RAINFALL PREDICTION IN MIZORAM USING LONG SHORT-TERM MEMORY NETWORKS (LSTM) AND PARTICLE SWARM OPTIMIZATION (PSO) TECHNIQUE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

C ZOREMSANGA

MZU REGISTRATION NO.: 1801172

Ph.D. REGISTRATION NO.: MZU/Ph. D./1268 of 28.08.2018

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

SCHOOL OF PHYSICAL SCIENCES

JUNE, 2024

RAINFALL PREDICTION IN MIZORAM USING LONG SHORT-TERM MEMORY NETWORKS (LSTM) AND PARTICLE SWARM OPTIMIZATION (PSO) TECHNIQUE

BY

C ZOREMSANGA

Department of Mathematics and Computer Science

Prof. Jamal Hussain

Submitted

In partial fulfillment of the requirement of the Degree of Doctor of Philosophy in Computer Science of Mizoram University, Aizawl

Dedicated to,

My family and friends. Your unwavering support was my anchor through this journey.

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

MIZORAM UNIVERSITY

Tanhril - 796004
Aizawl : Mizoram

Dr. Jamal Hussain
Professor

Post Box: 190
Gram: MZU
Mobile: +91 9436352389
Email: jamal.mzu@gmail.com
University Website: https://mzu.edu.in

CERTIFICATE

This is to certify that the thesis entitled "**Rainfall Prediction in Mizoram using Long Short-Term Memory Networks (LSTM) and Particle Swarm Optimization (PSO) Technique**", submitted by **C ZOREMSANGA** (Registration No: *MZU/Ph. D./1268 of 28.08.2018*), for the degree of Doctor of Philosophy (Ph. D.), **Department of Mathematics and Computer Science** at **Mizoram University** embodies the record of original investigation carried out by him under my supervision. He has been duly registered and the thesis presented is worthy of being considered for the award of the Ph. D. degree. This work has not been submitted for any degree in any other university/Institute.

……

PROF. JAMAL HUSSAIN

(Supervisor)

Place: Aizawl

Date: ……

DECLARATION
Mizoram University
June, 2024

I, **C ZOREMSANGA**, hereby declare that the subject matter of this thesis is the record of work done by me, that the contents of this thesis did not form basis of the award of any previous degree to me or to do the best of my knowledge to anybody else, and that the thesis has not been submitted by me for any research degree in any other University/Institute.

This is being submitted to the Mizoram University for the **degree of Doctor of Philosophy (Ph. D.) in Computer Science**.

(C ZOREMSANGA)
(Candidate)

(PROF. M. SUNDARARAJAN)
(Head of Department)
Dept. of Maths. and Comp. Sc.,
Mizoram University, Mizoram.

(PROF. JAMAL HUSSAIN)
(Supervisor)
Dept. of Maths. and Comp. Sc.,
Mizoram University, Mizoram.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Jamal Hussain, for his guidance, expert advice, and unwavering support throughout the research process. His expertise and insights greatly enriched the quality of this study and I am deeply grateful for his mentorship.

I am also thankful to the Department of Mathematics and Computer Science faculty members at Mizoram University for providing a stimulating academic environment and the resources necessary for conducting this research. I also thank the committee members of the Departmental Research Committee (DRC) and the Board of Studies (BOS) in Mathematics and Computer Science for providing valuable suggestions from time to time. I am indebted to my family and friends for their endless encouragement, love, and understanding throughout this journey. Their unwavering support sustained me during the challenging moments and motivated me to persevere.

I am profoundly grateful to the Almighty God for His guidance, unwavering support, and boundless blessings throughout this academic journey. I firmly believe this achievement would not have been possible without His divine intervention.

In conclusion, I am profoundly thankful to all those who have contributed directly or indirectly to this thesis. Your support has been invaluable, and I am truly honored and privileged to have had the opportunity to work on this research project.

Place: Aizawl

Date:.....

(C ZOREMSANGA)

PREFACE

The present thesis, titled "Rainfall Prediction in Mizoram using Long Short-Term Memory Networks (LSTM) and Particle Swarm Optimization (PSO) Technique" is the result of research conducted by the author under the guidance of Prof. Jamal Hussain, Professor in the Department of Mathematics and Computer Science at Mizoram University, Aizawl, Mizoram. The thesis is organized into seven chapters, with each chapter further divided into smaller sections.

The first chapter is an introduction to the context and rationale behind the study, accompanied by a statement of the problem and its objectives. This chapter also defines the scope and limitations of the study while highlighting its significance.

The second chapter presents an examination of the literature concerning rainfall prediction using deep learning since 2015. The literature is categorized based on the specific datasets utilized for training and testing the predictive models. This chapter thoroughly studies 45 papers from renowned publishers to gain insights into rainfall prediction. The classification of these papers focuses on the types of data implemented by the authors in their research. Furthermore, deep learning methods employed in these studies, the specific input data adopted for predicting rainfall, and metrics used to assess model performance are investigated extensively throughout this chapter. Additionally, both temporal and spatial distribution patterns of rainfall predictions are analyzed within this comprehensive exploration. The findings in this chapter indicate that deep learning techniques outperformed traditional machine learning models and shallow neural network architecture in the area of rainfall prediction. Therefore, deep learning methods are considered more favorable for this task.

The third chapter analyzes four LSTM models for predicting the monthly average rainfall in India from 1871 to 2016. The studied models include Model 1, which consists of a single LSTM hidden layer, Model 2 with a hidden LSTM layer and a hidden Dense layer, Model 3 consisting of two stacked LSTM layers and a dense output layer, and Model 4 constructed using two hidden LSTM layers, and one Hidden Dense layer. Model 4 outperformed the other models, achieving the lowest RMSE

value of 245.30 indicating that the stacked LSTM model has a strong ability to forecast rainfall patterns accurately. Additionally, increasing the timesteps further improved the performance of these models. In this study, all the proposed LSTM models outperformed the LSTM and RNN models proposed by ((Kumar *et al.*, 2019) in terms of RMSE. This shows that incorporating more neurons and additional timesteps can significantly enhance rainfall prediction accuracy.

The fourth chapter compares the predictive performance of the Bidirectional LSTM model by comparing it with the Vanilla LSTM, Stacked LSTM, and a benchmark model found in the existing literature. The main goal is to estimate the average rainfall for India one month ahead by utilizing the previous month's rainfall as a predictor. Furthermore, this study seeks to determine the optimal number of previous months' rainfall, epochs, and cells that produce the lowest RMSE when forecasting one-month rainfall into the future. The Bidirectional LSTM is found to achieve the lowest RMSE value. The findings indicated that achieving optimal prediction accuracy requires considering various factors. These factors include increasing the number of input timesteps, adding more LSTM cells, and extending the training epochs while closely monitoring for overfitting. Additionally, it is found that employing stacked LSTM layers and utilizing a Bidirectional LSTM approach significantly improved model performance compared to single-cell LSTM or Vanilla LSTM models.

In the fifth chapter of the study, various PSO-optimized deep learning and machine learning models including PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR are compared. The comparison was conducted using the Aizawl monthly rainfall dataset and the all-India monthly average rainfall dataset. This research is significant as it marks the first time machine learning and deep learning models have been employed to analyze Aizawl's rainfall data. Furthermore, this study also presents the first instance where the predictive capabilities of the models mentioned above (PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN) were compared in terms of their effectiveness in predicting rainfall patterns. In both datasets, the PSO-BiLSTM models outperformed the compared models, and the PSO-SVR model has the lowest

performance. Additionally, all PSO-optimized models outperformed those that utilized hyper-parameters determined through the grid search method or benchmarked from existing literature. This demonstrates that the Particle Swarm Optimization (PSO) algorithm is reliable for identifying optimal hyperparameters.

The sixth chapter presents an exhaustive comparison of twelve hybrid deep learning and machine learning models optimized using PSO to predict the amount of rainfall expected for the next day. To achieve this, four meteorological data from 1985 to 2018, collected by the Aizawl Weather Station in Mizoram, India, are utilized. Hybrid models demonstrated superior predictive abilities compared to the standalone models. Also, the BiLSTM models generally outperformed the compared model, proving that they were the most suitable model among the models under study. These experimental results demonstrate the viability of PSO-optimized hybrid deep learning models in rainfall prediction and show the advantage of the BiLSTM model over other deep learning models. This finding provides a foundational benchmark for future studies involving the Aizawl weather station data.

The seventh chapter of this thesis provides a summary and conclusion of the research discussed in the preceding chapters. It encapsulates the main findings of the study and highlight its key contributions. Additionally, a list of references is included at the end.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AE-MLP	Autoencoder Multilayer Perceptron
AFE	Arbitrage of Forecasting Expert
AI	Artificial Intelligence
AMD	Aizawl Meteorological Data
AMR	Aizawl Monthly Rainfall
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
AVNNET	Averaged Neural Networks
B-MAE	Balanced Mean Absolute Error
B-MSE	Balanced Mean Square Error
BiLSTM	Bi-directional Long Short-Term Memory
BMA	Bayesian Model Averaging
BPN	Back Propagation Network
BPTT	Backpropagation Through Time
BRNN	Bidirectional Recurrent Neural Network
CBP	Cascaded Back-Propagation
CLR	Clusterwise Linear Regression
CML	Commercial Microwave Link
CNN	Convolutional Neural Network
CNTK	Cognitive Toolkit
ConvGRU	Convolutional Gated Recurrent Unit

ConvLSTM	Convolutional Long Short-Term Memory
ConvRNN	Convolutional Recurrent Neural Network
COTREC	Continuous Tracking Radar Echo by Correlation
CSI	Critical Success Index
CTBT	Cloud-Top Brightness Temperature
CV	Coefficient of Variation
DAE	Denoising Autoencoder
DBLSTM	Double-Layer Bidirectional LSTM
DBN	Deep Belief Network
DBNPF	Deep Belief Network for Precipitation Forecast
DecTreeB	Decision Tree with Bagging algorithm
DenseROT	Dense Rotation
DFN	Dynamic Filter Network
DNN	Deep Neural Network
DSMIA	Dynamic Self-organizing Multilayer Network Inspired by the Immune Algorithm
EEMD	Ensemble Empirical Mode Decomposition
ELM	Extreme Learning Machine
ESN	Echo State Networks
ETS	Equitable Threat Score
EVS	Explained Variance Score
FAR	False Alarm Rate
FC-LSTM	Fully Connected Long Short-Term Memory
FNN	Feed-forward Neural Networks

FSS	Fractions Skill Score
GA-ConvGRU	Generative Adversarial ConvGRU
GBDT	Gradient Boosted Decision Tree
GRU	Gated Recurrent Unit
HRRR	High-Resolution Rapid Refresh
HSS	Heidke Skill Score
IEEE	Institute of Electrical and Electronics Engineers
IITM	Indian Institute of Meteorology
IMD	Indian Meteorological Department
IMR	India Monthly Rainfall
KNN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis
Logit Reg	Logistic Regression
LR	Linear Regression
LRN	Layer Recurrent Network
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAR-CNN	Multihead Attention Residual Convolutional Neural Network
MB	Multiplicative Bias
MGGP	Multi-Gene Genetic Programming
MIMO	Multi-Input Multi-Output
MISO	Multi-Input Single-Output
MLP	Multilayer Perceptron
MS-SSIM	Multi-Scale Structural Similarity

MSE	Mean Squared Error
MSMES	Master Super Model Ensemble System
MT-CNN	Multitasking Convolutional Neural Network
MT-GRU	Multitasking Gated Recurrent Unit
NCEP	National Centers for Environmental Prediction
NLP	Natural Language Processing
NNET	Neural Network
NOAA	National Oceanic and Atmospheric Administration
NRMSE	Normalized Root Mean Squared Error
NSE	Nash–Sutcliffe Efficiency coefficient
ORSS	Odds Ratio Skill Score
PAI	Platform for Artificial Intelligence
PCA	Principal Component Analysis
PCA-SVM	Principal Component Analysis- Support Vector Machine
PCC	Pearson Correlation Coefficient
PERSIANN-CCS	Precipitation Estimation from Remotely Sensed Imagery using an Artificial Neural Network Cloud Classification System
PFD	Probability of False Detection
POD	Probability of Detection
PRD	Percentage Root mean square Difference
PSO	Particle Swarm Optimization
QM	Quantile Mapping
R^2	Coefficient of determination
RB	Relative Bias

RBF	Radial Basis Function
RegTreeB	Regression Tree with Bagging algorithm
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROVER	Real-time Optical Flow using Variational methods for Echoes of Radar
RP	Random Prediction
S-SGD	Salp-Stochastic Gradient Descent
SAE	Stacked Autoencoder
SAE-AFM	Stacked Autoencoder - Anomaly Frequency Method
SCW	Severe Convective Weather
SDAE	Stacked Denoising Auto-Encoder
SEF	Stochastic Efficiency of Forecast
Seq2Seq	Sequence-to-Sequence
SGD	Stochastic Gradient Descent
SiLU	Sigmoid Linear Unit
SNN	Seasonal Neural Networks
SparseSD	Sparse Single Delta
SR	Standard Regression
SSA	Singular Spectrum Analysis
SSIM	Structural Similarity
SVM	Support Vector Machine

SVR	Support Vector Regression
TCN	Temporal Convolutional Networks
TL	Transfer Learning
TrajGRU	Trajectory Gated Recurrent Unit
TRMM	Tropical Rainfall Measuring Mission
TS	Threat Score
VAR	Vector Auto Regression
VECM	Vector Error Correction Model
WLSTM	Wavelet Long Short-Term Memory
WRF-NWP	Weather Research and Forecasting - Numerical Weather Prediction
WT	Wavelet Transform
XGB	XGBoost

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Rainfall, a fundamental meteorological parameter, shapes ecosystems, water resources, agricultural productivity, and regional development. Accurate rainfall prediction is essential for sustainable resource management, disaster preparedness, and climate adaptation strategies. Rainfall prediction is the process of forecasting the amount and timing of rainfall in a specific area. The ability to anticipate variations in rainfall patterns empowers decision-makers with valuable insights to make informed decisions, allocate resources efficiently, and mitigate the impact of extreme weather events (Barrera-Animas *et al.*, 2022; Ojo and Ogunjo, 2022; Pham *et al.*, 2020). Rainfall prediction is also crucial in countries like India that heavily depend on rainfall for water distribution and agriculture (Pal *et al.*, 2021). This chapter delves into the background of rainfall prediction, its associated challenges, and the motivation behind exploring novel approaches for enhancing accuracy in Mizoram, India.

