TOWARDS ACCELERATING & INTERPRETABLE WEATHER FORECASTING USING DEEP NETWORK LEARNING

A PROJECT REPORT

Submitted by

Durga Prasath J - RA2011003020283
Pavithra M G - RA2011003020238
Sudarsan S - RA2011003020256

Under the guidance of

Dr.S.Visnu Dharshini

(Assistant professor)

Department of Computer Science and Engineering

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM,CHENNAI-600089

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University U/S3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "TOWARDS ACCELERATING & INTERPRETABLE WEATHER FORECASTING USING DEEP NETWORK LEARNING" is the bonafide work of DURGA PRASATH J [REG NO: RA2011003020283], PAVITRA M G [REG NO: RA2011003020238], SUDARSAN S [REG NO:RA2011003020256] who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

Dr.S.Visnu Dharshini

Assistant professor

Computer Science and Engineering,
SRM Institute of Science and Technology,
Ramapuram,Chennai.

SIGNATURE

Dr.K.RAJA,M.E.,Ph.D.,

Professor and Head

Computer Science and Engineering,
SRM Institute of Science and Technology,
Ramapuram,Chennai.

Submitted for the project viva held on_____ at SRM Institute of Science and Technology, Ramapuram, Chennai-600089.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI-89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**TOWARDS ACCELERATING & INTERPRETABLE WEATHER FORECASTING USING DEEP NETWORK LEARNING**" has been carried out by **DURGA PRASATH J [REG NO:RA2011003020283], PAVITRA M G [REG NO: RA2011003020238], SUDARSAN S [REG NO:RA2011003020256]** at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Dr.S.Visnu Darshni**, assistant proffesor Department of Computer Science and Engineering.

Place: Chennai.

Date: **DURGA PRASATH J**

PAVITHRA M G

SUDARSAN S

SRM Institute of Science & Technology
Department of Computer Science and Engineering

Own Work Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech Computer Science and Engineering.

Student Name : DurgaPrasath J, Pavithra MG, SudarsanS.

Registration Number : RA2011003020283, RA2011002030238, RA2011002030256.

Title of Work : Towards Accelerating & Interpretable Weather Forecasting using Deep Network Learning.

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate.
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present.
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group. RA2011002030283 RA2011003020238 RA2011003020256

ACKNOWLEDGEMENT

We place on record our deep sense of gratitude to our lionized Chairman **Dr.R.SHIVAKUMAR** for providing us with the requisite infrastructure throughout the course.

We take the opportunity to extend our hearty and sincere thanks to our Dean, **Dr.M.MURALI KRISHNA, B.E, M.Tech, PhD., MISTE, FIE, C.Engg.,** for maneuvering us into accomplishing the project.

We take the privilege to extend our hearty and sincere guidance to the Professor and Head of the Department, **Dr.K.RAJA, M.E., PhD.,** for his suggestions, support andencouragement towards the completion of the project with perfection.

We express our hearty and sincere thanks to our Project Coordinator **Dr.S. SATHYA PRIYA, M.E., PhD., Associate Professor**for her fortification. We express our hearty and sincere thanks to our guide **Dr. S.VISNU DHARSINI, M.E., PhD., Associate Professor,** Computer Science and Engineering Department for his sustained encouragement, consecutive criticism and constant guidance throughout this project work.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering Department of SRM Institute of Science and Technology, Ramapuram Campus, who provided necessary resources for our project.

DURGA PRASATH J

PAVITHRA MG

SUDARSAN S

ABSTRACT

Global deep-learning weather prediction models have recently been shown to produce forecasts that rival those from physics-based models run at operational centers. It is unclear whether these models have encoded atmospheric dynamics, or simply pattern matching that produces the smallest forecast error. Answering this question is crucial to establishing the utility of these models as tools for basic science. We propose a novel, neural network-based method, which produces forecasts for all locations and lead times, jointly. To relax the distributional assumption made by many post-processing methods, our approach incorporates normalizing flows as flexible parametric distribution estimators. This enables us to model varying forecast distributions in a mathematically exact way. Deep learning has recently gained immense popularity in the Earth sciences as it enables us to formulate purely data-driven models of complex Earth system processes. Deep learning-based weather prediction (DLWP) models have made significant progress in the last few years, achieving forecast skills comparable to established numerical weather prediction (NWP) models with comparatively lesser computational costs. In order to train accurate, reliable, and tractable DLWP models with several millions of parameters, the model design needs to incorporate suitable inductive biases that encode structural assumptions about the data and modelled processes. When chosen appropriately, these biases enable faster learning and better generalisation to unseen data. Although inductive biases play a crucial role in successful DLWP models, they are often not stated explicitly and how they contribute to model performance remains unclear. Here, we review and analyse the inductive biases of state-of the-art DLWP models, involving a deeper look at five key design elements: input data, forecasting objective, loss components, layered design of the deep learning architectures, and optimisation methods. We show how the design choices made in each of the five design elements relate to structural assumptions.

TABLE OF CONTENTS

LIST OF FIGURES

List of Abbreviation

ML -Machine Learning

EL -Ensemble Learning

RF -Randon Forest

AB- Ada Boosting

NWP - Numerical Weather Prediction

EDA - Exploratory Data Analysis

DNN - Deep Neural Network

CNN - Convolutional Neural Network

RNN - Recurrent Neural Network

LSTM - Long Short-Term Memory

ML - Machine Learning

CHAPTER 1

INTRODUCTION

1.1 Introduction

Numerical weather prediction (NWP) is the dominant approach for weather forecasting. A weather forecast is the result of the numerical integration of partial differential equations starting from the best estimate of the current state of the Earth System. The idea that the physical laws of fluid dynamics and thermodynamics can be used to predict the state of the atmosphere dates back to the pioneering works. In a standard NWP framework, a weather prediction results from a deductive inference: a deterministic forecast is derived using the laws of physics starting from the best possible initial conditions, derived by optimally combining earth system observations and shortrange forecasts through data assimilation. However, our ability to perfectly know the initial conditions and numerically resolve the equations is limited. Hence, ensemble forecasting is used to account for uncertainty in both the initial conditions and the forecasting model, with the resulting ensemble forecast serving as a basis for probabilistic forecasting.

Numerical Weather Prediction (NWP) primarily combines Navier-Stokes equations governing fluid dynamics with principles of mass and momentum conservation, the idealised gas law, and the first law of thermodynamics. The result is a system of nonlinear, partial differential equations that model the physical processes unfolding in the atmosphere. To generate weather forecasts, given initial conditions that accurately describe the state of the atmosphere, the set of equations are numerically integrated in time on discretised, three-dimensional grid structures. State-of-the-art operational NWP models cover a wide range of spatial and temporal forecast scales: from local, sub-hourly kilometre-scales, to mid-term, weekly-scales, to global seasonal scales. In all cases, though, inherent nonlinearities of atmospheric dynamics impose a hard limit on how far ahead in time we can predict the weather. Variations in performance of different NWP models depend on various design choices, of which the two most relevant aspects are how subgrid processes are modelled and how the data is assimilated.

Accurate weather forecasts play an important role in different aspects of modern society including agriculture, infrastructure, transportation, aviation, and disaster mitigation. Ongoing climate change in particular increases the likelihood of extreme weather patterns on all scales. This creates the immediate demand for new models that can cope with these more erratic and extreme weather patterns in order to provide reliable forecasts for the public welfare in the next decades. Until now, operational weather

forecast models, which regularly issue predictions from minutes to days to weeks, rely on numerical weather predictions. NWP models are constructed by experts from physics, geoscience, and meteorology as well as from data analysis, statistics, and computer science to perform well. Over the past three decades, NWP models have made significant improvements in providing highly accurate weather forecasts, particularly up to weekly timescales. For instance forecast skills, which quantify improvements made over a baseline, for a 5-day forecast increased from slightly over 70% in the mid-nineties to over 90% in 2025. Termed the quiet revolution because of its incremental nature, the progress in NWP was made possible mainly by improvements in modelling subgrid processes, data assimilation and resulting model initialisation, and ensemble prediction. The gains in these areas crucially depended on concomitant advancements in computational technologies such as the availability of faster and more efficient computing chips, which has in recent years shown signs of slowing down, and has sparked new discussions of how NWP models can be adapted to the road ahead. Purely data-driven weather forecast models powered by deep learning are on the verge of triggering a major breakthrough in weather forecasting. An early study in this direction used convolutional Long Short Term Memory (LSTM) networks for precipitation nowcasting. Early attempts to model global atmospheric states include a multilayer perceptron and a convolutional neural network (CNN) architecture (with an LSTM variant) to forecast mid-tropospheric geopotential height, as well as an autoencoder setup to forecast the weather within a simple climate model. Many similar studies that followed over the last four years have demonstrated the potential of using deep learning models for weather prediction (DLWPs). Deep learning (DL) refers to artificial neural networks that contain many layers, thus transforming input data via abstract, compositional, and hierarchical representations into target predictions. Each layer in a neural network architecture consists of computational units called neurons, which receive information from other layers, combine this information, for example, via a weighted sum, and produce a consequent output with a typically non-linear but differentiable activation function. During training, the weights self-organise via error backpropagation such that the neurons and layers encode latent features for minimising a pre-specified, taskspecific performance criterion. Training is complete when no further performance gain is made or when overfitting is registered, which is the case when performance starts stalling or even declining on an evaluation set while it is still improving on the training set. The trained model can then be used with its learned set of weights to generate predictions about new data instances.

One fundamental reason why it is attractive to think of using DL for predicting the weather is its apparent simplicity we do not need to fully understand nor explicitly model the physics of the atmosphere as thoroughly as in NWPs. NWP models themselves are highly complicated objects with many interdependent handcrafted modules and tuning parameters, which make them hard to adapt to new situations. In contrast, DLWPs may selforganise and learn the necessary abstract representations of

atmospheric dynamics required to predict the weather accurately at scale, and infer the influences of subgrid processes and interactions between them while doing so. The fact that DL requires less modelling effort and imposes less representational constraints forms the basis for using it to solve complex problems. The self-organising nature of DL enables faster model design and more flexible inferences of complex correlation patterns as long as sufficient amounts of data are available. Moreover, once a DLWP model has been trained, the forward inference of a forecast can be conducted at a fraction of the energy and time investment of an NWP forecast, which may support larger ensembles and more accurate uncertainty quantification.

1.2 Problem Statement

It centers on the vital task of post-processing in weather forecasting, aiming to enhance the accuracy and reliability of numerical weather predictions (NWP). Numerical weather predictions inherently contain biases stemming from various sources, including initial condition errors, computational simplifications, and sub-grid parametrizations. These factors contribute to errors that amplify over time, leading to increased uncertainty in forecasts as the prediction horizon extends. Consequently, the forecast becomes inherently uncertain, posing significant challenges for decision-making and risk management in various sectors reliant on accurate weather forecasts. Addressing these issues requires innovative post-processing techniques to mitigate biases and improve the reliability of weather predictions, ultimately enhancing their utility in diverse applications.

1.3 Objective of the Project

The objective of the project is multifaceted, aiming to optimize the learning process and ensure stability in neural network models, particularly in the context of weather forecasting. The key objectives are as follows:

Individualized Normalization of Activations:

Normalize the activations of the previous layer individually for each sample within a batch, enhancing stability and optimizing the learning process. This normalization approach contributes to improved model performance by ensuring consistent activation scaling across different samples, thereby facilitating more effective gradient propagation during training.

