research is focused on addressing these challenges in time series analysis, particularly in the context of meteorological data, which differs from traditional time series data as it essentially constitutes image data.

Integration of Graph Convolution: In this study, graph convolution is integrated to enhance the encoder-decoder model, which combines traditional CNN and RNN, thereby improving the model's capacity to learn from time series data. Additionally, a low-parameter attention mechanism is employed to better optimize the feature distribution within the convolutional layer during the feature extraction and reconstruction stages of the model. This approach allows for a more detailed understanding of the complex interactions between spatial and temporal elements within meteorological data, leading to more accurate rainfall predictions.

2.3 Development of Short-term Precipitation Prediction Based on Deep Learning

Rainfall prediction, a critical component of meteorological forecasting, is essential for reducing the risks associated with natural disasters,

reducing socio-economic impacts, and addressing a range of related challenges [42]. In the context of China, where climatic variability can have profound effects on agriculture, water resources, and public safety, accurate precipitation forecasts are essential for devising effective flood and drought mitigation strategies [43]. Ground meteorological stations play a key role in this process, systematically collecting a wide array of meteorological data for subsequent analysis. This data includes, but is not limited to, air pressure, temperature, humidity, wind direction, wind speed, snow depth, sunshine duration, cloud cover, air quality indices, and precipitation measurements [44]. The collection of these diverse datasets facilitates real-time weather forecasting and contributes to a rich dataset for future meteorological research and prediction models.

The meteorological data gathered from various ground stations is characterized by its spatio-temporal nature, which presents both opportunities and complexities in analysis. Deep learning methods have emerged as a powerful tool for processing such multidimensional meteorological data, offering enhanced performance over traditional approaches [46]. However, the multidimensionality and the broad spectrum of input-output data types in meteorological datasets can limit the effectiveness of conventional deep learning methods, requiring innovative strategies in artificial neural network design.

The introduction of the Transformer model has marked a significant advancement in the field, demonstrating good performance across a spectrum of tasks, including natural language processing (NLP), computer vision (CV), and time series prediction [47] - [49]. The Transformer's ability to handle long-range dependencies and its flexibility in attention mechanisms have made it a popular choice for various predictive tasks. A recent innovation in this domain is the a-NDTT (adaptive-neighbor-distance-temperature-time) mechanism proposed by Meng, which has achieved state-of-the-art results in multiple time series classification, regardless of dataset size [50] - [54]. This achievement is due to the parallel application of a general, flexible, and efficient spatio-temporal attention module, which allows for a more detailed understanding of the data.

Progress in traffic prediction and multimodal forecasting, facilitated by the integration of Graph Neural Networks (GNNs) with Transformers, has enhanced the understanding of spatio-temporal dynamics and the inherent randomness in data [55]. The combined effect between Transformers and GNNs for effective spatio-temporal modeling in time series is a promising area for future research, offering potential insights into more accurate and robust forecasting models [50], [56]. Despite the potential, research on pre-trained Transformers for time series forecasting remains limited, with a current focus on time series classification [57].

A study by [58] identified several critical issues with Transformers, including high second-order time complexity, increased memory usage, and the inherent limitations of the encoder-decoder architecture, which can hinder their effectiveness in long-series time series forecasting (LSTF). To overcome these challenges, the authors proposed a probabilistic sparse self-attention mechanism and a structured generative decoder, aiming to alleviate the constraints of traditional transformer encoder-decoder frameworks [59]. Another study [60] utilized the Informer encoder for tool wear prediction, incorporating a distillation layer to refine feature extraction. The Informer's ability to efficiently extract global features was highlighted, offering a distinct advantage over Transformers. Furthermore, [61] suggested that data imputation strategies could address missing observations in time series, leveraging adversarial network model training and self-attention mechanisms to enhance the accuracy of multidimensional time series estimation models.

While Transformers and Informer have been extensively utilized in long-series time series forecasting tasks, their application in meteorological forecasting, particularly rainfall prediction, is less explored. This study aims to fill this gap by focusing on predicting data from ground observatories using Transformer and Informer models. The paper is structured as follows: the first section provides a comprehensive literature review, discussing the current advancements in prediction algorithms of Informer and Transformer in various fields. The second

section delves into the benefits and enhancements of employing Informer and Transformer in model design. The third section delineates the attributes of meteorological data and assesses the efficacy of both algorithms when applied to the dataset from ground observation stations.

To further elaborate, the predictive capabilities of these models are crucial in the context of meteorological forecasting, where the ability to anticipate precipitation patterns can have significant implications for disaster preparedness and resource management. The integration of advanced machine learning techniques, such as Transformers and Informer, with traditional meteorological data analysis methods can lead to more accurate and reliable predictions. This integration is particularly important given the increasing volume and complexity of meteorological data being collected.

Furthermore, employing deep learning in weather forecasting is not without difficulties. The high-dimensionality of meteorological data necessitates sophisticated algorithms capable of effectively discerning the intricate patterns and interrelationships within the data. The Transformer model, equipped with its self-attention mechanism, is especially apt for this purpose, as it can handle sequences of diverse lengths and identify complex dependencies. Nevertheless, the computational requirements of such models can be considerable, thus prompting the need for the development of more efficient training and inference methodologies.

In the case of rainfall prediction, the spatio-temporal dynamics of precipitation are influenced by a multitude of factors, including atmospheric conditions, topography, and ocean currents. Capturing these dynamics accurately is essential for generating reliable forecasts. The models presented in this study aim to tackle these complexities by harnessing the capabilities of deep learning to analyze and forecast precipitation patterns, utilizing the extensive dataset supplied by ground meteorological stations.

Furthermore, the study will also explore the potential of these models in handling large-scale meteorological datasets, which is a critical aspect of

modern meteorological research. The ability to process and analyze vast amounts of data in a timely manner is crucial for generating real-time forecasts and for understanding long-term climatic trends. The proposed models will be evaluated based on their ability to handle such datasets effectively and to provide accurate predictions.

In conclusion, the application of Transformer and Informer models in rainfall prediction represents a significant step forward in the field of meteorological forecasting. By harnessing the power of deep learning, these models have the potential to revolutionize the way we predict and respond to weather events, ultimately leading to more effective disaster management and resource allocation strategies.

CHAPTER 3

EXISTING THEORIES

Rainfall prediction is a critical component of disaster mitigation and readiness, acting as an essential tool for guiding decision-making and enhancing community resilience. This task is complex and goes beyond the scope of simple weather forecasting, delving into areas such as public safety, the stewardship of resources, and the preservation of the environment. The importance of precise rainfall forecasts is paramount, as they are essential for a variety of practical applications, including agricultural management, urban development, flood prevention, and drought relief efforts. Accurate predictions in this domain can lead to more effective strategies for disaster response and resource allocation, underscoring their fundamental role in both societal and environmental well-being.

In recent years, the meteorological field has witnessed remarkable advancements in technology, which have significantly transformed rainfall forecasting. These technological leaps have not only improved the precision of predictions but have also broadened our capacity to understand and respond to the complex dynamics of weather systems. This chapter focuses on exploring the intersection of advanced technologies and deep learning theories in the context of rainfall forecasting. The exploration will emphasize the enhancement of our comprehension of precipitation patterns and the optimization of forecast outcomes.

The chapter will scrutinize the complex relationship between radar echo data, ground observations, and the deployment of deep learning models within the forecasting process. Radar echo data, which provides a

snapshot of atmospheric conditions, is a vital input for rainfall prediction models. Ground observations, on the other hand, offer a complementary perspective, supplying empirical data that can corroborate and refine radar-based predictions. Combining these two types of data with the strong analytical capabilities of deep learning models could lead to the creation of more solid and dependable weather predictions.

Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have emerged as powerful tools in the meteorologist's toolkit. These models are skilled at identifying patterns and relationships within large datasets, which is essential for interpreting the complex and high-dimensional nature of meteorological information. The chapter will discuss the application of these models in the context of rainfall forecasting, highlighting their strengths and addressing their limitations.

Moreover, the chapter will explore the significance of data preprocessing and feature engineering for the effectiveness of deep learning models. The preprocessing of meteorological data, such as normalizing variables and extracting pertinent features, is an essential step that can substantially impact the performance of predictive models. Feature engineering, which entails crafting new variables that encapsulate key elements of the data, can further augment the models' predictive capabilities.

In short, this chapter seeks to explore the intersection of advanced technologies and deep learning theories in the context of rainfall forecasting, with a focus on enhancing our understanding of precipitation patterns and improving forecast outcomes. By examining the relationship between radar echo data, ground observations, and the application of deep learning models, this chapter aims to provide valuable insights into the evolving landscape of rainfall forecasting methodologies and contribute to the ongoing efforts to strengthen disaster preparedness and response strategies.

3.1 Knowledge of the field of rainfall forecasting

Rainfall forecasting is a critical discipline within the broader field of meteorology, tasked with the major challenge of predicting one of the most variable and impactful weather phenomena. In this field, a sophisticated set of methods is meticulously used to forecast rainfall patterns with exceptional accuracy and dependability. These strategies include a range of advanced techniques, from traditional statistical methods to state-of-the-art deep learning models, each carefully designed to manage the intricacies of weather-related information.

Embracing statistical methodologies, the field of rainfall forecasting delves into the complexities of regression analysis and time series examination. These methodologies combine historical rainfall datasets with a tapestry of meteorological variables, resulting in the construction of predictive models that reveal the underlying trends and patterns governing rainfall dynamics. The use of statistical techniques allows for the identification of correlations and dependencies within the data, providing a robust foundation for forecasting models. By analyzing past trends, these models can anticipate future rainfall events with a degree of confidence that is crucial for planning and preparedness.

In parallel, physical modeling stands as a strong key element in the domain of rainfall prediction. Numerical weather prediction models and hydrological models stand as embodiments of scientific rigor, smoothly integrating the foundational principles of atmospheric dynamics and thermodynamics to simulate the complex tapestry of atmospheric processes. These models are grounded in the laws of physics, providing a deterministic approach to forecasting that is essential for understanding the physical mechanisms driving rainfall. By solving these complexities, these models pave the way for astute predictions of future rainfall events and hydrological impacts, such as floods and droughts.

Venturing into remote sensing methodologies, the domain of rainfall forecasting magnifies the role of satellite and radar remote sensing

technologies. Satellite data scrutinization emerges as a linchpin, enabling a comprehensive monitoring of atmospheric and surface alterations crucial for discerning precipitation trends. The high-resolution imagery provided by satellites allows for the detection of minute changes in cloud patterns and humidity levels, which are critical indicators of impending rainfall. Paralleling this, radar technologies perform a key role in rainfall forecasting, delicately tracking and understanding rainfall cloud patterns and intensities, thereby fortifying real-time rainfall monitoring and alert mechanisms with unparalleled accuracy. The ability of radar to provide real-time, high-resolution data makes it an essential tool in the meteorologist's arsenal.

Incorporating deep learning techniques, especially neural network models, brings a contemporary viewpoint to the academic discourse on precipitation prediction. This method has sparked a revolutionary shift in our approach to forecasting rainfall patterns. Backed by artificial intelligence, these models meticulously handle vast quantities of meteorological data, elevating the precision and efficacy of rainfall forecasts to an unparalleled degree. The versatility and adaptability of neural networks allow them to glean insights from intricate, nonlinear associations within the data, providing a more nuanced prediction strategy. By harnessing the capabilities of deep learning, meteorologists are now able to produce forecasts that are not only more accurate but also more attuned to the dynamic nature of weather systems.

Moreover, the integration of these methodologies is not merely additive; it is synergistic. The combination of statistical analysis, physical modeling, remote sensing, and deep learning creates a multifaceted approach that is greater than the sum of its parts. This holistic strategy allows for the cross-verification of predictions, enhancing the reliability of forecasts and providing a more comprehensive understanding of the factors influencing rainfall.

The rich tapestry of these methodologies, meticulously woven with a synthesis of diverse technologies and data streams, heralds a new era in

meteorological forecasting, promising heightened accuracy, timeliness, and resilience in the field of rainfall predictions. As the field continues to evolve, the ongoing integration of advanced technologies and methodologies will undoubtedly lead to further advancements in our ability to predict and prepare for the rain.

3.1.1 The relationship between radar echo data and rain fall

The relationship between radar technology and the measurement of rainfall is essential in modern meteorological research. Radar serves as a precise tool for monitoring the atmosphere, playing a crucial role in accurately assessing rainfall intensity. Its incorporation into meteorological studies represents a significant advancement, enhancing our capacity to comprehend and forecast precipitation events.

Radar technology, an indomitable force in the meteorological arsenal, stands as a sentinel at the forefront of rainfall intensity assessment. Its strength lies in the complex web of signals it emits and receives, delicately capturing the minutiae of rainfall cloud dynamics and structure. The radar's ability to penetrate cloud layers and provide a detailed view of the internal structure of precipitation systems is unparalleled. By analyzing the returned signals, meteorologists can discern the size, shape, and density of raindrops, which are critical factors in determining rainfall intensity.

By solving the subtle nuances embedded within the radar returns, meteorologists glean profound insights into the spatial distribution, temporal evolution, and intensity gradients of rainfall events, painting a vivid tapestry of precipitation patterns with exquisite detail. The radar's temporal resolution allows for the tracking of rainfall events as they unfold, providing a dynamic view of precipitation that is crucial for short-term forecasting and nowcasting applications.

The correlation between radar data and rainfall intensity unfolds as a narrative of scientific acuity and analytical rigor. Through advanced algorithms and signal processing techniques, radar data undergoes a

metamorphosis into a rich tapestry of information, revealing the complex nuances of rainfall intensity variations. The radar's sensitivity to the vertical profile of rainfall, including the detection of melting layers and the differentiation between rain and snow, adds another layer of complexity and precision to the analysis.

From the gentle drizzles that caress the earth to the tumultuous downpours that cascade with unbridled ferocity, radar technology captures these nuances with a precision that borders on the sublime. The radar's ability to provide a continuous, real-time stream of data allows meteorologists to issue timely warnings and alerts, which are essential for public safety and disaster preparedness.

Moreover, radar-derived rainfall intensity data serves as a cornerstone in hydro-meteorological modeling, facilitating the adjustment and validation of predictive models that underpin flood forecasting and water resource management. The radar's role in these models is to provide a high-resolution, three-dimensional view of precipitation, which is critical for understanding the spatial and temporal variability of rainfall. This information is vital for the accurate simulation of runoff, infiltration, and evapotranspiration processes within hydrological models.

The interdependence between radar technology and rainfall intensity measurement constitutes a significant chapter in the annals of scientific research. Each radar echo carries the quintessence of meteorological insights, contributing to the advancement of our preparedness against the capriciousness of weather systems. The deepening comprehension of radar technology and its pivotal function in quantifying rainfall intensity concurrently augments our proficiency in forecasting and attenuating the ramifications of severe meteorological events.

In essence, the interplay between radar technology and rainfall intensity reveals a field of scientific inquiry characterized by precision, nuance, and unwavering dedication to solving the mysteries of atmospheric dynamics. As radar technology continues to evolve and innovate, its symbiotic relationship with rainfall intensity stands as a testament to the

unyielding quest for precision and accuracy in modern meteorological discourse. The ongoing advancements in radar technology, such as dual-polarization capabilities and phased array systems, promise to further enhance our ability to observe and interpret the complex world of precipitation.

3.1.2 The relationship between ground observation and rainfall forecast

The subtle relationship between ground station data and rainfall intensity is a central key element in meteorological studies, essentially important for the understanding and prediction of precipitation patterns. Within the complex field of meteorological analysis, this relationship is a key area of focus, capturing the essence of current hydro-meteorological research. Ground station data, stemming from a tradition of meticulous observation and empirical study, is essential for understanding the complex relationship between various atmospheric parameters and the force of rainfall events.

Ground station data forms the foundation of meteorological monitoring efforts, serving as a reliable source of empirical evidence for the assessment of rainfall intensity. These stations are equipped with an array of high-precision instruments designed to measure atmospheric variables including temperature, pressure, humidity, and wind speed. This data provides a detailed and accurate reflection of the atmospheric conditions that are critical in the formation and intensity of rainfall events. By meticulously analyzing the temporal changes in these parameters in conjunction with actual rainfall occurrences, meteorologists can derive a deeper comprehension of the factors influencing the variability in rainfall intensity.

The integration of ground station data with rainfall intensity analysis is a scientific process that demands rigorous methodology and precision. It involves the application of statistical methods, spatial interpolation techniques, and trend analysis algorithms to transform the raw data collected from ground stations into a detailed understanding of the interactions between meteorological variables and rainfall patterns. This

comprehensive analysis allows for the identification of subtle patterns and the prediction of rainfall events with a high degree of accuracy, thereby enhancing the reliability of weather forecasts.

Moreover, the incorporation of ground station data into hydro-meteorological models is crucial for various applications such as flood prediction, drought monitoring, and water resource management. These models rely on accurate ground station data to simulate and predict rainfall patterns, which is essential for anticipating extreme weather events. The use of this data in model adjustment ensures that forecasts are as precise and reliable as possible, providing critical information for decision-makers and stakeholders to mitigate the impacts of weather-related hazards.

The value of ground station data in rainfall intensity analysis is underscored by its role in validating and refining predictive models. As new data is collected and analyzed, models can be progressively adjusted to more closely reflect observed patterns, leading to improved forecast accuracy. This ongoing process of model refinement and validation is vital for advancing the field of meteorology and enhancing our predictive capabilities regarding weather events.

Furthermore, the spatial distribution of ground stations is a critical determinant of the quality and representativeness of the data collected. A well-planned network of stations can provide a more comprehensive overview of meteorological conditions across a region, leading to more accurate and reliable analyses. The strategic placement and density of these stations can significantly impact the resolution and dependability of the data, underscoring the importance of thoughtful station layout in meteorological research.

Technological advancements are continually enhancing the capabilities of ground stations. The development of new sensors and measurement techniques is providing even more detailed and precise data, which is invaluable for meteorological analysis. These innovations are expanding the scope and accuracy of ground station data, increasing its utility in meteorological studies and its contribution to the overall understanding

of weather patterns.

As ground station networks expand and technology progresses, the relationship between ground-based observations and the analysis of rainfall intensity becomes ever more significant. This synergy is a testament to the ongoing pursuit of knowledge and precision in the field of hydro-meteorological research, highlighting the dedication to enhancing our predictive models and better preparing for the variability of weather conditions.

3.2 Deep learning related theories

3.2.1 Basic Artificial neural networks

Artificial Neural Networks (ANNs) represent a advanced class of machine learning models, drawing their inspiration from the biological architecture and operational patterns of neurons within the human brain. These networks are designed to mimic the brain's ability to learn from observational data and adapt to new information, making them a powerful tool for a wide array of applications, from pattern recognition to complex decision-making processes [23].

ANNs are characterized by their layered structure, where each layer consists of multiple artificial neurons. These neurons are not merely simplified replicas of biological neurons but are mathematical functions that transform input data through a series of weighted connections. The process begins with the input layer, which receives the initial data. This data then propagates through a series of hidden layers, where each neuron processes the data it receives and passes it to the next layer until the final output is produced by the output layer.

The perceptron, as the fundamental computational unit of an artificial neuron, plays a essential role in this process. It computes a weighted sum of the input features it receives, each associated with a specific weight that determines the importance of that feature in the overall computation. Once the weighted sum is calculated, a nonlinear

activation function is applied. This function introduces the non-linear properties to the model, allowing it to learn and model complex patterns in the data.

The selection of an activation function is pivotal, as it dictates the range of problems that an Artificial Neural Network (ANN) can address effectively. Typical activation functions encompass the sigmoid function, apt for binary classification tasks, the hyperbolic tangent function, and the rectified linear unit (ReLU), which has gained widespread acceptance owing to its efficacy in deep learning architectures.

One of the key advantages of ANNs is their adaptability, achieved through a process known as backpropagation. This is an algorithm that allows the network to learn from its errors by adjusting the weights of the connections between neurons. Through iterative training, the network minimizes the difference between the predicted output and the actual target, improving its predictive accuracy over time.

The depth and complexity of ANNs can vary widely, from relatively simple feedforward networks with a few layers to more complex architectures such as convolutional neural networks (CNNs), which are particularly skilled at processing grid-like data such as images, and recurrent neural networks (RNNs), which are designed to handle sequential data.

ANNs have been successfully applied in various domains, including but not limited to image and speech recognition, natural language processing, medical diagnosis, and financial forecasting. Their ability to handle large volumes of high-dimensional data and to uncover complex patterns makes them essential in the era of big data and artificial intelligence.

Despite their success, ANNs also present challenges. Overfitting, where the network learns the training data too well and fails to generalize to new data, is a common issue. Regularization techniques and cross-validation are often employed to mitigate this problem. Additionally, the computational intensity of training deep networks

requires significant computational resources, which has led to the development of more efficient training algorithms and hardware accelerators.

As research progresses, the field of ANNs continues to evolve, with ongoing work focusing on improving network architectures, optimization algorithms, and understanding the theoretical underpinnings of their function and learning capabilities. The development of new types of ANNs, such as those incorporating attention mechanisms or memory elements, is expanding the scope of problems they can address.

In conclusion, ANNs are a versatile and powerful class of machine learning models that have proven their efficacy across a multitude of disciplines. Their capacity for learning complex representations and patterns, along with their adaptability and robustness, positions them at the forefront of modern artificial intelligence research and application. Mathematically, a perceptron can be defined as (3.1):

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{3.1}$$

where x_i are the input values, w_i are the corresponding weights, b is the bias term, and f is the activation function.

Neurons are often organized into layers, where each layer consists of multiple neurons that receive the same inputs and produce outputs that are fed into the next layer. The input layer receives the raw input data, while the output layer produces the final output of the network. The layers in between are called hidden layers. The illustration of ANN basic structure is presented in Fig. 3.1.

The Input Layer represents the entry point for data into the network. Each node in this layer corresponds to an input feature. The input features are the variables that the network will learn to associate with the desired output. In the context of an ANN, these could be pixel values in an image, measurements from sensors, or any other relevant data points.

Subsequent to the input layer are one or more Hidden Layers, which constitute the primary locus for computation and data manipulation. Within each hidden layer, neurons receive weighted inputs from the preceding layer's neurons, aggregate these inputs, and subsequently apply an activation function to produce an output signal. The role of hidden layers is to distill features and construct a more abstract portrayal of the input data. The intricacy and depth of the network may fluctuate in accordance with the complexity of the task under consideration. Networks with greater depth, comprising additional hidden layers, are capable of modeling more elaborate patterns; however, they necessitate a larger volume of data and increased computational resources for training.

Finally, the Output Layer is the final component of the ANN. It produces the predictions or decisions based on the processed information from the hidden layers. The number of neurons in the output layer typically corresponds to the number of output classes or the dimensionality of the prediction. For example, in a binary classification task, there might be a single neuron representing the probability of one class, while in a multi-class classification or regression task, there might be multiple neurons.

The diagram also indicates that the network has connections between the layers, with each connection having an associated weight. These weights are the parameters that the ANN learns by adjusting them during the training process to minimize the difference between the predicted output and the actual target.

In summary, the basic structure of an ANN consists of an input layer that feeds data into the network, one or more hidden layers that process the data through weighted connections and activation functions, and an output layer that produces the final prediction based on the learned features.

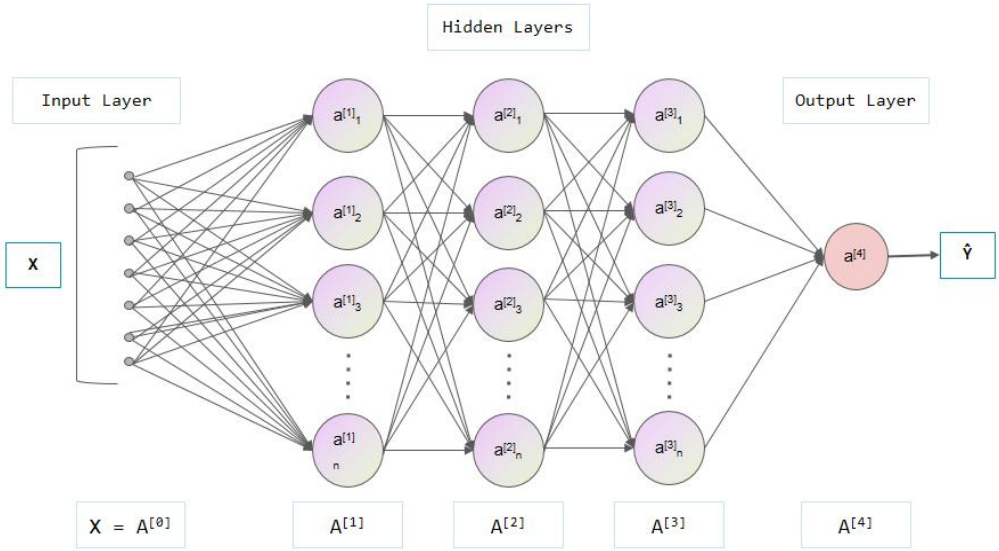


Figure 3.1: Basic structure of ANN

3.2.2 Activation functions

Activation functions are a key component of artificial neural networks (ANNs) that introduce nonlinearity into the output of a neuron. They are applied to the weighted sum of the inputs and biases of a neuron to produce its output, which is then passed on to the next layer of the network.

There are several types of activation functions that can be used in ANNs, including:

1. Sigmoid activation function is a popular choice in neural networks, particularly for binary classification problems. It is a smooth, S-shaped function that maps any input value to a range between 0 and 1.

The mathematical definition of the sigmoid function can be expressed as (3.2):

$$\text{Sigmoid}(x)=\sigma(x)=\frac{1}{1+e^{-x}} \tag{3.2}$$

where x is the input value.

The sigmoid function has several properties that make it useful in neural networks. First, it is differentiable, which allows for efficient backpropagation of errors during training. Second, it has a bounded output range, which can help prevent the network from becoming unstable or saturating (i.e., getting stuck at extreme values) during

training.

Nevertheless, the sigmoid function also exhibits certain drawbacks. A notable issue is that its output tends to saturate at 0 or 1 for large positive or negative input values, which can impede learning if the weights are not initialized optimally. Additionally, the output of the sigmoid function is not zero-centered, potentially decelerating the network's convergence during the training phase.

Despite these constraints, the sigmoid function continues to be a favored option for specific categories of neural networks, especially those employed in binary classification tasks.

2. The hyperbolic tangent (tanh) activation function is a commonly used activation function in neural networks. It is a smooth, S-shaped function that maps any input value to a range between -1 and 1.

The mathematical definition of the tanh function can be expressed as (3.3):

$$\text{Tanh}(x)=2\text{Sigmoid}(2x)-1 \qquad (3.3)$$

where x is the input value.

The tanh function has several properties that make it useful in neural networks. First, it is differentiable, which allows for efficient backpropagation of errors during training. Second, it has a zero-centered output range, which can help the network converge more quickly during training.

However, the tanh function also has some limitations. One issue is that its output saturates at -1 or 1 for large positive or negative input values, which can make it difficult for the network to learn if the weights are initialized poorly. Another issue is that the output of the tanh function is not as sparse as the ReLU function, which can make it less efficient for certain types of networks.

Despite these limitations, the tanh function remains a popular choice for certain types of neural networks, particularly those used for regression

tasks.

3. Rectified Linear Unit (ReLU) activation function is a commonly used activation function in deep learning. It is a piecewise linear function that returns the input if it is positive, and zero otherwise.

The mathematical definition of the ReLU function can be expressed as (3.4):

$$\mathrm{ReLU}(x)=\max(0,x) \tag{3.4}$$

where x is the input value.

The ReLU function has several properties that make it useful in neural networks. First, it is computationally efficient to compute, since it involves only simple thresholding of the input. Second, it has a bounded output range on one side (i.e., it is non-negative), which can help prevent the network from becoming unstable during training.

However, the ReLU function also has some limitations. One issue is that the output is not differentiable at zero, which can cause problems during backpropagation if the gradient is not handled carefully. Another issue is that some ReLU neurons can "die" during training, meaning that they become stuck at zero and stop contributing to the network's output.

Despite these limitations, the ReLU function remains a popular choice for deep neural networks, particularly those with many layers.

4. The softplus activation function is a smooth, nonlinear function that is often used as an alternative to the ReLU function in neural networks. It is defined as the logarithm of the exponential function plus one, and maps any input value to a range between 0 and infinity.

The mathematical definition of the softplus function can be expressed as (3.5):

$$\mathrm{Softplus}(x)=\ln(1+e^{x}) \tag{3.5}$$

where x is the input value.

The softplus function has several properties that make it useful in neural networks. First, it is differentiable, which allows for efficient backpropagation of errors during training. Second, it has a smooth output range that is similar to the ReLU function, which can help prevent the network from becoming unstable during training.

However, the softplus function also has some limitations. One issue is that its output range is unbounded on the positive side, which can make it less efficient for certain types of networks. Another issue is that it is more computationally expensive to compute than the ReLU function.

In spite of these limitations, the softplus function continues to be a favored option for specific types of neural networks, especially those utilized in regression tasks.

The Mish activation function is a recently proposed activation function for neural networks that has shown to provide improved performance over other commonly used activation functions such as ReLU and tanh. It is a smooth, nonlinear function that is defined as the product of the input and the hyperbolic tangent of the softplus function.

The mathematical definition of the Mish function can be expressed as (3.6):

$$\text{Mish}(x)=x(\text{Tanh}(\text{Softplus}(x))) \qquad (3.6)$$

Here, x denotes the input value.

The Mish function possesses several attributes that render it beneficial in neural networks. Firstly, it is differentiable, facilitating efficient backpropagation of errors during the training phase. Secondly, it exhibits a smooth output range akin to that of the ReLU function, which can aid in maintaining network stability during training. Thirdly, its non-monotonic nature can assist the network in learning more intricate patterns within the data.

The Mish function has been shown to provide improved performance over other commonly used activation functions on a variety of tasks,

including image classification and natural language processing.

3.2.3 Convolutional neural networks

Convolutional Neural Networks (CNNs) are a specialized form of deep learning algorithms that have risen to prominence due to their proficiency in processing and analyzing image and video data. CNNs are particularly skilled at leveraging the inherent spatial structure of images, which are essentially two-dimensional arrays of pixels. The power of CNNs lies in their ability to automatically and adaptively learn from data, making them highly suitable for a multitude of visual recognition tasks [29].

Central to CNNs are the convolutional layers, tasked with feature extraction. These layers employ a collection of learnable filters, or kernels, that are convolved with the input image to generate feature maps. Each filter is tailored to identify particular features across different scales and orientations, enabling the network to discern local patterns in the image. The convolution operation is computationally efficient and also contributes to forming a translation-invariant representation, implying that the network can identify features irrespective of their location within the image.