Rainfall prediction has been a longstanding pursuit in the field of meteorology. Historically, forecasting relied on empirical methods that used historical data and statistical techniques to estimate future precipitation. While these methods provided a basic understanding of rainfall trends, they often failed to capture the complex interactions between meteorological variables and the intricate spatial and temporal rainfall patterns (Ridwan *et al.*, 2021). These limitations were exacerbated in regions with unique geographical features, such as Mizoram, where local topography, monsoons, and other regional influences contributed to highly dynamic rainfall patterns.

In recent decades, advancements in computational power and data availability have encouraged the development of more sophisticated prediction techniques. Numerical

weather prediction models, based on physical equations governing atmospheric processes, provided more accurate forecasts by simulating the behavior of weather systems. However, these models were computationally intensive and required vast data and computational resources (Zhang, Zeng, *et al.*, 2020).

1.2 Challenges in Rainfall Prediction

Predicting rainfall remains a formidable challenge due to the inherent complexity of atmospheric processes and the dynamic interactions between various meteorological factors (Dash *et al.*, 2018). Many factors influence rainfall patterns, including temperature, humidity, wind patterns, geographical features, and global climate phenomena like El Niño and La Niña. These factors operate across various temporal scales, leading to intricate patterns that are difficult to capture using traditional methods (Broni-Bedaiko *et al.*, 2019; Haq *et al.*, 2021; Sahoo and Yadav, 2021; Singh, Gnanaseelan, and Chowdary, 2017).

Traditional methods, such as regression-based approaches, suffer from oversimplification of the underlying processes and an inability to account for non-linear relationships. Moreover, these methods often struggle to handle missing data, outliers, and noise in the datasets, leading to suboptimal predictions (Zerouali *et al.*, 2023; Zhang, Zeng, *et al.*, 2020). Statistical methods such as Linear Regression (LR) and Autoregressive Integrated Moving Average (ARIMA) models performed well with linear and stationary data. However, they are less effective for rainfall prediction compared to machine learning approaches, as rainfall data tend to be complex and exhibit non-linear characteristics (Ni *et al.*, 2020).

1.3 Motivation for Advanced Prediction

The limitations of conventional methods and the increasing availability of large-scale meteorological datasets have incited interest in applying machine-learning techniques such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), and

Extreme Learning Machines (ELM), to predict rainfall. Machine learning leverages the computational power of computers to identify complex patterns in data, making it well-suited for tasks like time series prediction. For instance, Abbot and Marohasy (2012, 2014, 2017) used ANN to forecast monthly rainfall in Queensland, Australia, and found it performed better than statistical models. Similarly, Chattopadhyay and Chattopadhyay (2010) demonstrated that neural networks outperformed the ARIMA model in forecasting summer monsoon rainfall in India. Dash *et al.* (2018) predicted monsoon rainfall for Kerala, India, using K-Nearest Neighbour (KNN), ANN, and ELM, with ELM showing superior performance. Mekanik *et al.* (2013) recommended ANN over Multiple Regression (MR) analysis for predicting rainfall in Victoria, Australia, citing ANN's better generalization ability. Furthermore, Meyer *et al.* (2016) compared algorithms like Random Forests (RF), Neural Networks (NNET), Averaged Neural Networks (AVNNET), and SVM for detecting rainfall areas and assigning rainfall rates in Germany, finding NNET and AVNNET to be the most effective.

Traditional machine learning models have a limitation in capturing dependencies within data, as they typically consider only the current input during processing. Deep learning models, such as Recurrent Neural Networks (RNN), address this issue by capturing dependencies within input data. However, RNNs face the vanishing gradient problem, limiting their ability to capture long-term dependencies. Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber (1997), solve this problem by maintaining long-term dependencies. Despite their advantages, both RNN and LSTM transmit knowledge only in the forward direction. The Bidirectional LSTM (BiLSTM) model, introduced by Graves and Schmidhuber (2005), improves performance by capturing past and future hidden states through a bidirectional network, enabling the model to learn long-term dependencies more effectively.

Recently, deep learning models have gained significant interest in various domains such as sound classification (Abayomi-Alli *et al.*, 2022; Khamparia *et al.*, 2019; Piczak, 2015; Raza *et al.*, 2019; Zhang *et al.*, 2022), social network analysis (Abbas, 2021; Balaji *et al.*, 2021; Jeong *et al.*, 2020; Mithoo and Kumar, 2023; Tan *et al.*, 2019; Wu,

Zhang, *et al.*, 2020), human action recognition (Chen *et al.*, 2021; Gu *et al.*, 2021; Jaouedi *et al.*, 2020; Wan *et al.*, 2020; Wang *et al.*, 2018; Zhang *et al.*, 2021), image classification (Affonso *et al.*, 2017; Dimitrovski *et al.*, 2023; Krishnapriya and Karuna, 2023; Li *et al.*, 2019; Sahu and Kashyap, 2023; Wang *et al.*, 2020; Zhang *et al.*, 2019), and electricity load forecasting (Bouktif *et al.*, 2018; Hafeez *et al.*, 2020; Shirzadi *et al.*, 2021; Yazici *et al.*, 2022; Zulfiqar *et al.*, 2022).

Deep learning models have also been applied to rainfall prediction. For example, Sharma *et al.* (2023) used a deep learning model called U-Nets to predict heavy rainfall events at the district level in Assam, India, demonstrating superior performance compared to the Weather Research and Forecasting (WRF) model. However, their study used a small dataset and did not compare other deep learning models. Hassan *et al.* (2023) combined machine learning models with feature selection methods to predict categorical rainfall using data from 49 weather stations in Australia. They compared models including Naïve Bayes, Decision Tree, SVM, RF, LR, ANN, and LSTM, finding that ANN achieved a 90% accuracy score, outperforming the other models. The hyperparameters were tuned using an iterative refinement process, which might not have found the optimal settings. Sengoz *et al.* (2023) compared five machine learning models—Multiple Linear Regression (MLR), Gradient Boosting Regression (GBR), Random Forest (RF), Feed Forward Neural Network (FNN), and Convolutional Neural Network (CNN)—using weather data from Canadian, American, and European weather agencies. The study found that FNN and CNN models performed better for predicting rainfall, though they did not employ optimization methods to determine the best hyperparameters.

The motivation for exploring LSTM networks and their variants in rainfall prediction is mainly due to their ability to capture temporal relationships and patterns crucial in understanding and forecasting rainfall patterns. LSTMs and their variants have demonstrated success in diverse fields, such as natural language processing (Choo and Kim, 2023; Khurana *et al.*, 2023; Mahadevaswamy and Swathi, 2023; Qing-dao-er-ji *et al.*, 2020; Wang *et al.*, 2023; Zhai *et al.*, 2023), speech recognition

(Ahmed *et al.*, 2023; Bhaskar and Thasleema, 2023; Daouad *et al.*, 2023; Geng, 2023; Ying *et al.*, 2020; Yuan *et al.*, 2023), and financial prediction (Bathla *et al.*, 2023; Fang *et al.*, 2023; Han and Fu, 2023; Jin *et al.*, 2020; Li *et al.*, 2023; Usmani and Shamsi, 2023).

LSTM networks are a powerful means of capturing time-dependent patterns in data. Applying LSTMs to rainfall prediction promises improved accuracy by modeling the intricate dependencies inherent in meteorological time series data. However, their hyperparameters heavily influence their effectiveness, including the number of cells, input timesteps, epochs, learning rate, number of layers, and dropout rate (Yu *et al.*, 2019). Determining the optimal combination of these hyperparameters through trial and error or grid search can be both costly and time-consuming (Li *et al.*, 2020; Long *et al.*, 2023). These methods can also get stuck in local optima and do not guarantee the discovery of the ideal hyperparameters.

Recently, optimization algorithms like Particle Swarm Optimization (PSO) have gained recognition. Particle Swarm Optimization (PSO) is an optimization technique that efficiently searches for optimal hyperparameters within a given parameter space (Du *et al.*, 2022). By pairing LSTM with PSO, automatic fine-tuning of hyperparameters becomes possible, enhancing the model's predictive accuracy without manual tuning (Aslan *et al.*, 2023). He, Wu, and Si (2022) applied PSO to optimize LSTM hyperparameters using historical sales data from an e-commerce company, finding that the optimized LSTM outperformed LR, SVR, MLP, M5 model trees, RF, KNN, ARIMA, Transfer Learning (TL), and RNN in reducing prediction error rates. He, Chen, *et al.* (2022) also used PSO to optimize parameters of an improved attention mechanism with Double-Layer Bidirectional LSTM (DBLSTM) for predicting electric energy consumption, confirming its effectiveness compared to ELM, Echo State Networks (ESN), BiLSTM, double-layer LSTM, EECP-CBL, CNN-LSTM, and Attention LSTM. Kim and Cho (2021) proposed a PSO-optimized CNN-LSTM model for classifying roles in a Role-Based Access Control (RBAC) system, which outperformed several deep learning and machine learning models including Naive

Bayes, AdaBoost, KNN, SVM, MLP, Decision Tree, RF, ELM, DNN, CNN, LSTM, LSTM-DNN, and CNN-LSTM. Ma *et al.* (2019) introduced WAT-NCL-PSO, a hybrid of Wavelet Analysis Technique (WAT), Negative Correlation Learning Neural Network (NCL-NN) optimized using PSO for wind speed prediction, demonstrating superior performance over nine other models including BPNN, SVM, bagging, AdaBoost, RF, LSTM, SARIMA, EEMD-SVM, and WAT-NCL. Additionally, PSO has been employed in various fields to identify optimal hyperparameters, such as sentiment analysis (Machová *et al.*, 2020; Mandal *et al.*, 2019; Primartha *et al.*, 2019; Suddle and Bashir, 2022; Wei *et al.*, 2024), water level prediction (Buyukyildiz and Tezel, 2017; Mozaffari *et al.*, 2022; Ruma *et al.*, 2023; Salari *et al.*, 2021), and sound classification (Al Hwaitat *et al.*, 2022; İnik, 2023; Ji *et al.*, 2020; Liu and Yeh, 2024; Zhang *et al.*, 2022).

Despite the success of PSO in optimizing proposed models, existing studies typically optimize only the proposed models while the hyperparameters of compared models are often determined through grid search, trial-and-error, or existing literature. This inconsistency indicates a gap where all models, both proposed and compared, should be uniformly optimized using PSO and tested on the same dataset, highlighting an area needing further investigation. Therefore, we aim to investigate whether PSO-optimized LSTM can accurately predict rainfall quantity using the meteorological dataset of the all-India monthly average rainfall and the Aizawl Weather Station datasets.

1.4 Problem statement

The main research problem addressed in this study is the need to enhance the accuracy of rainfall prediction in Mizoram, India. Mizoram's diverse terrain, which includes hills, valleys, and rivers, contributes to microclimates that lead to localized and highly variable rainfall patterns. The region's proximity to the Bay of Bengal makes it susceptible to monsoonal influences and tropical cyclones. Moreover, the region's economy heavily depends on agriculture, making accurate rainfall prediction crucial for crop planning and resource allocation.

While providing a baseline understanding of rainfall trends, traditional rainfall prediction methods fail to effectively capture the relationships between meteorological variables and the resulting rainfall patterns. Statistical approaches often oversimplify the underlying complex processes, neglecting the non-linear and time-varying interactions that characterize meteorological systems. Numerical weather prediction models, while more sophisticated, require substantial computational resources and extensive input data, which can be challenging to obtain in regions with limited meteorological infrastructure like Mizoram.

The motivation to address the research problem lies in the potential of advanced machine learning techniques to overcome the limitations of conventional methods. Specifically, the LSTM neural network offers promise due to its ability to capture temporal dependencies in time series data. LSTM networks have succeeded in various domains, including natural language processing and speech recognition, where sequences and patterns play a crucial role. By applying LSTMs to rainfall prediction, there is an opportunity to leverage their capability to capture sequential patterns in meteorological data and provide more accurate forecasts.

While applying LSTM networks to rainfall prediction is a promising path, the complex nature of LSTM architectures and their hyperparameters necessitates optimization techniques that can fine-tune the models for improved predictive performance. This study addresses this by integrating the PSO technique with

LSTM networks. PSO, a nature-inspired optimization algorithm, aims to find optimal hyperparameters for the LSTM model to enhance its predictive accuracy.

Mizoram, situated in the north-eastern part of India, lies between the geographical coordinates of 21° 58' and 24° 35' N latitude and 92° 15' and 93° 29' E longitude. This state shares its borders with Tripura, Assam, and Manipur within India and a border spanning approximately 722 kilometres with the neighbouring countries of Bangladesh and Myanmar. Mizoram experiences significant yearly rainfall, averaging between 2,500 mm to 3,000 mm. The terrain in this region is characterized by hills and rugged landscapes with varying altitudes ranging from sea level to slightly over 2,000 m above sea level.

Mizoram is directly influenced by the South-West Monsoon, which typically brings an adequate amount of rainfall. The rainy season, or summer monsoon, usually begins in April, with heavy rains occurring from May to September, and extends until late October. In contrast, the winter season, from November to February, is generally dry with minimal rainfall.

More than 70% of Mizoram's population relies on agriculture for their livelihood, with the majority practicing shifting cultivation. Given the state's topography and the significant rainfall it receives, Mizoram is highly vulnerable to abnormal climate variability and long-term climate changes. Climate-related hazards can severely impact all sectors in Mizoram, as its socio-economic conditions are relatively underdeveloped compared to other states in India. Both urban and rural populations heavily depend on agriculture and allied sectors, which are highly susceptible to climate variability and long-term changes. This vulnerability can lead to numerous issues, such as alterations in the timing of field preparation, sowing, harvesting, and overall yield. Additionally, climate-related hazards like landslides and flash floods, which are prevalent in Mizoram, can severely affect other developmental sectors (Lallianthanga *et al.*, 2018). Thus, it is crucial to have accurate rainfall predictions that can capture the rainfall patterns.