Extraction of Essential Features:

Facilitate the extraction of essential features from input sequences and deep layers, enhancing the accuracy of sequence prediction tasks. By normalizing activations individually within each sample, the model can more effectively discern relevant features from the input data, leading to more precise and informative representations of the underlying patterns and dynamics.

Automation of Data Characterization:

Automate the categorization of key data characteristics to enable faster computations and streamline the forecasting process. By automatically identifying and categorizing important data features, the model can efficiently process and analyze large volumes of data, reducing computational overhead and enhancing scalability.

Reducing Distributional Assumptions:

Reduce distributional assumptions in modeling, particularly in the context of weather forecasting, where many variables cannot be effectively described using simple probability distributions. By adopting a normalization approach that is less reliant on distributional assumptions, the model can better adapt to the complex and dynamic nature of atmospheric variables, resulting in more reliable and accurate predictions.

1.4 Project Domain

The project domain focuses on enhancing weather forecasting through the application of deep learning techniques, with a dual emphasis on speed and interpretability. By leveraging advancements in machine learning and integrating them with principles from meteorology, the project aims to accelerate forecasting models while maintaining high levels of accuracy. Central to this approach is the utilization of methods such as attention mechanisms, which enhance the transparency and interpretability of the models' predictions.

The domain encompasses the intersection of two fields: machine learning and meteorology. Machine learning techniques, particularly deep learning, offer powerful tools for pattern recognition and prediction, making them well-suited for analyzing complex weather data. By harnessing the computational capabilities of deep learning models, the project seeks to improve the efficiency and effectiveness of weather forecasting processes.

At the same time, the project remains firmly rooted in the domain of meteorology, drawing on

established principles and methodologies for understanding atmospheric dynamics and phenomena. By integrating machine learning with meteorological expertise, the project aims to develop forecasting models that not only leverage the computational power of deep learning but also reflect a deep understanding of the underlying physical processes driving weather patterns.

Overall, the project domain represents a convergence of cutting-edge technology and domain-specific knowledge, with the overarching goal of advancing the field of weather forecasting through innovative and interdisciplinary approaches. By harnessing the strengths of both machine learning and meteorology, the project seeks to push the boundaries of what is possible in weather prediction, ultimately benefiting various stakeholders and applications reliant on accurate and timely forecasts.

1.5 Scope of the Project

Neural networks have emerged as powerful tools in various aspects of weather forecasting, spanning nowcasting, data assimilation, model physics evaluation acceleration, and postprocessing of model outputs. The complex spatiotemporal nature of weather phenomena aligns well with the strengths of convolutional networks in capturing spatial relationships and recurrent networks in understanding temporal evolution. This versatility has sparked ambitious endeavors to expand the utilization of neural networks in meteorology further. Recognizing this potential, our project aims to contribute to this growing field by refining the architecture, training methodologies, and application frameworks of neural networks tailored specifically for weather forecasting. By enhancing these aspects, we envision improving the accuracy, efficiency, and interpretability of weather prediction models, thereby facilitating advancements across multiple domains within meteorology. Through our research and development efforts, we aim to not only contribute to the ongoing evolution of weather forecasting techniques but also to empower meteorologists with more reliable and actionable insights into atmospheric dynamics, ultimately bolstering our ability to mitigate the impacts of severe weather events and adapt to changing climate patterns.

1.6 Methodology

The methodology for the project "Towards Accelerating & Interpretable Weather Forecasting using Deep Network Learning" involves data collection from diverse sources, followed by preprocessing to handle anomalies and feature engineering for input data preparation. Subsequently, deep neural network architectures, such as CNNs or RNNs, are developed and trained on the preprocessed data, with hyperparameter tuning for optimization. Techniques for accelerating the forecasting process, like model

parallelism and hardware acceleration, are explored alongside methods to enhance model interpretability, such as attention mechanisms and feature importance analysis. Evaluation of model performance using metrics like MAE and RMSE is conducted, followed by deployment and integration into existing weather forecasting systems, with provisions for continuous monitoring and updating. Collaboration between machine learning and meteorology experts underpins each step to ensure both accuracy and relevance in the final forecasting solutions.

CHAPTER 2

LITERATURE REVIEW

Estimating Single-Epoch Integrated Atmospheric Refractivity From InSAR for Assimilation in Numerical Weather Models (Paper by Gert Mulde et al., 2022): It explores the estimation of atmospheric refractivity using Interferometric Synthetic Aperture Radar (InSAR) data. This information can be potentially integrated into numerical weather models to improve their accuracy by incorporating data on atmospheric bending of light signals.

Nationwide Radar-Based Precipitation Nowcasting—A Localization Filtering Approach and its Application for Germany (Paper by Ricardo Reinoso-Rondinel et al., 2022): This research focuses on nowcasting, which is the prediction of weather conditions for a very short time horizon (typically up to a few hours). This work investigates a radar-based approach for precipitation nowcasting in Germany. It proposes a localization filtering technique to improve the accuracy of nowcasting by considering the spatial distribution of precipitation data captured by radar.

A Novel Data-Driven Tropical Cyclone Track Prediction Model Based on CNN and GRU With MultiDimensional Feature Selection (Paper by Jie Lian et al., 2020): This work introduces a data-driven model for predicting the tracks of tropical cyclones. It utilizes a combination of Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) to learn complex patterns from historical cyclone track data and various atmospheric features. Additionally, the paper explores multi-dimensional feature selection techniques to identify the most relevant features for improving prediction accuracy.

Excess Path Delays From Sentinel Interferometry to Improve Weather Forecasts (Paper by Nazzareno Pierdicca et al., 2020): this research investigates techniques for incorporating additional data sources into weather forecasting models. It explores the use of excess path delay data obtained from Sentinel satellite interferometry. This data reflects changes in the atmosphere caused by weather phenomena. By including this information in weather models, the research suggests potential improvements in forecast accuracy.

A Newly Developed Integrative Bio-Inspired Artificial Intelligence Model for Wind Speed Prediction (Paper by Hai Tao et al., 2020): This work explores the application of bio-inspired artificial intelligence for wind speed prediction. It proposes a novel model that integrates different techniques inspired by biological processes, such as artificial bee colony optimization and a support vector machine. The research investigates the effectiveness of this bio-inspired approach for wind speed prediction.

Research on Ultra-Short-Term Wind Power Prediction Considering Source Relevance (Paper by Yong Sun et al., 2020): This research focuses on ultra-short-term wind power prediction, which is crucial for integrating wind power into the power grid effectively. The work proposes a method that considers the relevance of different data sources for improving prediction accuracy. It combines Numerical Weather Prediction (NWP) information with historical wind power data to identify time periods with potentially low forecasting accuracy from traditional methods. The model then utilizes a hybrid approach involving neural networks and persistence methods to address these specific timeframes

Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast (Paper by Hisham Mahmood et al., 2020): This work explores the use of Long Short-Term Memory (LSTM) neural networks for short-term photovoltaic power forecasting. Photovoltaic power generation is highly dependent on weather conditions. The research investigates the effectiveness of LSTMs in learning complex relationships between historical solar power data and synthetic weather forecasts. This approach allows for potentially more accurate predictions of photovoltaic power generation.

Parametric Reconstruction Method for the Long Time-Series Return-Stroke Current of Triggered Lightning Based on the Particle Swarm Optimization Algorithm (Paper by Xiangpeng Fan et al., 2020): This work might not be directly relevant to weather forecasting, it showcases an application of optimization algorithms within atmospheric research. The paper explores a method for reconstructing the long-time series current of lightning strikes using a Particle Swarm Optimization (PSO) algorithm. This technique could be potentially adapted for other applications in weather analysis or modeling specific weather phenomena.

Predicting Storm Outages Through New Representations of Weather and Vegetation (Paper by Diego Cerrai et al., 2019): This research focuses on predicting storm outages, which is crucial for utility companies to prepare and mitigate potential disruptions. The work explores the development of new representations for weather and vegetation data to improve the accuracy of storm outage prediction models. These new representations aim to capture the complex interactions between weather conditions and vegetation characteristics that can influence the likelihood of power outages during storms.

Weather Visibility Prediction Based on Multimodal Fusion (Paper by Chuang Zhang et al., 2019): This work investigates the use of multimodal fusion for weather visibility prediction. Multimodal fusion refers to combining data from different sources to create a more comprehensive understanding of a phenomenon. In this case, the research explores the fusion of weather station data, satellite imagery, and historical visibility data to improve the accuracy of weather visibility forecasts.

CHAPTER 3

PROJECT DESCRIPTION

3.1 Existing System

Numerical weather prediction (NWP) is a challenging task which involves working with micro and macro-scale spatio-temporal parameters susceptible to biases and accuracy problems. In recent years, machine learning has grown in popularity with the increasing demand in accurate weather predictions. In this study, the existing system authors adopt a multimodel (ensemble) forecasting approach by collecting precipitation data from multiple NWP models of Canadian, American and European weather agencies in an effort to deploy an optimal machine learning-based weather model for real-time precipitation forecasting that will outperform the baseline. We considered 8 NWP models as inputs and combined them to create ensemble predictors using 5 different machine learning techniques along with a baseline model (mean of eight input NWP models). We demonstrate that machine learning approaches can improve upon the results of the individual NWP models. The best results were obtained by the neural-network variants with 17% improvement in the mean absolute error, 3% in the root mean squared error, 47% in the median absolute error, 5% in the maximum error, 70% in the relative bias, 41% in the false alarm ratio and 8% in the critical score index over the baseline. Neural networks also complied with the practicality constraints, with minutes of training time and near-real time prediction time. In this study, the existing system authors have explored the capabilities of 5 different machine learning techniques to produce precipitation fore- casts. We collected and extracted sample precipitation data covering most of the continental USA and Canada. Extensive preprocessing and feature selection tasks were performed on the spatio-temporal dataset. We selected eight WMs as input for training six classical machine learning models. Experiments show that machine learning approaches can improve weather forecast predictions in terms of 9 different metrics considered, and the best performers are the neural networks. Future work includes considering additional input WMs, collection of a larger dataset over a longer span of time, and investigation of alternate deep neural network frameworks such as Generative Adversarial Networks, graph neural networks. A comparative performance analysis for longer range predictions is also a potential research area. The precipitation forecasts in this study were produced for daily (24-h) periods. While these are useful, many input WMs provide forecasts at increments as small as 1-h. To improve the usefulness of these forecasts, smaller temporal scales could be studied in the future. In addition, our forecasts were produced on a relatively coarse grid (0.125 degrees), which

convective storms. Using input models which are available at higher spatial and temporal resolutions may increase the utility of the resulting forecasts.

3.2 Proposed System

The experiment was performed on latest data and thus included some necessary preprocessing steps to reflect true model performance. The missing values in the data were imputed using linear interpolation in forward direction. Linear interpolation estimates the missing values in the increasing order from previous values. Smoothing of data using simple moving average with an appropriate window length is an effective technique in time series forecasting as it removes noise and random variations from data without neglecting the weather variations in time.

The data is split into training set and testing set of proportion 0.7 and 0.2 respectively. Both training and test sets are processed using moving window algorithm to obtain the input and output sequences. The input sequence contains features and the output sequence contains temperature values.