Beyond the convolutional layers, CNNs often incorporate pooling layers to reduce the spatial dimensions of the feature maps, which in turn reduces the complexity of the network. Pooling operations, such as max pooling, help in extracting the most important information while discarding less relevant details. This process also contributes to making the detected features more robust to variations in scale and orientation.

A pivotal element of CNNs is the incorporation of non-linear activation functions, which imbue the model with non-linear characteristics, enabling it to discern complex patterns. The Rectified Linear Unit (ReLU) is frequently employed as an activation function owing to its straightforwardness and efficacy in mitigating the vanishing gradient issue that may arise with conventional sigmoid or tanh functions.

As data traverses the convolutional and pooling layers, it ultimately arrives at the fully connected layers, where a sequence of dense neural network layers conduct high-level reasoning based on the extracted features. These layers are tasked with making the final decisions, such as categorizing the image into one of multiple classes or executing regression tasks.

One of the notable advantages of CNNs is their capacity for automatic feature learning, which stands in contrast to traditional machine learning approaches that frequently necessitate manual feature engineering. This capability not only conserves time and effort but also enables the network to uncover complex and abstract features that may be challenging to delineate manually.

CNNs have been applied in a multitude of domains beyond image classification, encompassing natural language processing, where they are utilized for sentence classification and machine translation, and in medical imaging, where they aid in the detection and diagnosis of diseases. The versatility of CNNs is a testament to their proficiency in handling diverse data types and addressing a broad spectrum of problems.

Despite their numerous advantages, CNNs also present challenges. Training deep CNNs requires a large amount of annotated training data, which can be a bottleneck in domains where data annotation is time-consuming or expensive. Additionally, CNNs can suffer from overfitting, especially when the network is complex and the amount of training data is limited. To combat overfitting, techniques such as dropout, data augmentation, and regularization are employed.

In conclusion, CNNs represent a significant advancement in the field of computer vision and machine learning. Their success is attributed to their ability to automatically learn rich feature representations from data, making them a go-to solution for a variety of image and video analysis tasks. Ongoing research continues to explore new architectures, training techniques, and applications for CNNs, ensuring their continued evolution and relevance in the ever-expanding field of artificial

intelligence.

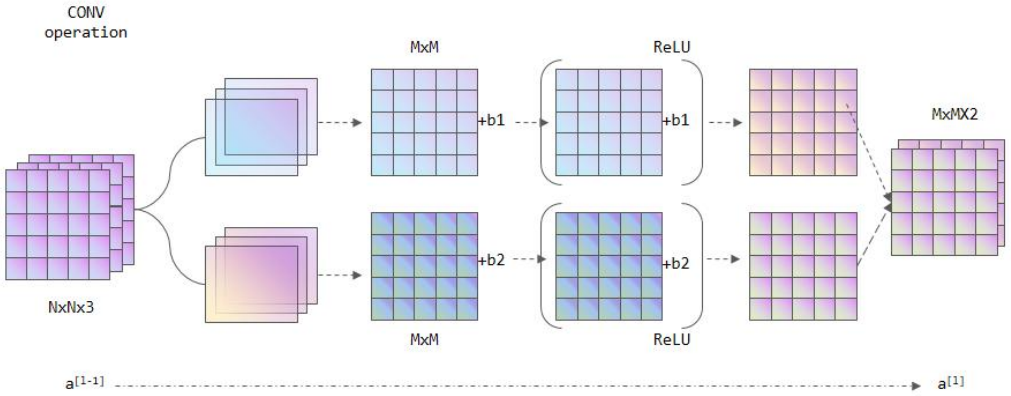


Figure 3.2: Basic structure of CNN

As depicted in Fig. 3.2, the fundamental component of a CNN is the convolutional layer, comprising a series of learnable filters that are applied to the input data. Each filter traverses the input data, calculating a dot product between its weights and the input values at each location, resulting in a 2D activation map that emphasizes the presence of specific features in the input. The outputs from multiple filters are subsequently combined to constitute the output of the convolutional layer.

After one or more convolutional layers, CNNs typically include one or more pooling layers, which downsample the output of the convolutional layers by taking the maximum or average value over small regions of the output. This helps to reduce the dimensionality of the feature maps and make the network more efficient.

Ultimately, one or more fully connected layers are generally employed to generate the final output of the network, such as a classification or regression outcome.

CNNs have been shown to be highly effective for a variety of computer vision tasks, including image classification, object detection, and semantic segmentation. Mathematically, a convolutional layer can be defined as:

$$y_{i,j,k} = \sum_{l=1}^{C} \sum_{m=1}^{H_f} \sum_{n=1}^{W_f} x_{i+m-1,j+n-1,l} \cdot w_{m,n,l,k} + b_k \tag{3.7}$$

where x is the input tensor, w is the filter tensor, b is the bias term, H_f and W_f are the height and width of the filter, C is the number of channels in the input, and y is the output tensor.

3.2.4 Recurrent neural networks

Recurrent Neural Networks (RNNs) represent a significant class of neural networks that are uniquely tailored for sequence analysis tasks, such as natural language processing (NLP), speech recognition, time series prediction, and more. Unlike traditional feedforward neural networks, which process input data in a single pass through the network, RNNs are designed to handle sequential data by maintaining a form of memory that captures information from previous time steps [30].

The defining characteristic of RNNs is their recurrent structure, which allows hidden units to connect back to themselves, forming a loop. This loop enables the network to process input data sequentially and to keep track of the context by propagating error gradients across time steps. The recurrent nature of RNNs is what gives them their strength in modeling tasks where the temporal order and context are crucial.

With each input processed, an RNN updates its hidden state, which subsequently serves as the basis for the next time step. This hidden state can be thought of as the network's memory, encoding information from past inputs to influence the output and decisions made at future time steps. The ability to maintain a dynamic memory allows RNNs to exhibit temporal dynamic behavior and to learn from data that has time-dependent features.

However, RNNs face challenges, particularly with the issue of vanishing or exploding gradients, which can hinder the network's ability to learn long-range dependencies in sequences. This issue emerges as the gradients may diminish or inflate significantly when propagated backward through time, thereby impeding the network's ability to adjust its weights effectively.

To address these challenges, various architectures have been developed,

such as the Long Short-Term Memory (LSTM) network and the Gated Recurrent Unit (GRU). Both LSTM and GRU introduce mechanisms to regulate the flow of information, preventing the vanishing and exploding gradients by using gating units that learn to forget what information to keep or discard at each time step.

LSTM networks consist of a cell state that acts as a conveyor belt, carrying information through time with three multiplicative gates: the input gate, the output gate, and the forget gate. These gates regulate the flow of information into and out of the cell state, allowing the network to maintain a long-term memory while still being able to update its hidden state based on new inputs.

Conversely, GRUs streamline the architecture by merging the input and forget gates into a singular update gate, and they lack a distinct cell state. Instead, the hidden state itself serves to convey information forward, rendering the model more computationally efficient.

The application of RNNs extends beyond sequence modeling and has found success in various domains. In NLP, RNNs are used for tasks such as language modeling, text classification, and machine translation. In speech recognition, they process the temporal sequence of audio signals to transcribe speech into text. RNNs are also applied in the field of bioinformatics for DNA sequence analysis and in financial markets for time series prediction.

Despite their versatility, RNNs require careful design and tuning. The choice of hyperparameters, such as the number of hidden units, learning rate, and the sequence length, can significantly affect the performance of the network. Additionally, the training of RNNs can be more complex compared to feedforward networks due to their sequential nature and the need to manage dependencies across time steps.

In summary, RNNs constitute a potent category of neural networks that have exhibited exceptional aptitude in managing sequential data. Their capacity to retain a memory of prior inputs and to learn from temporally dependent data renders them an invaluable instrument across diverse

domains. Ongoing research continues to explore new architectures and training techniques to further enhance the capabilities of RNNs and to address their limitations.The basic building block of an RNN is the recurrent unit, which consists of a set of learnable weights that are applied to the input and the previous hidden state to produce a new hidden state. The output of the recurrent unit is typically used as input to the next unit in the sequence, along with the next input. Mathematically, an RNN can be defined as:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{3.8}$$

where x_t is the input at time t, h_t is the hidden state at time t, W_{xh} and W_{hh} are the weight matrices for the input and recurrent connections, respectively, b_h is the bias term, and f is a nonlinear activation function such as the hyperbolic tangent or sigmoid function.

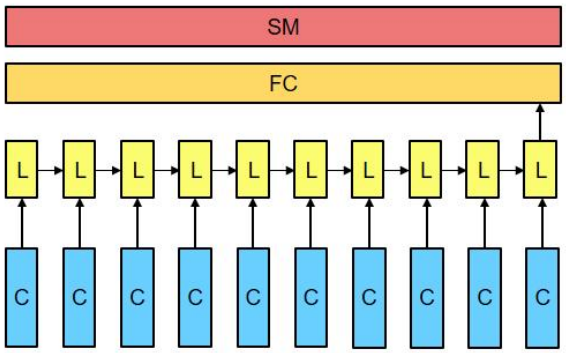


Figure 3.3: Basic structure of RNN

RNNs can be trained using backpropagation through time (BPTT), which involves computing gradients at each time step and accumulating them over the entire sequence before updating the weights. Nevertheless, RNNs may encounter the vanishing gradient problem, in which gradients diminish considerably as they are propagated backward through time, thereby impeding the learning of long-term dependencies.

To tackle this issue, several variants of RNNs have been introduced, such as the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. These utilize additional learnable gates to

regulate the flow of information within the network.

3.2.5 Graph convolutional networks

Graph Convolutional Networks (GCNs) represent a pattern shift in the field of machine learning, particularly designed to process and analyze data that is inherently structured as graphs. Graph-structured data is common in numerous domains, including social networks, citation networks, molecular graphs, and more, where the relationships between entities are as important as the entities themselves [6].

GCNs extend the powerful convolutional operations from the field of Euclidean data, such as images, to the non-Euclidean domain of graphs. The core concept is to perform a convolution-like operation that aggregates information from a node's neighbors, effectively capturing the local graph structure and the features of the connected nodes. This aggregation allows GCNs to learn representations of nodes that are informed by their surroundings, making them suitable for tasks such as node classification, graph classification, and link prediction.

One of the key innovations of GCNs is their ability to generalize the convolution operation to graph-structured data. In traditional convolutional neural networks, the convolution operation is defined in a spatial grid-like structure. In contrast, graph convolution must accommodate the irregular and different structures of graphs. This is achieved by using adjacency matrices that encode the connectivity patterns of the graph, along with node feature matrices that contain the attributes of the nodes.

The architecture of GCNs typically consists of a series of graph convolutional layers, where each layer applies a convolution operation that updates the node representations based on the features of their neighbors. The convolution operation in GCNs involves a weighted sum of the neighbor features, scaled by learnable weights that are adjusted during training. This process is repeated across layers, with each layer building upon the representations learned in the previous layer to

capture increasingly complex patterns.

GCNs also incorporate activation functions, such as the Rectified Linear Unit (ReLU), after each convolution operation. These activation functions introduce non-linearity into the model, allowing it to learn more complex relationships within the data.

A critical component of GCNs is their ability to handle variable-sized graphs and to be robust to the graph's structure. This flexibility is crucial for real-world applications, where graphs can range from small and dense to large and sparse. Techniques such as graph pooling and unpooling have been developed to enable GCNs to process graphs of different sizes and to learn hierarchical representations.

The success of GCNs in various applications has led to an increase in interest in graph-based machine learning. They have been applied to a wide range of tasks, including but not limited to, social network analysis, recommendation systems, drug discovery, and traffic prediction. For instance, in social network analysis, GCNs can help identify influential users or detect communities within the network. In drug discovery, they can predict the properties of molecules and suggest potential drug candidates.

Despite their advantages, GCNs also face challenges. One of the main challenges is dealing with large and complex graphs, which can be computationally expensive to process. Additionally, the quality of the learned representations is highly dependent on the graph's structure and the features of the nodes.

In conclusion, GCNs are a powerful tool for learning from graph-structured data. Their ability to capture the complex patterns and relationships within graphs has made them a valuable asset in many domains. Ongoing research continues to explore new architectures, training techniques, and applications for GCNs, ensuring their continued evolution and relevance in the field of machine learning.

The basic building block of a GCN is the graph convolutional layer,

which applies a linear transformation to the features of each node and its neighbors in the graph. The output of the convolutional layer is then passed through a nonlinear activation function, such as the ReLU function. Mathematically, a graph convolutional layer can be defined as:

$$H^{(l+1)} = f(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{3.9}$$

where H^l is the feature matrix for layer l, W^l is the weight matrix for layer l, $\hat{A}$ is the normalized adjacency matrix of the graph, $\hat{D}$ is the diagonal degree matrix of the graph, and f is a nonlinear activation function.

The Fig.3.4 illustrates the basic structural components of a Graph Convolutional Network (GCN). At its core, a GCN is designed to process and analyze data that is represented as a graph, where nodes signify individual entities and edges denote the relationships between these entities.

The Input Layer of the GCN, depicted at the bottom of the figure, is where the initial node features are introduced into the network. Each node in the graph is represented by a feature vector, which contains numerical attributes associated with that node. These features could be any relevant information depending on the application, such as the degree of a node in a social network graph or the atomic properties in a molecular graph. In our research work, the features are the weights between the pixel values of cloud layer thickness in radar images, ranging from 0 to 255.

Above the input layer, we have the Hidden Layers. These layers are where the primary computation of the GCN takes place. Within each hidden layer, a convolutional operation is performed over the nodes and their neighbors. The convolutional operation aggregates the features of a node and its neighbors, updating the node's feature representation to reflect the structure of the graph and the collective information of its local neighborhood. This process is repeated for each node in the layer, and the result is passed on to the next hidden layer, allowing the network

to learn increasingly complex patterns and relationships within the graph.

The Output Layer, shown at the top of the figure, is the final component of the GCN. After the features have been transformed through the hidden layers, the output layer produces the final predictions or classifications for each node. The specific form of this output will depend on the task at hand. For example, in a node classification task, the output layer might provide a probability distribution over possible node labels.

The figure also indicates that there are multiple hidden layers, suggesting a deep GCN architecture. Deep architectures allow the network to capture more complex patterns by learning hierarchical representations of the graph. However, designing deep GCNs requires careful consideration to avoid overfitting and to ensure that the network can effectively learn from the graph structure.

In essence, the fundamental architecture of a GCN comprises an input layer for initial feature representation, one or more hidden layers for feature transformation and aggregation, and an output layer for generating predictions based on the learned node features.

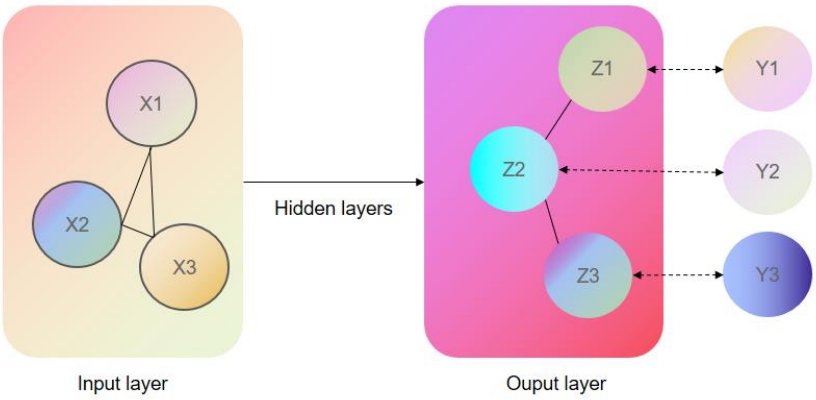


Figure 3.4: Basic structure of GCN

3.2.6 Transformer

The Transformer model, introduced in the 2017 paper "Attention Is All You Need" by Vaswani et al., has become a cornerstone in the field of deep learning, particularly for tasks involving the processing of sequential data [45]. Unlike the traditional recurrent neural networks (RNNs) that process input data in a sequential manner with a fixed-length context window, the Transformer model is built upon the innovative concept of self-attention. This mechanism enables the model to weigh the relevance of various parts of the input sequence dynamically, allowing for a more flexible and contextually aware processing of information.

The self-attention mechanism in Transformers is a significant departure from the sequential processing pattern. It allows the model to consider the entire sequence simultaneously, capturing global dependencies in a single operation. This is achieved by calculating attention scores that measure the relationship between different words or tokens in the sequence. The attention scores are then used to create a weighted sum of the input tokens, which is used to update the representation of each token.

One of the key advantages of the self-attention mechanism is its ability to handle long-range dependencies in the data. In many sequence tasks, such as language translation, the meaning of a word can depend on context that is far away in the sentence. Traditional RNNs struggle with this due to the vanishing gradient problem, but the Transformer's self-attention mechanism can easily capture these long-range dependencies.

The Transformer architecture also incorporates a multi-head attention approach, which allows the model to jointly attend to information from different representational subspaces at different positions. This provides a richer representation of the sequence and enables the model to capture a diverse set of contextual relationships.

Following the self-attention layers, the Transformer model includes feed-forward neural networks that apply a series of linear transformations to the output of the attention layers. These feed-forward networks are crucial for allowing the model to learn complex functions of the attended features.

The Transformer's encoder-decoder architecture represents another salient characteristic. The encoder is tasked with processing the input sequence, whereas the decoder is responsible for generating the output sequence. Both the encoder and decoder consist of a series of identical layers, facilitating a deep and hierarchical processing of sequential data. The encoder and decoder layers are interconnected via supplementary attention layers, which enable the model to concentrate on various segments of the input while generating each component of the output.

The Transformer model also employs positional encoding to convey information regarding the relative or absolute position of the tokens within the sequence. This is crucial as the self-attention mechanism does not inherently account for the order of the tokens. The efficiency of the Transformer model is further augmented by the implementation of layer normalization and residual connections. These techniques aid in stabilizing the learning process and enable the model to be trained with a substantial number of layers without the risk of gradients vanishing or exploding.

The Transformer's impact has been profound, not only in machine translation but also in a wide range of other applications such as language understanding, question answering, and text summarization. Its ability to handle complex sequences and capture subtle relationships between elements within the sequence has made it a versatile and powerful tool in the field of natural language processing.

Despite its numerous strengths, the Transformer model also faces challenges. The computational cost of the self-attention mechanism can be high, especially for long sequences, as it requires computing attention scores for all pairs of tokens in the sequence. Researchers are actively exploring methods to improve the efficiency of the Transformer, such as

using sparse attention patterns or incorporating inductive biases that mimic the hierarchical structure of the data.

In conclusion, the Transformer model has ushered in a new era of sequence modeling with its self-attention mechanism, multi-head attention, and encoder-decoder architecture. It has proven to be a highly effective and flexible framework for a variety of tasks that involve sequential data. As the field continues to evolve, the Transformer is likely to remain at the forefront of research and development in deep learning and artificial intelligence.

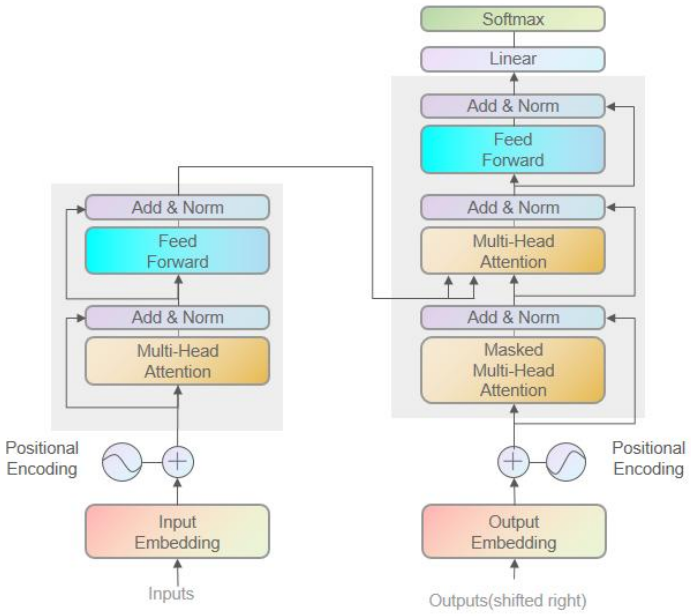


Figure 3.5: Basic structure of Transformer

The fundamental component of the Transformer is the self-attention mechanism, which calculates a weighted sum of the input sequence at each position, with the weights being adaptively learned based on the entire sequence's content. This enables the network to focus on various segments of the input sequence at each computational step and to capture long-range dependencies more effectively than conventional recurrent networks. Mathematically, the self-attention mechanism can be

defined as:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.10}$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors.

The Transformer also includes several other components, including multi-head attention, which allows the network to attend to multiple parts of the input simultaneously, and position-wise feedforward networks, which apply a nonlinear transformation to each position in the input sequence independently.

In this chapter, we have explored the landscape of neural network architectures that have shaped the field of artificial intelligence, particularly in the domain of machine learning. Each model—Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Graph Convolutional Networks (GCNs), and Transformers — offers unique capabilities tailored to specific types of data and tasks.

ANNs serve as the foundation, providing a general framework for learning from data through interconnected artificial neurons. They are the building blocks for more complex models and have been instrumental in a wide range of applications, from simple pattern recognition to complex decision-making processes.

CNNs have revolutionized the way we process image data, offering a powerful mechanism for feature extraction through convolutional operations. Their ability to capture local dependencies and spatial hierarchies has made them the go-to solution for computer vision tasks.

RNNs, with their inherent capability to maintain a memory of past inputs, excel in handling sequential data. They have been essential in advancing our ability to model time-series data and are particularly effective in natural language processing and speech recognition.

GCNs extend the power of neural networks to graph-structured data, allowing for the analysis of complex networks such as social networks and molecular structures. Their capacity to aggregate information from local neighborhoods within a graph has opened up new avenues in network analysis and data mining.

Lastly, Transformers have introduced a pattern shift with their self-attention mechanisms, enabling efficient processing of sequences and handling of long-range dependencies. They have become the architecture of choice for a variety of tasks in natural language processing, from translation to summarization.

Each of these models has contributed significantly to the advancement of AI, pushing the boundaries of what is computationally feasible and expanding our capability to solve real-world problems. As we continue to innovate and refine these architectures, we can expect further breakthroughs that will continue to reshape the landscape of AI research and application.

CHAPTER 4

DEEP LEARNING APPROACH COMBINING TRAJGRU AND GRAPH ATTENTION FOR ACCURATE CUMULONIMBUS DISTRIBUTION PREDICTION

4.1 Introduction

Drawing insights from the preceding chapter, this chapter introduces advanced technologies for a novel method of predicting cumulonimbus cloud distribution [61]. It is evident that accurate weather pattern prediction plays a crucial role in disaster mitigation and preparedness [62]-[64]. In subtropical regions, where heavy rainfall from cumulonimbus clouds can result in devastating flash floods and mudslides, conventional machine learning approaches have shown limitations in accurately forecasting cloud distribution.

Traditional machine learning techniques, while valuable in many applications, have exhibited limitations in the precise forecasting of cloud distribution, particularly in the complex and dynamic environment of subtropical weather patterns. These methods often rely on historical data and predefined models that may not fully capture the complexities and variability of atmospheric conditions. As a result, they may struggle to predict the formation and movement of cumulonimbus clouds with the required accuracy.

The limitations of conventional approaches have prompted researchers to explore new methodologies. One such advancement is the development of models that integrate advanced computational techniques, such as Graph Convolutional Networks (GCNs) and

Trajectory Gated Recurrent Units (TrajGRU), with attention mechanisms. These models are designed to better capture the spatial and temporal dynamics of cumulonimbus cloud formation and to provide more accurate and timely predictions.

GCNs, for instance, are skilled at processing graph-structured data, which is representative of the interconnected and complex nature of meteorological systems. They can analyze the relationships between different weather parameters and identify patterns that may indicate the development of cumulonimbus clouds. TrajGRU, on the other hand, is capable of handling the sequential nature of time series data, allowing the model to track the evolution of weather systems over time.

The integration of an attention mechanism further enhances the model's ability to focus on relevant features within the data, improving the prediction of cloud distribution. This is particularly important in subtropical regions, where the rapid development and movement of weather systems can lead to sudden and severe rainfall events.

The adoption of these advanced models represents a significant step forward in the field of meteorology, offering a more effective means of predicting the distribution of cumulonimbus clouds in subtropical regions. By improving the accuracy of weather forecasts, these models can contribute to more effective disaster preparedness and mitigation efforts, ultimately helping to protect communities from the potentially devastating effects of flash floods and landslides.

Therefore, this paper presents an innovative approach that integrates graph convolutional networks (GCN) and trajectory gated recurrent units (TrajGRU) with an attention mechanism to enhance the prediction of cumulonimbus cloud distribution using radar echo data. The promising results obtained from experiments utilizing both simulated and real-world datasets emphasize the potential of this approach to significantly improve prediction accuracy, highlighting its importance in protecting communities from weather-related disasters [65].

4.2 Methods

The proposed method contains three main structures: 1, anencoder–decoder network based on CNN and RNN; 2, anattention mechanism to enhance the feature extraction ability; and 3, a GCN layer to better build correlations between features. The structure diagram of the proposed GraphAT-NET is presented in Figure 4.1. In the following subsections, we introduce the mathematics and deployment of the details of the proposed method.

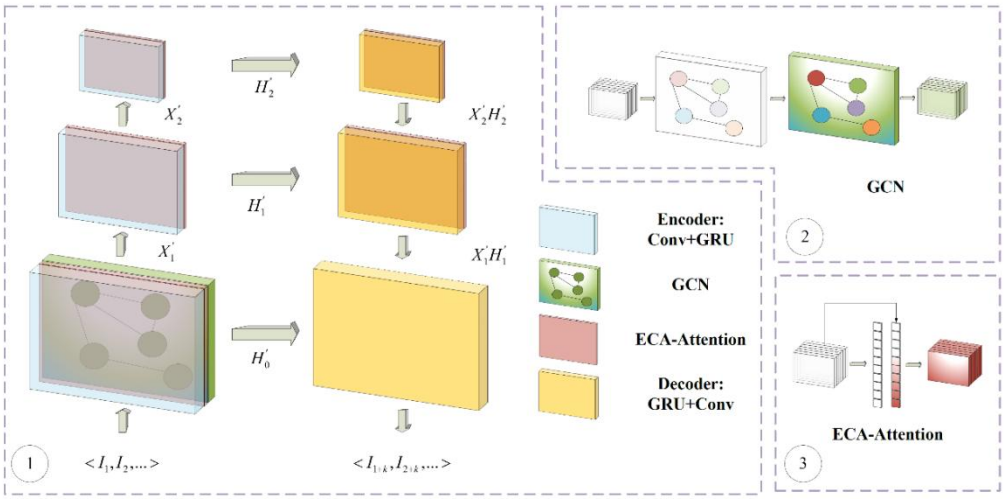


Figure 4.1: Framework of GraphAT-NET

As shown in Fig 4.1, part I is the overall architecture of the proposed model; part II is the structure diagram of GCN structure; and part III is the structure diagram of ECA-Attention.

4.2.1 Trajectory GRU Structure

As presented in the I part of Fig 4.1, the proposed method adopts RNN to build the correlation between time and radar data. The whole structure can be identified as a three-step encoding and decoding module.

We assume the input radar data I_1 are separated alone time dimension: $< I_1, I_2, \ldots >$. Then the prediction task can be arranged as forecasting k steps based on the inputs: $< I_{1+k}, I_{2+k}, \ldots >$. The task of

rainfall prediction is defined as a sequence learning-predicting mission. The main algorithm we adopted in the proposed method can be defined as follows:

The observations into n layers of RNN: $H_t^1, H_t^2, ..., H_t^n = h\left(I_{t-J+1}, I_{t-J+2}, ..., I_t\right)$ (here, h indicates operating history information), and then use another n layer of RNNs to generate the predictions based on these encoded states: $\hat{I}_{t+1}, \hat{I}_{t+2}, ..., \hat{I}_{t+K} = g(H_t^1, H_t^2, ..., H_t^n)$ (here, g indicates the gate operation of RNN).

Based on the introduction above, we define the methods of trajectory GRU as follows:

$$U_{t,}V_t = \gamma(X_t^{'}, H_{t-1}) \tag{4.1}$$

$$U_{t,}V_t = \gamma(X_t^{'}, H_{t-1}) \tag{4.2}$$

$$Z_t = \sigma(W_{xz} \times X_t^{'} + \sum_{l=1}^{L} W_{hz}^l) \times warp(H_{t-1}, U_{t,l}, V_{t,l}) \tag{4.3}$$

$$R_t = \sigma(W_{xr} \times X_t^{'} + \sum_{l=1}^{L} W_{hr}^l) \times warp(H_{t-1}, U_{t,l}, V_{t,l}) \tag{4.4}$$

$$H_t^{'} = f(W_{xh} \times X_t^{'} + R_t) \circ (\sum_{l=1}^{L} W_{hh}^l \times warp(H_{t-1}, U_{t,l}, V_{t,l})) \tag{4.5}$$

$$H_t = (1 - Z_t \circ H_t^{'} + Z_t) \circ H_{t-1} \tag{4.6}$$

Here, L is the total number of allowed links. $U_{t,}V_t \in \mathbb{R}^{L \times H \times W}$ are the flow fields that store the local connection structure generated by the structure generating network γ. And $W_{hz}^l, W_{hr}^l, W_{hh}^l$ are the weights for projecting the channels, which are implemented by 1×1 convolutions. The $warp(H_{t-1}, U_{t,l}, V_{t,l})$ function selects the position pointed out by $U_{t,l}, V_{t,l}$ from H_{t-1} via the bilinear sampling kernel. If we denote

$M = warp(I,U,V)$ where $M, I \in C \times H \times W$ and $\mathrm{U}, \mathrm{V} \in H \times W$, we have:

$$M_{c,i,j} = \sum_{m=1}^{H} \sum_{N=1}^{W} I_{c,m,n} \max(0, 1 - \left| i + \mathrm{V}_{i,j} - m \right|) max(0, 1 - \left| j + \mathrm{U}_{i,j} - n \right|) \quad (4.7)$$

The advantage of this framework is the ability to learn features through image sequences. However, radar images, as a complex data source, have features that cause problems.

4.2.2 GCN Structure

Radar imagery is a essential tool in meteorology, providing detailed snapshots of cloud structures and atmospheric conditions. These images are derived from the echo signals reflected by clouds, which are indicative of various meteorological phenomena. Among these, cumulonimbus clouds are of particular interest due to their association with heavy rainfall and thunderstorms. The formation and behavior of cumulonimbus clouds are influenced by a multitude of factors, including humidity, wind patterns, temperature variations, and the underlying topography. These factors intertwine to create a chaotic system, characterized by complex and often unpredictable dynamics.

In such a system, traditional linear models struggle to capture the complex relationships and interactions between different variables. To address this, nonlinear mapping components are essential for modeling the complex, nonlinear relationships inherent in meteorological data.