A recent analysis of meteorological data in Mizoram was carried out by Ralte and

Sil (2021) to study drought patterns within the state. In addition, Saha, Das, *et al.* (2021) conducted a quantitative assessment of rainfall forecasts at the district level for Mizoram. Several researchers have also utilized machine learning and deep learning models to predict rainfall in various parts of India (Chakraverty and Gupta, 2008; Dash *et al.*, 2018; Gope *et al.*, 2016; Poornima and Pushpalatha, 2019). However, there has been no previous application of machine learning or deep learning models to analyze meteorological data specifically for Mizoram. This research aims to bridge the gap by exploiting the power of LSTM and the PSO technique to improve rainfall prediction in Mizoram and contribute to the field of meteorology.

1.5 System Setup

The following are the software and hardware configurations for conducting this study-

1.5.1 Software Used

- Python 3.10
- Jupyter Notebook 6.5.4
- Keras and TensorFlow 2.10.0

1.5.2 Hardware Used

- Apple Mac Studio with M1 Max, 10 core CPU, 32 core GPU and 32GB RAM.
- Apple iMac with M1, 8 core CPU, 8 core GPU and 8 GB of RAM.
- Two (2) Google Cloud NVIDIA T4 with 128 GB RAM.
- Two (2) Google Cloud NVIDIA P100 with 128 GB RAM.
- HP Proliant server with Two 10 core 2.2 GHz CPU, 512 GB RAM.

1.6 Objectives

The aims and objectives of this study are as follows -

1. To analyze and preprocess the weather data collected from the Aizawl weather station and Data Supply Portal maintained by the National Data Center, IMD, Pune.

2. To analyze the input features of the weather data and network parameters for the LSTM networks using the PSO technique.

3. Predict rainfall using the LSTM network with the network parameters determined by PSO.

4. To compare the prediction results of LSTM and PSO with other neural network models such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Radial Basis Functions (RBF).

1.7 Scope and limitations

This research focuses on applying LSTM neural networks and their variants in combination with PSO optimization for rainfall prediction in Mizoram. While the LSTM-PSO approach shows promise, the study does not account for all the meteorological parameters to predict the amount of rainfall due to a lack of meteorological data within the state. The scope includes historical data analysis, model development, hyperparameter tuning of the model using PSO, and performance evaluation to demonstrate the feasibility and effectiveness of the proposed approach.

This research does not investigate the optimal hyperparameter search for the learning rate, optimizers, and activation function of the model due to the requirement for significant computing power. Additionally, the study acknowledges that the quality and availability of historical data may impact the model's accuracy.

1.8 Significance of the study

The significance of this study lies in its innovative approach to improving rainfall prediction. By integrating LSTM neural networks and Particle Swarm Optimization, the research aims to contribute to several areas:

- The LSTM-PSO hybrid model has the potential to provide more accurate rainfall predictions, which can lead to better decision-making in sectors such as agriculture, water resource management, and disaster preparedness.
- The study's outcomes can be directly applied to Mizoram's context, aiding local authorities in making informed decisions and responding effectively to weather-related challenges.
- Integrating LSTM networks with PSO optimization contributes to advancing machine learning techniques in meteorology and offers insights into practical hybrid approaches.
- Accurately predicting rainfall patterns assists in climate change adaptation strategies, helping communities prepare for changing weather patterns.
- The research demonstrates the feasibility of applying advanced machine-learning techniques to complex environmental phenomena, encouraging further exploration in the field.

In conclusion, combining LSTM neural networks and the PSO method can improve rainfall prediction. By addressing the region's specific challenges, this research contributes to scientific understanding and practical decision-making processes related to rainfall prediction.

Chapter 2

Literature Survey [1]

2.1 Introduction

Rainfall prediction methods include various techniques to forecast precipitation occurrence, intensity, and distribution over a specified period and geographical area. These techniques have progressed from rudimentary empirical models to advanced data-driven and computational methodologies (Dotse *et al.*, 2024; Latif *et al.*, 2023). The nature of the prediction methods can be classified in different ways. They can be classified based on the methodology, such as empirical, statistical, machine learning, and deep learning methods. It can also be classified based on the duration of the prediction, such as short-term or long-term predictions. In this chapter, we present a survey of literature on rainfall prediction using deep learning methods since 2015, categorizing it based on the type of dataset used for training and testing the models. The category includes weather parameters, radar images, and satellite images.

2.2 Methodology

The survey process includes: (a) Gathering relevant papers that focus on utilizing deep learning techniques for predicting rainfall. (b) Conducting a thorough investigation and analysis of the gathered papers. Initially, popular digital libraries like ScienceDirect, IEEE Xplore, Springer, and Google Scholar were explored to find journal articles and conference proceedings using the following keywords-

[1] The content of this chapter is published as a review article in: Hussain, J., and Zoremsanga, C. (2021). "A Survey of Rainfall Prediction Using Deep Learning". 2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), 1–10. https://doi.org/10.1109/ICECIE52348.2021.9664730

["deep learning" OR "machine learning"] AND ["rainfall" OR "precipitation"] AND ["forecasting" OR "prediction"]

Using the above-mentioned search criteria, 45 papers were chosen for an in-depth examination. The inclusion criteria consisted of the following factors:

1. Papers published since January 2015.
2. Peer-reviewed journal and conference papers.
3. Papers that focused on forecasting rainfall.
4. Papers that employed deep learning techniques.

These steps ensured a comprehensive selection process to obtain valuable research materials for further study and analysis. During the second phase, a thorough examination was conducted on 45 papers carefully chosen from the previous steps. The focus of this investigation is based on the following research questions:

1. What deep-learning techniques were employed to forecast rainfall?
2. Which metrics were utilized to assess the effectiveness of these methods?
3. What sources provided the datasets used in these papers?
4. What are the weather parameters used for training the models?
5. Do the deep learning methods offer superior prediction capabilities compared to alternative approaches?

In this section, we have classified the task of predicting rainfall into different categories based on the data utilized to train and evaluate the models. We have identified three main types of data: weather parameter data, radar image data, and satellite image data. Among the 45 papers analyzed, 26 employed weather parameters, 13 utilized radar images, and 6 relied on satellite images for model training and testing purposes.

2.3 Deep learning for rainfall prediction

2.3.1 Prediction using weather parameters

This section presents research papers that utilized weather data gathered from Meteorological Observation Stations, including surface, upper air, and ocean observations, to predict rainfall. Table 2.1 summarizes these papers, including details on the deep learning techniques employed, compared methods, frameworks utilized, country of origin, temporal resolution, and metrics used to evaluate the models.

Saha, Santara, *et al.* (2021) and Saha *et al.* (2016a, 2017) utilized a deep learning method called stacked autoencoder to identify the predictors for early-late Indian summer monsoons, monsoons in the homogeneous regions of India and aggregate Indian summer monsoons. Models such as Regression Tree with Bagging algorithm (RegTreeB) and Decision Tree with Bagging algorithm (DecTreeB) are used to forecast the monsoon. They compared the proposed prediction models with the Indian Meteorological Department (IMD) monsoon prediction models.

Hernández *et al.* (2016) used a Denoising Autoencoder (DAE) to extract non-linear features from the input data and a Multilayer Perceptron (MLP) to predict rainfall one day ahead. They compared their proposed model with MLP, a naive approach, a Back Propagation network (BP), a Layer Recurrent Network (LRN), a Cascaded Back Propagation (CBP), an Ensemble Empirical Mode Decomposition (EEMD), and Feed-forward Neural Networks (FNN).

Gope *et al.* (2016) proposed a Stacked Autoencoder (SAE) for feature reduction and a cost-sensitive SVM for classifying heavy rainfalls. They predicted heavy rain during the monsoon season (June, July, August, and September) and showed that the proposed model performed better than the Stacked Autoencoder—Anomaly Frequency Method(SAE-AFM), Principal Component Analysis—Support Vector Machine (PCA-SVM), and Fisher linear discriminant analysis (LDA).

Zhang *et al.* (2017) forecasted a short-term rainfall using the Deep Belief Network for Precipitation Forecast (DBNPF) model and compared its performance with the

Radial Basis Function (RBF) neural network, SVM, Autoregressive Integrated Moving Average (ARIMA), Extreme Learning Machine (ELM), and SAE models. Zhang *et al.* (2018) also applied the DBNPF model to forecast rainfall in four provinces of China. In these studies, the DBNPF model was superior to the compared models.

Echo State Networks (ESN) and Multi-Gene Genetic Programming (MGGP) were introduced by Ouyang and Lu (2018) as potential methods for predicting monthly rainfall. These models were compared to the Support Vector Regression (SVR) approach across different lead times: 1 month, 3 months, and 6 months. The results of Wavelet Transform (WT), Singular Spectrum Analysis (SSA), and Ensemble Empirical Mode Decomposition (EEMD) are compared by the authors. Among the forecasting models, ESN demonstrated superior performance compared to SVR and MGGP, while SVR outperformed MGGP. However, all three models proved suitable for monthly rainfall prediction. Regarding the data preprocessing techniques, WT showed promise in forecasting short-term rainfall, whereas SSA exhibited better results for long-term rainfall prediction. On the other hand, EEMD performed poorly in comparison to WT and SSA. The most successful method was SSA-ESN, which could accurately predict rainfall up to 2 years ahead.

In a study by Qiu *et al.* (2017), the authors employed a Deep Convolutional Neural Network (CNN) to forecast short-term rainfall based on data collected from rain gauges. The CNN extracted the input data properties, while a dense layer was implemented for the prediction task. The presented multi-task CNN model was compared with other models, including Quantitative Precipitation Forecast (QPF), Linear Regression (LR), MLP, AE-MLP, LSTM, CNN, Multi-task MLP and Multi-task RNN. Through experimental analysis, it was found that the proposed model outperformed all of these alternative models significantly.

A deep learning model called Deep Belief Network (DBN) was utilized for precipitation forecasting in a study conducted by Du *et al.* (2018). The researchers compared the performance of DBN with that of SVM models optimized using various algorithms, including PSO. The results showed that DBN required less time compared

to the SVM model. Furthermore, it was observed that the SVM models are better with small input, while the DBN model is more appropriate for large input.

The study by Yonekura *et al.* (2018) focused on predicting short-term precipitation. It compared the effectiveness of various methods, including Deep Neural Networks (DNN), SVM, XGBoost (XGB), Random Forest (RF), and Random Prediction (RP). The results showed that DNN achieved the best performance in predicting rainfall compared to all the methods.

The study by Weesakul *et al.* (2018) explored the effectiveness of DNN in predicting monthly rainfall. Through various tests involving different hidden layers and nodes, the authors concluded that a DNN model with five hidden layers, each containing 128 nodes, was most suitable for this task. The findings indicated that the DNN model performed well in forecasting monthly rainfall when given a one-month lead time. However, it was noted that as the lead time increased, the accuracy of the model decreased.

In the investigation conducted by Tran and Song (2019), performance comparisons were made between LSTM, MLP, and Seasonal Neural Networks (SNN) for rainfall prediction purposes. The results revealed that LSTM outperformed both MLP and SNN in terms of performance. Based on these findings, it was suggested that LSTM shows excellent potential as a reliable method for estimating precipitation.

LSTM was utilized by Kumar *et al.* (2019) to forecast monthly rainfall and determine the optimal time lag for the model. RNN was also evaluated against LSTM model, and the models were evaluated across various regions of India. The results indicated that LSTM output better results than RNN in terms of various fitness measures, with 12 to 15 antecedent rainfall events providing valuable information.

In Miao *et al.* (2019), a deep neural network consisting of convolutional layers and an LSTM network was developed to enhance Monsoon precipitation prediction. The convolutional layers were employed to extract the spatial characteristics from the input that are fed into the LSTM layers. Geopotential height emerged as the most significant predictor during their investigation into predictor effectiveness. A comparison was

made between the Convolutional Long Short-Term Memory (ConvLSTM) model and other methods such as Quantile Mapping (QM), SVM, and CNN. The ConvLSTM network displayed superior performance in estimating precipitation compared to alternative methods.

A study conducted by Viswanath *et al.* (2019) introduced the concept of identifying break and active monsoon periods in the central region of the Indian subcontinent using LSTM and Sequence-to-Sequence (Seq2Seq) models. The Seq2Seq model comprises two LSTM units, a dense soft-max layer, and an attention mechanism. The authors categorized each day as dry, wet, or normal. Subsequently, they utilized this classification data daily to identify break or active monsoon spells. Their analysis considered daily rainfall data from June to September, spanning 1948 to 2014, for detecting these monsoon spells. Comparisons were made between the proposed models, SVM and K-Nearest Neighbour (KNN). The result shows that both the LSTM and Seq2Seq outperformed SVM and KNN in terms of performance. Moreover, it was found that Seq2Seq exhibited superior capabilities compared to LSTM, precisely when it came to detecting monsoon spells during this study.

ESN and DeepESN were utilized by Yen *et al.* (2019) to forecast rainfall using meteorological data in Southern Taiwan. The authors conducted a study comparing the performance of these models with Back Propagation Network (BPN) and SVR. To determine the most crucial parameter for predicting rainfall, the authors employed Principal Component Analysis (PCA). It was discovered that rainfall, pressure, and humidity were the most significant parameters. The experimental results indicated that ESN and DeepESN exhibited higher correlation coefficients than BPN and SVR models. Additionally, it was observed that DeepESN outperformed all other models in terms of accuracy and overall performance.

Manokij *et al.* (2019) outlined a novel approach using cascading deep learning techniques to classify rain/no-rain events and predict rainfall quantities. The authors employed a Convolutional Neural Network (CNN) as the primary classification model to distinguish between rainy and non-rainy conditions. To predict the amount of

rainfall, they utilized a Gated Recurrent Unit (GRU) based on the classified rain category. The authors implemented focal loss with sigmoid activation to address bias towards non-rain instances. They compared their proposed cascaded model against various existing models, including ARIMA, Autoencoder Multilayer Perceptron (AE-MLP), Multitasking Convolutional Neural Network (MT-CNN), and Multitasking Gated Recurrent Unit (MT-GRU). The results demonstrated that the proposed model achieved lower Root Mean Squared Error (RMSE) values for single time-step predictions than its counterparts. However, when applied to multi-step forecasting with a rolling mechanism encompassing six subsequent time steps, limitations in accuracy were observed.