The spatial feature attention module is built in the decoder parallel to temporal layer to capture spatial feature correlations while attending most relevant time steps as it directly aligns with the output feature. Spatial feature embeddings are generated independently using feed forward neural network which are input to the spatial feature attention module. The feedforward neural network used to compute spatial feature embeddings include a series of computations where the data from previous hidden state of decoder is concatenated with input features acted upon by soft-max activation function to assign weights in the model.

3.2.1 Advantages

The advantages outlined pertain to a specific approach or model in the context of data analysis or machine learning. Here's an elaboration on each advantage:

Extract non-linear features and has good generalizing abilities:

This advantage implies that the approach or model is capable of capturing complex, non-linear relationships within the data. By extracting such features, it can effectively represent the underlying structure of the data, leading to better generalization performance. This is particularly beneficial when dealing with datasets that exhibit intricate patterns or relationships that cannot be adequately captured.

Does not require complex algorithms to process data:

Unlike some other methods that rely on complex algorithms or computations, this approach or model offers simplicity in data processing. This makes it easier to implement and deploy, reduces computational overhead, and facilitates scalability to large datasets. Moreover, the simplicity in processing may lead to faster training and inference times, making it suitable for real-time or near-real-time applications.

Adds complexity and increases the inference overhead as input sequences grow longer:

While this approach may be effective in capturing non-linear features, it may encounter challenges as the input sequences grow longer. The increased complexity can lead to higher computational overhead during inference, potentially impacting the model's efficiency and scalability. Additionally, longer input sequences may introduce greater computational complexity during training, requiring more computational resources and potentially longer training times.

Outperforms both baselines on most datasets as well as on average:

This advantage highlights the superior performance of the approach or model compared to baseline methods across various datasets. By consistently outperforming baselines, it demonstrates its efficacy and robustness in capturing the underlying patterns and relationships within the data. This makes it a preferred choice for tasks where high predictive accuracy and generalization performance are paramount, such as classification or regression tasks in machine learning.

3.3 Feasibility Study

A Feasibility study is carried out to check the viability of the project and to analyze the strengths and Weaknesses of the proposed system.The application of usage of mask in crowd are as must be evaluated. The feasibility study is carried out in three forms

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.3.1 Economic Feasibility

- The proposed system does not require any high cost equipment. This project can be developed within the available software. Computational Resources: Deep learning models for weather prediction can also be computationally intensive, particularly during the training phase. This can result in higher electricity costs associated with running the computations, especially if powerful GPUs are required. Additionally, there might be an initial investment in acquiring or upgrading hardware infrastructure to support the computational demands of the models.

- Data Acquisition: Acquiring high-quality weather datasets, including historical records, satellite imagery, and sensor data, may involve costs associated with data licensing, procurement, or collaborations with meteorological agencies or research institutions. The cost-effectiveness of data acquisition depends on factors such as the availability and quality of open-access datasets versus proprietary sources.

- Operational Costs: Beyond initial setup and training, operational costs include ongoing maintenance, monitoring, and optimization of the deep learning models. This may involve expenses related to software updates, data storage, and personnel training to ensure continued model performance and accuracy.

- Cost-Benefit Analysis: Despite the upfront and ongoing costs, deep learning-based weather prediction can offer significant benefits, including improved forecast accuracy, better resource allocation for disaster preparedness, and potential economic gains from more informed decision-making in various sectors such as agriculture, transportation, and energy. Conducting a cost-benefit analysis can help evaluate the overall economic feasibility of implementing deep learning-based weather prediction systems and justify investment decisions.

3.3.2 Technical Feasibility

- Technical Expertise: Implementing deep learning methods for weather prediction requires expertise in machine learning, specifically deep neural networks, time series analysis, and signal processing techniques. Additionally, knowledge of meteorology and climate science is beneficial to interpret model outputs accurately.

- Data Availability: The success of the project relies on the availability of extensive and high-

quality weather datasets, including historical weather records, satellite imagery, radar data, and atmospheric sensor data. These datasets should cover a wide range of geographical locations and weather conditions to ensure the model's robustness and generalization capability.

- Computational Resources: Training deep learning models for weather prediction demands significant computational resources, including high processing power and sufficient memory. Access to parallel processing units such as GPUs or TPUs is essential to accelerate model training and experimentation. Additionally, cloud computing services may be utilized to scale resources according to the project's needs and budget constraints.

3.3.3 Social Feasibility

- Ethical Considerations: Deep learning weather prediction models must adhere to ethical standards regarding data privacy and usage. This includes ensuring that personal data collected for weather monitoring is handled responsibly, with appropriate consent and protection measures in place to safeguard individuals' privacy rights.

- Acceptance by Meteorological Community: The successful adoption of deep learning weather prediction models relies on gaining acceptance from meteorologists and weather forecasters. Convincing these professionals of the model's accuracy, reliability, and ability to generalize across different regions and weather conditions is crucial. Collaborating with meteorological agencies and research institutions to validate and integrate these models into existing forecasting frameworks can facilitate their acceptance.

- Social Impact: Deep learning weather prediction models have the potential to have a significant positive social impact. By providing more accurate and timely weather forecasts, these models can help mitigate the impact of natural disasters, improve disaster preparedness and response efforts, and optimize resource allocation in sectors such as agriculture, transportation, and energy. Ultimately, these models contribute to enhancing public safety, economic stability, and environmental sustainability, thereby improving overall quality of life for individuals and communities.

3.4 System Specification

3.4.1 Hardware Specification

- Processor: Minimum i3 Dual Core
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 100 GB; Recommended 200 GB or more
- Memory (RAM): Minimum 8 GB; Recommended 32 GB or above

3.4.2 Software Specification

- Python
- Anaconda
- Jupyter Notebook
- TensorFlow
- Kera

CHAPTER 4

PROPOSED WORK

4.1 Introduction to the Proposed Work

The project titled "Towards Accelerating & Interpretable Weather Forecasting using Deep Network Learning" represents a pioneering effort to transform the landscape of weather forecasting through the application of advanced deep learning techniques. Amidst the escalating challenges posed by climate change and the increasing frequency of extreme weather events, there is a critical demand for more precise and understandable weather predictions. Our project aims to address this need by harnessing the potential of deep neural networks to not only expedite the forecasting process but also to provide meaningful insights into the underlying factors driving weather patterns.

At its core, our project recognizes the inherent complexity of weather data, which exhibits intricate spatio-temporal patterns that traditional forecasting methods often struggle to capture comprehensively. By integrating insights from diverse sources, including numerical models and observational data, we aim to develop a sophisticated framework capable of unlocking deeper insights into weather dynamics. Through the adept utilization of deep network learning, we seek to unveil hidden relationships and dependencies within the data, thereby enhancing the accuracy and reliability of our forecasts.

Crucially, our project does not prioritize speed at the expense of interpretability. We are committed to ensuring that the forecasting outcomes generated by our models are not only swift but also transparent and interpretable. To achieve this, we will explore innovative techniques such as attention mechanisms and feature importance analysis, which will allow us to elucidate the reasoning behind each forecasted outcome.

Throughout the project, rigorous experimentation and validation will be conducted to refine and validate our methodologies. By pushing the boundaries of weather forecasting through a combination of cutting-edge technology and domain expertise, we aspire to provide stakeholders, including emergency responders and agricultural planners, with actionable insights that can aid in decision-making and risk mitigation strategies. Ultimately, our project represents a bold step towards a future where weather forecasting is not only more accurate and efficient but also more understandable and accessible to all.

4.2 General Architecture

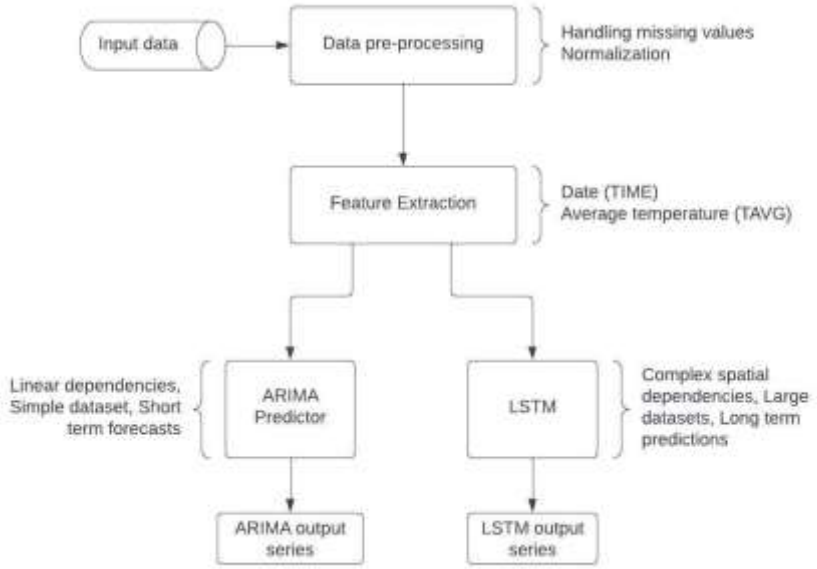

Figure 4.1: **Architecture Diagram**

Figure 4.1 represents the architecture diagram of the project. This machine learning training system takes data, cleans and prepares it, then trains various models on it. Once trained, the system tests the models on unseen data and picks the best performing one to make predictions on entirely new data, like forecasting sales or recommending products.

4.3 Design Phase

During the design phase of "Towards Accelerating & Interpretable Weather Forecasting using Deep Network Learning," we'll delineate the project's requirements and objectives, crafting a blueprint for architecture, model selection, and data pipelines. Our focus will be on selecting suitable deep learning models, designing efficient data preprocessing pipelines, and exploring techniques for model acceleration and interpretability. Additionally, we'll design experiments to evaluate model performance and mitigate potential risks. This phase aims to lay the groundwork for successful implementation, ensuring that our forecasting system is both accurate and interpretable, ultimately advancing the field of weather forecasting.

4.3.1 Data Flow Diagram

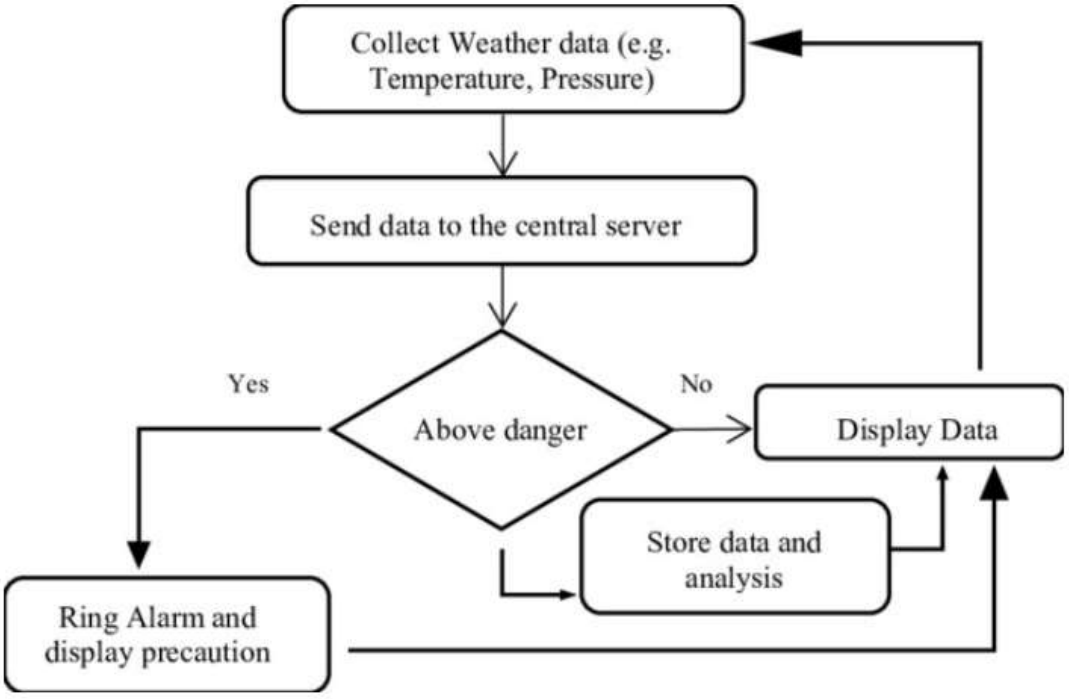

Figure 4.2: **Data Flow Diagram**

Figure 4.2 flowchart begins with the collection of weather data, including temperature and pressure. This data is transmitted to a central server for analysis. Upon analysis, if the data exceeds a predetermined danger threshold, an alarm is triggered, and a warning message is displayed.