These components enable the network to learn and represent complex patterns and dependencies that are not linearly separable.

Graph Convolutional Networks (GCNs) have emerged as a powerful solution for dealing with such irregular data. GCNs are skilled at handling data represented as graphs, where the nodes correspond to entities (in this case, meteorological features) and the edges represent the relationships between these entities. By applying convolution operations on the graph structure, GCNs can learn to aggregate

information from local neighborhoods, capturing the spatial and contextual relationships within the data.

In this work, we have integrated GCN layers into our proposed method, GraphAT-NET, to harness their ability to learn from the complex interplay of features associated with cumulonimbus clouds. The GCN layers enable the network to model the nonlinear interactions between humidity, wind, temperature, and topography, which are critical for understanding the development and movement of these clouds. By embedding GCN within our architecture, we aim to enhance the network's ability to predict the formation and behavior of cumulonimbus clouds with greater accuracy.

The GCN layers in our approach are crafted to acquire an enriched representation of meteorological features by discerning the nuanced relationships that may elude detection through linear analysis. This is crucial for tasks such as precipitation forecasting, where understanding the complex interactions between different atmospheric variables is key to generating reliable predictions.

The structure diagram is presented in the II part of Fig 4.1. We introduce the GCN components in the following paragraphs:

First, we define the transition pattern of GCN as presented in Equation (4.8, 4.9):

$$H^{(l+1)} = ReLU(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4.8}$$

$$\text{Here, } \hat{A} = A + I \tag{4.9}$$

A is the adjacency matrix,

I is the identity matrix,

$H^{(l)}$ is the graph-level outputs, $H^{(0)}$ is the input X,

$\hat{D}$ is the diagonal node degree matrix of $\hat{A}$, and

$ReLU(\cdot)$ is the ReLU activation function.

In this model, the GCN is embedded after the first convolutional RNN layer, and the features are reorganized, which balances the computational complexity and the effectiveness of the GCN. This part of the component is composed of two layers of GCNs, and the correlation between GCN and TrajGRU can be concluded as follows:

$$H_1 = GCN\left(h_1\right) \tag{4.10}$$

Here, theinput of the GCN layer is the state tensor of the first convolutional RNN layer.

To initialize the relations within features, we create a Gaussian distribution matrix as the adjacent matrix *A*. Then, we use the initialized weight matrix and bias matrix to transport and learn features during training. After the inner operations within the first layer of GCN, we use the ReLU activation function to enhance the nonlinear mapping capability of GCN. To avoid overfitting during training, we use dropout to randomly disable 50 % of the neurons. After that, we use another GCN layer to compose the bottleneck structure. Finally, we have the enhanced stage tensor H_1.

4.2.3 ECA Attention Structure

Efficient Channel Attention (ECA) is a mechanism designed to enhance the feature representation capabilities of convolutional neural networks (CNNs) by dynamically adjusting the importance of different channels within the network. Introduced by Wang et al. in 2020 [66], ECA is a lightweight and parameter-efficient approach that allows for better discrimination among features, thereby improving the overall performance of the network.

The core idea behind ECA is to apply a channel-wise attention mechanism that is both computationally efficient and easy to integrate into existing CNN architectures. Unlike traditional attention

mechanisms that may involve complex operations and a significant number of additional parameters, ECA uses a simple and effective approach based on the statistical properties of the channels.

In practice, ECA computes a global average pooling to generate a set of statistics that describe the distribution of activations across all channels. These statistics are then used to model the importance of each channel. Specifically, ECA employs a sigmoid function to convert the importance scores into a probability distribution, which is used to weight the channels. This mechanism effectively enables the network to concentrate on the most salient features while diminishing the impact of less informative ones.

The efficiency of ECA comes from its use of a single, shared set of parameters for all channels, reducing the computational overhead and the number of trainable parameters compared to other attention mechanisms. This makes ECA particularly attractive for applications where computational resources are limited or where the network architecture is already complex.

Moreover, ECA has been shown to be effective in various tasks, including image classification, object detection, and semantic segmentation. By permitting the network to dynamically concentrate on the most discriminative features, ECA can result in enhanced accuracy and robustness. The simplicity and effectiveness of ECA make it a valuable addition to the toolkit of techniques available for improving CNNs.

To further improve the convolutional layers in the encoder and decoder of the proposed method, we embed a lightweight attention layer after each convolutional layer. To reach the balance between performance and efficiency, we adopt efficient channel attention (ECA) attention in this method. The structure diagram is presented in the III part of Figure 4.1 , and the details of efficient channel attention (ECA) are as follows:

First, we use adaptive average pooling to generate the channel-wise weight of the feature maps:

$$\mathbf{w} = Adaavgpool(X_t) \tag{4.11}$$

After that, we use two layers of 1D convolutional layers to enhance the relationships of channel weight:

$$\mathbf{w}' = Conv1D(\mathbf{w}) \tag{4.12}$$

Then, we use sigmoid activation function to enhance the nonlinear mapping ability of $\mathbf{w}'$:

$$\mathbf{w}'' = \sigma(\mathbf{w}') \tag{4.13}$$

Finally, we use the inner product of $\mathbf{w}''$ and input feature maps as the enhanced feature:

$$X_t' = (\mathbf{w}'' \times X_t) \tag{4.14}$$

4.3 Experiment Settings

This section describes the experiment settings of this work, which evaluate the effectiveness of the proposed method using two datasets.

The moving MNIST dataset is a benchmark for testing and evaluating prediction models. The second dataset is real-world time sequence data used to assess the prediction ability of the proposed method. These datasets are crucial for evaluating the proposed method and its potential applications.

4.3.1 Dataset Information

Moving MNIST [67] is a handwriting digit dataset based on the MNIST dataset. It consists of 10,000 sequences, each containing 20 frames with a size of 64×64 pixels, where digits move inside each patch. The dataset is commonly used as a benchmark for testing and evaluating video prediction models due to its complexity and diversity.

The moving MNIST dataset is generated by adding random motion blur

with random speeds and directions to the MNIST digits. Examples of the moving MNIST dataset are presented in Figure 4.2.

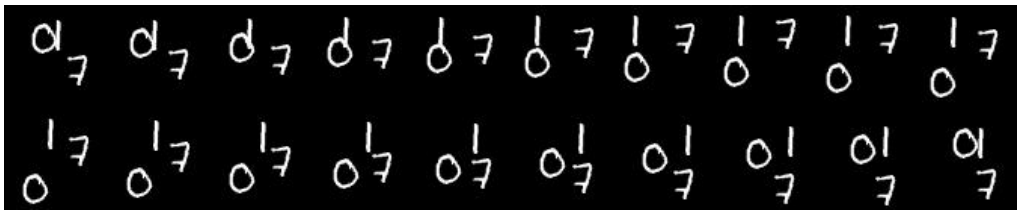


Figure 4.2: Examples of Moving MNIST.

The Guangxi constant altitude plan position indicating (GCAPPI) dataset is a high-time resolution record dataset of cumulonimbus cloud distribution in Guangxi province, China. The research area covers (102-114° E, 19-28° N). The experimental data consist of radar maps collected by 10 Doppler radars in Guangxi. The radar data are sampled and processed by the severe weather analysis and prediction system (SWAN) of the China Meteorological Administration to form a gridded reflectivity factor is surface mosaic, with a horizontal resolution of $0.01^{\circ} \times 0.01^{\circ}$ and an altitude spectrum ranging from 0.5 to 10.5 km.

In order to avoid ground interference and improve the reliability of data [62], the quality control algorithm was applied to remove isolated noise and ground echoes [68]. Specifically, the algorithm identified and removed echoes with low reflectivity values and those that were not contiguous with other echoes [69]. This step helped to reduce the impact of non-meteorological echoes on the analysis.

We selected the radar maps from June 2019, with a time resolution of 6 min. The original radar echo data are stored in bin format as echo data with amplitudes ranging from -128 to 127. To better form image information, we rescale the amplitudes to the range of 0 to 255. In addition, to avoid noise and abnormal values affecting the feature extraction process, we use the Daubechies8 wavelet for filtering. This results in a radar echo image with a size of 1200×900.

The GCAPPI dataset contains a total of 7200 frames, which were separated into a 6:4 training set and testing set, i.e., the GCAPPI dataset

includes a training set of 4320 frames and a validation set of 2880 frames. To improve the operation speed, we resized the radar map size to 256×256 . Examples of the GCAPPI dataset are presented in Figure 4.3.

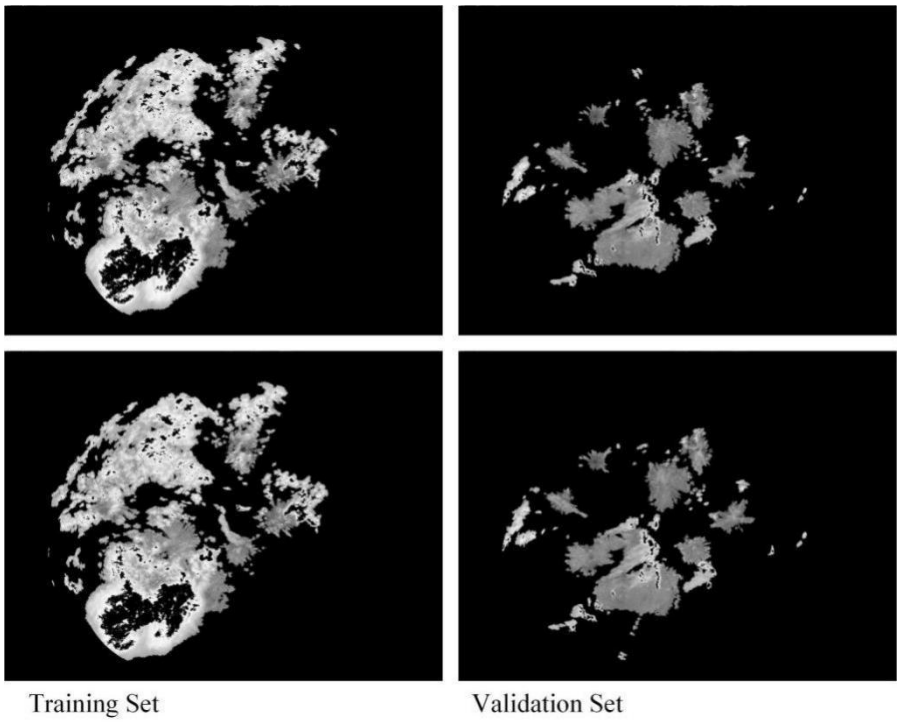


Figure 4.3: Examples of GCAPPI dataset (the brightness in figure represents the radar echo signal, which represents the thickness of the cloud).

4.3.2 Evaluation Metrics

MSE (mean square error) is the root of the deviation between the observed value and the ground truth value divided by the number of observations and is used to measure the deviation [70]. The standardized mean-variance is based on calculating the ratio of the accuracy between the model to be evaluated and the model based on the mean. The value range of the standardized mean-variance is usually 0 to 1. The smaller the ratio, the better the model is than the mean-based prediction strategy. The standard error is very sensitive to very large or very small errors in a group of measurements, so the standard error can reflect the precision of the measurement well. Therefore, this paper adopts MSE as the

evaluation method to evaluate the performance of each model.

$$MSE(x,y)=\frac{1}{n}\sum_{i=1}^{n}(x-y)^2 \tag{4.15}$$

SSIM (structure similarity index measure) [71], is used to evaluate the similarity between the target image and the generated image. SSIM mainly concerns three indicators: luminance, contrast, and structure.

First, luminance calculates the similarity between two patches; the closer two patches are, the larger luminance is. The definition of luminance is:

$$Luminance(x,y)=\frac{2\mu_x\mu_y}{\mu_x^2+\mu_y^2} \tag{4.16}$$

Here, μ indicates the average value of corresponding patch.

Second, contrast measures the distance between the texture of two patches; the definition of contrast is:

$$Contrast(x,y)=\frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2} \tag{4.17}$$

Here, σ means the variance of corresponding patch.

Third, structure is the correlation between the pixel values in two patches. The more edges with the same position and direction two patches contain, the higher score is. The definition of structure is:

$$Structure(x,y)=\frac{\sigma_{xy}}{\sigma_x\sigma_y} \tag{4.18}$$

Finally, when we add weight(ω) among three indicators, we have:

$$SSIM(x,y)=\omega_L Luminance(x,y)\times\omega_C Contrast(x,y)\times\omega_S Structure(x,y) \tag{4.19}$$

Obviously, the larger SSIM is, the better the model predicts.

4.3.3 Training Details

The experiments were conducted on a hardware platform that contains an Intel i5-9400f CPU, 24 GB RAM, and GTX 1080Ti GPU. All of the code was programmed and executed in PyTorch 1.8.

As introduced above, two datasets are adopted in this work. And the size of moving MNIST and GCAPPI data varies from each other. Thus, to achieve a balance between performance and hard platform, we use different training settings:

Moving MNIST: In the moving MNIST experiment, the batch size is 8, the learning rate is 1×10^{-4}, the input is 10 frames, and the predict the next 10 frames.

The total epoch is 100. We use Adam as the optimizer. We also use a learning rate scheduler to change the learning rate. The dynamic learning rate adjustment strategy we adopted in this experiment is ReduceLROnPlateau. As for training loss, we combine MSE loss with SSIM loss; the loss function is presented in Equation (4.16). In Equation (4.16), $\hat{I}_{t+k}$ represents the ground truth and I_{t+k} represents the sequence image predicted by the model. The loss function is mainly composed of the mean squared error (MSE) loss function and the structural similarity index (SSIM) loss function, which are added together and then divided by 2. This constrains the maximum value of the loss and improves training stability. To prevent overfitting, we adopted early stopping in the experiments, i.e., if the loss value does not decrease in 5 epochs, the training procedure will stop.

$$Loss = \ (MSE(\hat{I}_{t+k}, I_{t+k}) + (1 - SSIM(\hat{I}_{t+k}, I_{t+k}))/2)/2 \qquad (4.20)$$

GCAPPI dataset: In the GCAPPI experiment, we basically follow the same set of moving MNIST datasets. However, considering the size of GCAPPI dataset and the limitation of the hardware platform, we adjust the batch size to 1, and the amount of input frames and prediction frames is 4. We compare the performance of methods after 40 epochs of

training.

4.4 Performance

This section is designed to illustrate the advantages of the proposed method by contrasting it with several leading approaches in the field. The comparison is conducted with respect to the following methods: LSTM with fully connected layers (FC-LSTM), GCNnet, PSPNet, Seresunet, Smatunet, ConvLstm, and ConvGRU. These methods have been widely used in previous studies and are considered benchmarks for evaluating prediction models.

4.4.1 Performance of Methods on Moving-MNIST Dataset

We present the experiment results of the moving MNIST dataset in Table 4.1, sorted by MSE values. To more specifically present the advantage of the proposed method, we calculated the increased percentage (*ip*) of corresponding indicators (with $ip = |a - b|/b$). For MSE, the increase percentages are ConvGRU: 11.51 %; ConvLSTM: 17.45 %; Smatunet: 65.64 %; Seresunet: 72.30 %; FC-LSTM: 91.05 %; PSPNet: 82.43 %; GCNNet: 83.45 %. As for SSIM, the increase percentages are: ConvGRU: 3.47 %; ConvLSTM: 2.14 %; Smatunet: 15.84 %; Seresunet: 15.93 %; FC-LSTM: 25.88 %; PSPNet: 27.31 %; and GCNNet: 25.89 %.

Table 4.1: Comparison of the proposed model with other models based on the moving MNIST dataset (Vali Loss is short for validation loss).

Methods	MSE	SSIM	Vali Loss
GCNNet	7.43E-03	8.33E-04	4.83E-01
PSPNet	7.00E-03	8.17E-04	4.83E-01
FC-LSTM	6.49E-03	8.33E-04	4.82E-01
Seresunet	4.44E-03	9.45E-04	4.79E-01
Smatunet	3.58E-03	9.46E-04	4.78E-01
ConvLSTM	1.49E-03	1.10E-03	4.73E-01
ConvGRU	4.14E-02	1.09E-03	4.74E-01
GraphAT-Net	1.23E-03	1.12E-03	4.72E-01

In addition, we present the performance of different models on the

moving MNIST dataset in Figure 4.4 and Figure 4.5, respectively. Limited by the ability of different models, PSPNet, GCN, and FC-LSTM triggered the early stopping mechanism. Evidently, GraphAT-NET outperforms the other models in both the mean squared error (MSE) and structural similarity index (SSIM) metrics.

Firstly, GCN performs the worst, possibly because it cannot extract spatio-temporal feature information from sequence data alone. Additionally, fully CNN-based models such as PSPNet, Seresunet, and Smatunet, which use CNN to construct spatio-temporal feature correlation information, still do not achieve as high prediction accuracy as models that combine CNN and RNN.

Furthermore, as a typical machine learning method, FC-LSTM is not sufficient to learn feature representations from the moving MNIST dataset. Comparing the models, Convlstm, Convgru, and the proposed model show significantly improved accuracy compared to other models. This may be because CNN can extract feature information from images, and RNN can build spatio-temporal feature information based on the correlation between image features. Additionally, Table 4.1 shows that the proposed model achieves the highest accuracy.

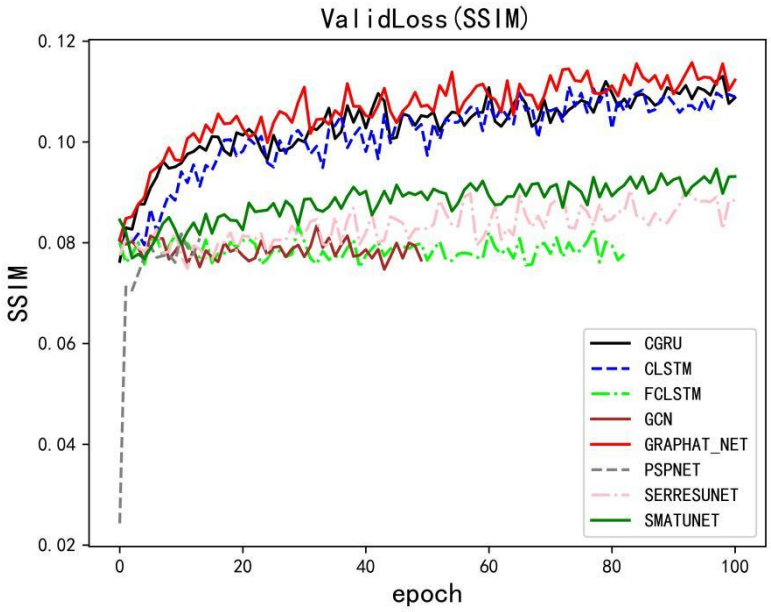


Figure 4.4: The change of MSE loss value for different methods.

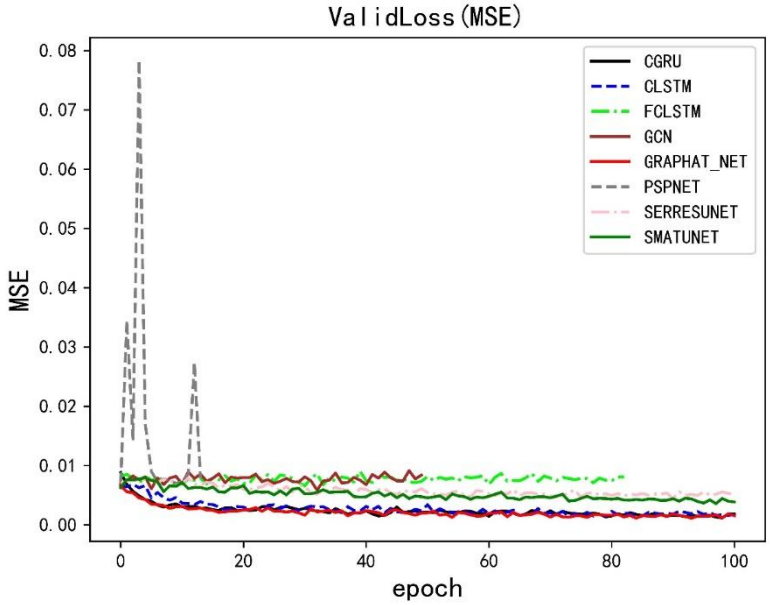


Figure 4.5: The change of SSIM loss value for different methods.

To better demonstrate the advantages of the proposed model, we display the visualization results in Figure 4.4, following the order of Table 4.1. It is evident that GCN can only achieve fuzzy predictions of the region, which is consistent with the previous speculation that an image-sensitive CNN structure is necessary to extract image details.

Notably, FC-LSTM learned nothing but the background and failed to make any predictions in the moving MNIST dataset. Furthermore, in the visualization results of PSPNet, Seresunet, and Smatunit in Figure 4.6, the short-term prediction results are better and clearer, while the long-term distribution becomes fuzzy and more disturbed. This may be because it is difficult for the model to build the spatio-temporal feature association relationship, making it challenging for the model built only by CNN to make long-term predictions.

Compared with ConvLSTM, ConvGRU, and the proposed model results, it is apparent that the architecture built by CNN and RNN can better extract spatio-temporal feature information and make more accurate predictions. However, the prediction results of ConvLSTM and ConvGRU still contain a lot of interference, which affects the prediction results. Additionally, it can be observed that the prediction accuracy of

all models decreases with time. Nevertheless, it is evident from the visualization results that the proposed model achieves the best prediction accuracy.

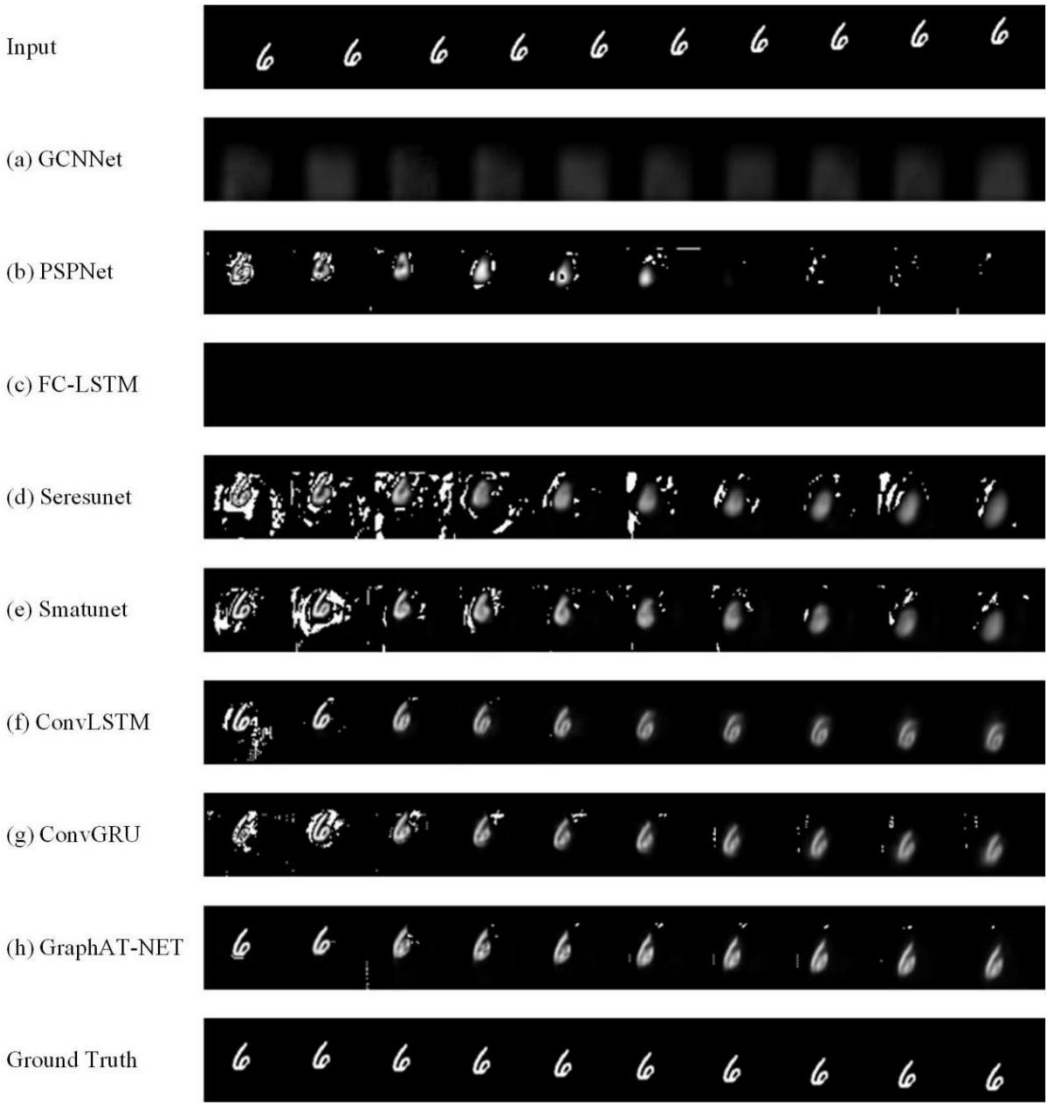


Figure 4.6: Performance of methods on moving MNIST dataset.

4.4.2 Performance of Methods on GCAPPI Dataset

Table 4.2 displays the performance of various methods on the GCAPPI dataset, arranged based on the MSE value. As indicated in Table 4.2, the proposed method surpasses the others in both MSE and SSIM. For MSE values, the proposed method has *ip* over other methods with: ConvLSTM: 10.92 %; ConvGRU: 24.72 %; FC-LSTM: 76.01 %;

GCNNet: 76.23 %; Seresunet: 83.59 %; Smatunet: 87.18 %; and PSPNet: 99.94 %. As for the results of SSIM, the proposed method outperforms other methods with: ConvLSTM: 4.92 %; ConvGRU: 0.08 %; FC-LSTM: 0.45 %; GCNNet: 0.57 %; Seresunet: 0.50 %; Smatunet: 0.99 %; and PSPNet: 64.51 %.

In the experiment on real-world data, pure CNN methods such as PSPNet, Smatunet, and Seresunet perform worse than methods based on GCN and RNN. This is because of their lack of spatio-temporal feature extraction ability. The results of other methods in Table 4.2 also verify that the combination of CNN, RNN, and GCN can help methods to extract the distributions of cumulonimbus clouds along the time dimension. Moreover, the proposed method performs the best on real-world data.

We demonstrate the performance of the compared algorithms on the CAPPI dataset over time in Figures 4.7 and 4.8. Due to hardware limitations, the training epoch of CAPPI is 40. Early stopping is not used in the CAPPI dataset experiment. It can be observed that PSPNet is ineffective in training on the CAPPI dataset. To better illustrate the training details, we excluded the loss curves of PSPNet, CLSTM, and CGRU and displayed the detailed plots on the right side of Figures 4.7 and 4.8. It can be seen that GCN, FC-LSTM, SERESUNET, SMATUNET, and GraphAT-NET perform similarly at the end of training, making it difficult to compare them intuitively based on loss values.

Table 4.2: Comparison of the proposed model with other models based on the GCAPPI dataset (Vali Loss is short for validation loss).

Methods	MSE	SSIM	Vali Loss
PSPNet	3.16E-04	3.55E-01	4.99E-01
Smatunet	1.54E-06	9.89E-01	4.98E-01
Seresunet	1.21E-06	9.94E-01	4.98E-01
FC-LSTM	8.24E-07	9.94E-04	4.98E-01
GCNNet	8.32E-07	9.93E-01	4.98E-01
ConvGRU	2.63E-07	9.98E-01	4.98E-01
ConvLSTM	2.20E-07	9.50E-01	4.98E-01

GraphAT-Net 1.98E-07 9.99E-01 4.98E-01

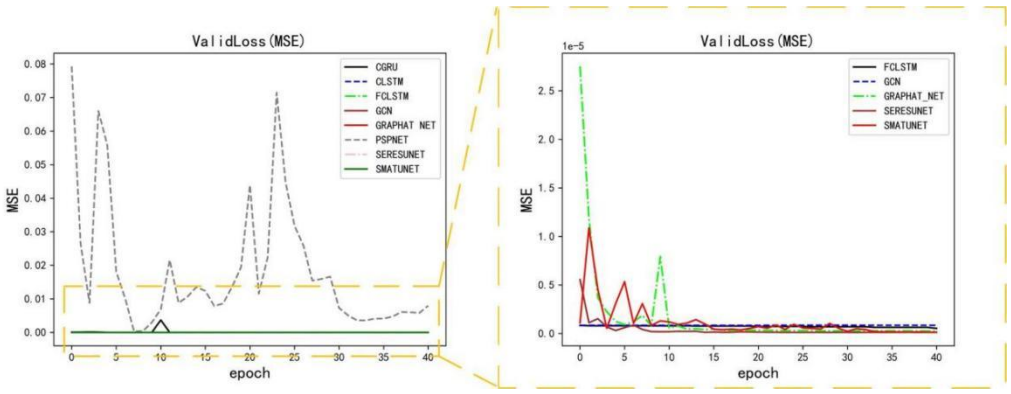


Figure 4.7: The change of MSE loss value for different methods.

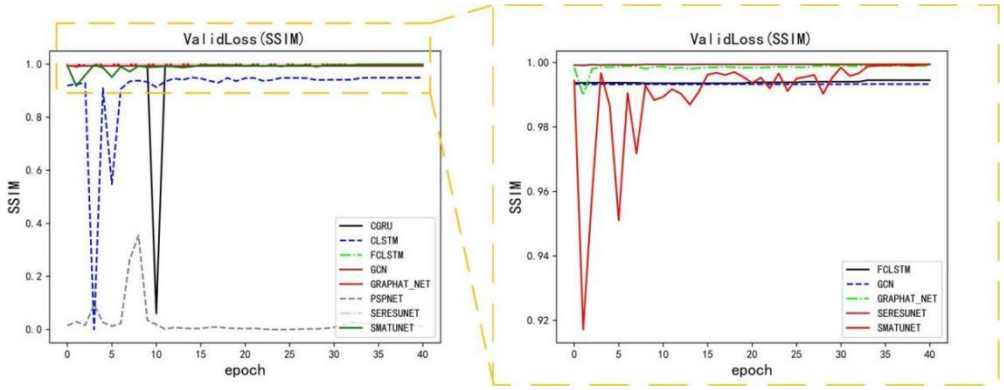


Figure 4.8: The change of SSIM loss value for different methods.

We present the visual results of corresponding methods in Figure 4.9. We analyze the results of Figure 4.9 row by row.

Firstly, the visual results of PSPNet show the worst performance on both indicators and visual results, making them unacceptable. Secondly, the visual results of Smatunet and Seresunet learn the prediction task as a segmentation task. This is evident from the minimal differences in contour and details, indicating that these models only learn spatial distribution rather than temporal correlations. Thirdly, FC-LSTM can make basic predictions of the distribution of cumulonimbus clouds, but there are still many noise interferences in the results. Fourthly, the unsatisfactory performance of pure GCN is due to its inability to study

spatial distributions. Fifthly, comparing the results of ConvGRU and ConvLSTM, both methods have predicted vague spatio-temporal distributions. Lastly, the proposed method generates the most accurate predictions, with clearer details than those of ConvGRU and ConvLSTM results.