The intensified LSTM model was introduced for rainfall prediction by Poornima and Pushpalatha (2019). To address the problem of the LSTM vanishing gradient, the authors made modifications to the LSTM architecture. They incorporated a sigmoid function and tanh function in the input gate and the candidate vector, both multiplied by input value. This alteration reduced the network's training time, leading to a higher learning rate and decreased losses. The performance of Intensified LSTM was compared against several other models such as Holt-Winters, ELM, ARIMA, RNN with Rectified Linear Unit (ReLU), RNN with Sigmoid Linear Unit (SiLU), and standard LSTM models. Based on their experimentation, it was observed that the intensified LSTM achieved better accuracy.

Two models, namely Wavelet Long Short-Term Memory (WLSTM) and CLSTM, were developed by Ni *et al.* (2020) to predict streamflow and rainfall. The WLSTM model combines wavelet transform with LSTM, while the CLSTM model utilizes CNN and LSTM. Compared to a three-layer MLP and LSTM, both WLSTM and CLSTM demonstrated superior performance in forecasting streamflow and rainfall. The integration of wavelet transforms and convolutional layers enhanced the accuracy of the LSTM model. These findings highlight the effectiveness of the hybrid models in improving forecasting accuracy for hydrological variables such as streamflow and rainfall.

A deep learning model, LSTM, and the K-Means clustering method were implemented by Zhang, Zeng, *et al.* (2020). The data was categorized into four groups using the K-means clustering method. Subsequently, different models were constructed using LSTM for each type of data. These models are compared with Frequency matching, Linear regression, SVM, and DBN based on RMSE and Threat Score (TS) to evaluate its effectiveness. The results showed that the proposed model reduced RMSE and enhanced TS for light and heavy rain conditions.

A deep CNN model was developed and trained by Zhou *et al.* (2019) to predict severe convective weather, including heavy rain, hail, convective gusts, and thunderstorms. Two databases were created using Severe Convective Weather (SCW) observations and National Centers for Environmental Prediction (NCEP) final analysis data. Compared to Logistic Regression (Logit Reg), RF, SVM, and MLP, the proposed CNN model demonstrated superior performance in forecasting SCW across China. These findings highlight the effectiveness of the deep CNN model over traditional machine learning algorithms for SCW prediction.

A study by Oswalt Manoj and Ananth (2020) proposed a ConvLSTM with a Salp-Stochastic Gradient Descent (S-SGD) algorithm to forecast rainfall in India. The S-SGD algorithm combines the Salp Swarm Algorithm (SSA) and Stochastic Gradient Descent (SGD) algorithm to determine the best weights for the ConvLSTM. The researchers employed the MapReduced for parallel processing to efficiently handle bulk input. The result of this proposed method was compared to other models such as ConvLSTM, Clusterwise Linear Regression (CLR), MLP, and Dynamic Self-organizing Multilayer Network Inspired by the Immune Algorithm (DSMIA). Results showed that the S-SGD-based ConvLSTM outperformed these models in terms of Percentage Root mean square Difference (PRD) and Mean Squared Error (MSE). The authors' findings highlight that their approach is more accurate in predicting rainfall patterns compared to existing methodologies. They improved forecast accuracy by leveraging a hybrid algorithm and parallel processing techniques. This research contributes significantly towards enhancing rainfall prediction methods for India.

A deep learning model was developed by Pudashine *et al.* (2020) using a two-layer LSTM model and trained on a dataset derived from a disdrometer. The researchers then utilized this model to enhance rainfall estimation by employing Commercial Microwave Link (CML) data. They could estimate rainfall using CML networks by analyzing the attenuation of the electromagnetic signal transmitted through rain. In their study, the authors compared the performance of the LSTM model with that of GRU and ANN models. The findings revealed that when it came to relative bias, GRU outperformed LSTM. However, in terms of RMSE, Mean Absolute Error (MAE), Coefficient of Determination (R2), and Coefficient of Variation (CV), the LSTM model slightly outperformed GRU and significantly outperformed ANN.

Hewage *et al.* (2021) introduced LSTM and Temporal Convolutional Networks (TCN) as methods for short-term rainfall forecasting using ten surface weather parameters. They conducted a comparison of these models with Standard Regression (SR), SVR, RF, ARIMA, Vector Auto Regression (VAR), Vector Error Correction Model (VECM), and Arbitrage of Forecasting Expert (AFE) models. The evaluation of the proposed models included two regression approaches: Multi-Input Multi-Output (MIMO) and Multi-Input Single-Output (MISO). MIMO-LSTM and MISO-LSTM demonstrated superior performance compared to the other models, prompting their selection as the proposed model. Given that there was minimal difference between MISO-LSTM and MIMO-LSTM in terms of results, along with its ease of handling and lower time/power consumption, MIMO-LSTM was chosen for further analysis. Subsequently, the author compared this selected model to the Weather Research and Forecasting - Numerical Weather Prediction (WRF-NWP) model. The findings revealed that MIMO-LSTM outperforms WRF-NWP by providing better predictions up to 12 hours ahead.

Table 2.1: Study of Rainfall Prediction Using Weather Parameters

Author	**Methods**	**Comparison**	**Country**	**Resolution**
(Gope *et al.*, 2016)	SAE-SVM, SAE-NN	SAE-AFM, PCA-SVM, Fisher LDA	India	Daily, 6-hours
(Hernández *et al.*, 2016)	DAE, MLP	MLP, Naïve, BPN, LRN, CBP, EEMD, FFNN	Colombia	Daily
(Saha *et al.*, 2016a)	SAE, RegTreeB	IMD Models	India	Monthly
(Saha *et al.*, 2016b)	Sparse AE, RegTreeB, DecTreeB	IMD Models	India	Monthly
(Ouyang and Lu, 2018)	ESN, MGGP	SVR	China	Monthly
(Qiu *et al.*, 2017)	MT-CNN	LR, MLP, AE-MLP, LSTM, CNN, MT-MLP, MT-RNN, ECMWF	Colombia, China	Daily
(Saha *et al.*, 2017)	SAE, RegTreeB	IMD Models	India	Monthly
(Zhang *et al.*, 2017)	DBNPF	RBF, SVM, ARIMA, ELM, Sparse AE	China	Daily
(Duong *et al.*, 2018)	LSTM	MLP, SNN	Vietnam	Monthly

Table 2.1: Study of Rainfall Prediction Using Weather Parameters

(Du *et al.*, 2018)	DBN	SVM, PSO-SVM	China	3 hours
(Weesakul *et al.*, 2018)	DNN	-	Thailand	Monthly
(Yonekura *et al.*, 2018)	DNN	SVM, XGB, RF, RP	Japan	10 minutes
(Zhang *et al.*, 2018)	DBNPF	SVM, RBF, ARIMA, ELM	China	-
(Kumar *et al.*, 2019)	LSTM	RNN	India	Monthly
(Manokij *et al.*, 2019)	CNN-GRU	ARIMA, AE-MLP, CNN, MT-GRU, CNN-GRU	Thailand	Hourly
(Oswalt Manoj and Ananth, 2020)	S-SGD-based convLSTM	convLSTM, CLR, MLP, DSMIA	India	Monthly, Quarterly, Yearly
(Miao *et al.*, 2019)	ConvLSTM	QM, SVM, CNN	China	Daily
(Ni *et al.*, 2020)	WLSTM, CLSTM	MLP, LSTM	China	Monthly
(Poornima and Pushpalatha, 2019)	Intensified LSTM	Holt–Winters, ELM, ARIMA, RNN-Relu, RNN-Silu, LSTM	India	-

Table 2.1: Study of Rainfall Prediction Using Weather Parameters

(Viswanath *et al.*, 2019)	LSTM, Seq2Seq	SVM, K-NN	India	Daily
(Yen *et al.*, 2019)	ESN, DeepESN	BPN, SVR, ECMWF	Taiwan	Hourly
(Zhang, Zeng, *et al.*, 2020)	LSTM	LR, SVM, DBN	China	-
(Zhou *et al.*, 2019)	CNN	Logit Reg, RF, SVM, MLP	China	6 hours
(Hewage *et al.*, 2021)	MIMO LSTM, MISO LSTM, MIMO TCN, MISO TCN,	SR, SVR, RF, ARIMA, VAR, VECM, AFE, WRF-NWP		3 hours
(Pudashine *et al.*, 2020)	Stacked LSTM	ANN, GRU	Australia	30 secs, 15 mins
(Saha, Santara, *et al.*, 2021)	SAE, RegTreeB, DecTree	IMD Models	India	-

Rainfall prediction using radar images has revolutionized the ability to forecast and monitor weather patterns. Radar technology employs radio waves to detect atmospheric precipitation, allowing meteorologists to create detailed images of rainfall intensity, movement, and distribution. By analyzing these radar images, meteorologists can make more accurate predictions about the timing, location, and amount of rainfall. This information is invaluable for various applications, including flood forecasting, drought management, and agriculture. With the advancements in radar technology and data analysis techniques, the understanding of complex weather systems has improved,

enabling the provision of timely and reliable rainfall predictions that help communities prepare for and respond to weather-related events effectively (Chen *et al.*, 2020). Researchers have dedicated extensive efforts to the comprehensive study and prediction of rainfall by harnessing the wealth of data gathered from strategically positioned radar stations. Table 2.2 provides a summary of research papers that utilize radar observations for rainfall prediction.

In their study, Shi *et al.* (2015) introduced the Convolutional LSTM (ConvLSTM) as a method for short-term rainfall prediction. This approach involved an expansion of the Fully Connected LSTM (FC-LSTM) by integrating convolutional layers within both the input-to-state and state-to-state transitions. The performance of ConvLSTM was evaluated with FC-LSTM and the Real-time Optical Flow using Variational methods for Echoes of Radar (ROVER). Notably, FC-LSTM exhibited suboptimal performance due to the spatial correlation inherent in radar data. While offering sharper predictions, ROVER exhibited the drawback of triggering more false alarms and displaying reduced precision compared to ConvLSTM. Consequently, the authors concluded that ConvLSTM excels in capturing spatiotemporal correlations and provides superior predictive capabilities compared to the ROVER algorithm.

The ConvLSTM technique was utilized Heye *et al.* (2017) to forecast short-term rainfall based on radar data. To determine the optimal hyper-parameters, such as the size of the convolutional kernel, number of convolutional filters, learning rate, and momentum, a search was conducted using Spearmint. The findings indicate that the traditional encoder-decoder method outperforms the attention model in terms of Probability of Detection (POD) and Critical Success Index (CSI), with a marginal difference in false alarms.

A Trajectory Gated Recurrent Unit (TrajGRU) was introduced by Shi *et al.* (2017) to predict short-term rainfall. The performance of TrajGRU was compared to other models, including Convolutional Gated Recurrent Unit (ConvGRU), Dynamic Filter Network (DFN), 2D and 3D Convolutional Neural Networks (CNNs), and two optical flow-based models called ROVER and its nonlinear variant. To address the issue of

imbalanced rainfall occurence with different rates of rainfall, a training and evaluation method using Balanced Mean Square Error (B-MSE) and Balanced Mean Absolute Error (B-MAE) are suggested by the authors. Experimental results validated that TrajGRU outperformed other models, indicating its effectiveness in short-term rainfall prediction. Furthermore, training the models with the balanced loss function yielded better results than training without it.

In their study, Singh, Sarkar, and Mitra (2017) investigated rainfall forecasting by integrating convolution operations into a vanilla recurrent neural network using a radar echo dataset. They compared the Convolutional Recurrent Neural Network (ConvRNN) model to ConvLSTM and Eulerian Persistence models. The authors argue that Conv-RNN can learn the characteristics of the Doppler weather radar phenomenon while utilizing fewer parameters than other hybrid methods. Additionally, convolutions within recurrence enable the encoding of spatiotemporal correlations.

A model called DeepRain was introduced by Kim *et al.* (2017) to forecast rainfall levels based on radar observations. The proposed method has been proven to have better accuracy in predicting rainfall compared to Linear Regression and FC-LSTM models. Furthermore, the results indicated that the two-stacked ConvLSTM model demonstrated greater stability than the one-stacked ConvLSTM.

To accelerate the development, testing, and deployment of new models, a study conducted by Samsi *et al.* (2019) explored a distributed learning method for training a precipitation nowcasting model. The study employed a data-parallel approach where the CNN and the training data were duplicated through several nodes. Notably, the CNN model utilized in this experiment was Fully Convolutional without dense layers. TensorFlow/Keras and Horovod frameworks were used to implement this model, utilizing 128 GPUs. Implementing this procedure significantly reduced training period reducing from an extensive 59 hours down to just approximately 1 hour. Furthermore, it was observed that as the number of GPUs increased up to 24 devices, there was a smooth decrease in validation loss. However, beyond 24 GPUs, erratic behaviour became apparent in the validation loss. This may be attributed to fewer training

images available for each device due to increased numbers. This research demonstrates how employing distributed learning can greatly expedite various stages of developing machine learning models while achieving desirable outcomes.

A Generative Adversarial ConvGRU (GA-ConvGRU) model was suggested by Tian *et al.* (2019). This model consists of two adversarial learning systems: a generator based on ConvGRU and a discriminator based on convolutional neural networks. The researchers used five radar echo images to forecast ten radar echo maps. The experimental outcomes demonstrated that GA-ConvGRU performed better than ConvGRU and optical flow methods.

In a study conducted by Bromberg *et al.* (2019), the effectiveness of U-Net CNN was compared to that of the Optical flow model, persistence model, and National Oceanic and Atmospheric Administration (NOAA) numerical one-hour High-Resolution Rapid Refresh (HRRR) in predicting short-term precipitation. The authors approached the forecasting issue as a translation problem from image to image, utilizing a sequence of radar images as input for their model. The results demonstrated that the proposed model outperformed all other models in performance.