Conversely, if the data falls within normal limits, it is stored for further analysis. This process ensures timely identification and communication of potential weather hazards, contributing to proactive risk management and decision-making.

4.4 UML Diagram

4.4.1 Use Case Diagram

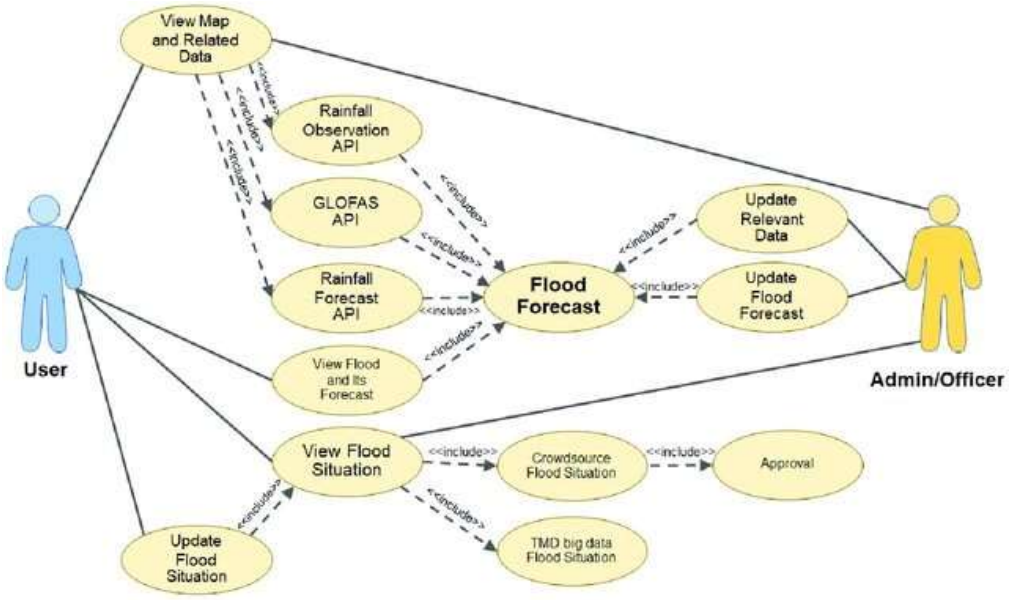

Figure 4.3: **Use Case Diagram**

Figure 4.4 represents the Use Case diagram of our model It gathers data on rainfall observations, forecasts, and potential flooding from multiple sources. This includes official resources like the Global Flood Monitoring System (GLOFAS) API and rainfall forecast APIs, alongside potentially crowd-sourced data from users. By analyzing this combined data, the system updates flood forecasts and warnings. User or official reports on flood situations also seem to factor into the process, requiring approval within the system. Additionally, the inclusion of TIMD big data suggests the system might leverage a large-scale data platform for advanced analysis, ultimately aiming to provide the most accurate and timely flood warnings possible.

4.4.2 Sequence Diagram

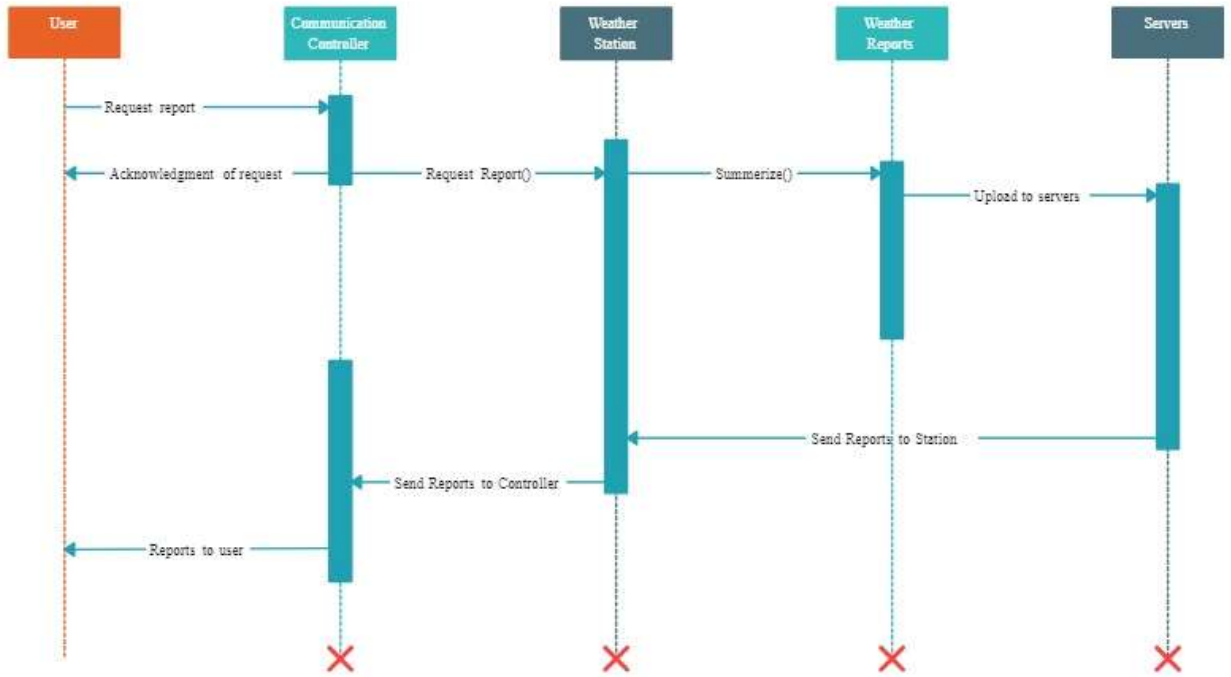

Figure 4.4: **Sequence Diagram**

The figure 4.5 represents the sequence diagram of our project. The sequence of interactions in the "Towards Accelerating & Interpretable Weather Forecasting Using Deep Network Learning" project involves a user initiating a weather forecast update request, prompting the system to gather data from official sources like the GLOFAS API and Rainfall Forecast APIs, as well as utilizing the TIMD Big Data Platform for advanced analysis. The system collects official flood observations, current and predicted rainfall data, and additional data from TIMD. Following data combination and advanced analysis, the system updates weather and flood forecasts, notifying the user of the updated forecast. In the event of a user-reported flood situation, the system processes the report, potentially requiring internal approval. This sequence ensures a comprehensive approach to gathering, analyzing, and updating forecasts, leveraging both official and potentially crowd-sourced data for accurate and timely warnings.

4.5 Module Description

Our entire project is divided into three modules.

- Module 1 : Exploratory Data Analysis

- Module 2 : Feature Analysis
- Module 3 : Weather Forecasting

4.5.1 MODULE 1 : Exploratory Data Analysis

Exploratory Data Analysis (EDA) stands as a fundamental pillar in the realm of data mining and analytics, offering a comprehensive approach to understanding and interpreting datasets before embarking on modeling tasks. In this module, we delve into the principles, techniques, and significance of EDA, which serves as the cornerstone for extracting meaningful insights and patterns from raw data.

Exploratory Data Analysis (EDA) serves as a crucial preliminary step in the data analysis pipeline, enabling analysts to gain a deeper understanding of the dataset's structure, characteristics, and relationships. Unlike formal modeling tasks, which involve predictive modeling or hypothesis testing, EDA focuses on exploring the data's inherent qualities through qualitative investigation and visual methods. By employing EDA techniques, analysts can uncover patterns, anomalies, and trends that may inform subsequent analytical tasks.

Univariate Analysis

Univariate analysis constitutes one of the foundational components of EDA, focusing on the examination of individual variables within the dataset. This approach involves analyzing the distribution, central tendency, and variability of a single variable at a time. Visualizations such as histograms, bar charts, and box plots are commonly used to summarize and visualize the distribution of univariate data. Through univariate analysis, analysts can identify outliers, detect skewness or asymmetry in the data, and assess the presence of any notable patterns or trends.

Multivariate Analysis

Multivariate analysis extends beyond the examination of individual variables to explore the relationships between two or more variables within the dataset. This approach aims to uncover complex interactions and dependencies between variables, providing deeper insights into the underlying data structure. Bivariate analysis, which involves examining the relationship between two variables, is a common technique in multivariate analysis. Visualizations such as scatter plots, heatmaps, and correlation matrices are employed to visualize the relationships between variables and identify potential associations or correlations. Additionally, multivariate analysis may involve the exploration of higher-

dimensional relationships through techniques such as principal component analysis (PCA) or cluster analysis.

Significance and Applications of Exploratory Data Analysis

Exploratory Data Analysis plays a pivotal role in various domains and applications, ranging from business analytics and finance to healthcare and social sciences. By providing a holistic understanding of the dataset's characteristics and relationships, EDA enables analysts to make informed decisions, identify actionable insights, and uncover hidden patterns or trends. Moreover, EDA serves as a critical step in the data preprocessing pipeline, helping analysts identify and address data quality issues, missing values, or outliers before proceeding with formal modeling tasks. In summary, EDA serves as a foundational tool for data exploration, hypothesis generation, and decision-making, paving the way for more robust and reliable analytical insights.

Exploratory Data Analysis serves as an indispensable component of the data analysis process, offering a systematic approach to uncovering insights and patterns within datasets. Through techniques such as univariate and multivariate analysis, analysts can gain a deeper understanding of the data's characteristics, relationships, and trends. By leveraging EDA, analysts can make informed decisions, generate hypotheses, and extract actionable insights that drive data-driven decision-making across various domains and applications. As datasets continue to grow in size and complexity, the significance of EDA in unlocking the potential of data analytics will only continue to grow, making it an essential skill for data scientists, analysts, and decision-makers alike.

4.5.2 Module 2 : Feature Analysis

Feature selection is a dimensionality reduction technique that selects a subset of features (predictor variables) that provide the best predictive power in modeling a set of data. Feature selection can be used to:

Prevent overfitting: avoid modeling with an excessive number of features that are more susceptible to rote learning specific training examples.

Reduce model size: increase computational performance with high-dimensional data or prepare model for embedded deployment where memory may be limited.

Improve interpretability: use fewer features, which may help identify those that affect model behavior.

There are several common approaches to feature selection (we will choose one of the appropriate approaches according to our dataset).