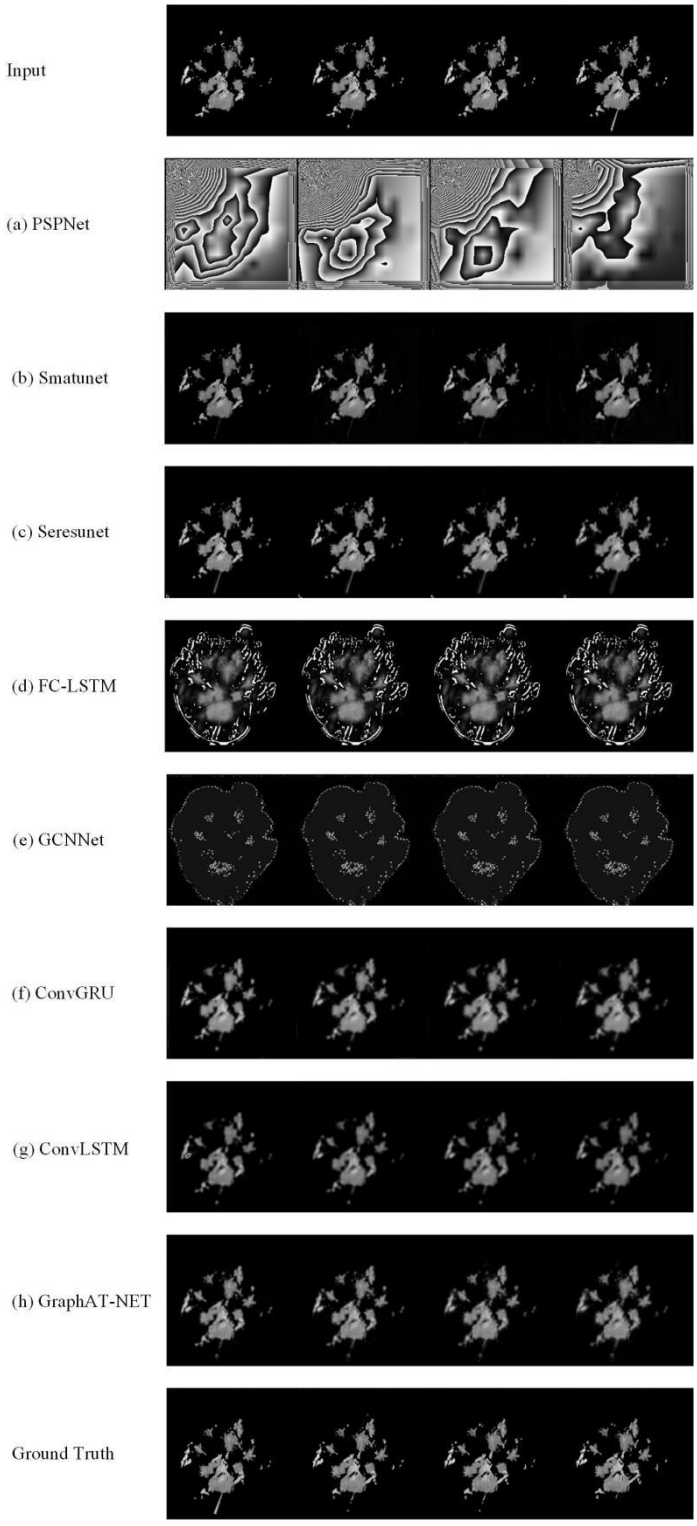


Figure 4.9: Performance of methods on GCAPPI dataset.

4.5 Ablation Study

In this section, we analyze the effectiveness of the modules in the proposed method. We conduct experiments on the GCAPPI dataset to discuss the validity of the corresponding modules. The details are presented in the following sections.

4.5.1 Effectiveness of GCN

To validate the effectiveness of the GCN module, we conducted experiments among GCNNet, ConvGRU, ConvGRU + GCNNet, and GraphAT-Net. The purpose of this experiment is to evaluate the contribution of the GCN module to the performance of GraphAT-Net.

As shown in Table 4.3, ConvGRU with GCN increased about 10.9 % on MSE and showed the same performance on SSIM. This indicates that GCNNet could enhance the model to learn more accurate features. Specifically, the GCN module can capture the spatial dependencies among the input data and pass the information to the subsequent layers, which leads to more accurate predictions.

To better illustrate the enhancement, we present visual results in Figure 4.10. As shown in Figure 4.10, the results of pure GCN are still unreadable, while ConvGRU + GCN performs much better than ConvGRU. This verifies the effectiveness of GCN as an embedded module in the CNN+RNN-based framework. Specifically, the GCN module can effectively capture the spatial correlations in the input data and enhance the feature representation of the model, which leads to more accurate predictions.

The experimental results demonstrate that the GCN module can effectively enhance the feature representation of the model and improve the accuracy of the predictions. Therefore, the GCN module is a valuable addition to the proposed method and can be used to improve

the performance of other CNN + RNN-based models.

Table 4.3: Performance of methods in ablation study on GCN module (Vali Loss is short for validation loss).

Methods	MSE	SSIM	Vali Loss
GCNNet	8.32E-07	9.93E-01	4.98E-01
ConvGRU	2.63E-07	9.98E-01	4.98E-01
ConvGRU + GCN	2.34E-07	9.98E-01	4.98E-01
GraphAT-Net	1.98E-07	9.99E-01	4.98E-01

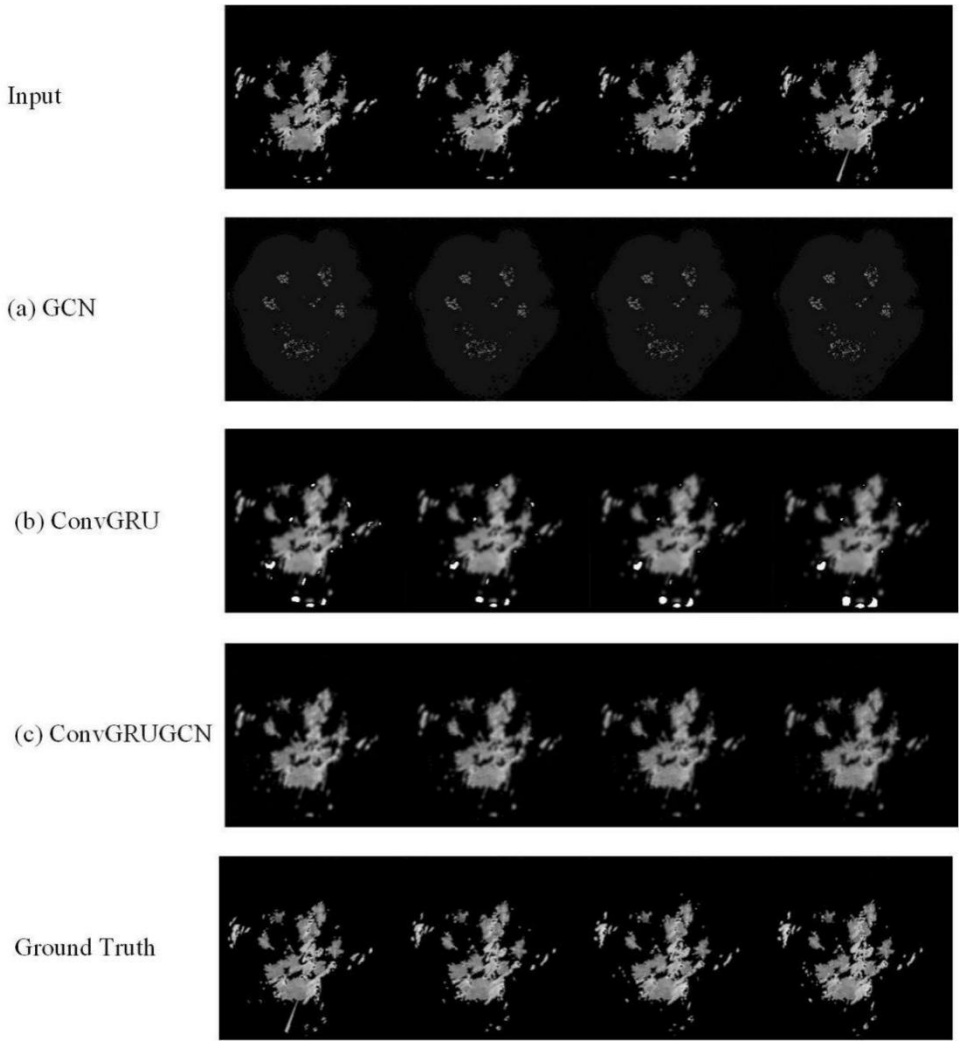


Figure 4.10: Visual results of ablation study on GCN module.

4.5.2 Effectiveness of ECA

In this subsection, we discuss the effectiveness of the ECA module. We present the experimental results in Table 4.4 and the visual results in Figure 4.11. The purpose of this experiment is to evaluate the contribution of the ECA module to the performance of the proposed method.

Table 4.4: Performance of methods in ablation study on ECA module (Vali Loss is short for validation loss).

Methods	MSE	SSIM	Vali Loss
ConvGRU	2.63E-07	9.98E-01	4.98E-01
ConvGRU + ECA	2.12E-07	9.98E-01	4.98E-01
GraphAT-Net	1.98E-07	9.99E-01	4.98E-01

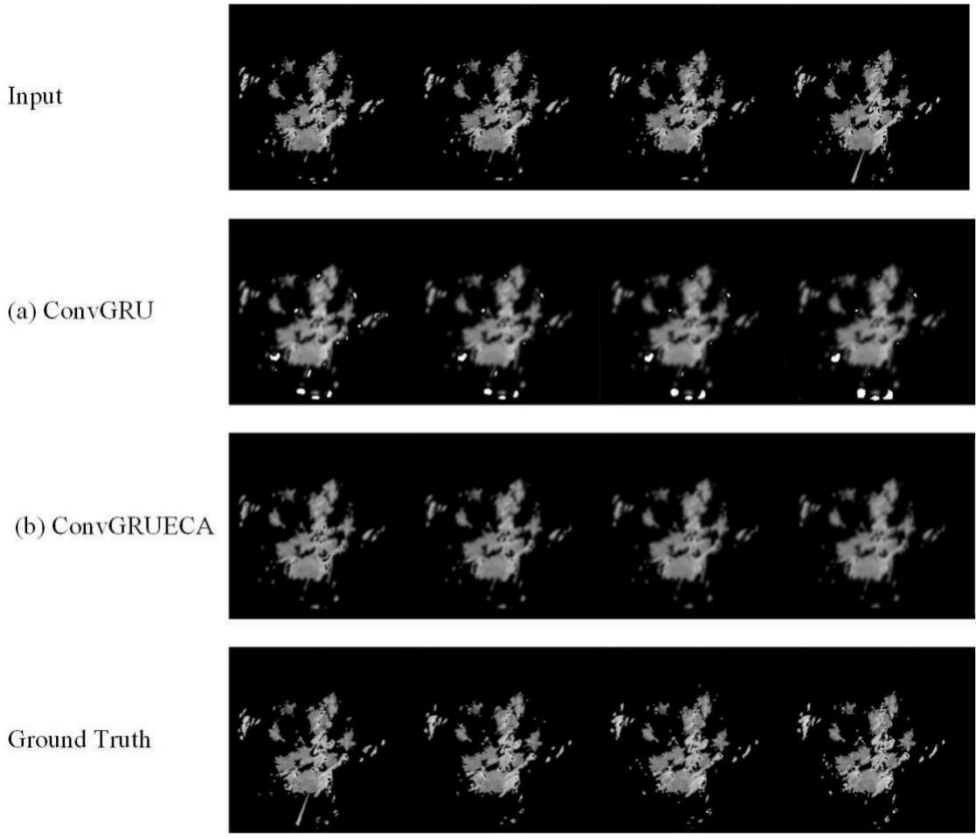


Figure 4.11: Visual results of ablation study on ECA module.

Table 4.4 shows that the ECA module helps ConvGRU improve by about 19.28 % on MSE. This indicates that the ECA module can effectively enhance the feature representation of the model. Specifically, the ECA module can selectively emphasize informative features and suppress irrelevant ones, which leads to more accurate predictions.

To better illustrate the effectiveness of the ECA module, we present visual results in Figure 4.11. As shown in Figure 4.11, the details of the ConvGRU+ECA results are more accurate than those of the ConvGRU results. In particular, the ConvGRU+ECA results have clearer boundaries and more accurate shapes, which indicates that the ECA module can effectively capture the spatio-temporal correlations in the input data.

In summary, the experimental results demonstrate that the ECA module can effectively enhance the feature representation of the model and improve the accuracy of the predictions. Therefore, the ECA module is a valuable addition to the proposed method and can be used to improve the performance of other CNN+RNN-based models.

4.6 Conclusions

The proposed method, GraphAT-NET, represents a significant advancement in the field of meteorological data analysis, displaying its strength through the integration of multiple deep learning technologies. This model stands out due to its synergistic combination of Convolutional Neural Networks (CNNs), Graph Convolutional Networks (GCNs), Recurrent Neural Networks (RNNs), an advanced attention mechanism, and a refined loss function. The ensemble of these components has proven to be particularly effective, as evidenced by its performance on both the moving MNIST dataset and real-world radar echo data.

In the moving MNIST experiment, GraphAT-NET's capabilities were put to the test against other state-of-the-art methods. The results were remarkable, with a significant reduction of 59.12% in the Mean Squared

Error (MSE), a standard measure of prediction accuracy. Moreover, the model demonstrated a 16.26% improvement in the Structural Similarity Index (SSIM), which assesses the similarity between two images and is crucial for visual data. These improvements highlight the model's ability to capture the nuances of sequential data, a critical factor in meteorological predictions.

The effectiveness of the proposed model was further substantiated through visualization results on the moving MNIST dataset. The visual outputs provided a clear and compelling demonstration of the model's superior performance, displaying its ability to accurately predict and track the movement of objects, similar to predicting the evolution of weather patterns.

In the GCAPPI experiment, which involved real-world radar echo data, GraphAT-NET continued to excel. It demonstrated a substantial average enhancement of 65.40% in MSE, a testament to its robustness when applied to practical meteorological data. Furthermore, the model outperformed other models in the SSIM index assessment, showing a noteworthy improvement of 10.29%. This outcome highlights the model's competence in preserving the structural integrity of the data, a critical factor for accurate and reliable predictions.

The visualization results from the GCAPPI experiment were particularly telling, as they demonstrated that the proposed method surpassed other methods in terms of prediction accuracy and the level of detail in the distribution. The ability to predict with high precision and detail is critical in meteorology, where the smallest variations can have significant implications.

To delve deeper into the contributions of different modules within GraphAT-NET, an ablation study was conducted. This study methodically removed individual components to assess their individual impacts. The findings were insightful, revealing that the GCN module, which is responsible for learning complex feature relationships, resulted in an average 10.09% improvement in MSE. This highlights the importance of capturing the complex interactions within meteorological

data.

Additionally, the ECA module, which is tasked with enhancing the accuracy of the model, showed a 19.28% improvement in MSE. This substantial improvement underscores the value of the ECA module in refining the model's focus on the most relevant features for prediction.

Based on these comprehensive results, we are confident in the potential of the proposed GraphAT-NET method to enhance the prediction of cumulonimbus cloud distribution. The prediction of these clouds is a critical component in the broader process of rainfall nowcasting, which involves predicting the distribution and intensity of rainfall in the immediate future.

However, it is important to recognize that predicting cumulonimbus clouds is merely the first step in this process. For comprehensive rainfall predictions, it is necessary to link radar echo data, which provides information about cloud structures, with actual rainfall data. Our future work will be dedicated to establishing these connections, aiming to develop an end-to-end prediction system that can provide more complete and accurate forecasts.

In summary, the proposed GraphAT-NET method, through its integration of advanced deep learning technologies, has demonstrated significant potential in the field of meteorological prediction. The method's performance on both synthetic and real-world data, along with the insights gained from the ablation study, positions it as a powerful tool for improving the prediction of cumulonimbus cloud distribution and, by extension, rainfall forecasting.

CHAPTER 5

PRECIPITATION FORECASTING BASED ON INFORMER ON DATA OF GROUND STATIONS IN GUANGXI, CHINA

5.1 Introduction

Building upon the research findings from the preceding chapters, this study introduces cutting-edge technologies to explore a novel method for predicting cumulonimbus cloud distribution. As previously discussed, accurate weather pattern prediction is crucial for disaster mitigation and preparedness. Particularly in subtropical regions, the heavy rainfall from cumulonimbus clouds can lead to devastating flash floods and mudslides. Traditional machine learning methods have shown limitations in accurately forecasting cloud distribution. Therefore, this paper proposes an innovative approach that integrates graph convolutional networks (GCN) and trajectory gated recurrent units (TrajGRU) with an attention mechanism to enhance predictive accuracy using radar echo data. Analysis of experimental results demonstrates successful application of this method on simulated and real-world datasets, underscoring its potential to improve prediction accuracy significantly and emphasizing its importance in safeguarding communities from weather-related disasters.

Time-Series Forecasting (TSF) tasks are still used in many industrial scenarios, such as transaction volume forecasting for banks, electricity consumption forecasting for power systems, access forecasting for cloud services, and regional rainfall forecasting. If rainfall can be predicted in advance for a period of time in the future, resources can be deployed in advance to prevent damage to economic and social infrastructure caused by heavy rainfall.

There are currently three types of deep network models applied to

convective weather forecasting: one is convolutional neural network, which takes the input grid weather elements as the form of images, and performs feature learning through image filters, fully considering the spatial structure. However, the disadvantage is that it lacks the ability to process sequence data and is only suitable for processing ”fixed-length” data [71]. The other is the recurrent neural network. This model, which is often used in natural language processing, can flexibly process sequence data through an autoregressive structure and perform effective learning in the time dimension. The shortcomings are also more obvious, and multi-layer perception Like the machine, the input features can only be represented by one dimensional vector, so the inherent spatial characteristics of the grid data are lost, and the learning ability is weak [72]. The last one is to combine the above two models in different forms, which can learn spatial and temporal features at the same time, which is more suitable for solving the problem of convective weather forecasting. oth the Convolutional Long Short-Term Memory (ConvLSTM) unit model and the Convolutional Gated Recurrent Unit (ConvGRU) model examined in this study fall under this category of networks [73].

However, the above methods do not consider integrating the temporal dimension with the high-level semantic information in the image, thereby helping the model to learn more accurate feature representations. Transformer is mainly used to solve the problem of training parallelism, and it also has a good effect on long-term dependencies. The disadvantage is that for long sequences, the amount of parameters will be too large due to self-attention between all vectors.

Transformer introduces more elements and structures of deep learning in structure (residual network, Norm, EncoderDecoder, feedforward network, multi-head self-attention, position encoding, Embedding), but the most important structure is multi-head self-attention , and the basic unit of multi-head selfattention is Self-Attention. Self-Attention realizes that each vector can pay attention to the information of other vectors by calculating the Attention Value between the input (matrix composed of multiple vectors). The multi-head self-attention is to perform Self-Attention on the vector from different subdimensions, and then use

the concat result as the output. There is currently no reasonable explanation for the purpose of the multi-head. Existing explanations are: 1. The multi-head acts as a filter to extract richer information from sub-dimensions; 2. The multi-head is equivalent to the integration of multiple basic learners, reducing overfitting.

The advent of the Transformer model addresses this bottleneck issue. Transformers have demonstrated exceptional performance in numerous tasks within natural language processing and computer vision, thereby garnering considerable attention in the time series community (Zhang et al., 2021) [74]. Zhang et al. (2022) and Naseem et al. (2020) utilize a Transformer encoder architecture to encapsulate the impact of historical events and to calculate the intensity function for event forecasting [75][76]. Subsequently, a more flexible attentional neural data log across time (a-NDTT) is proposed [77]. Some research learn multivariate time series for downstream tasks by reusing existing data samples (with or without other unlabeled data) [78]-[83] Recently, multiple studies have shown that the combination of GNN and Transformer/attention can not only significantly improve traffic prediction [84]and multimodal prediction [85], but also better understanding of spatiotemporal dynamics and underlying arbitrariness. Combining Transformers and global neural networks for efficient spatiotemporal modeling in time series is an important future direction [86] [87]. However, research on pre-trained Transformers for time series is limited, and existing research mainly focuses on time series classification [88].

According to Wang, et al. (2022), they proposed an upgraded version of Transformer for long-period forecasting. In order to improve the operation efficiency of Transformer in long-term estimation, ProbSparse self-attention is proposed, which reduces a large amount of computation by forming sparse attention between key values and key queries [89].

During the encoding phase, the encoder receives an extended sequence of inputs and acquires feature representations via the ProbSparse self-attention module and the self-attention distillation module. The Informer provides three types of embedding representations: Local Time

Stamp serves as a fixed positional embedding in the Transformer; Global Time Stamp constructs a table of information and represents each entry through an embedding feature; Scaler transforms the input sequence into a vector using one-dimensional convolution. In the decoding phase, the decoder receives a long sequence input (with the prediction target segment initialized to 0), interacts with the encoded features through multi-head attention, and ascertains the output target segment. This approach is more expeditious than the step-by-step method [90].

Zhou, et al. (2021) proposed that Transformer has several serious problems, such as: high quadratic time complexity, high memory usage, and inherent limitations of encoderdecoder architecture, which prevent it from being directly applied to long-series time series forecasting (LSTF) requirements [91]. Therefore, the authors design a probabilistic sparse self-attention mechanism and a well-designed generative decoder to alleviate the limitations of traditional transformer encoder-decoder structures. Li, et al. (2022) used the Informer encoder for tool wear prediction, a distillation layer is employed at the end of the basic block, eliminating redundant features [92]. In contrast to the Transformer, the Informer is capable of efficiently extracting global features. Qian et al. (2022) posited that data imputation can address missing observations in time series data. Leveraging adversarial network training and a self-attention mechanism, this approach can effectively tackle the multidimensional time series interpolation issue and enhance the precision of time series data interpolation models [93].

Subsequently, this study will delve deeper into the field of precipitation forecasting. With the transition from numerical weather prediction (NWP) and optical flow methods to deep learning techniques, precipitation forecasting has made notable progress, resulting in enhanced accuracy. This research will investigate the application of spatio-temporal sequence models in precipitation forecasting. By analyzing the structures of Transformer and Informer, and selecting relevant features using the Pearson correlation coefficient, the aim is to further enhance the accuracy of precipitation forecasting.

5.2 Data and methods

5.2.1 Methods

1)Description of Transformer

The Transformer model diverges from conventional CNN and RNN structures, opting instead for the utilization of the Attention mechanism to automatically capture relational dependencies at diverse positions within the input sequence. Each encoder consists of two foundational components: self-attention and a feed-forward network. The Attention mechanism serves to comprehensively capture the context by aggregating input elements with assigned weights. It processes the time series data, evaluates the importance of each feature, and identifies their interactions. Consequently, the feed-forward networks adjust their parameters in alignment with these determined weights.

Moreover, in addition to self-attention and the feed-forward network, the Decoder includes an encoder-decoder attention layer. In the field of rainfall prediction, each temporal node evaluates not only the anticipated rainfall but also considers the rainfall in the encoder both before and after the time series. Self-Attention can be segmented into multiple segments, giving rise to Multi-Head Attention. This multi-head self-attention process involves conducting Self-Attention on vectors from various sub-dimensions, and subsequently utilizing the concatenated results as the final output. The amalgamated outcomes from multiple attention mechanisms serve to fortify the model, facilitating the network's acquisition of more complex features and extensive information.

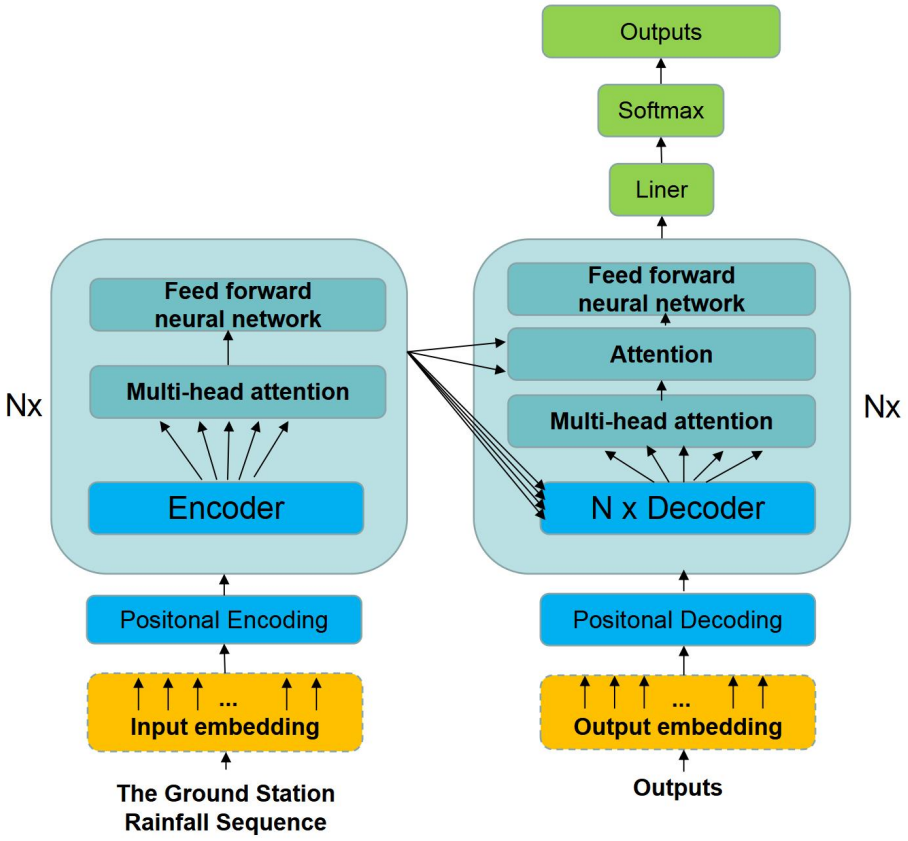


Figure 5.1: The Structure of Transformer.

The encoder and decoder consist of N=6 identical layers, each comprising two sub-layers: a multi-head self-attention mechanism and a fully connected feed-forward network. Each sub-layer includes a residual connection and normalization, allowing the output of the sub-layer to be expressed as:

$$sub\_layer\_output = LayerNorm(x + (SubLayer(x))) \quad (5.1)$$

The calculation of attention necessitates three specified inputs: Q (query), K (key), V (value), and the resultant attention calculation is obtained using the following formula.

$$Attention(Q,K,V) = sofi\max(\frac{QK^T}{\sqrt{d_k}})V \quad (5.2)$$

$$Q = W^Q X, K = W^K X, V = W^V X \quad (5.3)$$

The attention calculation result of the current time step is the

accumulation of a group coefficient multiplied by the feature vector value of each time step. This coefficient is obtained by the inner product of the query of the current time step and the keys corresponding to other time steps. This process is equivalent to querying the keys of other time steps with your own query. This is equivalent to judging the similarity, and deciding what proportion to inherit the information of the corresponding time step. The reason for dividing by dk is that when the dimensions of the two vectors are very large. Therefore, the variance of the point multiplication result will also be large, and the gradient of the softmax will approach 0, that is, the gradient will disappear. If the result of the dot product is scaled, that is, divided by dk, the variance can be effectively controlled from dk back to 1. In addition, the problem of gradient disappearance can be effectively controlled.

Multi-head attention projects Q, K, and V through h different linear transformations, and finally stitches together different attention results:

$$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \tag{5.4}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5.5}$$

Both the encoder and the decoder of the Transformer contain inputs, and the input structure of the two parts is the same, but the usage during inference is different. The output includes the probability distribution of the output word corresponding to position i. The input includes the output of the encoder and the output of the decoder corresponding to the i-1 position. So the middle attention is not self-attention, its K and V come from the encoder, and Q comes from the output of the previous position decoder. The decoding during training and prediction is not the same. During training, the decoding is decoded all at once, and the ground truth of the previous step is used to predict. When predicting, because there is no ground truth, you need to predict one by one.

2)Description of Informer

In order to improve the operation efficiency of Transformer in long-term estimation, ProbSparse self-attention is proposed, which reduces a large

amount of computation by forming sparse attention between key values and key queries.

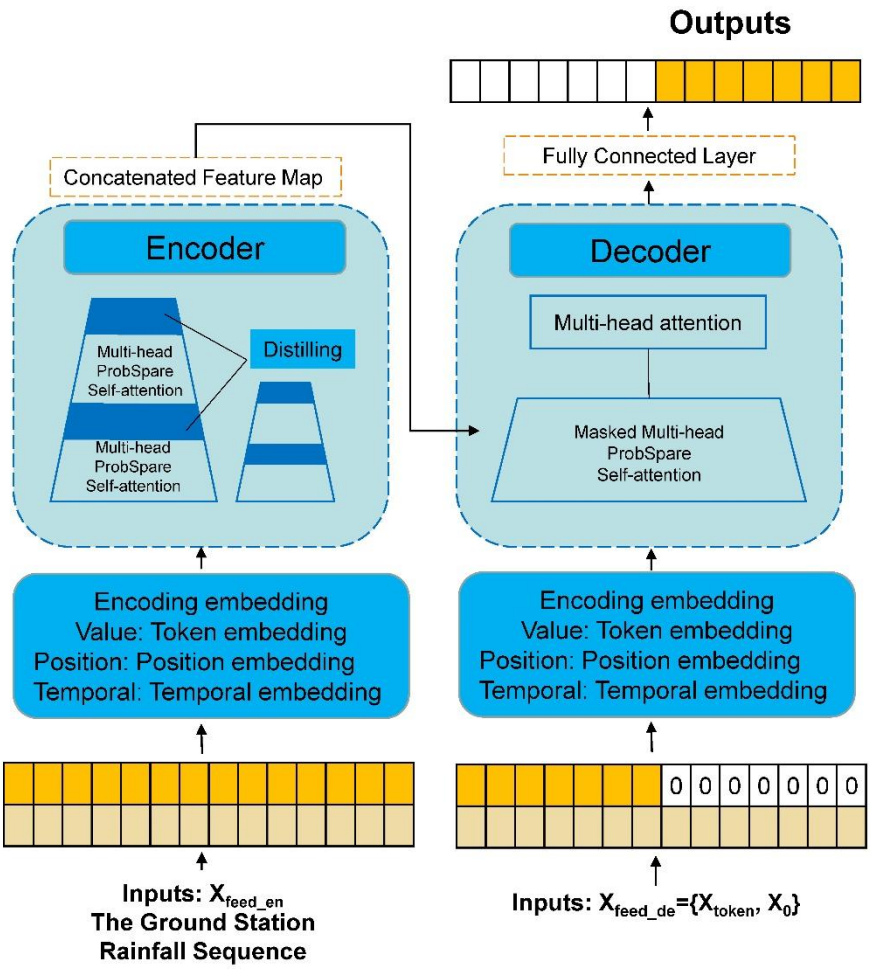


Figure 5.2: The Structure of Informer.