A proposed model called dec-seq2seq has been introduced by Tran and Song (2019). This model is comprised of dec-TrajGRU, dec-ConvGRU, and dec-ConvLSTM. Compared to TrajGRU, ConvGRU, and ConvLSTM models, the dec-seq2seq models have significantly improved. Among these models, dec-TrajGRU outperformed the others. To address the issue of blurry images caused by loss functions like MAE or MSE, the authors have suggested using image quality assessment metrics such as Structural Similarity (SSIM) and Multi-Scale Structural Similarity (MS-SSIM). The experimental results indicate that combining SSIM with MSE and MAE yields the best results in terms of loss function. Additionally, it has been observed that the dec-seq2seq models are capable of handling high levels of uncertainty while maintaining their performance.

To enhance the precision of Doppler radar's ability to predict short-term rainfall, a research study by Zhang, Wang, *et al.* (2020) introduced an innovative approach

called Tiny-RainNet. This method combines Bi-directional Long Short-Term Memory (BiLSTM) and CNN techniques to obtain temporal and spatial data. The researchers analyzed Tiny-RainNet and other models such as ConvLSTM, LSTM, FC-LSTM, and AlexNet. The results revealed that Tiny-RainNet outperformed all the other models in terms of performance quality.

Table 2.2: Summary of papers that predict rainfall using radar images

Author	**Methodology**	**Comparison**	**Country**	**Temporal Resolution**
(Shi *et al.*, 2015)	ConvLSTM	FC-LSTM, ROVER	-	-
(Heye *et al.*, 2017)	ConvLSTM	Attention	USA	-
(Kim *et al.*, 2017)	ConvLSTM	LR, FC-LSTM	China	6 mins
(Singh, Sarkar, and Mitra, 2017)	ConvRNN, Multi-layer ConvRNN	ConvLSTM, Eulerian persistence	USA	-
(Shi *et al.*, 2017)	TrajGRU	ConvGRU, 2D and 3D CNN, ROVER (nonlinear variant)	China	-
(Agrawal *et al.*, 2019)	U-Net CNN	MRMS persistence, HRRR Optical flow method,	USA	2 mins
(Samsi *et al.*, 2019)	CNN	-	USA	-

Table 2.2: Summary of papers that predict rainfall using radar images

(Tian *et al.*, 2019)	GA-ConvGRU	Optical flow method, ConvGRU	China	-
(Tran and Song, 2019)	dec-ConvLSTM, dec-ConvGRU, dec-TrajGRU	TrajGRU, ConvGRU, ConvLSTM	China	6 mins
(Zhang, Wang, *et al.*, 2020)	CNN-BiLSTM	ConvLSTM, LSTM, FC-LSTM, AlexNet	China	6 mins
(Ayzel *et al.*, 2020)	RainNet (deepCNN)	CNN	Germany	5 mins
(Chen *et al.*, 2020)	ConvLSTM	COTREC, ConvLSTM (with cross entropy loss)	China	6 mins
(Yan *et al.*, 2020)	MAR- CNN	dual-channel CNN attention, dual-channel CNN, single-channel CNN, GBDT, SVM	China	6 mins

RainNet, a deep convolutional neural network designed for short-term precipitation forecasting using radar data, was developed by (Ayzel *et al.*, 2020). The RainNet model adopts a standard encoder-decoder structure with stacked CNN and incorporates skip connections between branches. Initially, RainNet predicts precipitation for a lead time of 5 minutes and can extend its prediction to 60 minutes by recursively utilizing the previous output as the subsequent input. The experimental findings demonstrate that RainNet consistently outperforms benchmark models such as the Rainymotion and

persistence method across all lead times within the 60-minute range.

Chen *et al.* (2020) implemented a star-shaped bridge architecture in ConvLSTM for precipitation nowcasting. This model was analyzed in comparison to Continuous Tracking Radar Echo by Correlation (COTREC), and ConvLSTM with cross-entropy loss. They also implemented the Group Normalization method to enhance the performance for ConvLSTM. Additionally, they employed a unique multisigmoid loss function. The results demonstrated that the proposed technique scored exceptional performance, surpassing existing methods in this field.

A short-term precipitation forecasting technique called Multihead Attention Residual Convolutional Neural Network (MAR-CNN) was proposed by Yan *et al.* (2020). This method utilized two CNN architectures. The first model focused on extracting deep characteristics from radar images, while the second model captured deep features from non-image inputs. To highlight the significant areas related to precipitation, the authors introduced multihead attention and implemented a residual connection to prevent the loss of global information caused by the attention layer. The performance of MAR-CNN was compared with other models such as dual-channel convolutional attention, dual-channel convolutional, single-channel CNN, Gradient Boosted Decision Tree (GBDT), and SVM. The results showed that MAR-CNN outperformed all the compared models in terms of prediction accuracy.

2.3.2 Prediction using satellite image

While commonly employed for rainfall observation, rain gauges, and radar systems are inherently constrained by their limited spatial coverage. In contrast, satellite observations offer a distinct advantage by providing extensive coverage across vast geographic regions at consistent intervals (Kidd *et al.*, 2009). This expansive coverage has been made possible by deploying multiple satellites to observe meteorological phenomena. For several decades, scientists and researchers have harnessed the invaluable data furnished by these orbiting instruments to enhance our understanding of weather patterns, enabling more accurate predictions of rainfall and other

meteorological events. This fusion of cutting-edge technology and comprehensive data acquisition has significantly advanced the field of meteorology, allowing us to anticipate better and respond to a wide range of atmospheric conditions. This section provides an overview of the existing literature that employs deep learning techniques to predict rainfall using satellite images. The papers found within the literature are summarized in Table 2.3.

In their study, Tao *et al.* (2016) employed a Stacked Denoising Auto-Encoder (SDAE) to address bias issues in satellite precipitation data. Specifically, they utilized the SDAE to enhance the accuracy of the Precipitation Estimation from Remotely Sensed Imagery using an Artificial Neural Network Cloud Classification System (PERSIANN-CCS). The evaluation of this model encompassed detecting Rain or No-Rain pixels and determining the amount of rainfall during both warm and cold seasons. The findings indicate that the proposed approach successfully identifies false alarm pixels within PERSIAN-CCS and effectively corrects biases in overall precipitation levels for both warm and cold seasons.

The Cloud-Top Brightness Temperature (CTBT) was utilized by Akbari Asanjan *et al.* (2018) to predict short-term precipitation. A deep learning algorithm called LSTM was introduced to forecast the upcoming CTBT image value. To estimate precipitation based on the forecasted CTBT image, the Precipitation Estimation using Remotely Sensed Information using Artificial Neural Networks (PERSIANN) algorithm was employed. The performance of this model was compared with other methods such as RNN with PERSIANN, Persistency method with PERSIANN, Farneback optical flow with PERSIANN algorithm, and Rapid Refresh (RAPv1.0). Results revealed that the proposed model exhibited better accuracy in short-term precipitation forecasting than its counterparts.

An ensemble approach using a deep neural network was suggested for rainfall prediction by Souto *et al.* (2018). To predict rainfall, the Convolutional LSTM method was compared to Bayesian Model Averaging (BMA) and Master Super Model Ensemble System (MSMES). The experiment results demonstrated that the proposed

model is 50% more accurate than the models it compared.

A precipitation nowcasting model was suggested by Sato *et al.* (2018), utilizing the PredNet network architecture and comparing it to the TrajGRU method. Instead of using the ConvLSTM unit, ConvGRU was employed as the PredNet unit. The experiment demonstrated that this proposed model attained exceptional performance in the MovingMNIST++ dataset and produced satisfactory results with actual precipitation data. Additionally, compared to the TrajGRU model, this new model consumed less GPU memory.

Wu, Yang, *et al.* (2020) suggested combining the Tropical Rainfall Measuring Mission (TRMM) 3B42 V7 satellite image, rain gauge output, and thermal infrared images to improve the accuracy of quantitative precipitation estimation. The authors combined CNN and LSTM models in this study to extract the merged dataset's spatial characteristics and time dependence. The accuracy of this approach was compared with that of CNN, LSTM, and MLP models. It was found that the CNN-LSTM model, which considered both time and space dependence of precipitation, outperformed the other models that only considered either spatial or temporal information. Furthermore, the authors demonstrated that under different levels/intensities of rainfall, the CNN-LSTM model could correct errors in TRMM data and improve its quality.

Convcast, a model for short-term precipitation prediction, was introduced by Kumar *et al.* (2020). This model used ConvLSTM layers to capture spatial and temporal features. Additionally, a 3D convolutional layer was employed to forecast precipitation. The prediction process involved inputting ten consecutive sequences of precipitation data at 30-minute intervals and then predicting the eleventh sequence. Furthermore, the predicted precipitation sequence was utilized to forecast precipitation up to 150 minutes in advance. The performance of Convcast was evaluated by comparing it with LSTM and four optical flow-based methods: Sparse Single Delta (SparseSD), Sparse, Dense, and Dense Rotation (DenseROT). According to the authors' experiment results, LSTM did not perform well when dealing with spatial and temporal information data. On the other hand, Convcast outperformed all of

Table 2.3: Summary of papers that predict rainfall using satellite images

Author	**Methods**	**Comparison**	**Country**	**Temporal Resolution**	**Lead time**
(Tao *et al.*, 2016)	SDAE	PERSIANN-CCS, Stage IV	USA	Hourly	-
(Akbari Asanjan *et al.*, 2018)	LSTM-PERSIANN	RNN-PERSIANN, Persistency PERSIANN, Farneback optical flow-PERSIANN, Rapid Refresh	USA	GOES-IR: 30 mins Q2 dataset: 5 mins	6 hours
(Sato *et al.*, 2018)	PredNet network, ConvGRU	TrajGRU	Japan	5 mins	10 frames (50 mins)
(Souto *et al.*, 2018)	ConvLSTM	BMA, MSMES	Brazil	Daily	1 day
(Kumar *et al.*, 2020)	ConvLSTM	LSTM	-	30 mins	150 mins
(Wu, Yang, *et al.*, 2020)	CNN-LSTM	CNN, LSTM, MLP	China	Daily	1 Day

the compared models. In summary, the proposed model Convcast utilizes advanced techniques like ConvLSTM layers and 3D convolutional layers for accurate short-term precipitation forecasting. Its superior performance compared to LSTM suggests its effectiveness in handling spatio-temporal data.

Figure 2.1 illustrates the temporal distribution of the 45 papers chosen for analysis. The graph demonstrates a rising pattern in the number of papers published on rainfall prediction utilizing deep learning techniques. Notably, there was a substantial growth in 2019, accounting for 51% of all papers. This growth signifies an escalating interest in employing deep-learning approaches for rainfall prediction. The decrease in paper count from 2019 to 2020 is because data was only considered until June 2020.

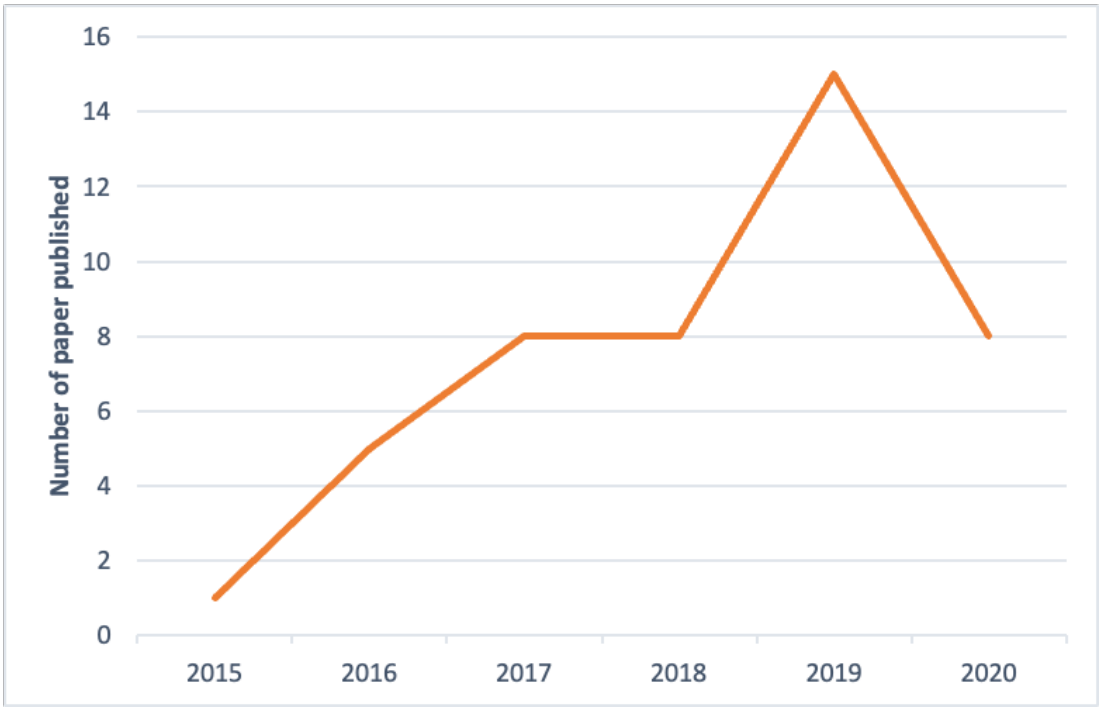

Figure 2.1: Frequency of publication during the year 2015-2020

2.3.3 Deep learning methods used for rainfall prediction

The utilization of different types of deep learning methods for rainfall prediction is presented in Figure 2.2. This figure reveals that LSTM (10 papers) and ConvLSTM (9 papers) are the most frequently employed techniques. When faced with many input variables, authors are found to employ AE and SAE to identify predictor variables. By utilizing convolution operators, it becomes possible to acquire spatial information in addition to temporal information. Thus, in cases where the input comprises a satellite image or radar image, convolutional layers are predominantly used by authors to learn the input's features. Our findings indicate that deep learning models outperform traditional machine learning models in terms of accuracy when it comes to rainfall prediction, as they effectively capture temporal and spatial information from the input data. Traditional machine learning models fall short when retaining past information is necessary in cases involving temporal data as inputs.