Iteratively change features set to optimize performance or loss: Stepwise regression sequentially adds or removes features until there is no improvement in prediction. It is (c) Wisen IT Solutions Page 11 of 27 used with linear regression or generalized linear regression algorithms. Similarly, sequential feature selection builds up a feature set until accuracy (or a custom performance measure) stops improving.

Rank features based on intrinsic characteristic: These methods estimate a ranking of the features, which in turn can be used to select the top few ranked features. Minimum redundance maximum relevance (MRMR) finds features that maximize mutual information between features and response variable and minimize mutual information between features themselves. Related methods rank features according to Laplacian scores or use a statistical test of whether a single feature is independent of response to determine feature importance.

Neighborhood Component Analysis (NCA) and Relief: These methods determine feature weights by maximizing the accuracy of prediction based on pairwise distance and penalizing predictors that lead to misclassification results.

Learn feature importance along with the model: Some supervised machine learning algorithms estimate feature importance during the training process. Those estimates can be used to rank features after the training is completed. Models with built-in feature selection include linear SVMs, boosted decision trees and their ensembles (random forests), and generalized linear models. Similarly, in lasso regularization a shrinkage estimator reduces the weights (coefficients) of redundant features to zero during training.

4.5.3 Module 3 : Weather Forecasting

The dataset is partitioned into training, testing, and validation sets in the ratio of 7:2:1, respectively. The training set, constituting 70% of the data, is utilized to train the model, while the validation set (10%) is employed to fine-tune the model parameters. Finally, the testing set (20%) evaluates the model's performance on unseen data, providing insights into its generalization ability.

Neural network models typically comprise three layers: input, hidden, and output. The input layer receives the data, which is then processed in the hidden layer, leveraging nonlinear activation functions to enhance the model's capacity to capture complex relationships in the data. The output layer presents the results of the analysis.

In this study, the neural network architecture includes three dense layers and two dropout layers, aimed at preventing overfitting by randomly dropping out some neurons during training. Additionally, a deeper neural network (DNN) configuration is employed, consisting of five dense layers and three dropout layers, further enhancing the model's capacity to capture intricate patterns in the data.

Classification accuracy serves as a key performance metric, quantifying the model's ability to correctly classify data instances. Metrics such as overall accuracy (OA), recall, and precision are commonly utilized to assess classification performance, providing insights into the model's efficacy in accurately labeling data instances across different classes. Overall accuracy measures the proportion of correctly classified instances, while recall and precision offer insights into the model's ability to correctly identify instances of a specific class and minimize false positives, respectively. Evaluating classification accuracy is crucial for assessing the model's performance and informing decision-making processes in various applications, ranging from healthcare to finance and beyond.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

STATION	NAME	LATITUDE	LONGITUDE	ELEVATION	PRCP	SNWD	TAVG	TMAX	TMIN	DATE
1993-01-01	PLM00012375	OKECIE, PL	52.166	20.967	110.3	0.0	10.0	-8.3	NaN	NaN
1993-01-02	PLM00012375	OKECIE, PL	52.166	20.967	110.3	NaN	10.0	-14.9	NaN	NaN
1993-01-03	PLM00012375	OKECIE, PL	52.166	20.967	110.3	0.0	10.0	-13.6	-9.7	NaN
1993-01-04	PLM00012375	OKECIE, PL	52.166	20.967	110.3	0.0	10.0	-10.5	-6.5	-13.3
1993-01-05	PLM00012375	OKECIE, PL	52.166	20.967	110.3	0.0	10.0	-12.0	-8.9	-14.1

Table 5.1: Dataset Snippet

5.2 Testing

The testing process of the deep learning-based weather prediction system is essential to ensure its effectiveness in real-world scenarios. Key considerations in this testing methodology include:

- Data Source: High-quality weather data from reputable sources is crucial. This data should encompass diverse geographical regions, weather phenomena, and temporal scales to ensure the model's robustness and generalization capability.

- Data Preprocessing: Similar to the training data, the testing data undergoes preprocessing steps to ensure consistency and quality. This may involve data cleaning, normalization, and feature extraction to prepare the data for model input.

- Ground Truth Labeling: Each weather event in the testing set requires accurate ground truth

labels, such as observed weather conditions or forecasted outcomes. These labels serve as benchmarks for evaluating the model's predictions and performance.

- Data Splitting: The available weather data is typically split into training, validation, and testing sets. The training set is used to train the model, the validation set helps optimize model parameters and prevent overfitting, and the testing set is reserved for final performance evaluation.

- Various metrics are employed to assess the performance of the weather prediction system. These metrics may include accuracy, precision, recall, and F1-score, which measure the model's ability to correctly predict different weather conditions.

5.2.1 Unit Testing:

- Objective: To verify the functionality of individual components within the weather prediction system, such as data preprocessing modules, feature extraction algorithms, and model layers.
- Importance: Unit testing ensures that each component operates correctly and reliably, identifying any errors or bugs early in the development process.

5.2.2 Integration Testing:

- Objective: To assess how well different components of the weather prediction system interact and function together. This includes testing data flow between preprocessing, model inference, and post-processing steps.
- Importance: Integration testing ensures the seamless operation of the entire system, identifying any compatibility issues or communication breakdowns between modules.

5.2.3 Functional Testing:

- Objective: To verify that the weather prediction system delivers accurate and reliable forecasts across different weather conditions and geographical regions.
- Importance: Functional testing ensures that the system performs its intended purpose effectively, meeting the requirements of users and stakeholders.

5.2.4 Performance Testing:

- Objective: To evaluate the system's performance characteristics, such as prediction speed, memory usage, and scalability. This includes measuring processing time for generating forecasts and assessing the system's ability to handle large datasets or increased computational demands.
- Importance: Performance testing ensures that the weather prediction system is efficient and practical for real-world use, identifying any bottlenecks or resource constraints that may impact its performance.

5.2.5 User Acceptance Testing:

- Objective: To gather feedback from users, such as meteorologists and emergency responders, on the system's usability, accuracy, and overall effectiveness in supporting decision-making.
- Importance: User acceptance testing provides valuable insights into the system's usability and user experience, helping to identify areas for improvement and ensuring successful adoption in operational settings.

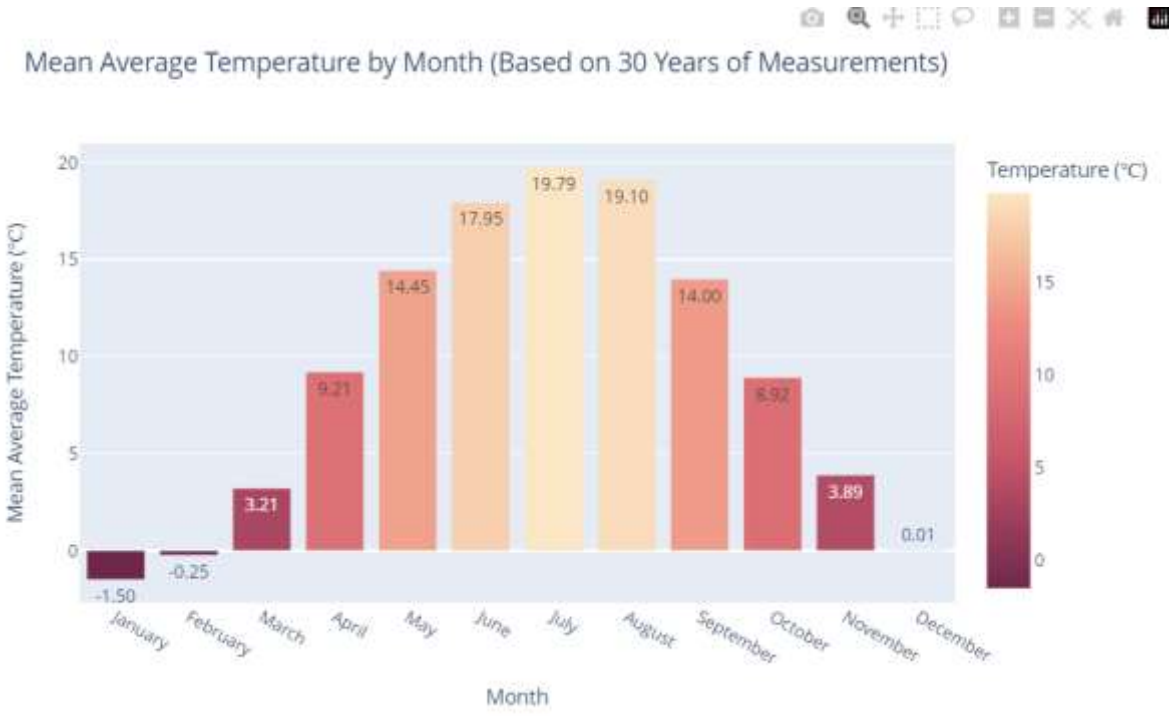

Figure 5.1: Mean avg. temp by Month

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of Proposed System

The proposed framework for deep learning-based weather prediction offers several key advantages that contribute to its efficiency compared to traditional numerical weather prediction systems:

- Consolidated Methodology: Our framework integrates various stages of weather prediction, including data preprocessing, feature extraction, and model inference, into a unified pipeline. This consolidation streamlines the prediction process and eliminates the need for separate, often manual, procedures, thus saving time and resources.

- Automation: Unlike traditional numerical weather prediction methods that rely heavily on manual intervention for data assimilation and model parameterization, our framework automates these processes using deep learning algorithms. This automation reduces human error and accelerates the prediction process, leading to more efficient and timely forecasts.

- Feature Learning: Deep learning models have the ability to automatically learn relevant features from raw weather data, such as satellite imagery, radar data, and atmospheric sensor readings. This feature learning capability enables the model to extract complex patterns and relationships from the data, improving prediction accuracy and efficiency compared to handcrafted feature engineering approaches.

- Ensemble Learning: Ensemble methods, such as combining predictions from multiple deep learning models or integrating deep learning with traditional numerical weather prediction models, can enhance prediction accuracy and robustness. By leveraging the diversity of individual models, ensemble techniques improve the overall efficiency and reliability of weather forecasts.

- Cross-Domain Generalization: Deep learning models trained on diverse weather datasets can generalize well to unseen regions or weather conditions, thus reducing the need for region-specific model calibration or parameter tuning. This cross-domain generalization capability enhances the efficiency and scalability of ourframework for widespread deployment and operational use.