During the encoding process, the encoder receives a long sequence of inputs, and obtains feature representations through the ProbSparse self-attention module and self-attention distillation module. Three embedding representations are provided by the informer: Local Time Stamp is a fixed position embedding in the Transformer; Global Time Stamp builds a table of information and represents each piece through the embedding feature; the scaler transforms the input sequence into a vector via one-dimensional convolution. During the decoding phase, the decoder receives an extended sequence input (with the prediction target segment initialized to 0), engages with the encoded features through multi-head attention, and ascertains the output target segment. This

approach is more rapid than the step-by-step method and yields comparable results. The Informer introduces three enhancements based on the Transformer:

1. ProbSparse self-attention

The quadratic computational complexity of self-attention in Transformer results in $O(L^2)$ time complexity and memory usage per layer. ProbSparse self-attention replaces canonical self-attention, achieves $O(L \log L)$ in time complexity and memory usage, and has sequences dependency alignment.

ProbSparse self-attention sorts the queries according to the importance of each token, and selects only the largest top-u queries (a query vector actually corresponds to a point in a sequence).

$$A(Q,K,V) = \text{Softmax}(\frac{\overline{Q}K^T}{\sqrt{d}})V \qquad (5.6)$$

ProbSparse self-attention is implemented by allowing each key to focus only on the u dominant query vectors. In practice, the input lengths of Query and Key are usually equal in the sub-attention calculation, that is, $L_Q = L_K = L$, which makes both the time and space complexity of Prob Sparse self-attention $O(L \ln L)$.

2. Self-attention distillation mechanism.

3. The Self-Attention Distillation mechanism can reduce the input sequence length for each layer. With a shorter sequence length, the computational and storage requirements naturally decrease. Self-Attention Distillation halves the input to the subsequent layer through MaxPool with a stride of 2, which is akin to the output dimension of a transformer layer being half of the input dimension of that layer. This process highlights the dominant attention and effectively

manages extremely long input sequences. The blue section in the figure represents the Self-Attention distillation operation for feature compression.

4. Generative decoder mechanism

The Generative Decoder mechanism enables the acquisition of results in a single step rather than through a step-by-step process when predicting sequences (including during the inference phase), which directly reduces the prediction time complexity from $O(N)$ to $O(1)$. The generative style decoder also has input, and the output is shaped in one shot, rather than predicting long sequences in a "step-by-step" fashion. It allows processing of longer sequence inputs under memory usage constraints, which greatly improves inference speed for long sequence predictions.

5.2.2 Data

The study presented in this thesis is geographically focused on the Guangxi region, specifically within the coordinates of 104E-112E and 21N-27N. This area, known for its subtropical climate, presents unique meteorological patterns that are critical to analyze and understand. The primary objective of this research is to enhance the accuracy of precipitation forecasts, particularly on a half-hourly basis for the numerous stations within this region. The study's emphasis on short-term forecasts is driven by the need for timely and precise weather predictions that can aid in disaster preparedness and response.

The empirical data underpinning this study is derived from a comprehensive set of real-time observations collected from 2928 ground-level meteorological stations across Guangxi. These stations have been meticulously recording hourly meteorological data from January 2019 to October 2019, providing a rich dataset that spans over nine months. The temporal resolution of the data, with hourly readings, is crucial for capturing the nuances of daily weather variations, which are essential for short-term precipitation forecasting.

The cumulative precipitation data, which is a total of all precipitation recorded over the study period, has been meticulously quantified and is presented in an accompanying hourly precipitation chart. This chart serves as a visual representation of the rainfall patterns observed throughout the year. A thorough analysis of this chart reveals a distinct seasonal trend in the occurrence of rainfall, with a notable concentration of increased rainfall events occurring from March through September.

During the early months of the study, from March to May, the region predominantly experienced light to moderate rainfall. This period is characterized by a gradual increase in precipitation, setting the stage for the heavier rainfall events that typically follow. While no extensive regions within Guangxi faced heavy rainfall during this time, the intermittent nature of the showers, alternating between heavy downpours and lighter rain, posed logistical challenges for local residents and required careful monitoring.

As the study period progressed into June and July, the Guangxi region was influenced by a persistent high-temperature weather system that extended across a broad area. This climatic condition led to a significant increase in the frequency and intensity of heavy rainfall events. The high temperatures contributed to the formation of atmospheric conditions conducive to intense precipitation, resulting in a period of heightened rainfall activity.

Following the peak rainfall months, the precipitation levels observed in the data began to decrease post-August. This reduction in rainfall is indicative of the transition into the latter part of the year, where the meteorological patterns shift, and the region experiences a relative lull in precipitation. The tapering off of rainfall in the latter months is an important aspect to consider for long-term water resource management and agricultural planning.

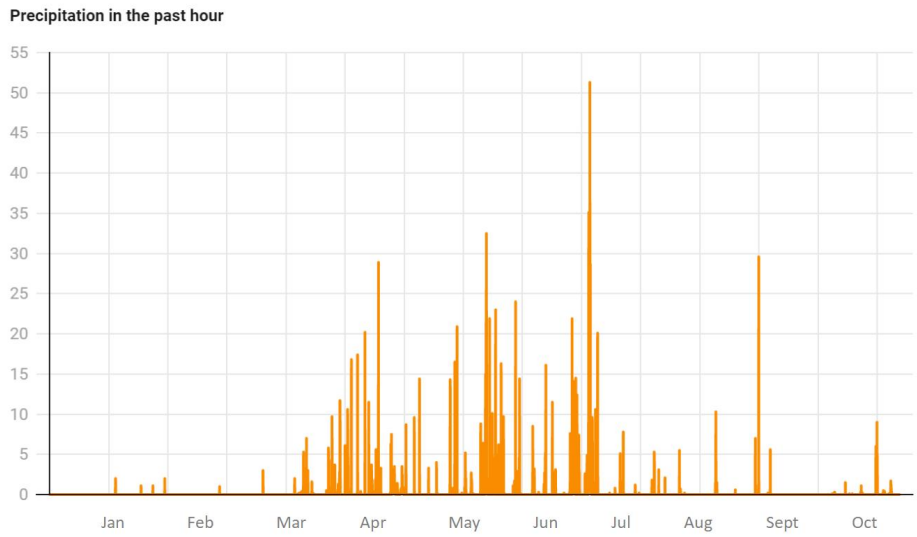


Figure 5.3: Rainfall statistics in Guangxi in 2019

5.2.2.1 Data preprocessing

(1) Description of Ground Station Datasets

Automatic weather stations, essential to meteorological observation networks, serve as the primary interface for collecting a wide array of atmospheric data. These observation sites are equipped with an interconnection of various sensors, all feeding into a centralized data acquisition system. The array of standard monitoring equipment is designed to provide a comprehensive snapshot of current weather conditions, including but not limited to a barometer for measuring atmospheric pressure, a rain gauge for precipitation levels, an anemometer for wind speed, a weather vane indicating wind direction, a louver box for housing delicate instruments, a wind direction anemometer for precise wind direction measurements, wet and dry bulb thermometers for assessing humidity, a thermo-hygrometer for temperature and humidity readings, an evaporating dish for measuring evaporation rates, a sunshine meter for recording solar radiation, and a ground temperature meter for soil temperature.

The deployment of these stations allows for the systematic capture of key meteorological elements such as atmospheric pressure, air temperature, humidity, and precipitation. These elements are fundamental to understanding weather patterns and are critical for the operation of meteorological observation systems. The precision of these measurements is paramount, as they form the basis for weather analysis and forecasting models.

The sensors within automatic weather stations are capable of discerning not only the presence of precipitation but also its type and intensity. This is achieved through the integration of multiple sensor signals, which when combined, provide a detailed account of the weather conditions. For instance, the rain gauge measures the amount of rainfall, while the wet and dry bulb thermometers can indicate the likelihood of rain, snow, or sleet based on the difference in temperature readings.

In the complex field of weather forecasting, the data gathered from these stations is subjected to advanced algorithms that are particularly designed to meteorological observations. These algorithms are designed to integrate the findings from individual stations with a broader network of observations. By considering a multitude of meteorological parameters such as air temperature, humidity, ground temperature, and cloud presence, these algorithms can produce more reliable and accurate diagnostic results. The integration of such diverse data sources is essential for developing a complete understanding of weather systems and for issuing timely and precise weather forecasts.

Moreover, the advancements in sensor technology and data processing techniques have significantly enhanced the capabilities of automatic weather stations. Modern stations are equipped with sensors that can provide real-time, high-resolution data, which is crucial for short-term weather prediction and nowcasting. The real-time nature of this data allows for the detection of rapidly changing weather conditions, enabling meteorologists to issue timely warnings and alerts.

The importance of these stations extends beyond routine weather monitoring and forecasting. They also play a critical role in climate research, where long-term data collection is necessary for understanding climate patterns and trends. The data collected by these stations contributes to climate models and helps scientists predict the impacts of climate change.

The dataset from the ground observation site in Guangxi has been chosen as a sample dataset, including pertinent meteorological observation data collected on an hourly basis spanning from 0:00 on January 1, 2019 to 23:00 on October 30, 2019. A portion of the original ground observation data is presented in the Table 5.1.

Table 5.1: Definitions, units and ranges of individual meteorological elements.

Title	Name	definition	Unit	Scope
station_ID	station number	There are 2928 sites in Guangxi		59053-846805
Lat	latitude	Guangxi is located at north latitude 20°54'～26°23'	Degree	20°54'～26°23'
Lon	longitude	Guangxi is located in east longitude 104°29'～112°04'	Degree	104°29' ～ 112°04'
PRS_Sea	sea level pressure	The weight of the air column per unit area from sea level to the upper atmosphere of the atmosphere	hPa	995hPa-1024hPa
PRS_Change_3h	3-hour variable pressure	3-hour barometric pressure change	hPa	-4.1hPa-4.4hPa
PRS_Change_24h	24-hour variable pressure	24-hour barometric pressure change	hPa	-7.7hPa-8.8hPa
TEM_ChANGE_24h	temperature change in the past 24 hours	24-hour temperature change	℃	-13.5℃-10℃
DPT	dew point temperature	The temperature at which the air is cooled to saturation when the water vapor content in the air remains unchanged and the air pressure is kept constant	℃	3.4℃-28.9℃

RHU	relative humidity	The ratio between the vapor pressure e' and the saturated vapor pressure e'w.	%	20%-100%
PRE_1h	precipitation in the past 1 hour	Water falling from the sky in liquid or solid form, also including water deposited in the form of dew, ice, etc.	mm	0mm-250mm
PRE_3h	precipitation in the past 3 hours	Water falling from the sky in liquid or solid form, also including water deposited in the form of dew, ice, etc.	mm	0mm-250mm

The observation data from the Guangxi weather station, as detailed in the table, is meticulously updated on an hourly basis, providing a precise and timely record of the atmospheric conditions for this specific region. The data set is comprehensive, including a variety of parameters that are crucial for meteorological analysis and forecasting. Specifically, the recorded information includes the station number, geographic coordinates defined by latitude and longitude, sea-level pressure, and variations in air pressure over both 3-hour and 24-hour periods. These measurements are complemented by temperature change observations over the past 24 hours, dew point temperature, relative humidity, and precipitation levels recorded for the last hour and the last 3 hours. It is important to recognize that the geographical coordinates are essential for understanding the spatial distribution of weather phenomena, with each degree of latitude and longitude representing an approximate distance of 111 kilometers on Earth's surface.

The collection of these atmospheric state variables is essential for studying precipitation patterns. With 8 distinct parameters being measured hourly, the data provides a detailed account of the atmospheric conditions related to rainfall. Each record in the dataset is a snapshot of the atmospheric state at a given time, containing quantitative information that is critical for meteorologists and data scientists. The station number serves as a unique identifier for each set of observations, while the latitude and longitude offer precise geographical context. The sea-level pressure and pressure changes are vital indicators of weather

systems' movement and development. Temperature variations, dew point, and relative humidity are key to understanding the thermodynamic properties of the atmosphere, which in turn influence precipitation formation.

Time series data, such as the hourly observations from the Guangxi station, consist of three primary elements: the time point of observation, the subject of measurement (in this case, the atmospheric conditions at the station), and the actual measurement value. In the realm of data science, time series data are typically arranged in a tabular format, such as a CSV file. Here, each row corresponds to a specific time point, and the columns represent various measured parameters, with the cells containing the respective values.

Ground station data is highly regarded for its reliability in providing observational data. However, the distribution of meteorological stations is not uniform, leading to potential gaps in the temporal resolution of the collected data. These gaps can result in incomplete meteorological records within the study area, which may affect the accuracy of weather analysis and forecasting. Furthermore, the data obtained from weather stations may sometimes contain errors or inconsistencies, such as erroneous readings or duplicated entries, which can compromise the quality of the dataset.

To ensure the accuracy and reliability of the meteorological data for analysis and forecasting, it is crucial to undertake a rigorous preprocessing step. This preprocessing may involve data cleaning to remove errors and duplicates, data interpolation to fill in gaps, and data normalization to ensure consistency across different parameters. By implementing these preprocessing techniques, researchers can enhance the quality of the meteorological data, thereby improving the reliability of weather forecasts and the accuracy of meteorological studies.

(2)Missing Value Processing

In the meticulous process of data preprocessing for the meteorological dataset, the initial step involves addressing the presence of repeated

values within the data table. The occurrence of duplicates can lead to inaccuracies in the analysis by skewing the representation of the data. It is essential to identify and eliminate these redundancies to ensure that each data point is unique and accurately reflects an independent observation. This step is crucial for maintaining the integrity and reliability of the dataset, which forms the foundation for any subsequent analysis or modeling.

Following the elimination of repeated values, the next critical task is to tackle the issue of missing values in the dataset. Missing data is a common challenge in meteorological observations due to various factors such as equipment malfunction, power outages, or environmental conditions that may hinder data collection. In this dataset, certain factor values are marked as 999999 and 999998, which are indicative of missing data, signifying that the respective sites did not collect data for these factors during the specified periods.

To handle these missing values, the article employs a strategic approach using the Python programming language, which is renowned for its robust data manipulation libraries. Specifically, the fillna function from the pandas library is utilized. This function is instrumental in filling in the gaps within the dataset by replacing the missing values with a statistic that is representative of the overall distribution of the data.

In this context, the choice of the median as a replacement for missing values is a deliberate and data-informed decision. The median is a robust statistical measure that is less sensitive to extreme values or outliers compared to the mean. By using the median, the dataset maintains a balance that is not skewed by potential extreme data points, thus preserving the representativeness of the meteorological observations.

Moreover, the integration of the date information is a essential step in the organization of the meteorological time series data. By setting the date as the index, the data is chronologically ordered, which facilitates the analysis of trends and patterns over time. This chronological

organization is essential for time series forecasting models that rely on the sequential nature of the data to make predictions.

The preprocessing steps outlined above—removing duplicates, imputing missing values with the median, and using the date as the index—are fundamental to the preparation of the meteorological dataset. These steps not only cleanse the data but also enhance its quality, ensuring that it is ready for advanced analytical techniques and modeling, which are critical for accurate weather forecasting.

(3)Data Standardization

In this research, we have identified 11 key meteorological elements as input variables for our predictive model. These elements include a diverse range of measurements, each with its own unique magnitude and units. The heterogeneity of these units poses a significant challenge when integrating the data into a unified model, as the differences in scale can skew the model's learning process and lead to suboptimal predictions.

The variance in magnitude and units across meteorological elements such as temperature, pressure, and humidity can create an imbalance in the contribution of each variable to the model's output. For instance, a small change in atmospheric pressure may be of similar significance to a large change in relative humidity, but due to their differing scales, the model might not accurately capture this relationship without appropriate preprocessing.

To address this issue, the article employs min-max normalization, a widely-used technique in data preprocessing that rescales the data to a common scale, typically [0, 1]. This method linearly transforms the original values of each meteorological element to a normalized form, ensuring that all input variables contribute equally to the model's predictions. The formula for min-max normalization is given by:

$$x' = \frac{x - \min A}{\max A - \min A} \tag{5.7}$$

Among these, "max" denotes the maximum value of each meteorological element within the cleaned sample data, while "min" represents the minimum value of each meteorological element in the cleaned sample data. After normalization, each input variable is in the same order of magnitude, which can speed up the training speed and reduce the output error.

Normalization is a critical step in the data preprocessing phase, especially when dealing with models that are sensitive to the scale of input data, such as neural networks. By applying min-max normalization, we ensure that the model is not biased towards variables with larger magnitudes and that all variables are treated equally during the training process.

Normalization can also improve the convergence speed of the model during training, as it helps in maintaining the gradients within a specific range, thus facilitating faster and more stable optimization. After normalization, the dataset is more amenable to analysis and can be directly compared across different variables. This allows for a more meaningful interpretation of the model's weights and biases, as they are no longer influenced by the original scale of the data.

Table 5.2: Standardized meteorological elements.

	PRS_Sea	PRS_Change_3h	PRS_Change_24h	TEM_Chang_24h	DPT	RHU	PRE_1h	PRE_3h
variance	1024.2	4.4	8.8	10.0	28.9	100.0	51.3	101.6
mean	995.7	-4.1	-7.7	-13.5	3.4	20.0	0.0	0.0

Traverse all files, get the data of the specified site, clean the data, set the machine outliers to 0, delete the unused columns: site, 'latitude',

'longitude', before data analysis, we need to standardize the data, using the standardized data for data analysis.

Table 5.3: Standardized dataset

station	year	month	date	time	PRS_Sea	PRS_Change_3h	PRS_Change_24h	TEM_Chang_24h	DPT	RHU	PRE_1h	PRE_3h
803798	2019	1	1	0	1024	1.7	5.9	-2.3	12.4	95	0.1	0.8
59631	2019	1	1	1	1024.2	1.7	5.1	-1.1	12.8	94	0	0.1
57949	2019	1	1	2	1023.9	0.7	4.4	-0.6	12.8	90	0	0.1
821708	2019	1	1	3	1023.3	-0.6	3.4	0.5	12.1	83	0	0
57821	2019	1	1	4	1022.4	-1.7	2.8	1.8	12.3	82	0	0
846849	2019	1	1	5	1021.5	-2.4	2	1.5	11.9	79	0	0
805199	2019	1	1	6	1020.6	-2.6	1.7	1.5	11.7	77	0	0
803659	2019	1	1	7	1020	-2.3	1.1	0.9	12.3	82	0	0
786434	2019	1	1	8	1019.6	-1.8	0.4	0.6	12.5	85	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
57971	2019	10	30	22	1021.8	1.2	-1.1	0.7	10.1	100	0	0
801461	2019	10	30	23	1019.1	1.2	-0.5	0.6	9.3	100	0	0

(4) Data Correlation Calculation

Following the meticulous data preprocessing steps outlined previously, the dataset is now devoid of outliers, which can often distort analysis and lead to unreliable conclusions. With the removal of these extreme values, the data presents a cleaner, more accurate representation of the

underlying patterns and relationships. The absence of outliers allows for a more focused examination of the inherent structure within the data.

The data now exhibits a linear correlation between the two groups, indicating that as one variable changes, the other variable changes at a constant rate in the same or opposite direction. This linear relationship is a fundamental aspect of statistical analysis, providing a basis for understanding the potential causality or association between variables. The presence of a linear correlation suggests that a simple and interpretable model may be sufficient to capture the relationship between the variables in question.

Covariance, which is the basis of Pearson's correlation coefficient, is a measure of the joint variability of two random variables. It represents the numerical characteristics of the relationship between two variables, X and Y. A positive covariance indicates that as X increases, Y tends to increase as well, while a negative covariance suggests that as X increases, Y tends to decrease. The magnitude of the covariance is influenced by the scale of the variables, which is why it is normalized to produce the correlation coefficient.

The Pearson correlation analysis will enable us to discern the degree to which different meteorological elements are interrelated. This understanding is invaluable for feature selection in predictive modeling, as it helps in identifying which variables have the most significant influence on the target variable, in this case, temperature prediction.

Covariance is Pearson's correlation coefficient. Covariance represents the numerical characteristics of the relationship between two variables X and Y. The calculation formula of Covariance is :

$$\mathrm{COV}(X,Y) = \frac{1}{n-1}\sum_{1}^{n}(X_i - \overline{X})(Y_i - \overline{Y}) \tag{5.8}$$

The formula for the pearson correlation coefficient is as follows:

$$\mathrm{COR}(X,Y)=\frac{\sum_1^n(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_1^n(X_i-\overline{X})^2\sum_1^n(Y_i-\overline{Y})^2}} \tag{5.9}$$

From the formula, it is evident that the Pearson correlation coefficient is derived by dividing the covariance by the standard deviations of the two variables. While covariance can indicate the degree of correlation between two random variables (a covariance greater than 0 suggests a positive correlation, and less than 0 implies a negative correlation), its value is significantly influenced by the scale of measurement. The degree of correlation between variables cannot be simply assessed based on the covariance value alone. In order to eliminate the influence of this dimension, there is the concept of correlation coefficient. The correlation coefficient is only meaningful when the variance of the two variables is not zero, and the value range of the correlation coefficient is [-1, 1].

The Pearson correlation coefficient can calculate the contribution of each feature. Eliminating irrelevant data columns can reduce the burden of system operation and improve the operation speed. We pick out features that are positively correlated with precipitation, which are 3-hour variable pressure, dew point temperature, relative humidity, the past 1 hour and precipitation in the past 3 hours. The 5 Meteorological Elements were preserved. As presented in Fig 5.4, 0.6-0.8: Strong correlation, 0.4-0.6: Moderate correlation, 0.2-0.4: Weak correlation, 0.0-0.2: Very weak correlation or no correlation, negative numbers are negative correlations.

As can be seen from the Fig 5.4, sea level pressure has a very weak correlation with 3-hour variable pressure and 24-hour variable pressure, and is not directly related to other elements. The 3-hour variable pressure has a weak correlation with relative humidity and very weak correlation with other elements. The features of 24-hour variable pressure and temperature change in the past 24 hours contributed little

and were basically negative correlations. Dew point temperature has a strong relationship with relative humidity and a weak relationship with precipitation in the past 1 hour and precipitation in the past 3 hours. The characteristics of relative humidity contribute significantly to both precipitation in the past 1 hour and precipitation in the past 3 hours. The relationship between precipitation in the past 1 hour and precipitation in the past 3 hours is the strongest. This relationship reached 0.75, indicating that precipitation at this time has a strong Pearson correlation.

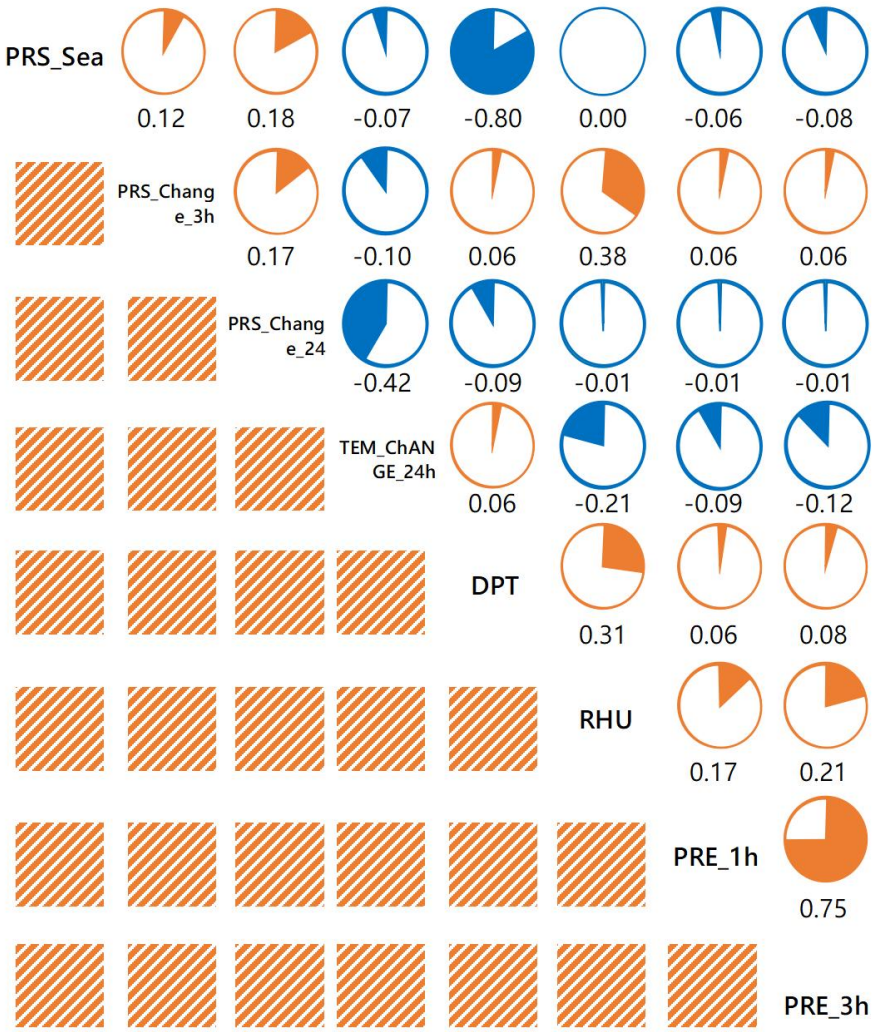


Figure 5.4: Pearson correlation coefficient between various Meteorological elements

5.3 Experiments

5.3.1 Division of Datasets

Convert the data into tensor format, split and generate two sets of data sets for model training and testing. We take the first 292 days (1 year) of data for model training and the following 73 days for testing. Generate a data loader, set the batch size, batch_size=128 for the training set, and batch_size=512 for the test set. The model parameters are optimized after each batch, so the larger the batch size, the shorter the training time of each epoch; however, the batch size is too large and it is not easy to converge, resulting in more training times (epochs) required, so the total training time may be longer. How much is appropriate? This seems to be a situation where there is no definite algorithm, and it has to be adjusted slowly as it learns from experience.

Table 5.4: Experimental setup of Transformer modeling and informer modeling.

	Transformer modeling and informer modeling
Number of Stations	2928
Meteorological Elements	5
Training Date	01/01/2019-31/05/2019
Validating Date	01/06/2019-31/08/2019
Testing Date	01/09/2019-30/10/2019
Loss Function	MSELoss
Evaluation Metric	MAE, MSE, RMSE, MAPE

In this experiment, we use historical 15-hour ground observatory data as input to predict the past 1-hour rainfall over the next 5 hours. In the actual observation, the data of the ground observation station is updated every hour, and the data of 15 time points are input into the Transformer and the informer. The input dimension of the encoder of the two models is set to 8, and the input dimension of the decoder is set to 7. The output dimension of the decoder is 1, and the time length is 5. The number of heads for multi-head attention is 8, and the number of layers of the

encoder and decoder is both 2.

5.3.2 Experimental Environment and Strategies

The experiments in this paper use PyTorch 1.7 as the deep learning framework and use Python 3.7 for coding; the experiments are implemented on a PC, Windows 10 operating system, AMD Ryzen 7 3700X CPU and ROG STRIX GeForce RTX 2080Ti GPU). We take the average loss value over 100 iterations as the final loss value for each epoch. We test the model to get the final prediction result on the test set. MSELoss is used as the loss function of the model; Adam is used as the optimizer of the model. This paper uses 15 hours of historical ground observation data to predict rainfall for the next 5 hours. The parameter settings of the model are shown in the table.

Table 5.5: The values of hyper-parameters.

Hyper-parameters	Values
Batch size	512
Number of heads in multi-head attention	8
Dropout rate	0.1
Encoder layers	6
Decoder layers	6
Intermediate layer dimension of feedforward network model	2048
Activation functions for the encoder/decoder intermediate layers	relu

5.3.3 Experiment results and analysis

In this section, we present the experimental results and corresponding analysis based on the settings previously introduced. Figure 6 provides a visual representation of the training phase, where it is observed that the Informer model demonstrated a more pronounced convergence compared to its Transformer counterpart. This enhanced convergence can be primarily attributed to the innovative distilling self-attention mechanism integrated within the Informer's architecture. This

mechanism is designed to distillate the self-attention maps, leading to the creation of more refined and precise feature representations. The distillation process allows the model to focus on the most informative aspects of the data, thereby enhancing the quality of the learned representations.

The improved feature representations, in turn, facilitate a more rapid and accurate convergence during the training process. This is a critical factor, as it reflects the model's ability to learn from the training data efficiently and to adjust its parameters in response to the error signals received.

Moreover, the performance of the Informer model is further underscored by its achievement of a lower loss value when compared to the Transformer model. The loss value is a essential metric in evaluating the performance of a model during training, as it quantifies the difference between the predicted outputs and the actual targets. A lower loss value indicates that the Informer model's predictions are closer to the true values, signifying higher accuracy and better generalization capabilities.

The superior performance of the Informer model during training can be linked to its ability to capture long-term dependencies in the data, which is a common challenge in sequence modeling tasks. The Informer's architecture, which includes the distilling self-attention mechanism, is particularly skilled at handling such dependencies, enabling it to make more accurate predictions over extended time horizons.

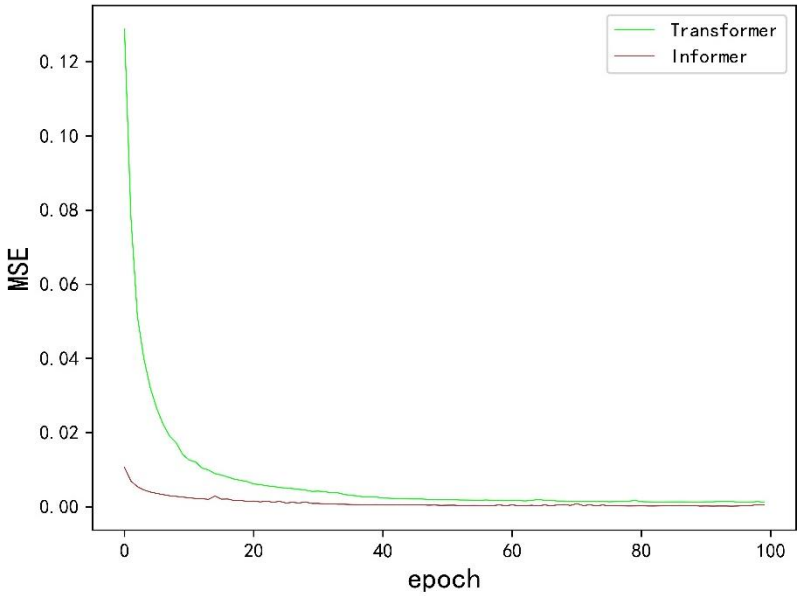


Figure 5.5: The convergence of two methods

Following the completion of training, we proceeded to evaluate the forecast capabilities of both models on the AWS dataset, and the average performance is summarized in Table 5.5. Notably, the Transformer-based approaches outperformed the RNN-based models substantially. Specifically, the Transformer model exhibited performance to the GRU model by -2.04%, outperformed the LSTM model by 33.68% and outperformed the Feedforward Neural Network(FFN) by 38.12%. Additionally, the Informer model outperformed the GRU model by 14.16%, outperformed the LSTM model by 229.5%, and outperformed the FFN by 240.5%. Comparing the Transformer and Informer models, the Informer model performed better than the Transformer model by 14.65% on average.