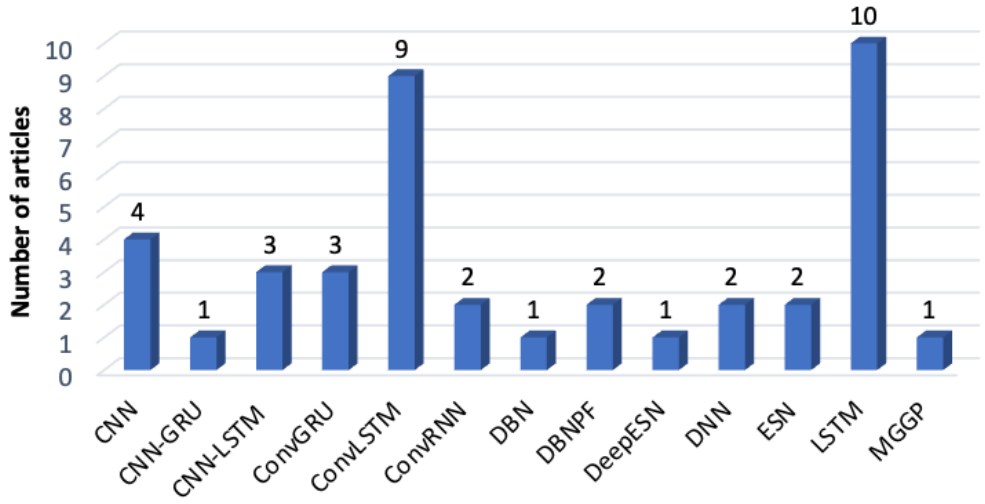

Figure 2.2: Frequency of deep learning methods in the surveyed papers

2.3.4 Spatial distribution of studies

Figure 2.3 depicts the global distribution of the papers analyzed. This representation was based on the countries where weather data collection and rainfall prediction occurred. China has the most published papers, with 17, followed by India with 9 and the USA with 6.

2.3.5 Software used

Figure 2.4 presents the details regarding the software for implementing the machine learning models, including their type and frequency. Most papers (57.8%) explicitly mention the software utilized, while 42.2% did not provide such information. Our analysis revealed that MATLAB emerged as the most commonly used software in seven papers, followed by Tensorflow (six papers) and Keras (five papers). It appears that authors prefer Tensorflow and Keras due to their user-friendly interfaces and extensive support for machine learning, which are also offered by MATLAB. We also observed that Theano, Python, PyTorch, Platform for Artificial Intelligence (PAI), MapReduce, Hovorod, Microsoft Cognitive Toolkit (CNTK), and Chainer were utilized in this study's paper.

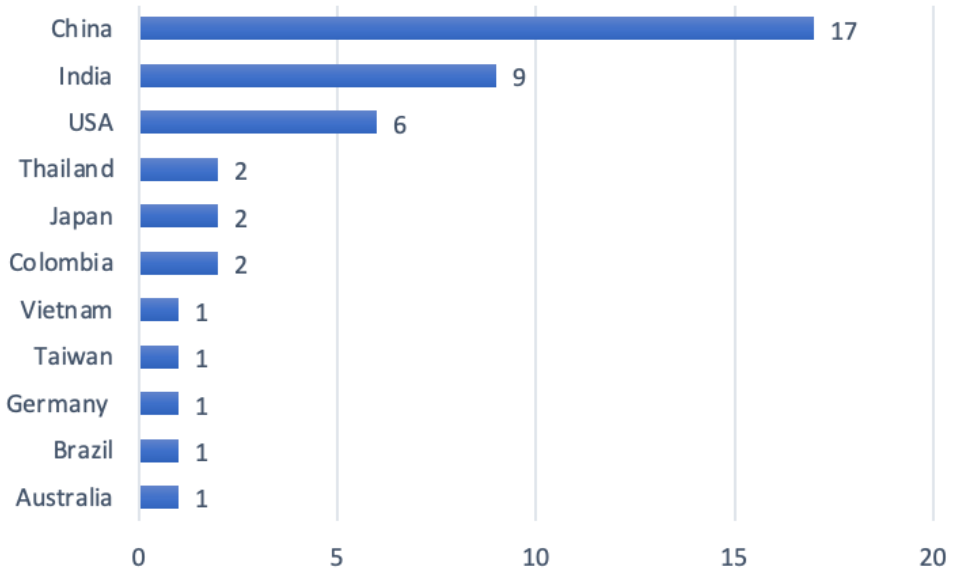

Figure 2.3: Global distribution of the papers under study

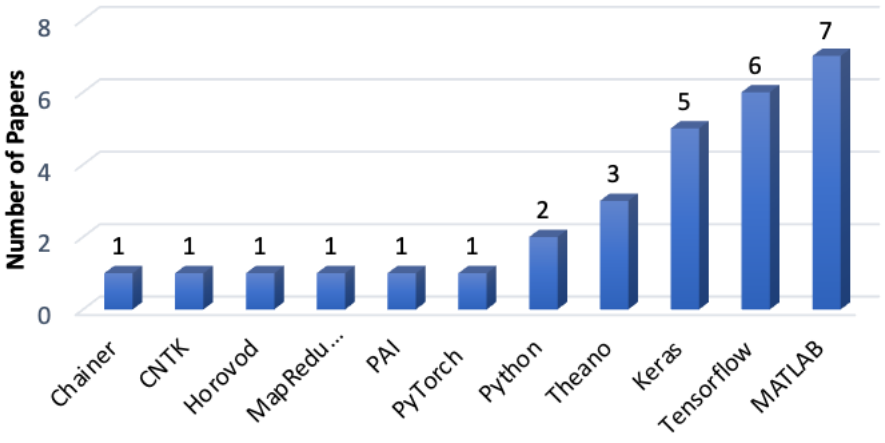

Figure 2.4: Software used for implementing the machine learning models

2.3.6 Performance metrics

Table 2.4 presents the distribution of various performance metrics utilized in the papers that were studied. Some papers did not specify the metrics employed, so they have been excluded from this analysis. While some authors relied on a single metric, others used multiple metrics to compare model performance. It was observed that RMSE emerged as the most frequently utilized metric (appearing in 20 papers), followed by MAE (in 13 papers), CSI (in 10 papers), and False Alarm Rate (FAR) (also in 10 papers). Other metrics included Correlation (CORR), B-MAE, B-MSE, CV, Equitable Threat Score (ETS), Explained Variance Score (EVS), Fractions Skill Score (FSS), Heidke Skill Score (HSS); Multiplicative Bias (MB), Multi-Scale Structural Similarity (MS-SSIM), Normalized Root Mean Squared Error (NRMSE), Odds Ratio Skill Score (ORSS), Pearson Correlation Coefficient (PCC), Probability of False Detection (PFD), Percentage Root mean square Difference (PRD), Stochastic Efficiency of Forecast (SEF), Nash–Sutcliffe Efficiency coefficient (NSE), Relative Bias (RB), SSIM, Success ratio and Variance.

2.4 Conclusions

Rainfall prediction remains challenging due to weather variables' intricate and nonlinear characteristics. However, even though rainfall has a significant impact on our daily lives, it continues to be an area of extensive research. In this study, we have examined 45 papers published by reputable publishers to gain insight into different approaches for predicting rainfall. One key aspect we analyzed is the type of data utilized by authors in their predictions. Rainfall and other weather phenomena are typically used for various weather parameters, radar images, or satellite images. We also investigated the deep learning methods employed, the types of input data used in these predictive models, the metrics employed for evaluating model performance, and the software used for implementation. Additionally, we examined both temporal and spatial aspects related to these studies.

Table 2.4: Metrics used in the paper under study

Sl. No.	Metrics	Papers	Sl. No.	Metrics	Papers
1	RMSE	20	18	RB	2
2	MAE	13	19	B-MAE	1
3	CSI	10	20	B-MSE	1
4	FAR	10	21	CV	1
5	POD	9	22	EVS	1
6	CORR	8	23	FSS	1
7	MSE	7	24	MB	1
8	HSS	5	25	MS-SSIM	1
9	TS	5	26	NRMSE	1
10	F1-score	4	27	ORSS	1
11	NSE	4	28	PCC	1
12	Precision	4	29	PFD	1
13	Recall	4	30	PRD	1
14	Accuracy	3	31	SEF	1
15	Bias	3	32	SSIM	1
16	ETS	3	33	Success Ratio	1
17	R-Squared	2	34	Variance	1

This study revealed that research interest in predicting rainfall using deep learning methods has grown significantly since 2015. LSTM and ConvLSTM are the most used methods in this study, and RMSE is the most common metric for analyzing the results. Additionally, MATLAB, Tensorflow, and Keras are the software most commonly utilized by the authors. However, in the majority of papers, model hyperparameters are determined through trial and error, with few employing optimization algorithms to achieve optimal settings. Future study could explore integrating optimization algorithms with deep learning methods to develop hybrid models. It was also found that weather data in Mizoram has not been used to predict rainfall.

Chapter 3

Prediction of monthly rainfall using LSTM [2]

3.1 Introduction

Rainfall plays a crucial role in providing fresh water, and its impact extends to all forms of life on our planet. Additionally, it has significant implications for transportation, agriculture, and the management of renewable energy (Markuna *et al.*, 2023). The complexity of predicting rainfall is mainly due to the many factors that influence it (Dash *et al.*, 2018). This chapter presents a study on four variations of LSTM models to predict the monthly average rainfall in India. Furthermore, this study compares the performance of these proposed models with a benchmark model found in the existing literature by Kumar *et al.* (2019).

In recent years, the LSTM model has gained significant attention in the research community for its ability to predict time series data accurately. One of the critical challenges in utilizing LSTM effectively lies in determining the optimal hyperparameters for the model. These hyperparameters include the number of LSTM layers, the number of cells within each layer, the number of epochs, and the appropriate input timesteps. This chapter aims to identify the best hyperparameters for an LSTM model to forecast one month ahead of rainfall using the all-India monthly average rainfall dataset from 1871 to 2016. Through extensive experimentation and analysis, this chapter aims to find the optimal configuration of the models and provide valuable

[2] The content of this article is published as a research article in: Zoremsanga, C., and Hussain, J. (2023).“A Comparative Study of Long Short-Term Memory for Rainfall Prediction in Indi”. In S. N. Singh, S. Mahanta, and Y. J. Singh (Eds.), Proceedings of the NIELIT’s International Conference on Communication, Electronics and Digital Technology, Lecture Notes in Networks and Systems (pp. 547–558). https://doi.org/10.1007/978-981-99-1699-3_38

Table 3.1: Statistical Illustration of Average Monthly Rainfall (1871 - 2016)

Area		All-India
Months		1752
No. of features		1 (Rainfall)
	Mean	904.94
	Standard Deviation	951.71
	Minimum	3
Rainfall (in 10^{th} of mm)	**Maximum**	3460
	25^{th} Percentile	146.75
	50^{th} Percentile	425.0
	75^{th} Percentile	1632.5

insights for improving LSTM-based rainfall predictions.

3.2 Study area and data source

In this study, the monthly rainfall data for all-India was obtained from the dataset created by Kothawale and Rajeevan (2017). This dataset includes average monthly, seasonal, and annual rainfall values from 1871 to 2016, weighted according to the geographical areas they cover. The dataset comprises thirty-six meteorological sub-divisions in India, as illustrated in Figure 3.1. For each sub-division, rainfall data from thirty rain-gauge stations within that area were considered during the compilation process. However, the four sub-divisions located in the Himalayas were excluded due to insufficient rain gauge availability and limited coverage in hilly regions. Similarly, islands in both the Bay of Bengal and the Arabian Sea were also not included in this dataset to ensure a continuous data flow (Kothawale and Rajeevan, 2017).

Figure 3.2 illustrates the average monthly precipitation in India from 1871 to 2016 in tenths of millimetres. The set of data contains 1752 months of precipitation data.

Table 3.1 gives the statistical description of the dataset for all-India average monthly rainfall. The mean monthly precipitation for all-India is 904.94, with a standard deviation of 951.71. The least rainfall is 3, with the highest amount at 3460. This study trains and tests four LSTM models using this dataset.