6.2 OUTPUT

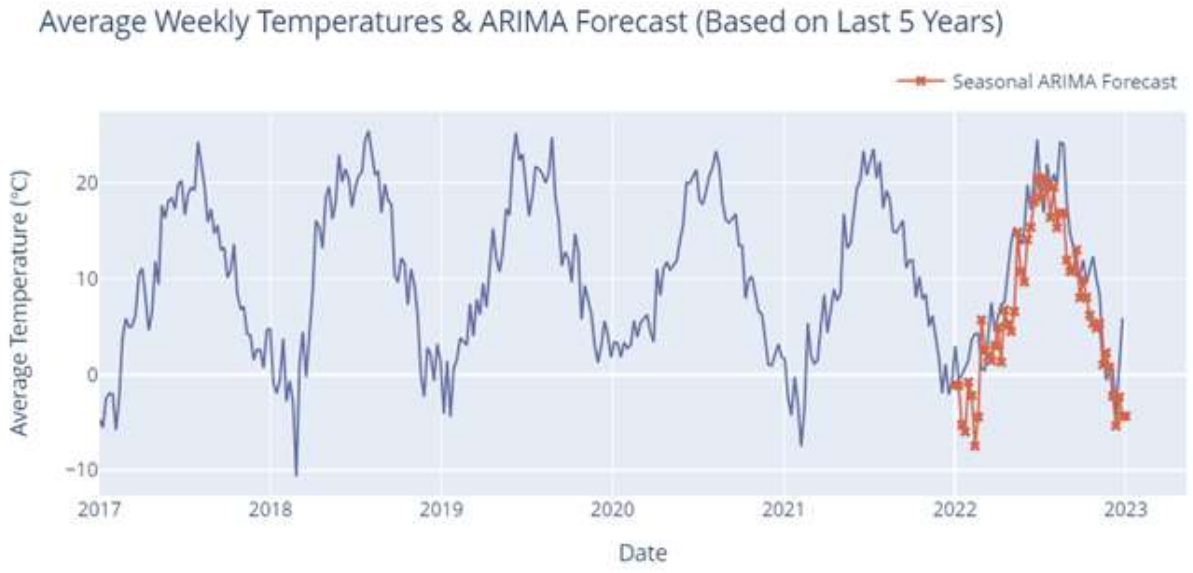

Figure 6.1: Average Weekly Temperatures & ARIMA Forecast

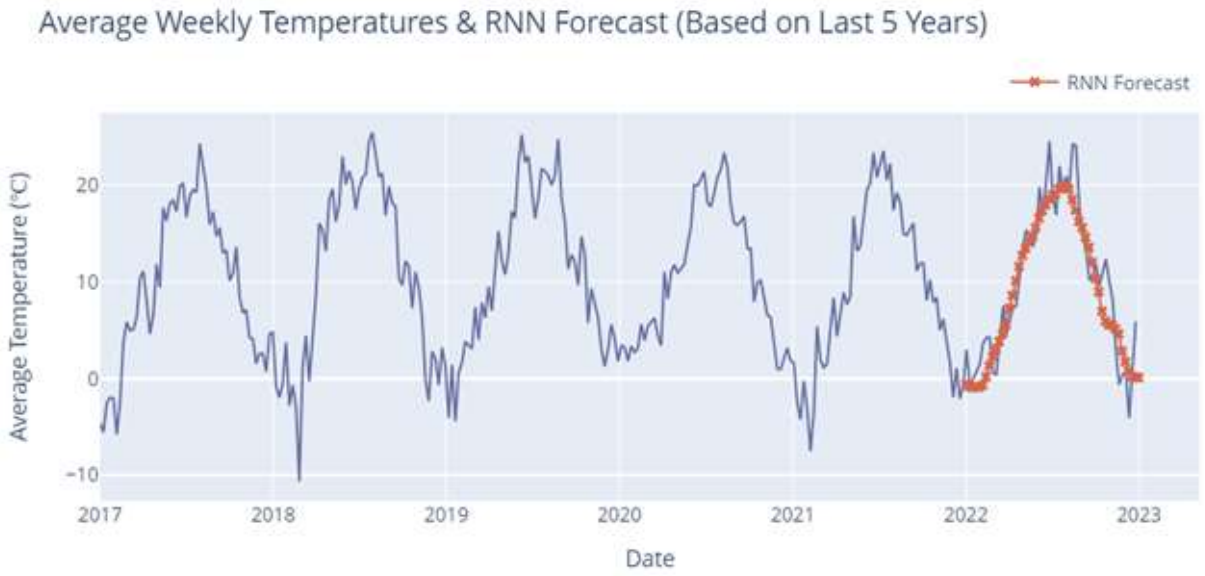

Figure 6.2: Average Weekly Temperatures & RNN Forecast

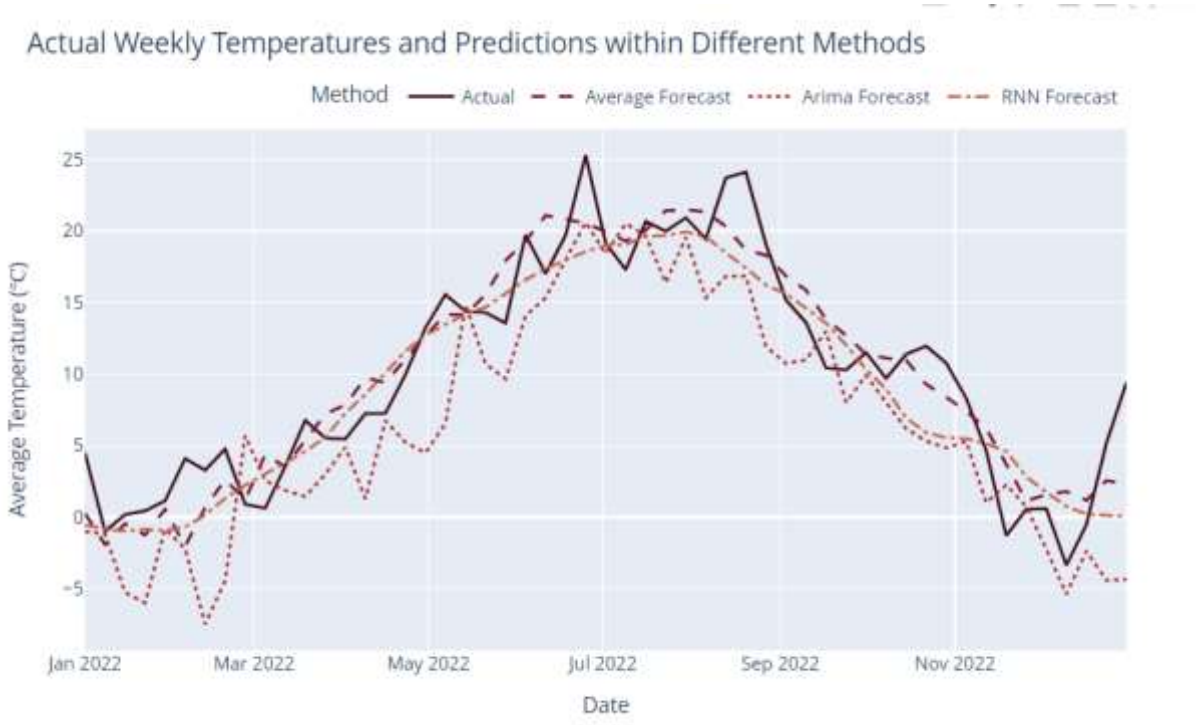

Figure 6.3: Actual Weekly Temperatures & Predictions Within Different Methods

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

We underscore the indispensable role of weather forecasting, underpinned by scientific rigor and technological sophistication, in anticipating atmospheric conditions. Meteorological data, encompassing a spectrum of variables like atmospheric pressure, temperature, humidity, wind speed, and precipitation, serves as the foundational bedrock for understanding the dynamic interplay of atmospheric processes. Through meticulous analysis and interpretation of this data, we gain insights that empower us to forecast future weather states, thereby facilitating informed decision-making and proactive measures to mitigate potential weather-related risks. Our proposed model introduces a groundbreaking spatial feature attention mechanism, strategically engineered to capture the nuanced interdependencies and spatial correlations embedded within multivariate time series data. This innovative mechanism augments forecasting precision by prioritizing the quantitative mutual influence of input features on the target feature, particularly adept at predicting abrupt shifts in weather dynamics. Moreover, by harnessing the collective power of multivariate weather data to forecast a singular target weather feature, our model adeptly discerns the spatial feature influence across diverse variables, thereby elevating the accuracy and reliability of our predictions. Through the deployment of such pioneering methodologies, we aspire to propel the frontiers of weather forecasting, furnishing stakeholders with indispensable tools to navigate and adapt to the evolving complexities of atmospheric dynamics.

7.2 Future Enhancements

In considering future enhancements to our study, one avenue worth exploring involves the integration of regularization and normalization techniques. While these methods did not yield significant benefits in the current model, they have demonstrated notable success in similar contexts by other researchers. Regularization techniques, such as L1 or L2 regularization, offer the potential to mitigate overfitting and enhance the model's ability to generalize beyond the training data. Additionally, normalization methods like standardization or min-max scaling can contribute to improved stability and convergence during the training process. Despite their limited impact in our study, further investigation into the application of these techniques holds promise for enhancing model performance, particularly in optimizing predictive accuracy and robustness. By delving into these avenues for refinement, we can potentially unlock new

insights and strategies to bolster the efficacy of our forecasting model, advancing its utility and applicability in real-world scenarios.