Table 5.6: Experiment results of different methods

Method	MAE	MSE	RMSE	MAPE
LSTM(8 layers)	0.0236	0.003	0.055	0.126
GRU(8 layers)	0.023	0.003	0.055	0.0712
FFN(8 layers)	0.0212	0.0025	0.05	0.1408
Transformer	0.0174	0.0012	0.034	0.1027
Informer	0.0077	0.0002	0.0129	0.0422

5.4 Analysis of the experimental results

5.4.1 Contrast in terms of temporal aspects

The results for both the Transformer and Informer models are depicted in Fig 5.6 and Fig 5.7, respectively. Notably, the Informer model achieved more accurate forecasting, with the predicted rainfall more accurately reflecting the actual rainfall.

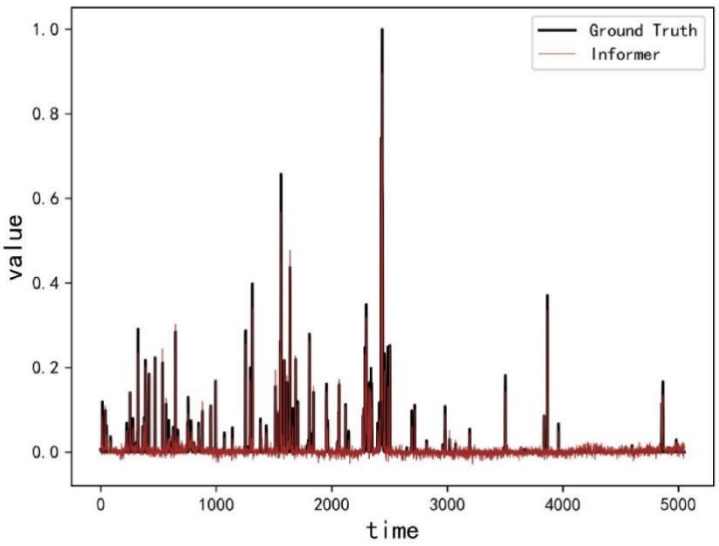


Figure 5.6: Forecasting result of Informer

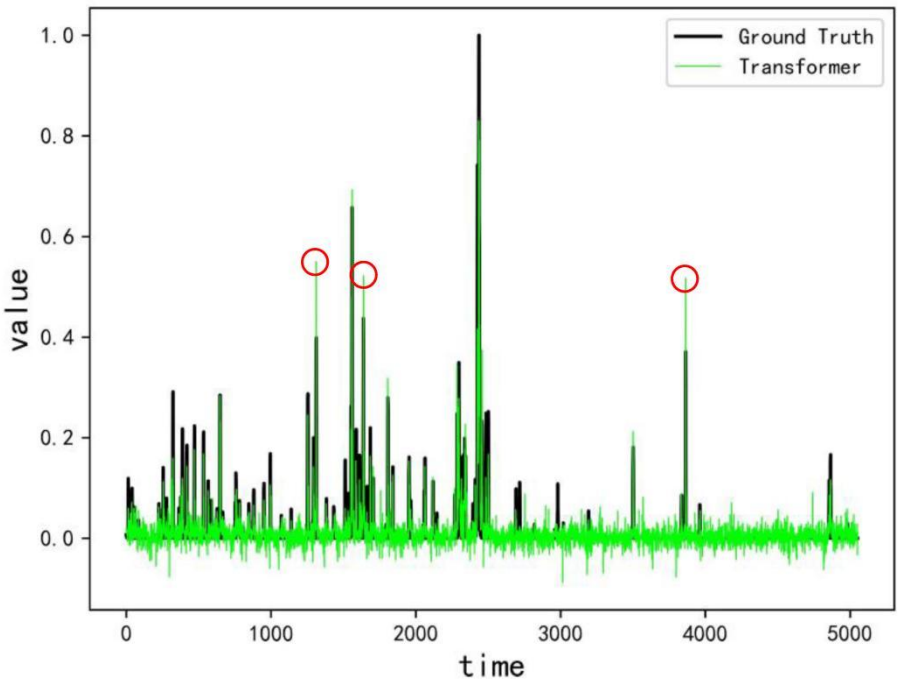


Figure 5.7: Forecasting result of Transformer

5.4.2 Contrast in terms of spatial aspects

To better illustrate the predictive capabilities of the models, we utilized the weights obtained from the training phase to conduct one-step forecast. Specifically, we input the data from time points t-1 to t-5 as the input data and generated predictions for the rainfall at time point t. The results for both the Transformer and Informer models are depicted in Fig. 5.8 and Fig. 5.9, respectively. Notably, the Informer model achieved more accurate forecasting, with the predicted rainfall more accurately reflecting the actual rainfall.

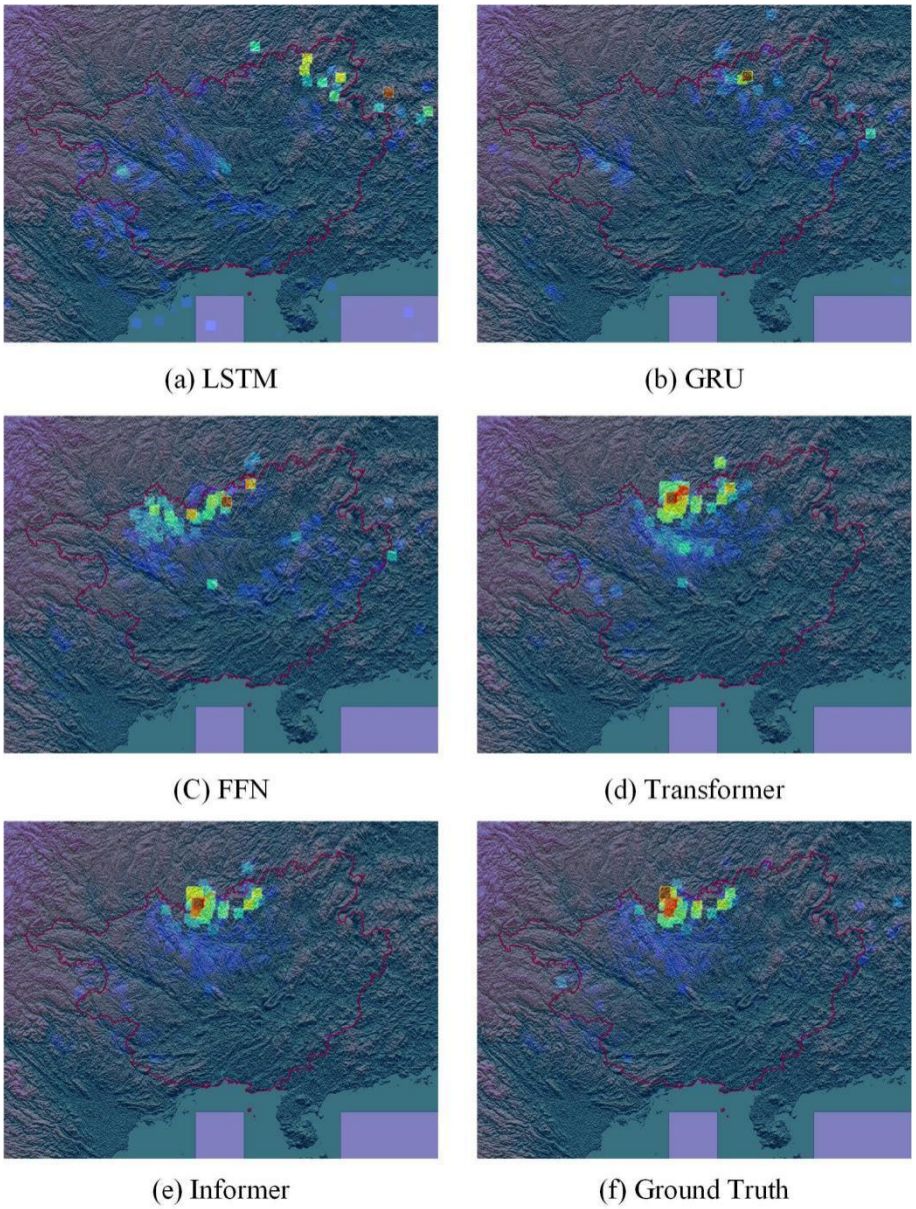


Figure 5.8: Visual results of combining forecast with topographic maps

To elucidate the comparative forecasting strength of various models, we have selected a specific dataset from the Guangxi region, recorded on August 2, 2019, for an in-depth prediction analysis. This date was chosen to provide a clear and focused evaluation of the models' capabilities under a set of environmental conditions. The selection of this dataset allows us to examine the models' performance under real-world circumstances, offering a robust basis for comparison.

As illustrated in Figure 5.8, the RNN-based models, namely (a) Long

Short-Term Memory (LSTM) and (b) Gated Recurrent Unit (GRU), demonstrate a notable shortfall in capturing the spatial nuances of the rainfall data. These models, despite their strengths in handling sequential data, struggle to accurately represent the geographical distribution of precipitation. This limitation leads to a considerable discrepancy between the predicted and actual rainfall, indicating a less effective performance in this context.

Furthermore, the performance of the RNN-based models is found to be outperformed by the (c) Feedforward Neural Network (FFN) model, which, despite its simpler architecture, provides more reliable predictions in terms of rainfall range. The FFN model's ability to make more accurate forecasts may be attributed to its focus on the input-to-output mapping without the complexity of maintaining a state, which, in this case, seems to be advantageous.

In stark contrast, the Transformer and Informer models, which are designed to handle sequential data with a heightened sensitivity to temporal dynamics, achieve commendable results in the forecasting task. These models leverage self-attention mechanisms and gating structures to process the temporal dependencies within the data, leading to a more subtle understanding of the rainfall patterns.

Upon a meticulous comparison of the forecasting outcomes, it is observed that the Informer model surpasses the Transformer model in terms of prediction accuracy. The Informer model's architecture, which incorporates a distilling self-attention mechanism, allows it to generate more precise and temporally coherent feature representations. This capability enables the Informer to better capture the complex spatial and temporal relationships within the meteorological data, resulting in superior predictive performance.

The results underscore the importance of selecting the appropriate model for a given task. While RNN-based models may excel in certain domains, their limitations in spatial representation are exposed in this analysis. The Transformer and Informer models, with their advanced processing of temporal sequences, demonstrate a clear advantage in

forecasting tasks, particularly when fine-grained spatial and temporal accuracy is required.

5.5 Conclusions

Weather forecasting, a essential domain within meteorology, has seen a surge in significance with the continual progression of technological capabilities. This study pioneers an interdisciplinary approach by amalgamating the fields of meteorology and artificial intelligence (AI), with a particular focus on leveraging Transformer and Informer models for the purpose of near-term precipitation forecasting. Utilizing ground observation data as the foundation for these models, this research endeavors to enhance the precision and efficiency of weather predictions.

The study's essential contributions are outlined as follows:

1. A meticulous and extensive review of existing literature on weather forecasting was conducted. This scholarly review served as the foundation for the selection of the Transformer and Informer models, which were identified based on their proficiency in predicting a spectrum of meteorological elements. The selection process was grounded in the models' demonstrated capabilities and their relevance to the objectives of this study.

2. The research navigated the complexities of handling vast volumes of multi-dimensional meteorological time series data. To refine the data, a noise reduction technique was applied to mitigate the interference of irrelevant fluctuations and to enhance the signal-to-noise ratio. Furthermore, the selection of high-correlation meteorological events centered on precipitation as the key feature variable was prioritized. This methodological choice aimed to curtail the influence of uncertain factors on the observed data, thereby furnishing more potent and precise data for the subsequent utilization by the neural network. This, in turn, bolsters the network's capacity to represent and learn from the data, leading to an enhanced predictive accuracy.

3. The empirical results garnered from both the Transformer and Informer models substantiate their efficacy in the field of precipitation forecasting. The findings indicate that the amalgamation of meteorological knowledge with AI, specifically through the deployment of these models, confers substantial benefits for near-term precipitation forecasting. This integration has the potential to offer more reliable predictions, which is invaluable for various applications such as disaster preparedness, agriculture planning, and urban management.

Moreover, this study also delves into the operational dynamics of the Transformer and Informer models, elucidating how these models process sequential data and capture temporal dependencies within the meteorological context. The research provides insights into the models' architecture, their ability to handle irregular time intervals, and their adaptability to the unique challenges posed by meteorological data.

CHAPTER 6

ANALYSIS BASED ON RADAR CHARTS AND GROUND STATION DATA PREDICTION

6.1 Introduction

As meteorological technology advances, the field of weather forecasting has evolved to incorporate two main approaches for rainfall prediction: radar-based and ground station-based methods. Radar-based rainfall prediction involves the analysis of radar signals to assess atmospheric moisture and cloud patterns, which are critical for anticipating future precipitation events. This method provides a real-time, spatial view of developing weather systems, allowing for the tracking of rain-bearing clouds and the potential for localized heavy rainfall.

On the other hand, ground station rainfall prediction is grounded in the collection of meteorological data from terrestrial observation stations. These stations measure essential parameters such as temperature, humidity, wind speed, and air pressure. The collation and examination of this data provide an extensive comprehension of the prevailing weather conditions, which subsequently serve as the basis for predicting imminent rainfall events. This approach is particularly valuable for its detailed insights into local weather variations and its ability to capture small-scale weather phenomena that may not be detected by radar alone.

The primary research focus of this chapter is to explore the integration of these two rainfall prediction methods. The aim is to enhance the accuracy and reliability of precipitation forecasts by leveraging the strengths of both approaches. We will delve into the technical and methodological aspects of combining radar-based and ground station-based data to create a hybrid model for rainfall prediction.

This involves a detailed examination of data fusion techniques, where the high-resolution spatial data from radar systems is complemented by the precise, localized data from ground stations. The integration of these datasets requires careful consideration of data synchronization, spatial and temporal resolution, and the handling of potential discrepancies between the two sources of information.

Moreover, the chapter will address the development of algorithms that can process and analyze the combined data effectively. These algorithms will be designed to identify patterns and trends in the data that are indicative of rainfall, and to learn the complex relationships between various meteorological parameters and precipitation events.

The performance and accuracy of the proposed rainfall prediction model will be rigorously evaluated through a series of experiments. These experiments will simulate real-world weather conditions and assess the model's ability to predict rainfall accurately over different timescales and geographical areas. The results will be compared against independent datasets to validate the model's predictive capabilities.

6.2 Data selection

In this chapter, the radar distribution data and ground station sample data from June 3, 2019, and June 16, 2019, are selected for algorithm verification and comparison. To enhance the analysis and better understand the interplay between terrain features and actual rainfall, we incorporate an integrated analysis of the terrain map of the Guangxi region. This approach is critical because the topography of an area can significantly influence rainfall distribution, affecting the accuracy of our predictions.

For this study, we utilize the 30-meter accuracy elevation data known as CLDASGRID-DEM, which is publicly released by the China Meteorological Administration. This dataset provides a detailed representation of the topographical landscape, which is essential for our analysis. We focus on the Guangxi Zhuang Autonomous Region, which

is characterized by its unique geographical coordinates: East longitude ranging from 104.43333 to 112.06666 and North latitude from 20.9 to 26.4.

By extracting the corresponding regional elevation information based on the geographical information of the Guangxi Zhuang Autonomous Region, we are able to map out the terrain's influence on rainfall patterns with greater precision. The coordinate system transformation applied to the DEM data allows us to represent the area of interest within Guangxi in grid data format, specifically within the range of [500: 1200, 3400: 4300].

This transformation into grid data is a crucial step as it standardizes the elevation information, facilitating the integration with other datasets such as radar and ground station data. The grid format also enables us to perform spatial analysis, which is vital for understanding the spatial distribution of rainfall and its correlation with the underlying topography. The results obtained are as follows:

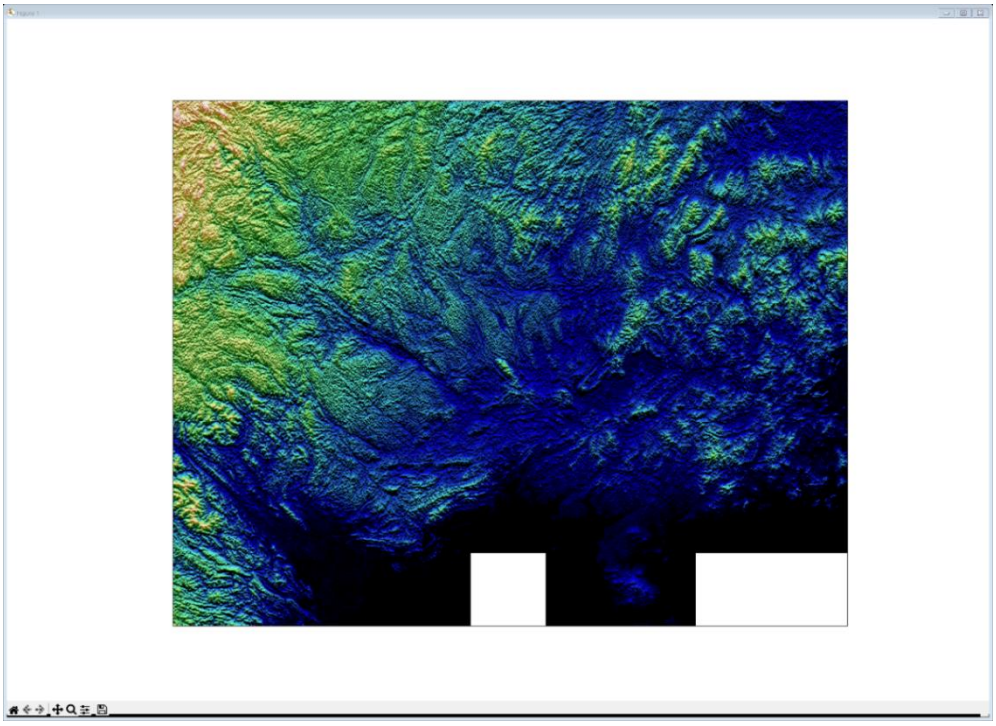


Figure 6.1: Topographic Map of South China Region

Utilizing geographical data for the administrative boundaries of provinces, cities, and counties in China from the cnmaps library, polygons are constructed to outline the actual administrative region of the Guangxi Zhuang Autonomous Region. The results are displayed using matplotlib, as shown in the figure:

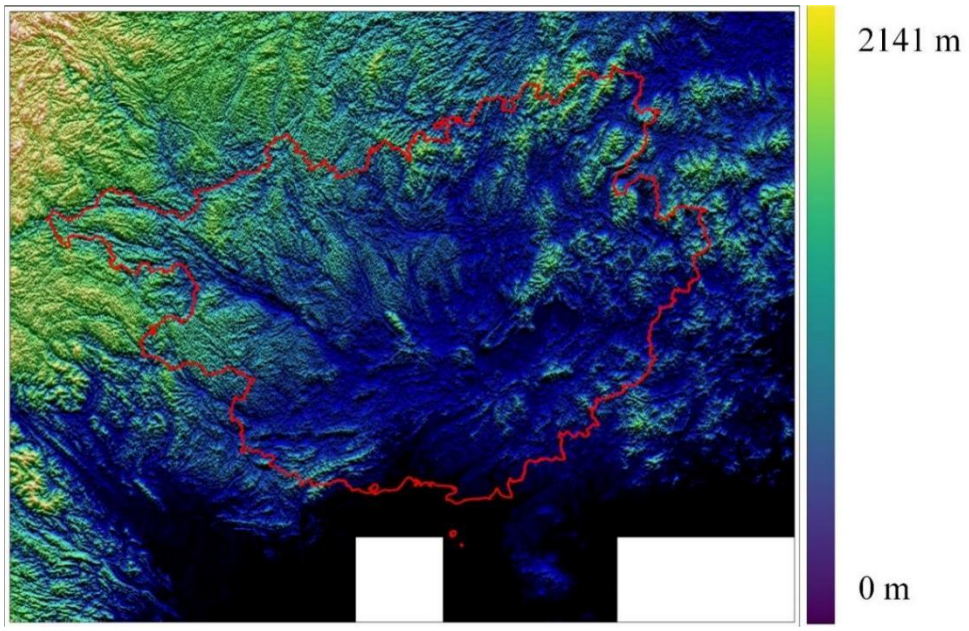


Figure 6.2: Guangxi Topographic Map

Building upon the radar reflectivity cloud data acquired in Chapter 4, which is contained in binary (bin) files as unsigned binary data, the following code is written utilizing Python's 'struct' and 'csv' libraries:

Table 6.1: Code for converting .bin to images

Code content
import os import struct import csv file_path = 'test\\Z_OTHE_RADAMCR_20190601000000.bin' data_bin = open(file_path, 'rb+') data_size = os.path.getsize(file_path) data_list = [] for i in range(data_size):   data_i = data_bin.read(1) # To output one byte at a time

```
            num = struct.unpack('b', data_i) #
        data_list.append(num[0])
data_bin.close()

new_data_list = data_list[0:(900*1200)]
mmax = max(new_data_list)
mmin = min(new_data_list)
ang = []
# print(mmin,mmax)

grouped_data = [data_list[i:i+900] for i in range(0, len(data_list), 900)]

with open('output.csv', 'w', newline='') as f:
        writer = csv.writer(f)
        for row in grouped_data:
            writer.writerow(row)
f.close()
print("Done")
```

The provided code snippet serves the purpose of processing binary radar distribution data and converting it into a readable format, specifically a CSV file. The code is written in Python and utilizes several standard libraries to achieve its goal. Below is an in-depth elucidation of each segment of the code:

1. Importing Libraries: The code begins by importing necessary libraries. os is used for interacting with the operating system, struct is used for packing and unpacking binary data, and csv is used for writing data to a CSV file.

2. Defining the File Path: The variable file_path is assigned the string value of a file path. This path points to a binary file (.bin) that contains radar data from June 3, 2019. The file is expected to be in the current working directory under a folder named 'test'.

3. Opening the Binary File: The data_bin variable is used to open the file specified by file_path in read-binary mode ('rb+'). This mode allows reading the file in binary format.

4. Determining File Size: The os.path.getsize(file_path) function is called to obtain the size of the file in bytes, which is stored in the variable data_size.

5. Reading and Storing Binary Data: A for loop is used to iterate over the file size times. In each iteration, the read(1) method reads one byte from the file. The struct.unpack('b', data_i) function is then used to convert the read byte into a signed char (an 8-bit integer). The result is appended to the data_list list.

6. Extracting Relevant Data: After the loop, data_list contains all the data from the file. The code then slices the list to extract only the first 900x1200 elements, assuming the radar data is organized in a 900x1200 grid. This is stored in new_data_list.

7. Finding Maximum and Minimum Values: The max and min functions are used to find the maximum and minimum values in new_data_list, which could be used for normalization or scaling purposes.

8. Preparing Data for CSV: The grouped_data list comprehension is used to group the elements of data_list into sublists of 900 elements each, corresponding to the rows of the radar data grid.

9. Writing to CSV: The csv.writer is used to write the grouped data to an output file named 'output.csv'. Each row in grouped_data is written as a separate row in the CSV file.

10. Closing the File: After the loop, both the binary data file and the CSV file are closed using the close() method to free up system resources.

11. Completion Message: Finally, the code prints "Done" to indicate that the process of reading the binary file and writing its contents to a CSV file has been completed.

We obtain the file and present the rest in Fig. 6.3:

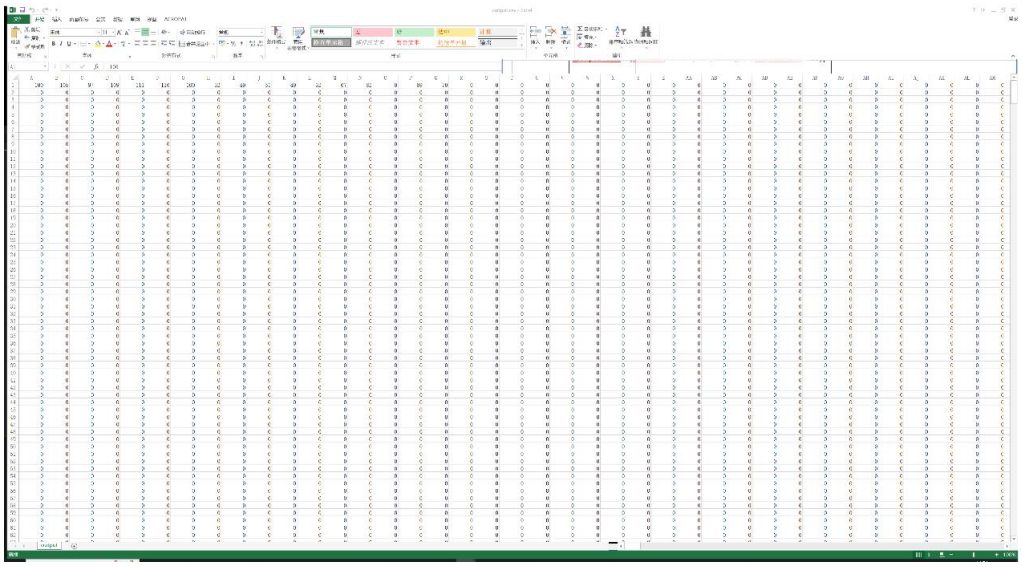


Figure 6.3: Legend of the content of CSV file.

In the realm of image processing for meteorological data, the raw data distribution frequently does not meet the prerequisites for direct visualization or further analysis. As shown in Figure 6.3, the initial data distribution may not be amenable to image processing due to factors such as its scale, range, or format. To remedy this, interpolation methods are utilized to convert the data into a format that is more appropriate for the ensuing processing stages.

Interpolation is a mathematical method that estimates unknown data points within the range of a discrete set of known data points. In the case of meteorological data, this process is crucial for generating a continuous and smooth representation of the data, which can then be effectively utilized in image processing algorithms.

The specific approach taken in this study involves linearly mapping the data to a 0-255 range, which corresponds to the standard grayscale values used in image processing. This linear transformation ensures that the original data's relative differences are preserved while scaling the data to fit within the displayable range of most imaging systems. The result of this interpolation is a grayscale image that provides a visual representation of the data, as illustrated in Figure 6.4.

The grayscale image produced through interpolation offers several advantages. Firstly, it allows for the visualization of data in a format that is easily interpretable by both humans and machine learning algorithms. Secondly, the grayscale representation can enhance the contrast and clarity of the data, making subtle variations more discernible. This is particularly important in meteorological applications, where the ability to detect small changes in data can have significant implications for weather prediction and analysis.

The use of interpolation to process meteorological data for image generation is a critical step in the pipeline of data analysis. It transforms the data into a format that is not only visually informative but also suitable for advanced image processing techniques. The resulting grayscale image, as demonstrated in Figure 6.4, is a testament to the effectiveness of the interpolation process in preparing data for downstream applications in meteorology.

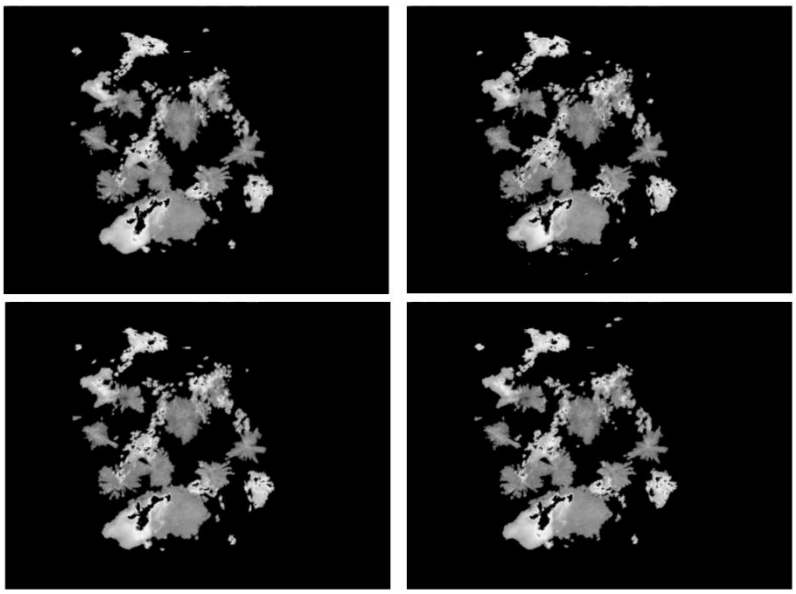


Figure 6.4: Sample of Cloud Distribution in Guangxi on June 16, 2019

The data covers the following geographical range: East longitude: 102° to 114°, North latitude: 19° to 28°, with a horizontal resolution of 0.01° × 0.01°. Therefore, it is necessary to clip and reproject the data according to the administrative boundaries of Guangxi as described above.

6.3 Prediction results based on radar data

We made predictions based on proposed GRAPHAT_NET method and we obtain the results as shown in Fig 6.5:

Figure 6.5: Guangxi Cloud Distribution Forecast by GRAPHAT-NET from 0 to 23 on June 16, 2019.