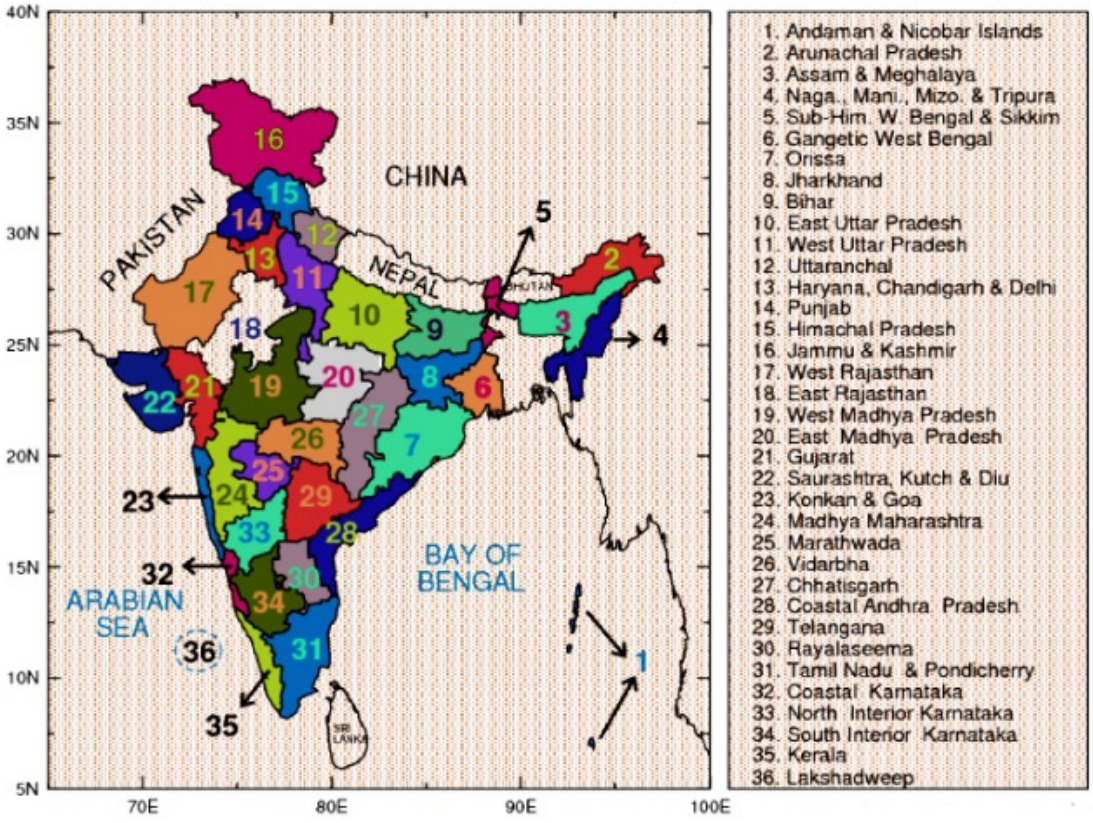

Figure 3.1: Meteorological Sub-Divisions of India

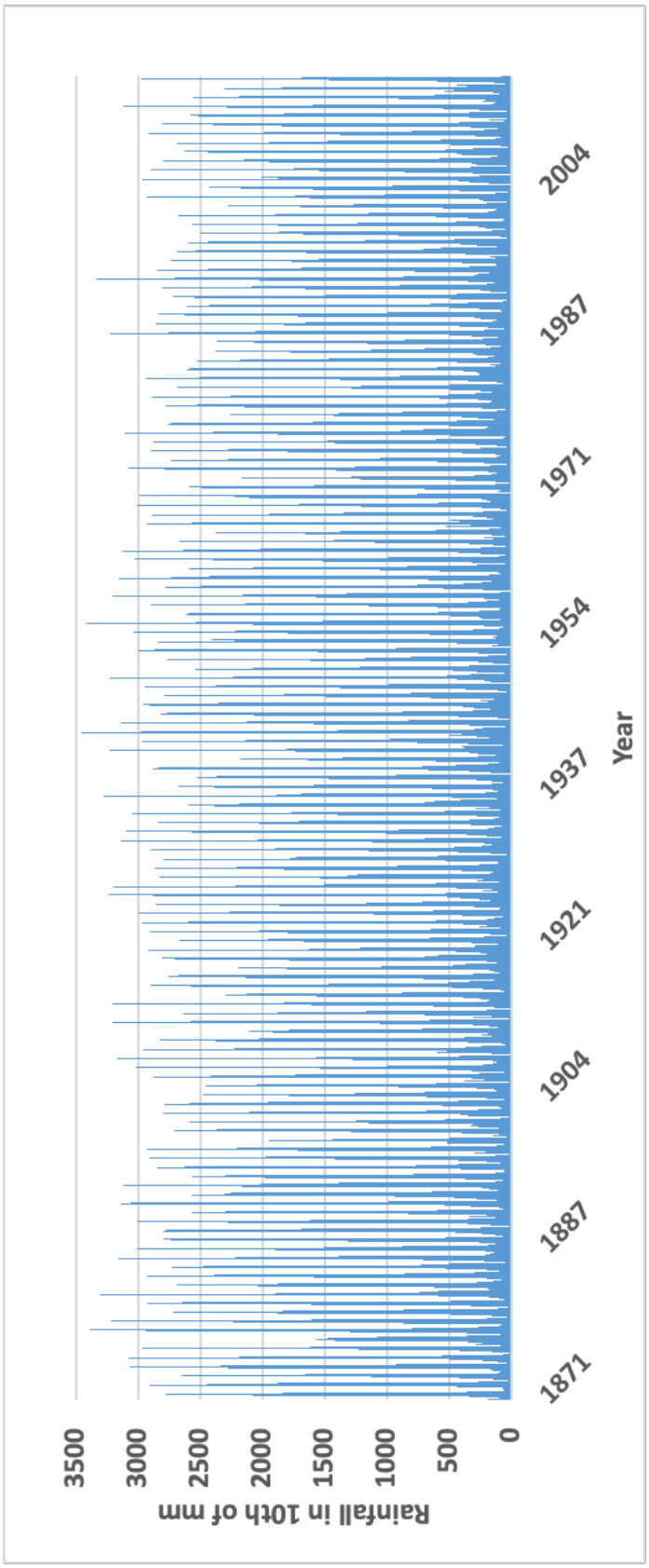

Figure 3.2: Average Monthly Rainfall in India (1871 - 2016)

3.3 Methodology

3.3.1 Data preprocessing

Training and testing data

The division of rainfall data into training and testing sets is a common practice in machine learning. The rainfall data is divided into two parts - one for training the models and the other for evaluating their performance. In this study, 70% of the available data was used to train the models, while the remaining 30% was used to assess their ability to predict monthly rainfall. Comparing the predicted rainfall values with the actual measurements in the testing set allows us to determine the accuracy of the models.

Normalization

To enhance the predictive capabilities of the models, the Min-Max normalization technique is utilized to transform the data. The normalization involved scaling the data values from 0 to 1, following the formula given in equation (3.1). To ensure that no information from the training dataset leaks into the testing dataset, the normalization coefficients are estimated based only on the training data. However, for the final predictions, the inverse transformation of the data is performed to convert the data back to its original scale.

$$D' = d_{min} + (d_{max} - d_{min}) + (x - x_{min})/(x_{max} - x_{min}) \tag{3.1}$$

The normalized value D' is obtained by using equation (3.1) given above. In this formula, the minimum and maximum values of the data (0 and 1, respectively) are represented as d_{min} and d_{max}. The variable x represents the value that needs to be scaled, while x_{min} and x_{max} represent its minimum and maximum values.

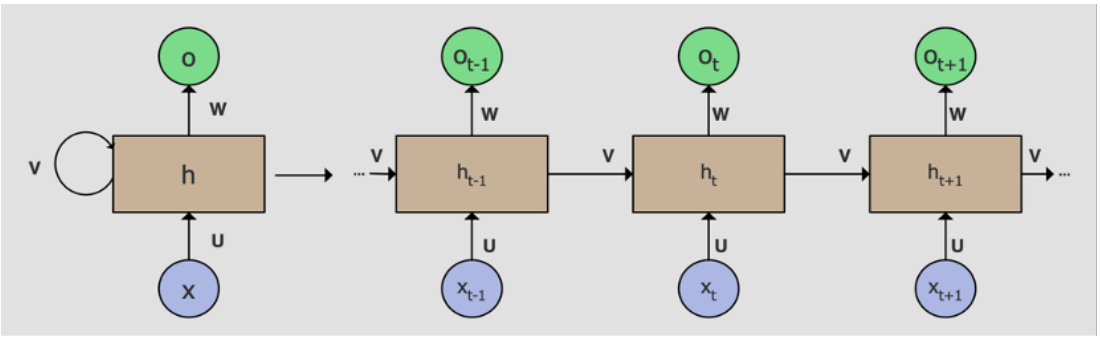

Figure 3.3: The standard RNN and unfolded RNN

3.3.2 RNN

A proposal by Rumelhart *et al.* (1986) introduced the concept of an RNN, a form of artificial neural network designed to process sequential or time series data. These deep learning algorithms are commonly employed in various applications such as language translation, Natural Language Processing (NLP), speech recognition, and image captioning. Unlike traditional deep neural networks, RNN possess memory that enables them to utilize information from previous inputs to influence current inputs and outputs.

Another characteristic of RNN is that they share parameters across each network layer, distinguishing them from feedforward networks with different weights for each node. The Backpropagation Through Time (BPTT) algorithm determines the gradients within RNN. This algorithm resembles traditional backpropagation but is specifically designed for sequence data. BPTT aggregates errors at each time step, unlike feedforward networks, which do not require error summation as they lack parameter sharing. Figure 3.3 illustrates the fundamental structure of both a standard RNN and an unfolded RNN. In this figure, x_t represents the input while h_t signifies the hidden state, and o_t denotes the network's output. W, U, and V are all parameters shared across timesteps within the network.

3.3.3 LSTM

Hochreiter and Schmidhuber (1997) introduced LSTM as a type of RNN utilized in deep learning and NLP. Its primary function is to develop robust models capable of comprehending intricate data sequences, including text, audio, and video. What sets LSTM apart from other neural networks is its ability to retain long-term dependencies. This ability to retain long-term dependencies is made possible through the utilization of specialized units known as "memory cells", which store and access information over time. Researchers have extensively employed LSTM for various tasks involving understanding temporal patterns, such as time series forecasting. By analyzing how data evolves, LSTM networks can effectively predict future events. LSTM has been employed in various classification tasks, such as analyzing sentiment (Feng *et al.*, 2019) and natural language processing (Ying *et al.*, 2020). It can also be utilized for generating sequences, like producing text (Song *et al.*, 2019) or translating languages (Qing-dao-er-ji *et al.*, 2020). In addition, LSTM models are extensively used for forecasting time series data (Abdel-Nasser and Mahmoud, 2019; Bouktif *et al.*, 2018; Broni-Bedaiko *et al.*, 2019) as well as recognizing images (Dorbe *et al.*, 2018). A recent study by Jin *et al.* (2020) also implemented an LSTM model to predict the closing price of stocks based on investors' emotional tendencies.

The fundamental structure of LSTM is illustrated in Figure 3.4, and it consists of several components: an input gate (i_t), a forget gate (f_t), an output gate (o_t), and a memory cell with a state c_t at time t. The input gate is responsible for determining which information should be stored in the memory cell, while the output gate determines what information should be retrieved from it. On the other hand, the forget gate plays a crucial role in discarding irrelevant information. The memory cell is perhaps the most critical element of the LSTM architecture. It is a continuous vector that can retain and retrieve information over extended periods. This capability enables the network to store and access data across lengthy sequences effectively. To regulate how information flows into and out of the memory cell, both input and output gates

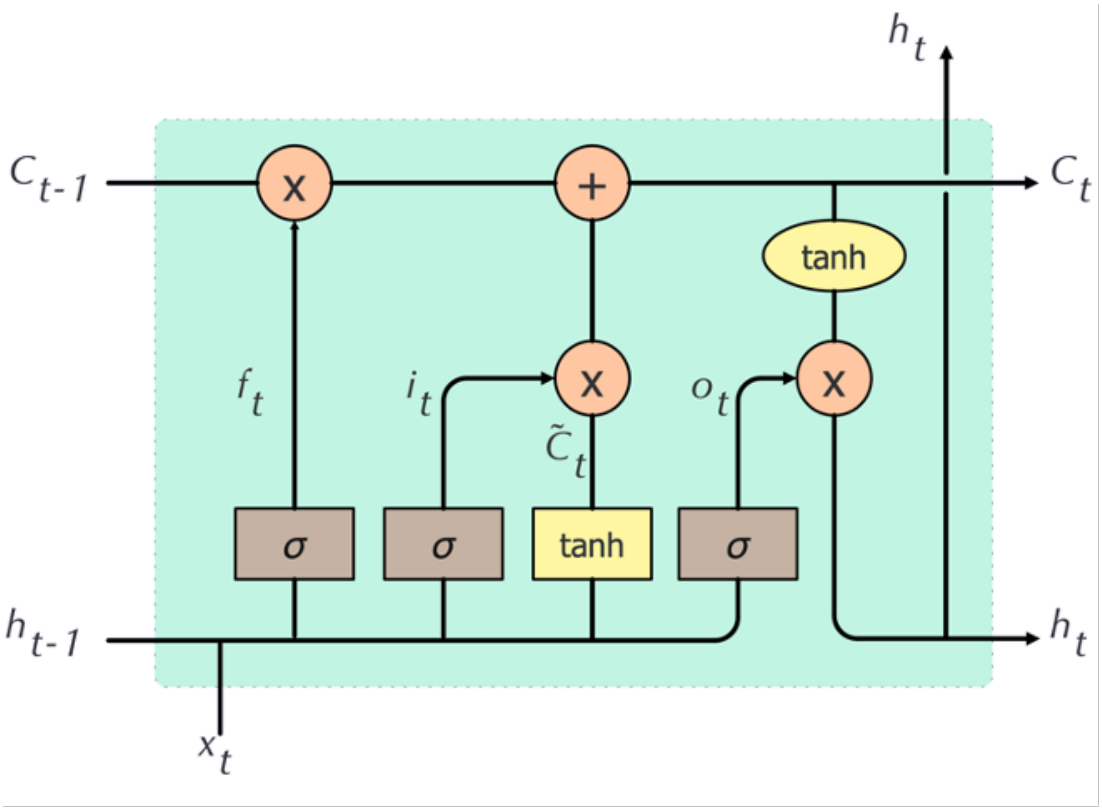

Figure 3.4: The basic architecture of LSTM Block

employ fully connected neural network layers that utilize sigmoid functions coupled with pointwise multiplication operations (Olah, 2015).

The memory cell comprises two distinct categories of weights: input and recurrent weights. The input weights control the transmission of information from the input gate to the memory cell. In contrast, the recurrent weights regulate the transmission of information from the memory cell to the output gate (Abdel-Nasser and Mahmoud, 2019). Equations (3.2), (3.3), (3.4), (3.5), (3.6), and (3.7) provide a representation of how information flows and the functioning mechanism of these gates.

$$f_t = \sigma\left[W_f \cdot (h_{t-1}x_t) + b_f\right] \tag{3.2}$$

$$i_t = \sigma\left[W_i \cdot (h_{t-1}x_t) + b_i\right] \tag{3.3}$$

$$\tilde{C}_t = \tanh\left[W_c \cdot (h_{t-1}x_t) + b_c\right] \tag{3.4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.5}$$

$$o_t = \sigma\left[W_o \cdot (h_{t-1} x_t) + b_o\right] \tag{3.6}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{3.7}$$

The range of values for the forget gate (f_t), input gate (i_t), and output gate (o_t) is between 0 and 1, which is determined by the sigmoid function σ. The variables $C_{t-1}, \tilde{C}_t$, and C_t represent the previous state of the cell, the new candidate value of the cell, and the updated cell state, respectively. At time *t*, h_t represents the output of the LSTM block. The weight matrices are denoted as W_f, W_i, W_c, and W_o , while b_f, b_i, b_c and b_o are referred to as bias vectors. The * symbol signifies element-wise multiplication between two vectors. Additionally, sigmoid activation functions are represented as σ, while tangent activation functions are denoted as tanh (Bouktif *et al.*, 2018).