CHAPTER 8

SOURCE CODE

```
import os
import warnings
warnings.filterwarnings("ignore")
from pathlib import Path
import plotly.graph_objects as go
import plotly.express as px
import matplotlib.pyplot as plt
import numpy as np
from itertools import product
import pandas as pd
import seaborn as sns
import tensorflow as tf
from colorama import Fore, Style
from IPython.core.display import HTML
from plotly.subplots import make_subplots
from tensorflow import keras
from tensorflow.keras import layers
from scipy.stats import gaussian_kde
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error
MODELS_PATH = Path("models")
MODELS_PATH.mkdir(exist_ok=True)
weather = pd.read_csv(
    "Dataset.csv",
    parse_dates=["DATE"],
    index_col="DATE"
)

weather.head()

weather.sample()

weather.tail()

weather.info()

temp = weather[["TAVG"]].copy()
temp.describe().T.drop("count", axis=1).rename(columns=str.title)

temp.query("TAVG == -22.3")

temp.query("TAVG == 29.1")

fig = px.line(
```

```
    temp,
    y="TAVG",
    labels={"DATE": "Date", "TAVG": "Average Temperature (\u2103)"},  # Celsius degree.
    title="Average Daily Temperatures in Warsaw, Poland in 1993-2022 (Last 30 Years)",
    height=420,
    width=840,
)
fig.update_layout(

    title_font_size=18
)
fig.update_traces(
    line=dict(width=1.0, color="#2A357D"),
    opacity=0.5,
)
fig.show()

resampled_2022 = temp.loc["2022"].resample("7D").mean()

fig = px.area(
    temp.loc["2022"],
    y="TAVG",
    labels={"DATE": "Date", "TAVG": "Average Temperature (\u2103)"},
    title="Average Daily Temperatures in Warsaw, Poland in 2022",
    height=420,
    width=840,
)
fig.update_traces(line=dict(width=1.5, color="#67727e"))
fig.add_scatter(
    x=resampled_2022.index,
    y=resampled_2022.TAVG,
    showlegend=False,
    text="7 Day Frequency",
    line_shape="spline",
    line=dict(dash="solid", color="#d4674c", width=3),
)
fig.update_layout(

    title_font_size=18,
)
fig.show()

tavg_kde = gaussian_kde(temp.TAVG)

tavg_range = np.linspace(temp.TAVG.min(), temp.TAVG.max(), len(temp))

kde_estimated = tavg_kde.evaluate(tavg_range)

fig = px.histogram(
    temp,
    x="TAVG",
    marginal="box",
    histnorm="probability density",
```

```
    title="Probability Density of Daily Temperatures (Based on 30 Years of Measurements)",
    color_discrete_sequence=["#2A357D"],
    nbins=100,
    height=540,
    width=840,
)
fig.add_scatter(
    x=tavg_range,
    y=kde_estimated,
    showlegend=False,
    text="Average Temperature KDE",
    line=dict(dash="solid", color="#d4674c", width=4),
)
fig.update_layout(

    title_font_size=18,

    bargap=0.25,
    xaxis_title_text="Average Temperature (\u2103)",
    yaxis_title_text="Probability Density",
)
fig.show()

tavg_monthly = (
    temp.groupby(temp.index.month_name(), sort=False).mean(numeric_only=True).reset_index()
)

fig = px.bar(
    tavg_monthly,
    x="DATE",
    y="TAVG",
    labels={"TAVG": "Mean Average Temperature (\u2103)", "DATE": "Month"},
    title="Mean Average Temperature by Month (Based on 30 Years of Measurements)",
    text_auto=".2f",
    color="TAVG",
    color_continuous_scale=px.colors.sequential.Burgyl_r,
    height=500,
    width=840,
)
fig.update_layout(

    title_font_size=18,

    coloraxis_colorbar_title_text="Temperature (\u2103)",
)
fig.show()

adf_result = adfuller(temp.TAVG)

print("Augmented Dickey-Fuller Test for data within daily frequency:")

print("● critical value:".ljust(52), f"{adf_result[0]:.2f}")
print("● p-value:".ljust(52), f"{adf_result[1]:.1e}")
```

```
print("● number of lags used:".ljust(52), f"{adf_result[2]}")
print("● number of observations:".ljust(52), f"{adf_result[3]}")
print(
    "● t-values at 1%, 5% and 10% confidence intervals:".ljust(52),
    f"{np.array(list(adf_result[4].values())).round(2)}",
)

adf_result = adfuller(temp.TAVG.resample("3M").mean())

print("Augmented Dickey-Fuller Test for data within three-month frequency:")
print("=" * 72)
print("● critical value:".ljust(52), f"{adf_result[0]:.2f}")
print("● p-value:".ljust(52), f"{adf_result[1]:.2f}")
print("● number of lags used:".ljust(52), f"{adf_result[2]}")
print("● number of observations:".ljust(52), f"{adf_result[3]}")
print(
    "● t-values at 1%, 5% and 10% confidence intervals:".ljust(52),
    f"{np.array(list(adf_result[4].values())).round(2)}",
)
print("=" * 72)


fig = px.line(
    temp.diff(365),
    y="TAVG",
    labels={"DATE": "Date", "TAVG": "Temperature Difference (\u2103)"},
    title="Average Temperature Difference on a Given Day Year-by-Year (Stationary Signal)",
    height=420,
    width=840,
)
fig.update_layout(

    title_font_size=18,

)
fig.update_traces(
    line=dict(width=1.0, color="#2A357D"),
    opacity=0.5,
)
fig.show()

decomposition = seasonal_decompose(temp.TAVG, model="additive", period=365)

fig = make_subplots(
    rows=4,
    cols=1,
    shared_xaxes=True,
    vertical_spacing=0.1,
    x_title="Date",
    y_title="Temperature (\u2103)",
    subplot_titles=["Observed Values", "Trend", "Seasonality", "Residuals"],
)
```

```
observed = go.Scatter(
    x=decomposition.observed.index,
    y=decomposition.observed,
    name="Observed Temperature",
    line=dict(width=0.7, color="#2A357D"),
)
trend = go.Scatter(
    x=decomposition.trend.index,
    y=decomposition.trend,
    name="Trend",
    line=dict(color="#2A357D"),
)
seasonal = go.Scatter(
    x=decomposition.seasonal.index,
    y=decomposition.seasonal,
    name="Seasonality",
    line=dict(color="#2A357D"),
)
residuals = go.Scatter(
    x=decomposition.resid.index,
    y=decomposition.resid,
    name="Residuals",
    mode="markers",
    marker_size=1,
    line=dict(color="#2A357D"),
)

fig.add_trace(observed, row=1, col=1)
fig.add_trace(trend, row=2, col=1)
fig.add_trace(seasonal, row=3, col=1)
fig.add_trace(residuals, row=4, col=1)

fig.update_annotations(font_size=14)
fig.update_layout(

    title_font_size=18,

    title_text="Average Daily Temperatures - Seasonal Decomposition",
    showlegend=False,
    height=800,
    width=840,
)
fig.show()

ef draw_acf(series, n_lags, marker_size=12):
    corr_array = acf(series, alpha=0.05, nlags=n_lags)
    corr_values = corr_array[0]
    lags = np.arange(len(corr_values))
    lower_y = corr_array[1][:, 0] - corr_array[0]
    upper_y = corr_array[1][:, 1] - corr_array[0]

    fig = go.Figure()
```

```
    for l in lags:
        fig.add_scatter(
            x=(l, l), y=(0, corr_values[l]), mode="lines", line_color="black"
        )

    fig.add_scatter(
        x=lags,
        y=corr_values,
        mode="markers",
        marker_color="#2A357D",
        marker_size=marker_size,
        name="ACF",
    )
    fig.add_scatter(x=lags, y=upper_y, mode="lines", line_color="rgba(255,255,255,0)")
    fig.add_scatter(
        x=lags,
        y=lower_y,
        mode="lines",
        fillcolor="rgba(32, 146, 230, 0.3)",
        fill="tonexty",
        line_color="rgba(255, 255, 255, 0)",
    )
    fig.update_traces(showlegend=False)
    fig.update_xaxes(range=[-1, n_lags + 1])
    fig.update_yaxes(zerolinecolor="black")
    fig.update_layout(

        title_font_size=18,

        title_text="Average Temperature Autocorrelation (ACF)",
        xaxis_title="Lag (Months)",
        yaxis_title="ACF",
        height=500,
        width=840,
    )
    fig.show()

df_monthly = temp.resample("M").mean()
draw_acf(df_monthly, n_lags=12)

draw_acf(df_monthly, n_lags=120, marker_size=6)

train_weekly = temp.TAVG["2017":"2021"].resample("7D").mean()
train_daily = temp.TAVG["2017":"2021"]

avg_forecast = (
    train_daily.groupby(train_daily.index.isocalendar().week)
    .mean()
    .set_axis(pd.date_range("2022-01-01", periods=53, freq="7D"))
)

train_test_weekly = temp.TAVG["2017":"2022"].resample("7D").mean()
train_with_forecast = pd.concat([train_weekly, avg_forecast])
```

```
fig = px.line(
    x=train_test_weekly.index,
    y=train_test_weekly,
    labels={"x": "Date", "y": "Average Temperature (\u2103)"},
    title="Average Weekly Temperatures & Average Forecast (Based on Last 5 Years)",
    height=420,
    width=840,
)
fig.update_traces(line=dict(width=1.5, color="#2A357D"), opacity=0.7)
fig.add_scatter(
    x=avg_forecast.index,
    y=avg_forecast,
    mode="markers+lines",
    marker=dict(symbol="x", size=6),
    line=dict(color="#d4674c", width=1.5),
    name="Average Forecast",
)
fig.update_layout(

    title_font_size=18,

    legend=dict(orientation="h", yanchor="bottom", xanchor="right", y=1.02, x=1),
)
fig.show()

actual_weekly = temp.TAVG["2022"].resample("7D").mean()
avg_method_mae = mean_absolute_error(actual_weekly, avg_forecast)
print("Average Method - Mean Absolute Error: ", f"{avg_method_mae:.2f}")

naive_forecast = temp.TAVG["2021"].resample("7D").mean()
naive_method_mae = mean_absolute_error(actual_weekly, naive_forecast)
print("Naive Method - Mean Absolute Error: ", f"{naive_method_mae:.2f}")

p = d = q = range(0, 2)
pdq = list(product(p, d, q))
s_pdq = list((*combination, 53) for combination in pdq)

np.random.seed(42)

for i_pdq, i_spdq in zip(np.random.permutation(pdq), np.random.permutation(s_pdq)):
    print(
        "ARIMA",
         "({}, {}, {})".format(*i_pdq),
        "x", "({}, {}, {}, {})".format(*i_spdq),
    )

train = temp.TAVG["2017":"2021"].resample("7D").mean()
model = ARIMA(train, order=(1, 1, 1), seasonal_order=(1, 2, 1, 53)).fit()
model.summary()

fig = plt.figure(figsize=(11.7, 11), tight_layout=True)
model.plot_diagnostics(fig=fig)
```

```
ax = fig.get_axes()[2]
ax.get_lines()[0].set_markersize(4.0)
ax.get_lines()[0].set_alpha(0.5)
ax.get_lines()[1].set_linewidth(3.0)
ax.get_lines()[1].set_color("#67727e")
plt.show()

arima_forecast = model.predict(start="2022-01-01", end="2022-12-31")

fig = px.line(
    x=train_test_weekly.index,
    y=train_test_weekly,
    labels={"x": "Date", "y": "Average Temperature (\u2103)"},
    title="Average Weekly Temperatures & ARIMA Forecast (Based on Last 5 Years)",
    height=420,
    width=840,
)
fig.update_traces(line=dict(width=1.5, color="#2A357D"), opacity=0.7)
fig.add_scatter(
    x=arima_forecast.index,
    y=arima_forecast,
    mode="markers+lines",
    marker=dict(symbol="x", size=6),
    line=dict(color="#d4674c", width=1.5),
    name="Seasonal ARIMA Forecast",
)
fig.update_layout(

    title_font_size=18,

    legend=dict(orientation="h", yanchor="bottom", xanchor="right", y=1.02, x=1),
)
fig.show()

arima_method_mae = mean_absolute_error(actual_weekly, arima_forecast)
print("ARIMA Method - Mean Absolute Error: ", f"{arima_method_mae:.2f}")

example_series = np.arange(0, 10)
seq_length = 5  # For example [0, 1, 2, 3, 4].
ahead_steps = 2  # For example [5, 6].
batch_size = 2

example_ds = keras.preprocessing.timeseries_dataset_from_array(
    example_series,
    targets=None,
    sequence_length=seq_length + ahead_steps,
    batch_size=batch_size,
).map(lambda series: (series[:, :-ahead_steps], series[:, -ahead_steps:]))

list(example_ds)

train = temp.TAVG["2007":"2016"].resample("7D").mean()
valid = temp.TAVG["2017":"2021"].resample("7D").mean()
```

```
seq_length = 104  # Window over 2 years.
ahead_steps = 53  # Forecast for the next year.
batch_size = 32

tf.random.set_seed(42)

train_ds = keras.preprocessing.timeseries_dataset_from_array(
    train.to_numpy(),
    targets=None,
    sequence_length=seq_length + ahead_steps,
    batch_size=batch_size,
    shuffle=True,
    seed=42,
).map(lambda series: (series[:, :-ahead_steps], series[:, -ahead_steps:]))

valid_ds = keras.preprocessing.timeseries_dataset_from_array(
    valid.to_numpy(),
    targets=None,
    sequence_length=seq_length + ahead_steps,
    batch_size=batch_size,
).map(lambda series: (series[:, :-ahead_steps], series[:, -ahead_steps:]))

tf.random.set_seed(42)

rnn_model = keras.Sequential(
    [
        layers.Normalization(input_shape=[None, 1]),
        layers.LSTM(32, return_sequences=True, dropout=0.2),
        layers.LayerNormalization(),
        layers.LSTM(32, return_sequences=True, dropout=0.2),
        layers.LayerNormalization(),
        layers.LSTM(32, dropout=0.2),
        layers.LayerNormalization(),
        layers.Dense(ahead_steps),
    ]
)

n_epochs = 200
n_steps = n_epochs * len(list(train_ds))

early_stopping_cb = keras.callbacks.EarlyStopping(
    monitor="val_mae", patience=50, restore_best_weights=True
)

scheduled_lr = keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.05, decay_steps=n_steps, decay_rate=0.1
)

optimizer = keras.optimizers.RMSprop(learning_rate=scheduled_lr)

rnn_model.compile(loss="huber", optimizer=optimizer, metrics=["mae"])
```

```
history = rnn_model.fit(
    train_ds,
    validation_data=valid_ds,
    epochs=n_epochs,
    callbacks=[early_stopping_cb],
    verbose=0,
)

rnn_history = pd.DataFrame(history.history)

fig = px.line(
    rnn_history,
    labels={"variable": "Variable", "value": "Value", "index": "Epoch"},
    title="RNN Training Process",
    color_discrete_sequence=px.colors.diverging.balance_r,
    height=420,
    width=840,
)
fig.update_layout(

    title_font_size=18,

    legend=dict(orientation="h", yanchor="bottom", xanchor="right", y=1.02, x=1),
)
fig.show()

rnn_test = valid.to_numpy()[np.newaxis, :]
rnn_forecast = pd.Series(
    rnn_model.predict(rnn_test, verbose=0).flatten(),
    index=pd.date_range("2022-01-01", periods=ahead_steps, freq="7D"),
)

fig = px.line(
    x=train_test_weekly.index,
    y=train_test_weekly,
    labels={"x": "Date", "y": "Average Temperature (\u2103)"},
    title="Average Weekly Temperatures & RNN Forecast (Based on Last 5 Years)",
    height=420,
    width=840,
)
fig.update_traces(line=dict(width=1.5, color="#2A357D"), opacity=0.7)
fig.add_scatter(
    x=rnn_forecast.index,
    y=rnn_forecast,
    mode="markers+lines",
    marker=dict(symbol="x", size=6),
    line=dict(color="#d4674c", width=1.5),
    name="RNN Forecast",
)
fig.update_layout(

    title_font_size=18,
```

```
    legend=dict(orientation="h", yanchor="bottom", xanchor="right", y=1.02, x=1),
)
fig.show()

nn_method_mae = mean_absolute_error(actual_weekly, rnn_forecast)
print("RNN Method - Mean Absolute Error: ", f"{rnn_method_mae:.2f}")

actual_plus_forecats = pd.DataFrame(
    {
        "Actual": actual_weekly,
        "Average Forecast": avg_forecast.to_numpy(),
        "Arima Forecast": arima_forecast.to_numpy(),
        "RNN Forecast": rnn_forecast.to_numpy(),
    },
    index=pd.date_range("2022-01-01", periods=53, freq="7D", name="Date")
)

fig = px.line(
    actual_plus_forecats,
    labels={"variable": "Method", "value": "Average Temperature (\u2103)"},
    title="Actual Weekly Temperatures and Predictions within Different Methods",
    color_discrete_sequence=px.colors.diverging.balance_r,
    line_dash="variable",
    line_dash_sequence=["solid", "dash", "dot", "dashdot"],
    height=520,
    width=840,
)
fig.update_layout(

    title_font_size=18,

    legend=dict(orientation="h", yanchor="bottom", xanchor="right", y=1.02, x=1),
)
fig.show()
```