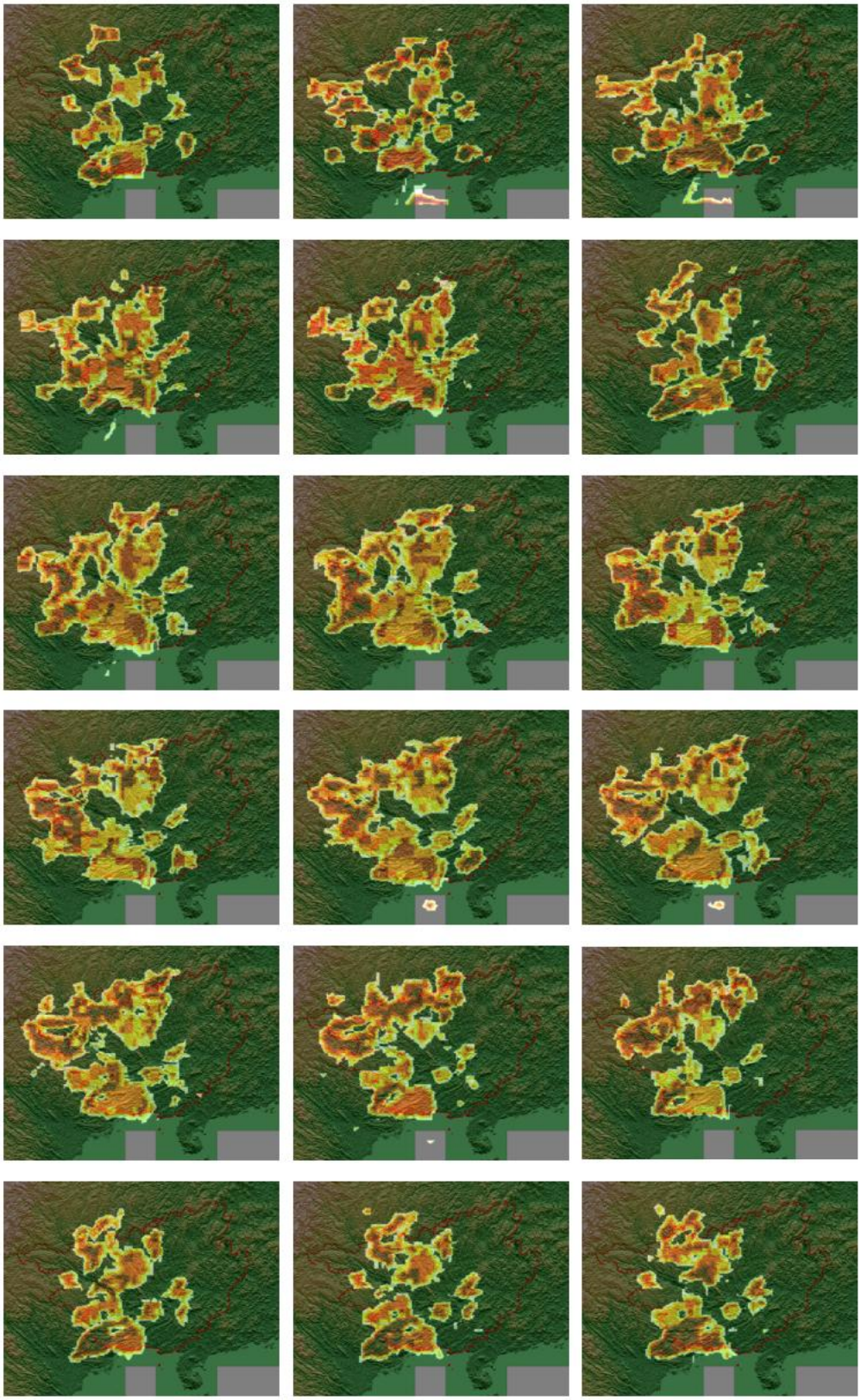
The process of generating a cloud distribution prediction map from radar data involves several critical steps, each designed to enhance the accuracy and interpretability of the final output. Given that the radar data's distribution range is confined to the radar's detection scope, the initial step is to crop the data to match the actual geographical coordinates of the Guangxi region. This cropping is essential for aligning the radar data with the specific area of interest, ensuring that the subsequent analysis and visualization are geographically relevant.

Once the data is cropped to the region of interest, the next step involves the transformation of the radar data into a format that can be visually represented. This transformation is achieved through the application of a color mapping technique, which assigns colors to the data matrix values based on their intensity. The color mapping method used in this study is the Bronze method, a specific color scheme designed to provide a clear and intuitive visual representation of the data.

In the Bronze method, the color red is used to denote areas of high intensity, such as regions where there is a high probability of cloud formation or areas with dense cloud cover. Conversely, the color yellow is used to represent areas of low intensity, indicating regions with less cloud cover or lower probabilities of cloud formation. This color-coding system allows for a rapid and effective visual assessment of the cloud distribution across the Guangxi region.

After applying the Bronze color mapping, the final step in the process is to overlay the color-coded matrix onto geographical imagery of Guangxi. This overlaying step integrates the radar data with visual features of the landscape, such as landforms, water bodies, and urban areas, providing a comprehensive and geographically accurate representation of the predicted cloud distribution. The resultant cloud distribution prediction map serves as a valuable instrument for meteorologists and researchers, facilitating a more precise and lucid analysis and interpretation of weather patterns and potential cloud formations.

The following figure presents the cloud distribution prediction result, which is a testament to the effectiveness of the described methodology. The map not only serves as a visual aid but also as a data-rich resource that can be used for further analysis, such as the identification of weather trends, the prediction of rainfall events, and the assessment of potential weather-related hazards.



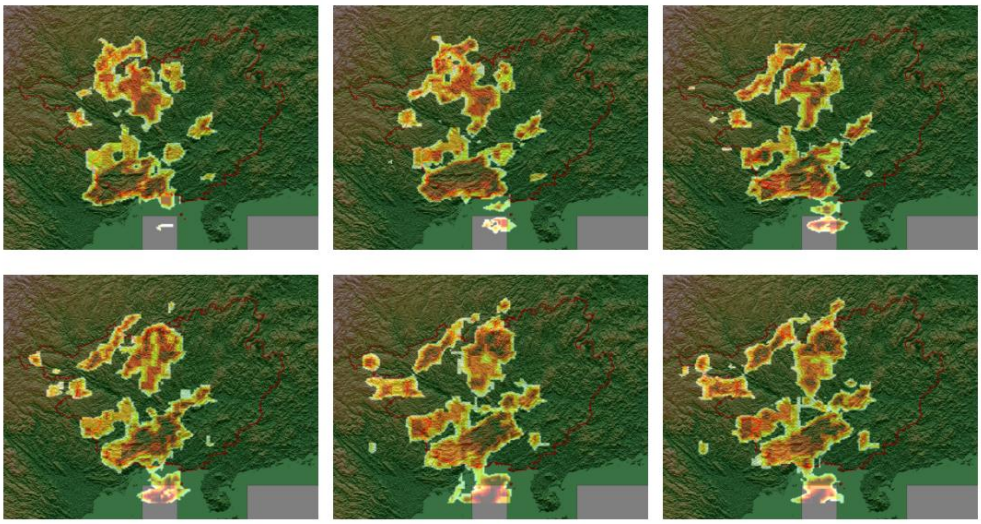


Figure 6.6: Guangxi Cloud Distribution Forecast Results from 0 to 23 on June 16, 2019

6.4 Prediction results based on ground station data

The ground station data is categorized into 11 components: 'Station', 'Latitude', 'Longitude', 'Sea Level Pressure', '3-hour Pressure Change', '24-hour Pressure Change', 'Past 24-hour Temperature Change', 'Dew Point Temperature', 'Relative Humidity', 'Past 1-hour Precipitation', and 'Past 3-hour Precipitation'. The data is saved in a TXT document format, and is specifically demonstrated in the Fig 6.7:

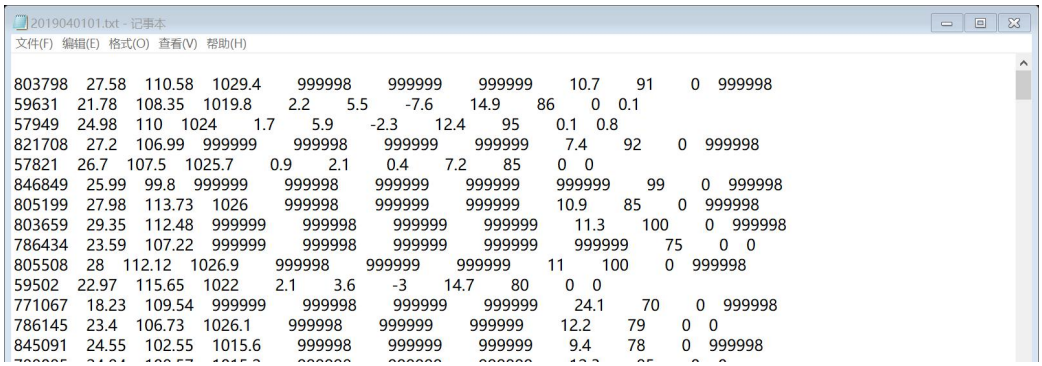
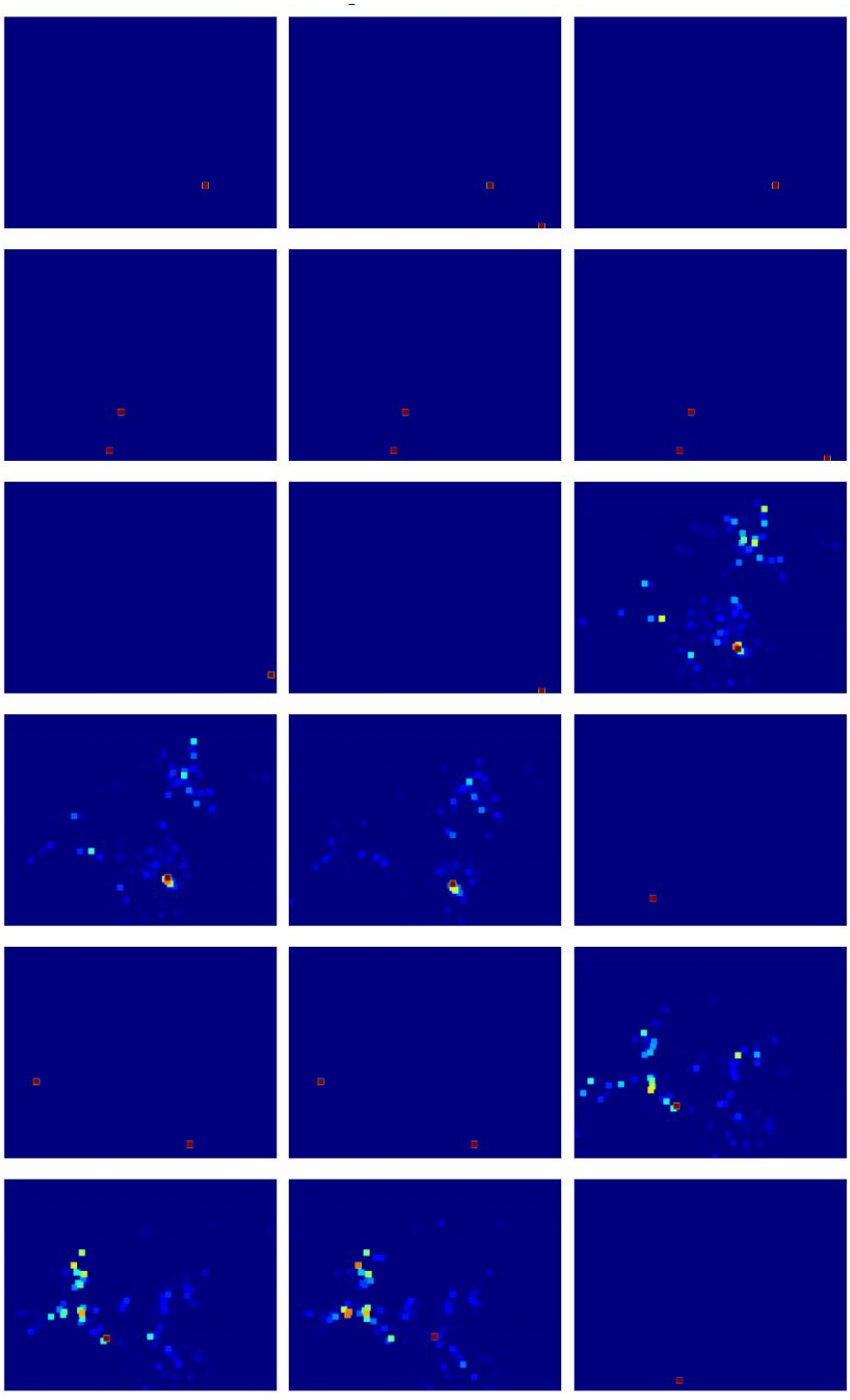


Figure 6.7: Example of ground station data

Utilizing the pandas library in Python, we select the following items:

'Latitude', 'Longitude', and 'Past 3-hour Precipitation'. We read these data, filter and remove any outliers, and perform cubic interpolation based on their coordinates to obtain rainfall data that matches the geographical coordinate distribution of the Guangxi region. The rainfall data for June 16, 2019, is presented as follows:



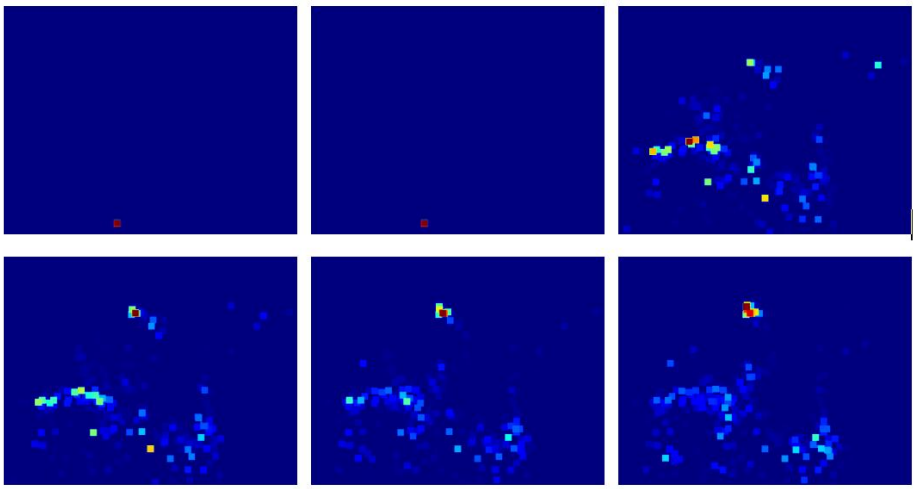
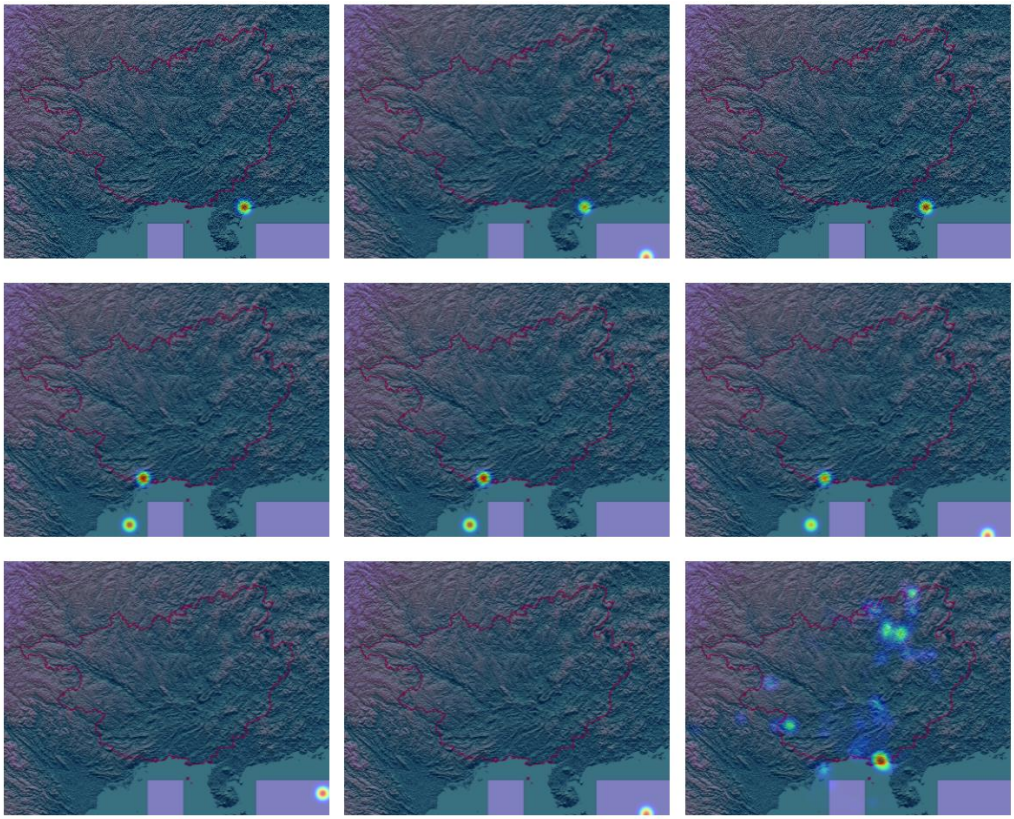


Figure 6.8: Utilization of Ground Station Rainfall Information in Guangxi on June 16, 2019

Using the Informer method adopted in Chapter 5, a forecast was conducted and geospatially matched, yielding the following forecast results(Fig 6.9):



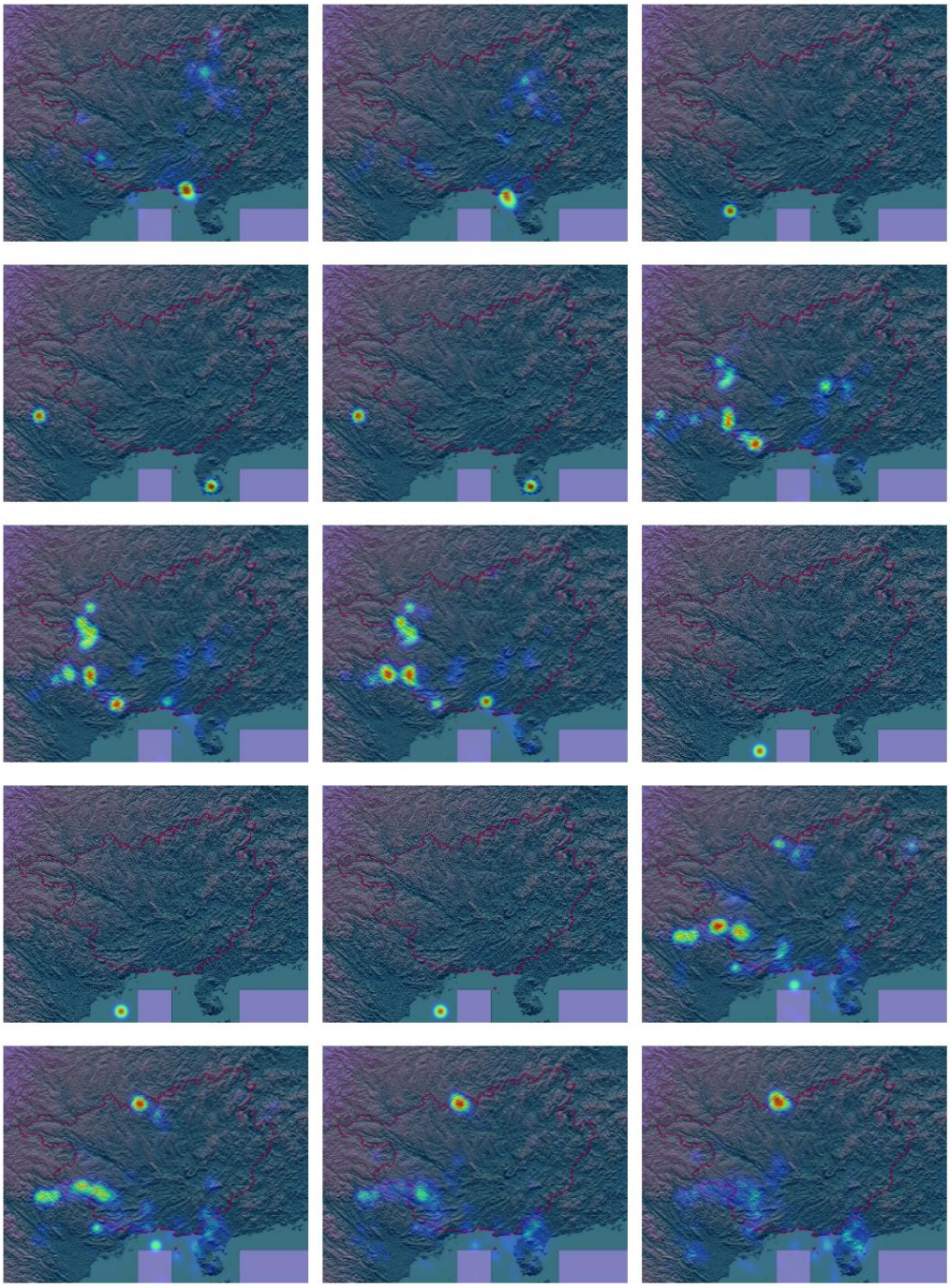


Figure 6.9: Ground Station Rainfall Forecast in Guangxi Utilizing the Informer Model

6.5 Analysis on the results from two data sources

Since this section involves radar echo data, geographical coordinate information for Guangxi region, and ground station coordinate

information, it is necessary to unify all three within the range of Guangxi's geographical coordinates for calculation. Given that the radar echo range (east longitude: 102° to 114°, north latitude: 19° to 28°) exceeds the geographical coordinate range used in this section (east longitude: 104.43333 to 112.06666, north latitude: 20.9 to 26.4), it is essential to crop this data accordingly. Moreover, since the distribution of ground station coordinate information is not spatially equidistant, it is important to retain the ground station information within the geographical range for analysis. Hence, we filter according to the geographical coordinate range and retain the ground station data for further computation. Through the steps above, we can map the ground station data and radar echo data onto the topographical data of Guangxi for further analysis.

In the field of meteorology, the prediction and analysis of rainfall are critical for understanding weather patterns and preparing for potential weather-related events. One of the key methodologies utilized for this purpose is Z-R modeling, which involves leveraging the relationship between the radar reflectivity factor (Z) and the precipitation intensity (R). This relationship, known as the "Z-R" relationship, is an empirical one that has been established through extensive research and observation.

The "Z-R" relationship is fundamental in rainfall prediction as it quantifies the link between what is observed through radar technology and the actual rainfall experienced on the ground. Radar reflectivity factor (Z) is a measure of the intensity of the radar echoes returned from a weather system, which is directly related to the size and concentration of the precipitation particles within the system. Precipitation intensity (R), on the other hand, refers to the rate of rainfall, typically measured in millimeters per hour.

The empirical relationship is typically expressed mathematically, allowing for the conversion of radar reflectivity data into estimates of rainfall intensity. This conversion is essential for various meteorological applications, including short-term rainfall prediction, flood warning

systems, and hydrological modeling. The equation that represents the "Z-R" relationship can vary depending on the region and the specific climatic conditions.

Using the distribution of accumulated rainfall from clouds and ground stations, we perform Z-R modeling, where the "Z-R" relationship is an empirical relationship between the radar reflectivity factor (Z) and the precipitation intensity (R). It describes the correlation between the radar reflectivity factor and the precipitation intensity, typically used to convert observed radar reflectivity factors into estimated rainfall intensity. The "Z-R" relationship is usually expressed in the form of the following equation (6.1):

$$R = a \times Z^b \tag{6.1}$$

In the context of radar meteorology, R represents the precipitation intensity, and Z is the radar reflectivity factor; *a* and *b* are empirical constants. The values of these coefficients are usually adjusted and determined according to different regions and types of precipitation.

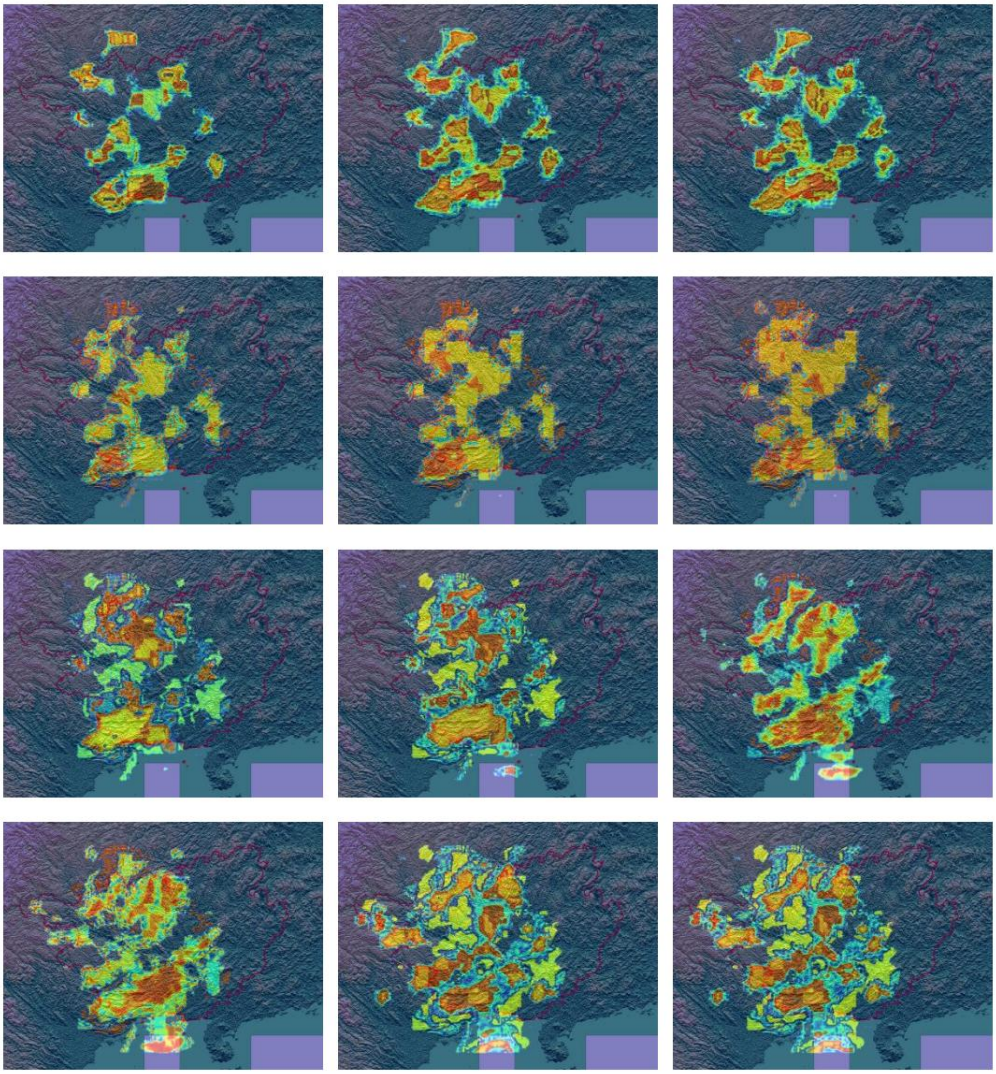
To establish the Z-R relationship based on cloud distribution and ground station information, a polynomial fitting method can be employed to solve for the coefficients a and b. Polynomial fitting is a common data fitting technique that can be used to fit a curve within a dataset. In polynomial fitting, we use a polynomial function to approximate the dataset, which produces a curve similar to the original dataset.

Specifically, we take the datasets of R (the precipitation intensity) and Z (the radar reflectivity factor) as inputs. The objective is to transform the Z-R relationship into logarithmic form, i.e., fit the *log(Z)* to *log(R)* relationship using polynomial fitting to obtain the values for a and b. This method can effectively establish the coefficients within the Z-R relationship.

In this context, Z data comes from actual radar echo measurements, which have been geographically cropped to provide the relevant distribution. R data represents precipitation information, sourced from

the "1-hour precipitation" item at ground stations. The precipitation data is then subjected to interpolation using the station coordinates to generate a fitting for the Z data. By combining both datasets, the polyfit method in the NumPy library can be utilized for polynomial fitting calculations.

Using the above methods, a and b have been fitted to be a=197.062 b=1.660. Thus, the empirical Z-R relationship is formulated as: $R = 197.062 \times Z^{1.660}$. According to this relationship, the following rainfall distribution forecast based on convective cloud is obtained:



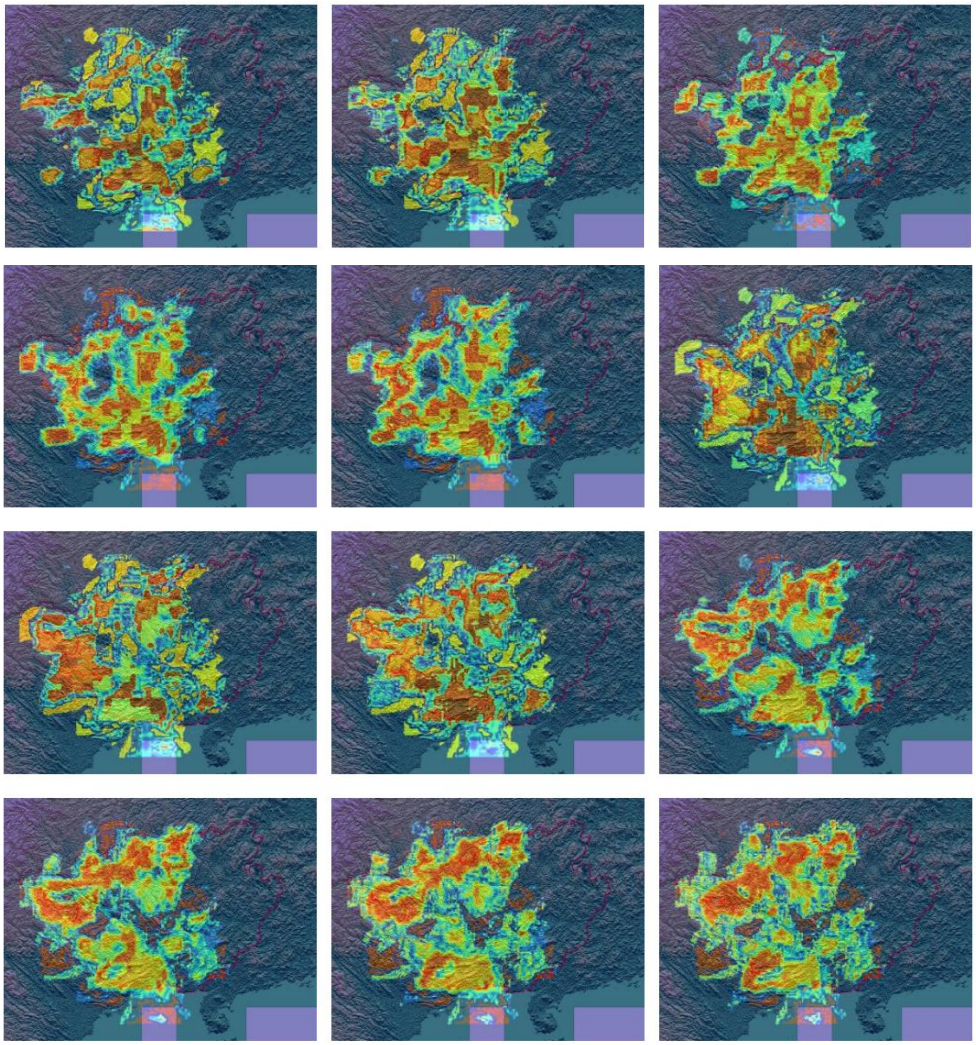


Figure 6.10: Predicted Rainfall Distribution Map for June 16th, 2019

6.6 Accuracy of Rainfall Prediction Based on Base Station Information

Using Mean Squared Error (MSE) to compare the forecasted results with the actual rainfall records, as shown in Table 6.2:

Table 6.2: MSE of forecasted results based on ground station

Time	Error(%)	Time	Error(%)
0	0.06	12	0.12
1	0.09	13	0.12
2	0.06	14	0.17
3	0.12	15	0.22

4	0.12	16	0.29
5	0.15	17	0.06
6	0.06	18	0.06
7	0.04	19	0.06
8	0.15	20	0.23
9	0.09	21	0.20
10	0.07	22	0.12
11	0.06	23	0.15

In the data analysis, the Mean Square Error (MSE) is a crucial measure for assessing the model's performance, especially in regression tasks where the objective is to forecast continuous outcomes. The observed MSE values in the dataset under review exhibit fluctuations, which is a common characteristic in time-series data where the underlying phenomena are subject to various influences over time.