REFERENCES

1) Cenker Sengoz,Sheela Ramanna,Scott Kehler,Rushil Goomer,Paul Pries Machine Learning Approaches to Improve North American Precipitation Forecasts IEEE Access, 2023

2) Chuang Zhang,Ming Wu,Jinyu Chen,Kaiyan Chen,Chi Zhang,Chao Xie,Bin Huang,Zichen He Weather Visibility Prediction Based on Multimodal Fusion IEEE Access, 2019

3) Gert Mulder,Freek J. van Leijen,Jan Barkmeijer,Siebren de Haan,Ramon F. Hanssen Estimating Single-Epoch Integrated Atmospheric Refractivity From InSAR for Assimilation in Numerical Weather Models IEEE Transactions on Geoscience and Remote Sensing, 2022

4) Nazzareno Pierdicca,Ida Maiello,Eugenio Sansosti,Giovanna Venuti,Stefano Barindelli,Rossella Ferretti,Andrea Gatti,Mariarosaria Manzo,Andrea Virgilio Monti-Guarnieri,Federica Murgia,Eugenio Realini,Simona Verde Excess Path Delays From Sentinel Interferometry to Improve Weather Forecasts IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020

5) Diego Cerrai,David W. Wanik,Md Abul Ehsan Bhuiyan,Xinxuan Zhang,Jaemo Yang,Maria E. B.Frediani,Emmanouil N. Anagnostou Predicting Storm Outages Through New Representations of Weather and Vegetation IEEE Access, 2019

6) Hai Tao,Sinan Q. Salih,Mandeep Kaur Saggi,Esmaeel Dodangeh,Cyril Voyant,Nadhir Al-Ansari,Zaher Mundher Yaseen,Shamsuddin Shahid A Newly Developed Integrative Bio-Inspired Artificial Intelligence Model for Wind Speed Prediction IEEE Access, 2020

7) Yong Sun,Zhenyuan Li,Xinnan Yu,Baoju Li,Mao Yang Research on Ultra-Short-Term Wind Power Prediction Considering Source Relevance IEEE Access, 2020

8) Mohammad Safayet Hossain,Hisham Mahmood Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast IEEE Access, 2020

9) Xiangpeng Fan,Wen Yao,Yang Zhang,Liangtao Xu,Yijun Zhang,Paul R. Krehbiel,Dong Zheng,Hengyi Liu,Weitao Lyu,Shaodong Chen,Zhengshuai Xie Parametric Reconstruction Method for the Long Time-Series Return - Stroke Current of Triggered Lightning Based on the Particle Swarm Optimization Algorithm IEEE Access, 2020

10) Ricardo Reinoso-Rondinel,Martin Rempel,Markus Schultze,Silke TrÃ¶mel Nationwide Radar-Based Precipitation Nowcastingâ€”A Localization Filtering Approach and its Application for Germany IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022

11) Jie Lian,Pingping Dong,Yuping Zhang,Jianguo Pan,Kehao Liu A Novel Data-Driven Tropical Cyclone Track Prediction Model Based on CNN and GRU With Multi-Dimensional Feature Selection IEEE Access, 2020

12) Z. Pu and E. Kalnay, Numerical Weather Prediction Basics: Models Numerical Methods and Data Assimilation, Berlin, Germany:Springer, pp. 67-97, 2019.

13) A.Y. Hou, The global precipitation measurement mission, Bull. Amer. Meteorol. Soc., vol. 95, pp. 701-722, May 2014.

14) T. M. Hamill, E. Engle, D. Myrick, M. Peroutka, C. Finan and M. Scheuerer, The U.S. national blend of models for statistical postprocessing of probability of precipitation and deterministic precipitation amount, Monthly Weather Rev., vol. 145, no. 9, pp. 3441-3463, Sep. 2017.

15) T. M. Hamill, D. R. Stovern and L. L. Smith, Improving national blend of models probabilistic precipitation forecasts using long time series of reforecasts and precipitation reanalyses. Part I: Methods, Monthly Weather Rev., vol. 151, no. 6, pp. 1521-1534, Jun. 2023.

16) D. R. Stovern, T. M. Hamill and L. L. Smith, Improving national blend of models probabilistic precipitation forecasts using long time series of reforecasts and precipitation reanalyses. Part II: Results, Monthly Weather Rev., vol. 151, no. 6, pp. 1535-1550, Jun. 2023.

17) Y. He, X. Sun, L. Gao and B. Zhang, Ship detection without sea-land segmentation for large-scale highresolution optical satellite images, Proc. IEEE Int. Geosci. Remote Sens. Symp., pp. 717-720, Jul. 2018.

18) R. F. Chevalier, G. Hoogenboom, R. W. Mcclendon and J. A. Paz, Support vector regression with reduced training sets for air temperature prediction: A comparison with artificial neural networks, Neural Comput. Appl., vol. 20, pp. 151-159, 2011.

19) S. Visnu Dharsini & S. Babu (2022): Profit Suggestion Technique in Agronomics Using a New Heuristic-Based Barnacle Mating Honey Badger Algorithm I-BMHBA, Cybernetics and Systems, DOI: 10.1080/01969722.2022.2157612.,

20) S.Visnu Dharsini., S.Babu., ((2022): Advanced agronomics model with species classification, minimum support price prediction, and profit suggestion using enhanced deep learning strategy/ Knowledge and Information Systems https://doi.org/10.1007/s10115-022- 01787-1

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed tobeUniversityu/ s3of UGCAct,1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORT FOR UG /PG PROGRAMMES

(To be attached in the dissertation/project report)

1	NameoftheCandidate**(INB LOCKLETTERS)**	Durga Prasath J Pavithra M G Sudarsan S
2	Address ofCandidate	Chennai600003 Chennai600089 Chennai600089 **MobileNumber:** 8825774248, 8110043667, 7200638483
3	RegistrationNumber	RA2011003020283,RA2011003020238,RA2011003020256
4	DateofBirth	08/07/2002, 10/11/2002, 18/11/2002
5	Department	Computer Science and Engineering
6	Faculty	DR. VISNU DHARSINI S, M.E., Ph.D., Assistant Professor
7	Title of the Dissertation / Project	TOWARDS ACCELERATING & INTERPRETABLE WEATHER FORECASTING USING DEEP NETWORK LEARNING
8	Whether the above project / dissertation is done by	Individualor group : Group a) If the project / dissertation is done in group, then how many students together completed the project : 03 b) Mention the Name & Register number of other candidates : DURGA PRASATH J RA2011003020283 PAVITHRA M G RA2011003020238 SUDARSAN S RA2011003020256
9	Name and address of the Supervisor / Guide	Dr. VISNU DHARSINI S, M.E., Ph.D., ASSISTANT PROFESSOR Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai-89. **Mail ID:** visnudhs@srmist.edu.in **Mobile Number:** 8870115862

10	Software Used	Turnitin		
11	Date of Verification	26/04/2024		
12	**Plagiraism Details:(to attach the final report from the software)**			
Chapter	**Title of the Report**	**Percentage of Similarity index (including self citation)**	**Percentage of Similarity index (Excluding self citation)**	**% of plagiarism After excluding Quotes, Bibliography, etc.,**
1	TOWARDS ACCELERATING & INTERPRETABLE WEATHER FORECASTING USING DEEP NETWORK LEARNING	NA	NA	9%
Appendices		NA	NA	NA

I/ We declare that The above information have been verified and found true to the best of my/our knowledge.

Signature of the Candidate	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor /Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Dr.K.Raja

Name & Signature of the HOD

turnitin

Similarity Report ID: oid:3618:58281706

PAPER NAME

BATCH E6 REPORT.docx

WORD COUNT	CHARACTER COUNT
6540 Words	**46464 Characters**
PAGE COUNT	FILE SIZE
35 Pages	**323.8KB**
SUBMISSION DATE	REPORT DATE
Apr 26, 2024 12:31 PM GMT+5:30	**Apr 26, 2024 12:32 PM GMT+5:30**

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)

turnitin

Similarity Report ID: oid:3618:58281706

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	mathworks.com Internet	4%
2	researchgate.net Internet	3%
3	careerfoundry.com Internet	2%
4	export.arxiv.org Internet	1%
5	Masooma Ali Raza Suleman, S. Shridevi. "Short-Term Weather Forecas... Crossref	1%
6	researchonline.federation.edu.au Internet	1%
7	deepai.org Internet	1%
8	hedimanai.medium.com Internet	1%