The range of MSE values recorded throughout the entire observation period is indicative of the variability in the model's predictive accuracy. The minimum MSE value observed is 0.04, which suggests that at certain points in time, the model's predictions were very close to the actual values, resulting in a low error. Conversely, the maximum MSE recorded is 0.29, indicating instances where the model's predictions deviated significantly from the actual outcomes. This wide range underscores the dynamic nature of the data and the inherent challenges in achieving consistent accuracy in predictions.

The average MSE of the dataset is calculated to be around 0.12, which provides a central tendency measure of the model's overall performance. However, the standard deviation of the MSE values is found to be 2.08, a figure that is notably high relative to the mean. This high standard deviation indicates a substantial degree of dispersion in the MSE values, suggesting that while the model performs adequately on average, there are considerable variations in its performance across different time points.

In the time series analysis, it is essential to identify and examine points

of deviation from the norm, as these can provide insights into potential anomalies or areas where the model's performance is less reliable. Notably, at time points 15 and 16, the MSE values spike to 0.22 and 0.29, respectively. These values are not only significantly higher than the average but also represent a substantial deviation from the expected performance. Such outliers may warrant a closer examination to determine if they are the result of model limitations, data anomalies, or external factors affecting the predictions.

Further investigation into these outliers could involve a detailed examination of the data and model parameters at these specific time points. It may also be necessary to consider the broader context of the data, such as changes in the input variables or the presence of unforeseen events that could have influenced the model's performance. Understanding the reasons behind these deviations is crucial for improving the model's robustness and reliability.

6.7 Accuracy of Rainfall Prediction Based on Radar Echo Signals

Using Mean Squared Error (MSE) to compare the forecasted results with the actual rainfall records, as shown in Table 6.3:

Table 6.3: MSE of forecasted results based on Radar data

Time	Error(%)	Time	Error(%)
0	0.77	12	0.56
1	0.60	13	0.56
2	0.59	14	0.55
3	0.59	15	0.55
4	0.59	16	0.69
5	0.59	17	0.69
6	0.59	18	0.68
7	0.59	19	0.68
8	0.56	20	0.66
9	0.56	21	0.65
10	0.56	22	0.64
11	0.71	23	0.62

It can be observed from the data that at most time points, the mean

square error (MSE) is concentrated between 0.55 and 0.77, with relatively minor fluctuations overall. The maximum MSE is 0.77, with the minimum being. It can be observed from the data that at most time points, the MSE is concentrated between 0.55 and 0.77, with relatively minor fluctuations overall. The maximum MSE is 0.77, with the minimum being 0.55. The data's average MSE is around 0.62, with a standard deviation of 0.35, indicating that the data's dispersion is relatively small.

In the time series, there are no significant outliers that deviate from the average level, and the MSE for most data points is relatively close to the mean. This may indicate a higher stability in the system, but further analysis is needed to confirm this.

In general, this set of data shows the trend of the MSE over time with small overall fluctuations and no obvious outliers. These analyses help us gain a deeper understanding and evaluation of the data.

6.8 Conclusion

The prediction data from ground stations exhibit significant fluctuations in terms of MSE, which may reflect the instability and uncertainty of ground station forecasts. This necessitates a more in-depth evaluation and analysis. Such instability should be approached with caution, as it may impact real-world decision-making and applications.In contrast, the predictions from radar echo data show much smaller fluctuations overall, indicating higher stability and consistency. This stability could make radar echo data more reliable in practical applications, especially in scenarios that require accurate forecasting.Considering the characteristics of both types of data, it is evident that ground station and radar echo data exhibit different features and performances in predictions. In practical applications, it is necessary to weigh the pros and cons of the two types of data based on the specific situation to support more accurate and reliable decision-making and application requirements.

CHAPTER 7

CONCLUSION

This chapter serves as a comprehensive conclusion to the current doctoral thesis, which can be divided into two principal components: 1. A brief reiteration of the methodologies introduced throughout this research; 2. An exploration of potential avenues for future scholarly pursuits.

7.1 Background

The key distinguishing characteristics between short-term prediction and long-term prediction lie in their time horizons and the nature of the information they rely on. Short-term prediction focuses on forecasting events or trends within a relatively short timeframe, such as minutes or hours. It typically relies on real-time or near-real-time data, emphasizing the immediate and transient dynamics of the system under study. This type of prediction is crucial for applications requiring rapid response and decision-making, such as weather forecasting for the next few hours or traffic flow prediction for the upcoming rush hour.

In contrast, long-term prediction extends its scope to a much longer time period, ranging from days to years or even decades. It often incorporates historical data and considers broader trends, patterns, and underlying factors that influence the system over time. Long-term prediction is essential for strategic planning and decision-making in areas like climate change modeling, economic forecasting, and infrastructure development. The inherent uncertainty and complexity involved in long-term prediction make it more challenging, as it requires accounting for various potential scenarios and the evolution of external factors.

7.2 Summary of the Proposed Methodology

In this thesis, the fusion of meteorological science with the cutting-edge field of artificial intelligence has been meticulously examined, specifically through the application of Transformer and Informer models for the forecasting of near-term precipitation based on data collected from ground observations. An extensive review of existing literature was conducted to identify these models, which have garnered recognition for their adeptness in predicting a diverse array of meteorological variables. The methodology employed in this study included advanced noise reduction techniques and the pinpointing of meteorological events with high correlation, both of which significantly bolstered the models' capacity for data representation. The experimental outcomes were highly promising, with the Transformer model achieving a Mean Absolute Error (MAE) of 0.0174 and an R-squared (R^2) value of 0.034, while the Informer model outperformed it with an MAE of 0.0077 mm and an R^2 value of 0.0129. These results underscore the profound efficacy of integrating meteorological knowledge with AI in the domain of short-term precipitation forecasting.

Furthermore, the innovative GraphAT-NET approach, which amalgamates Convolutional Neural Networks (CNN), Graph Convolutional Networks (GCN), Recurrent Neural Networks (RNN), an attention mechanism, and an advanced loss function, has been demonstrated to surpass other modern methods in its performance on both the moving MNIST dataset and real-world radar echo data. On the moving MNIST dataset, GraphAT-NET achieved a Mean Squared Error (MSE) of 0.0123 and a Structural Similarity Index (SSIM) of 0.0112, surpassing the performance of established methods such as Smatunet (MSE: 0.00358, SSIM: 0.000946) and TrajGRU (MSE: 0.0414, SSIM: 0.0109). When applied to real-world radar echo data, GraphAT-NET delivered an MSE of 1.98E-7 and an SSIM of 99.9%, outperforming ConvLSTM (MSE: 2.2E-7, SSIM: 95%) and Seresunet (MSE: 1.21E-6, SSIM: 99.4%). The visual outcomes of the model's application corroborated its effectiveness, displaying a significant reduction in MSE

and an enhancement in SSIM. An ablation study was conducted to dissect the contributions of the GCN and ECA modules to the model's predictive accuracy. It was discovered that the removal of the GCN module resulted in a 10.9% decrease in MSE, while the removal of the ECA module led to a 19.28% reduction in MSE.

The implications of these findings are profound, suggesting that the GraphAT-NET method holds substantial potential for advancing the prediction of cumulonimbus cloud distribution, a critical precursor in the rainfall nowcasting process. However, it is important to acknowledge that predicting cumulonimbus clouds is merely the first step in this process. Future research endeavors will aim to establish a more comprehensive connection between radar echo data and actual rainfall information, facilitating end-to-end predictions that are more accurate and reliable.

In the field of prediction data, ground stations have exhibited considerable variability in MSE, indicating a degree of instability and uncertainty. Specifically, the MSE for ground station data ranged from 0.04 to 0.29, suggesting that while some stations provide stable predictions, others may not. Conversely, radar echo data, despite being more consistent, showed an MSE range from 0.55 to 0.77, which, while lower than that of ground stations, still indicates a certain level of unreliability in forecasts. Given the distinct characteristics and performance levels of both ground station and radar echo data, it is imperative to conduct a thorough evaluation of their respective strengths and weaknesses. This assessment is essential for making informed decisions and for the effective application of these data sources in practical meteorological forecasting scenarios.

The integration of AI with meteorology, as demonstrated through the methodologies and models explored in this thesis, opens up opportunities for future research. The potential for further refinement of these models, the exploration of additional meteorological variables, and the development of more advanced data processing techniques are all areas ripe for investigation. As the field of AI continues to evolve, so too

does its capacity to enhance our understanding and prediction of complex meteorological phenomena, paving the way for more accurate and timely weather forecasts that can benefit society at large.

7.3 Discussion

In this thesis, different datasets were employed across various chapters due to the distinct prediction methods utilized. Each prediction approach necessitates specific types of data to effectively capture the relevant features and dynamics of the system being predicted. Consequently, the evaluation metrics also vary accordingly.

Drawing from the reference literature, mainstream evaluation methods were selected to ensure the scientific rigor and consistency of the assessments. For image-based predictions, the Structural Similarity Index Measure (SSIM) was adopted as it provides a more comprehensive evaluation of image quality compared to traditional metrics like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR). SSIM takes into account the luminance, contrast, and structural information of the images, making it particularly suitable for assessing the similarity between the predicted and actual images. This choice aligns with the scientific standards for image quality assessment, allowing for a more accurate and reliable evaluation of the prediction performance.

7.4 Contributions

The research endeavor investigates the application of deep learning techniques for predicting meteorological data, drawing on insights from a multitude of sources. In the field of radar data for meteorological prediction, the research goes beyond traditional methods by integrating the dimension of time with detailed semantic information that is usually extracted from images. This integration is crucial as it allows for a more detailed understanding of the complex patterns found in meteorological data. To achieve this, the study uses Graph Convolutional Networks (GCN), which are good at broad application, thus enabling the model to

identify and predict meteorological phenomena with greater accuracy.

The thesis introduces a new predictive model, GraphAT-NET, which is based on graph neural networks and includes a channel attention mechanism. This model, short for graph-Attention-trajgru, is designed to improve prediction results by getting key point information from the feature map of TRAJGRU, a type of neural network that is effective in arranging sequences. By building a Graph structure, the model captures the complex relationships within the data. In addition, the use of an ECA (Enhanced Channel Attention) mechanism allows for the dynamic re-distribution of weights to the graph information, making sure that the model focuses on the most relevant features for accurate predictions.

In the field of ground-based meteorological prediction, the thesis faces the challenges that trouble traditional forecasting methods. These challenges include difficulties in making accurate predictions, achieving high accuracy, and managing the calculation needs due to the large amount of meteorological data. To lessen these issues, the study starts by removing noise from the raw meteorological data, a critical step in ensuring the quality of the input data. Then, the selection of important meteorological elements is made easier using the Pearson correlation coefficient algorithm, a statistical method that measures the linear relationship between two variables. This selection process specifically aims at meteorological elements that are related to the height of rainfall, thereby decreasing the impact of other uncertain factors on the observational data.

Building on this foundation, the study creates networks using the latest Transformer and Informer technologies. These technologies allow for the exploration of deeper knowledge and more representative expressions of the meteorological data, leading to a more complete understanding of the basic patterns. The Transformer model, known for its ability to handle long-range dependencies in data, and the Informer, which is good at processing sequential data with irregular intervals, are used to increase the model's predictive power.

In the final analysis, the thesis presents a side-by-side comparison of

the experimental results of the two models. This comparison is important as it provides insights into the relative strengths and weaknesses of each model, offering a basis for further refinement and optimization. The results show the potential of deep learning techniques in meteorological forecasting, demonstrating their ability to handle the complexity and variability of meteorological data.

7.5 Future work

Future research directions include several key areas for further exploration and advancement. Building upon the successful integration of meteorology and artificial intelligence using Transformer and Informer models for near-term precipitation forecasting, future studies can focus on refining these models, exploring additional AI algorithms, and incorporating more advanced meteorological data sources to enhance forecasting accuracy. Additionally, the optimization and enhancement of the GraphAT-NET method for cloud distribution prediction hold promise for improving cumulonimbus cloud forecasting precision. Further investigations can delve into refining model architectures, preprocessing techniques, and the integration of additional atmospheric variables. Connecting radar echo data with actual rainfall information provides an avenue for advancing end-to-end predictions. Future endeavors can focus on developing robust methodologies for linking these data sources effectively, optimizing data fusion techniques, and leveraging combined information for more accurate precipitation forecasts. Lastly, systematic evaluations comparing the strengths and weaknesses of ground station versus radar echo data under various scenarios will aid in informed decision-making for selecting the most suitable data sources for specific forecasting tasks and applications.

The trend of utilizing multi-modal data for rainfall prediction is on the rise. By integrating data from diverse sources, including radar, satellite

imagery, ground-based sensors, and meteorological models, a more holistic understanding of the atmospheric conditions and dynamics that precipitate rainfall events can be achieved. This multi-modal approach holds the potential to markedly improve the accuracy and reliability of rainfall predictions.

Looking ahead, there will be efforts to establish more unified evaluation standards to assess the performance of different multi-modal tasks. Developing a cohesive framework that accounts for the unique characteristics and contributions of each modality will enable a fair and comprehensive comparison of multi-modal prediction models. This will facilitate the identification of the most effective approaches and drive further advancements in the field of rainfall prediction and other related applications.

REFERENCES

[1] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 802–810.

[2] C. I. Mavrogiannis and R. A. Knepper, “Multi-agent trajectory prediction and generation with topological invariants enforced by hamiltonian dynamics,” in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 744–761.

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference*, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015.

[4] K. Trebing, T. Stanczyk, and S. Mehrkanoon, “SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture,” *Pattern Recognit*. Lett., vol. 145, pp. 178–186, 2021.

[5] Song, Kuan, et al. "Deep learning prediction of incoming rainfalls: An operational service for the city of Beijing China." *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2019.

[6] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[7] K. Tan and D. Wang, "Learning Complex Spectral Mapping With Gated Convolutional Recurrent Networks for Monaural Speech Enhancement," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 380-390, 2020.

[8] T.-L., Dinh , et al. "A new approach for quantitative precipitation estimation from radar reflectivity using a gated recurrent unit network." *Journal of Hydrology* (2023):624.

[9] Kimura, R. (2002). Numerical weather prediction. *Journal of Wind Engineering and Industrial Aerodynamics*, *90*(12-15), 1403-1414.

[10] Fente, D. N., & Singh, D. K. (2018, April). Weather forecasting using artificial neural network. In *2018 second international conference on inventive communication and computational technologies (ICICCT)* (pp. 1757-1761). IEEE.

[11] Moraux, A., Dewitte, S., Cornelis, B., & Munteanu, A. (2021). A deep learning multimodal method for precipitation estimation. *Remote Sensing*, *13*(16), 3278.

[12] Yang, C., Mei, H., & Eisner, J. (2022, April). Transformer embeddings of irregularly spaced events and their participants. In *Proceedings of the tenth international conference on learning representations (ICLR)*.

[13] D. Nam, K. Udo, and A. Mano, "Uncertainty assessment for short-term flood forecasts in central vietnam," in *6th International Conference on River Basin Management including all aspects of Hydrology, Ecology, Environmental Management, Flood Plains and wetlands*, RM 2011. WITPress, 2011, pp. 117–130.

[14] E. J. Becker, H. Van Den Dool, and M. Peña, "Short-term climate extremes:Prediction skill and predictability," *Journal of climate*, vol. 26, no. 2, pp. 512–531, 2013.

[15] M. Ripoche, C. Campagna, A. Ludwig, N. H. Ogden, and P. A. Leighton, "Shortterm forecasting of daily abundance of west nile virus vectors culex pipiensrestuans (diptera: Culicidae) and aedes vexans based on weather conditions in southern québec (canada)," *Journal of medical entomology*, vol. 56, no. 3, pp. 859–872, 2019.

[16] J. Hong, S. Lee, J. H. Bae, J. Lee, W. J. Park, D. Lee, J. Kim, and K. J. Lim, "Development and evaluation of the combined machine learning models for the prediction of dam inflow," *Water*, vol. 12, no. 10, p. 2927, 2020.

[17] L. Jia, Q. Qingtai, L. Chuanzhe, J. Yufei, W. Wei, and Y. Fuliang, "Advances of precipitation nowcasting and its application in hydrological forecasting," , vol. 31, no. 1, pp. 129–142, 2020.

[18] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "Scent: A new precipitation nowcasting method based on sparse correspondence and deep neural network," *Neurocomputing*, vol. 448, pp. 10–20, 2021.

[19] Y. Zhang, J. Han, G. Pan, Y. Xu, and F.Wang, "A multi-stage predicting methodology based on data decomposition and error correction for ultra-short-term wind energy prediction," *Journal of Cleaner Production*, vol. 292, p. 125981, 2021.

[20] J. Lee, J. Lee, M. Lee, M. Lee, Y. Kim, J. Hyung, K. Kim, Y. Cha, and J. Koo, "Development of a short-term water quality prediction model for urban rivers using real-time water quality data," *Water Supply*, vol. 22, no. 4, pp. 4082–4097, 2022.

[21] S. Sunardi, A. Yudhana, and G. Z. Muflih, "Sistem prediksi curah hujan bulanan menggunakan jaringan saraf tiruan backpropagation," *JSINBIS (Jurnal Sistem Informasi Bisnis)*, vol. 10, no. 2, pp. 155–162, 2020.

[22] M. Kulakarni, M. Nemichandrappa, G. S. Reddy, P. S. Kulkarni, and C. Ramachandra, "Factors affecting groundwater level fluctuation: A case study at manvi," *IJCS*, vol. 8, no. 2, pp. 803–806, 2020.

[23] Q. Li, J. Wang, S. Yang, F. Wang, J. Wu, and Y. Hu, "Sub-seasonal

prediction of rainfall over the south china sea and its surrounding areas during spring–summer transitional season," *International Journal of Climatology*, vol. 40, no. 10, pp. 4326–4346, 2020.

[24] V. Q. Tran and I. Prakash, "Prediction of soil loss due to erosion using support vector machine model," *Science of the Earth*, vol. 42, no. 3, pp. 247–254, 2020.

[25] B. T. Pham, L. M. Le, T.-T. Le, K.-T. T. Bui, V. M. Le, H.-B. Ly, and I. Prakash, "Development of advanced artificial intelligence models for daily rainfall prediction," *Atmospheric Research*, vol. 237, p. 104845, 2020.

[26] L. Brocca, C. Massari, T. Pellarin, P. Filippucci, L. Ciabatta, S. Camici, Y. H. Kerr, and D. Fernández-Prieto, "River flow prediction in data scarce regions: soil moisture integrated satellite rainfall products outperform rain gauge observations in west africa," *Scientific Report*s, vol. 10, no. 1, p. 12517, 2020.

[27] D. Naidu, B. Majhi, and S. K. Chandniha, "Development of rainfall prediction models using machine learning approaches for different agro-climatic zones," in *Handbook of Research on Automated Feature Engineering and Advanced Applications in Data Science*. IGI Global, 2021, pp. 72–94.

[28] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 802–810.

[29] Qiu, M., Zhao, P., Zhang, K., Huang, J., Shi, X., Wang, X., & Chu, W. (2017, November). A short-term rainfall prediction model using multi-task convolutional neural networks. In *2017 IEEE international conference on data mining (ICDM)* (pp. 395-404). IEEE.

[30] Akbari Asanjan, A., Yang, T., Hsu, K., Sorooshian, S., Lin, J., & Peng, Q. (2018). Short-term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks. *Journal of Geophysical Research: Atmospheres*, *123*(22), 12-543.

[31] Y. M. Souto, F. Porto, A. M. de Carvalho Moura, and E. Bezerra, "A spatio-temporal ensemble approach to rainfall forecasting," in *2018 International Joint Conference on Neural Networks*, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018. IEEE, 2018, pp. 1–8.

[32] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Deep learning for precipitation nowcasting: A benchmark and A new model," in Advances in *Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5617–5627.

[33] L. Tian, X. Li, Y. Ye, P. Xie, and Y. Li, "A generative adversarial gated

recurrent unit model for precipitation nowcasting," *IEEE Geosci. Remote. Sens. Lett*., vol. 17, no. 4, pp. 601–605, 2020.

[34] P. Xie, X. Li, X. Ji, X. Chen, Y. Chen, J. Liu, and Y. Ye, "An energy-based generative adversarial forecaster for radar echo map extrapolation," *IEEE Geosci. Remote. Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[35] T. Yu, Q. Kuang, and R. Yang, "Atmconvgru for weather forecasting," *IEEE Geosci. Remote. Sens. Lett*., vol. 19, pp. 1–5, 2022.

[36] W. He, T. S. Xiong, H. Wang, J. X. He, X. Y. Ren, Y. L. Yan, and L. Y. Tan, "Radar echo spatio-temporal sequence prediction using an improved convgru deep learning model," *ATMOSPHERE*, vol. 13, no. 1, JAN, 88.

[37] K. Trebing, T. Stanczyk, and S. Mehrkanoon, "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture," *Pattern Recognit. Lett*., vol. 145, pp. 178–186, 2021.

[38] K. Song, X. Yu, Z. Gu, W. Zhang, G. Yang, Q. Wang, C. Xu, J. Liu, W. Liu, C. Shi, Y. Wang, and G. Zhang, "Deep learning prediction of incoming rainfalls: An operational service for the city of beijing china," in *2019 International Conference on Data Mining Workshops*, ICDM Workshops 2019, Beijing, China, November 8-11, 2019. IEEE, 2019, pp. 180–185.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations*, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[40] Y.Wu, Y. Tang, X. Yang,W. Zhang, and G. Zhang, "Graph convolutional regression networks for quantitative precipitation estimation," *IEEE Geosci. Remote. Sens. Lett*., vol. 18, no. 7, pp. 1124–1128, 2021.

[41] M. Ghamariadyan and M. A. Imteaz, "A wavelet artificial neural network method for medium-term rainfall prediction in queensland (australia) and the comparisons with conventional methods," *International Journal of Climatology*, vol. 41, no. S1, pp. E1396–E1416, 2021.

[42] X. Wang, L. Liu, P. Hu, Z. Gong, and G. Feng, "Improving the prediction skill for china summer rainfall through correcting leading modes in beijing climate center's climate system model," *International Journal of Climatology*, vol. 39, no. 11, pp. 4329 – 4339, 2019.

[43] J. B. Ekanayake, T. Dananjali et al., "Variability in rainfall prediction quality over time," *Journal of Agricultural Informatics* (ISSN 2061-862X), vol. 11, no. 1, pp. 24–33, 2020.

[44] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," arXiv preprint arXiv:2202.07125, 2022.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," CoRR, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[46] H. Zhang, Y. Xia, T. Yan, and G. Liu, "Unsupervised anomaly detection in

multivariate time series through transformer-based variational autoencoder," in *2021 33rd Chinese Control and Decision Conference (CCDC)*. IEEE, 2021, pp. 281–286.

[47] L.-n. Zhang, J.-w. Liu, Z.-y. Song, and X. Zuo, "Temporal attention augmented transformer hawkes process," *Neural Computing and Applications*, vol. 34, no. 5, pp. 3795–3809, 2022.

[48] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.

[49] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," arXiv preprint arXiv:2001.08317, 2020.

[50] Z. Gao, X. Shi, H. Wang, Y. Zhu, Y. Wang, M. Li, and D.-Y. Yeung, "Earthformer: Exploring space-time transformers for earth system forecasting," arXiv preprint arXiv:2207.05833, 2022.

[51] H. Yin, Z. Guo, X. Zhang, J. Chen, and Y. Zhang, "Rr-former: Rainfall-runoff modeling based on transformer," *Journal of Hydrology*, vol. 609, p. 127781, 2022.

[52] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[53] C. I. Mavrogiannis and R. A. Knepper, "Multi-agent trajectory prediction and generation with topological invariants enforced by hamiltonian dynamics," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 744–761.

[54] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[55] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[56] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, andW. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 106 ‑ 11 115.

[57] H.-K. Wang, K. Song, and Y. Cheng, "A hybrid forecasting model based on cnn and informer for short-term wind power," *Frontiers in Energy Research*, p. 1041, 2022.

[58] W. Li, H. Fu, Z. Han, X. Zhang, and H. Jin, "Intelligent tool wear prediction based on informer encoder and stacked bidirectional gated recurrent unit," *Robotics and Computer-Integrated Manufacturing*, vol. 77,

p. 102368, 2022.
[59] Y. Qian, L. Tian, B. Zhai, S. Zhang, and R. Wu, “Informer-wgan: High missing rate time series imputation based on adversarial training and a self-attention mechanism,” *Algorithms*, vol. 15, no. 7, p. 252, 2022.
[60] C. G. Collier, “Developments in radar and remote-sensing methods for measuring and forecasting rainfall,” Phil. Trans. R. Soc. A, vol. 360, p. 1345–1361, 2002.
[61] N. R. DALEZIOS, “Digital processing of weather radar signals for rainfall estimation,” *International Journal of Remote Sensing*, vol. 11, no. 9, pp. 1561–1569, 1990.
[62] S. Michaelides, V. Levizzani, E. Anagnostou, P. Bauer, T. Kasparis, and J. Lane, “Precipitation: Measurement, remote sensing, climatology and modeling,” *Atmospheric Research*, vol. 94, no. 4, pp. 512–533, 2009.
[63] E. Ghaemi, M. Kavianpour, S. Moazami, Y. Hong, and H. Ayat, “Uncertainty analysis of radar rainfall estimates over two different climates in iran,” *International Journal of Remote Sensing*, vol. 38, no. 18, pp. 5106–5126, 2017.
[64] W. Peng, S. Alan, S. Lao, O. Edel, L. Yunxiang, and O. Noel, “Short-term rainfall nowcasting: Using rainfall radar imaging,” in In: *Eurographics Ireland 2009: The 9th Irish Workshop on Computer Graphics*, 11 December 2009, Dublin, Ireland., 01 2009.
[65] G. Scarchilli, V. Gorgucci, V. Chandrasekar, and A. Dobaie, “Self-consistency of polarization diversity measurement of rainfall,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 1, pp. 22–26, 1996.
[66] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* , pp. 11534-11542.2020.
[67] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model,” in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 26-30, 2010, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.
[68] T. Wu, Y. Wan, W. Wo, and L. Leng, “Design and application of radar reflectivity quality control algorithm in swan,” *Meterological science and technology*, vol. 41, no. 5, p. 809, 10 2013.
[69] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? −˘ Arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.
[70] Z.Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans.*

Image Process., vol. 13, no. 4, pp. 600–612, 2004.
[71] K. Ioannou, D. Karampatzakis, P. Amanatidis, V. Aggelopoulos, and I. Karmiris, "Low-cost automatic weather stations in the internet of things," *Information*, vol. 12, no. 4, p. 146, 2021.
[72] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.
[73] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," arXiv preprint arXiv:1801.01078, 2017.
[74] R. Liu, H. Zhou, D. Li, L. Zeng, and P. Xu, "Evaluation of artificial precipitation enhancement using unet-gru algorithm for rainfall estimation," *Water,* vol. 15, no. 8, p. 1585, 2023.
[75] H. Zhang, Y. Xia, T. Yan, and G. Liu, "Unsupervised anomaly detection in multivariate time series through transformer-based variational autoencoder," in *2021 33rd Chinese Control and Decision Conference (CCDC). IEEE*, 2021, pp. 281–286.
[76] L.-n. Zhang, J.-w. Liu, Z.-y. Song, and X. Zuo, "Temporal attention augmented transformer hawkes process," *Neural Computing and Applications*, pp. 1–15, 2022.
[77] U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer based deep intelligent contextual embedding for twitter sentiment analysis," *Future Generation Computer Systems*, vol. 113, pp. 58–69, 2020.
[78] H. Meng, Y. Zhang, Y. Li, and H. Zhao, "Spacecraft anomaly detection via transformer reconstruction error," in *Proceedings of the International Conference on Aerospace System Science and Engineering* 2019. Springer, 2020, pp. 351–362.
[79] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song, "Gated transformer networks for multivariate time series classification," arXiv preprint arXiv:2103.14438, 2021.
[80] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.
[81] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," arXiv preprint arXiv:2001.08317, 2020.
[82] Z. Gao, X. Shi, H. Wang, Y. Zhu, Y. B. Wang, M. Li, and D.-Y. Yeung, "Earthformer: Exploring space-time transformers for earth system forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 390–25 403, 2022.
[83] H. Yin, Z. Guo, X. Zhang, J. Chen, and Y. Zhang, "Rr-former: Rainfallrunoff modeling based on transformer," *Journal of Hydrology*, vol.

609, p. 127781, 2022.
[84] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, “Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting,” *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.
[85] C. I. Mavrogiannis and R. A. Knepper, “Multi-agent trajectory prediction and generation with topological invariants enforced by hamiltonian dynamics,” in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics* 13. Springer, 2020, pp. 744–761.
[86] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.
[87] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, “Pre-trained image processing transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.
[88] Kang, Jeon-Seong, et al. "Neural architecture search survey: A computer vision perspective." *Sensors* 23.3 (2023): 1713.
[89] H.-K. Wang, K. Song, and Y. Cheng, “A hybrid forecasting model based on cnn and informer for short-term wind power,” *Frontiers in Energy Research*, vol. 9, p. 1041, 2022.
[90] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.
[91] W. Li, H. Fu, Z. Han, X. Zhang, and H. Jin, “Intelligent tool wear prediction based on informer encoder and stacked bidirectional gated recurrent unit,” *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102368, 2022.
[92] Y. Qian, L. Tian, B. Zhai, S. Zhang, and R. Wu, “Informer-wgan: High missing rate time series imputation based on adversarial training and a self-attention mechanism,” *Algorithms*, vol. 15, no. 7, p. 252, 2022.
[93] Alzubaidi, Laith, et al. "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications." *Journal of Big Data* 10.1 (2023): 46.