3.4 Model development

In this study, four LSTM models shown in Table 3.2 were compared to predict the average monthly rainfall in India. These models are referred to as LSTM Model_1, LSTM Model_2, LSTM Model_3, and LSTM Model_4. The grid search method is used to evaluate the number of input timesteps, LSTM cells, and epochs required for each model. The models were compiled using the Adam optimization algorithm, and MSE was used as the loss function. To account for variability in results, we repeated each model ten times using the same training and evaluation dataset. The average of these ten outputs was then calculated to determine the final performance of each model.

The optimal parameters returned by the grid search method are also given in Table 3.2. The number of input time steps is searched in the range of 1 to 30, the number of LSTM cells in the range of 1 to 100, and the number of epochs in the range of 50 to 1000. The first model, LSTM Model_1, comprises an LSTM with a single hidden layer containing fifty cells. It also has a Dense output layer with just one neuron. LSTM

Table 3.2: Summary of LSTM Models under study

LSTM Model	Input Timesteps	Model Architecture	Epochs
RNN (Kumar *et al.*, 2019)	20	RNN (1) – Dense (1)	500
LSTM (Kumar *et al.*, 2019)	20	LSTM (1) – Dense (1)	500
LSTM Model_1	12	LSTM (50) – Dense (1)	400
LSTM Model_2	24	LSTM (10) – Dense (10) – Dense (1)	500
LSTM Model_3	24	LSTM (10) – LSTM (10) – Dense (1)	500
LSTM Model_4	28	LSTM (12) – LSTM (12) – Dense (12) – Dense (1)	500

Model_2 includes an additional dense layer. Therefore, it consists of both a hidden LSTM layer and a hidden Dense layer. The hidden LSTM layer contains ten cells, while the hidden Dense layer consists of ten neurons. Similar to the previous models, the output layer in LSTM Model_2 is a Dense layer with one neuron. LSTM Model_3 differs from the last two models, comprising two stacked LSTM layers and a dense output layer. Both LSTM layers in this model contain ten cells each. On the other hand, the dense output only has one neuron. Lastly, LSTM Model_4 comprises two LSTM hidden layers with twelve cells each and one hidden Dense Layer consisting of twelve neurons.

3.4.1 Performance Metrics

The evaluation of the LSTM models' performance was carried out by utilizing statistical metrics such as MAE and RMSE. These parameters serve as indicators of the accuracy of the models, with lower values representing superior performance. To determine the final MAE and RMSE values for a particular model, the average of ten MAE and ten RMSE outputs was computed. This approach ensures a robust evaluation process that accounts for any potential variability in the results. Equation

(3.8) provides the formula for calculating MAE, which involves calculating the absolute differences between predicted and actual values and averaging them. On the other hand, equation (3.9) outlines the calculation of RMSE, which involves squaring the differences between predicted and actual values, averaging them, and taking the square root.

$$MAE = \left(\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \right) \tag{3.8}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2} \tag{3.9}$$

Where y_i represents the rainfall observed at the ith instance. On the other hand, $\hat{y}_i$ refers to the rainfall predicted by the model for that same instance. Additionally, n denotes the total number of monthly rainfalls observed.

3.5 Results and Discussion

Accurate prediction of rainfall is an essential task with broad implications for agriculture, transportation, disaster management, and the daily lives of all living beings. This study evaluated the effectiveness of four LSTM models (LSTM Model_1, LSTM Model_2, LSTM Model 3, and LSTM Model_4) in forecasting average monthly rainfall in India. The rainfall data used for this analysis was obtained from the Indian Institute of Meteorology (IITM), Pune. Additionally, we compared the performance of our models (LSTM Model_1 to LSTM Model_4) with a previously proposed RNN and LSTM model by Kumar *et al.* (2019). We assessed forecast accuracy using MAE and RMSE metrics. We employed a grid search approach to determine optimal parameters such as input timesteps, number of epochs, and number of cells for each model configuration. In this approach, the input timesteps are searched in the range of 1 to 30, the number of epochs in the range of 50 to 1000, and the number of cells in the range of 1 to 100. Calculation of MAE and RMSE values followed standard formulae provided in the scikit-learn package documentation.

The architecture and summary of the LSTM models under investigation can be found in Table 3.2. Table 3.3 compares the performance between the RNN and LSTM models from Kumar *et al.* (2019) and our proposed models. It can be seen from these results that our proposed LSTM models, namely LSTM Model_1, LSTM Model_2, LSTM Model_3, and LSTM Model_4, achieved lower RMSE values than those mentioned in Kumar *et al.* (2019). However, it is worth noting that there may be differences in MAE due to variations in the formula used for this study. Figure 3.5 illustrates the plot of MSE loss for the LSTM models during the training and validation stages.

Among the various LSTM models that have been proposed, LSTM Model_4 stands out as it has achieved the lowest RMSE value of 245.30. LSTM Model_1 had an RMSE value of 250.47, LSTM Model_2 had a value of 246.36, and LSTM Model_3 had a value of 246.48. Upon analyzing the performance metrics for each model, it becomes evident that including more previous timesteps leads to improved model performance. To visually depict these findings, Figure 3.6 to Figure 3.9 show the plot of prediction results for LSTM Model_1 to LSTM Model_4 respectively. Examining these figures reveals that LSTM Model_4 demonstrates greater accuracy in generalizing the testing data compared to the other models.

3.6 Conclusions

This study applied four LSTM models to predict monthly average rainfall in India from 1871 to 2016. In addition to evaluating the performance of the LSTM models using statistical metrics like MAE and RMSE, this study also compared the results with those of previous research. The lowest RMSE value of 245.30 was achieved by LSTM Model_4, outperforming the values obtained by LSTM Model_1 (250.47), LSTM Model_2 (246.36), and LSTM Model_3 (246.48). This result indicates that stacked LSTM models can accurately predict rainfall patterns in India over a long period. Moreover, it was observed that increasing the timesteps further enhanced the

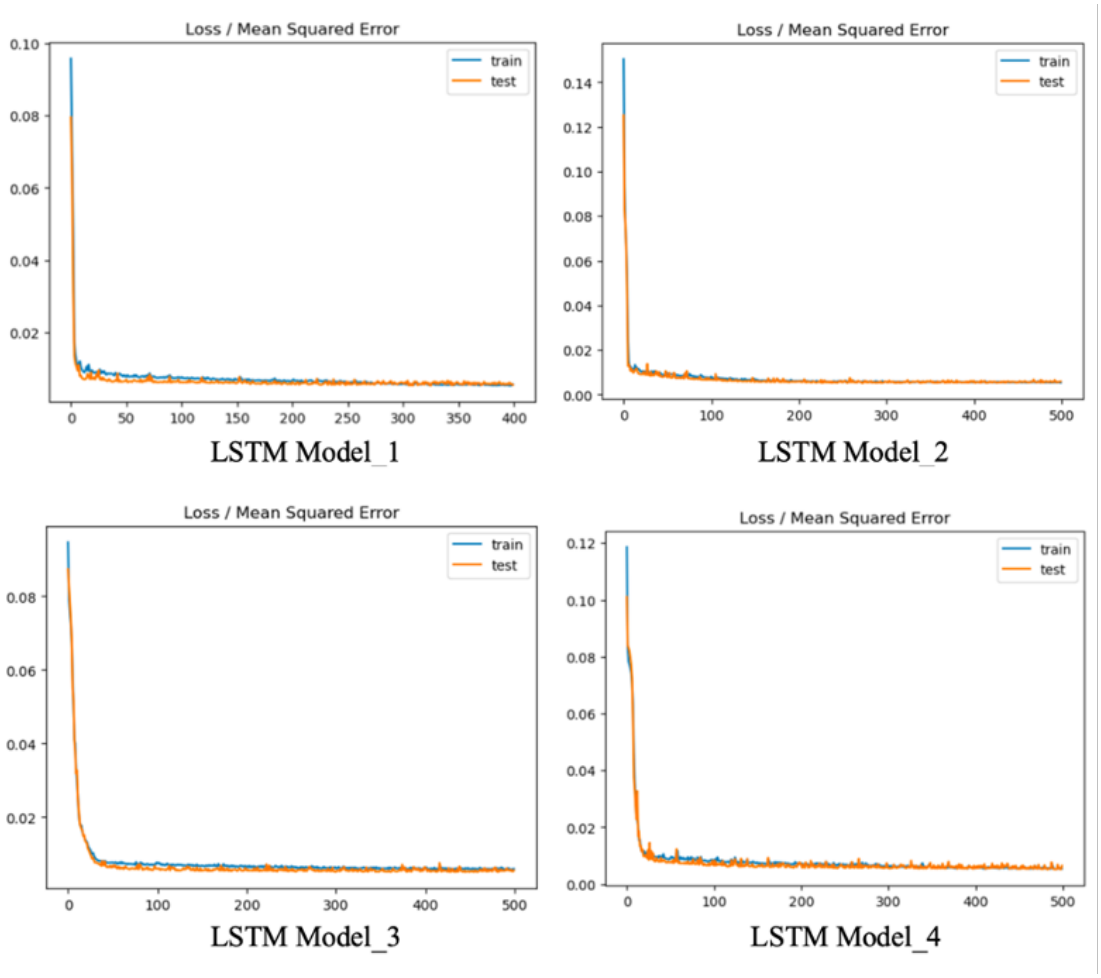

Figure 3.5: Performance (MSE) Plot of LSTM Models for training and validation

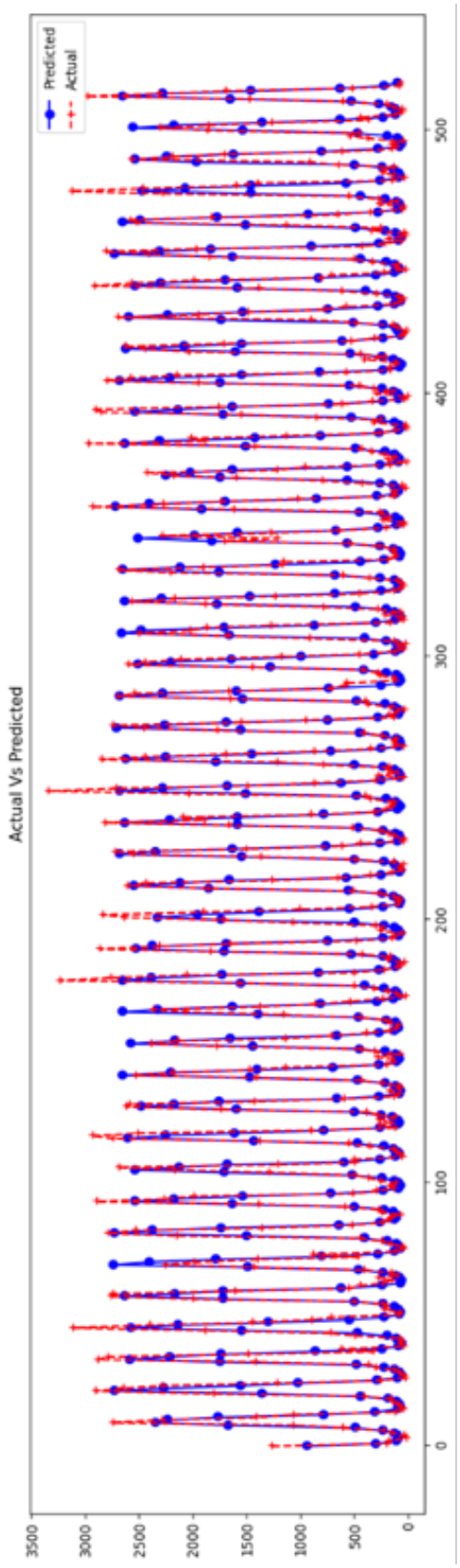

Figure 3.6: Prediction result of LSTM Model_1

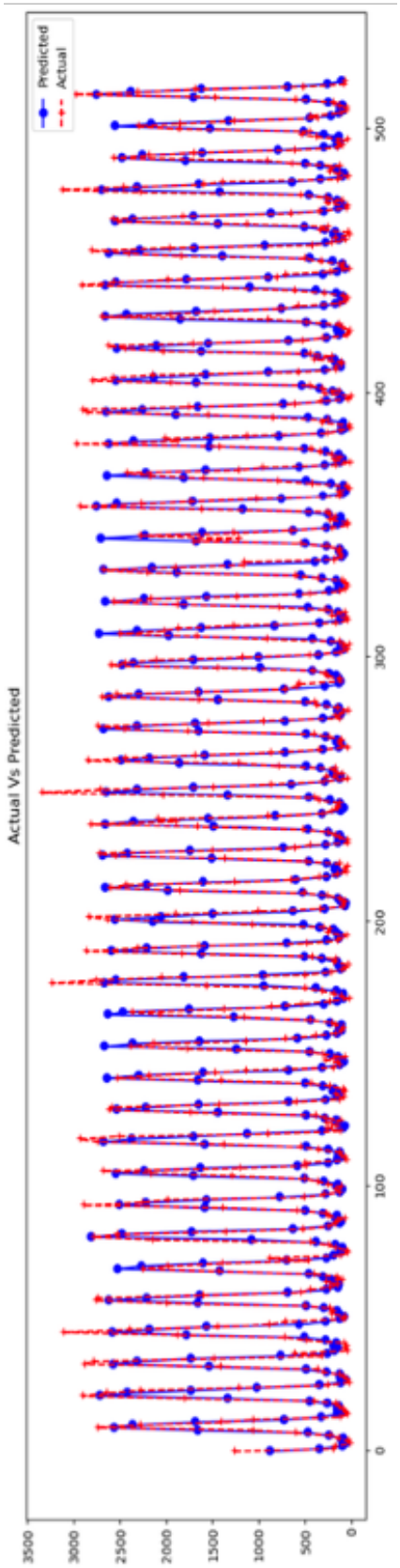

Figure 3.7: Prediction result of LSTM Model_2

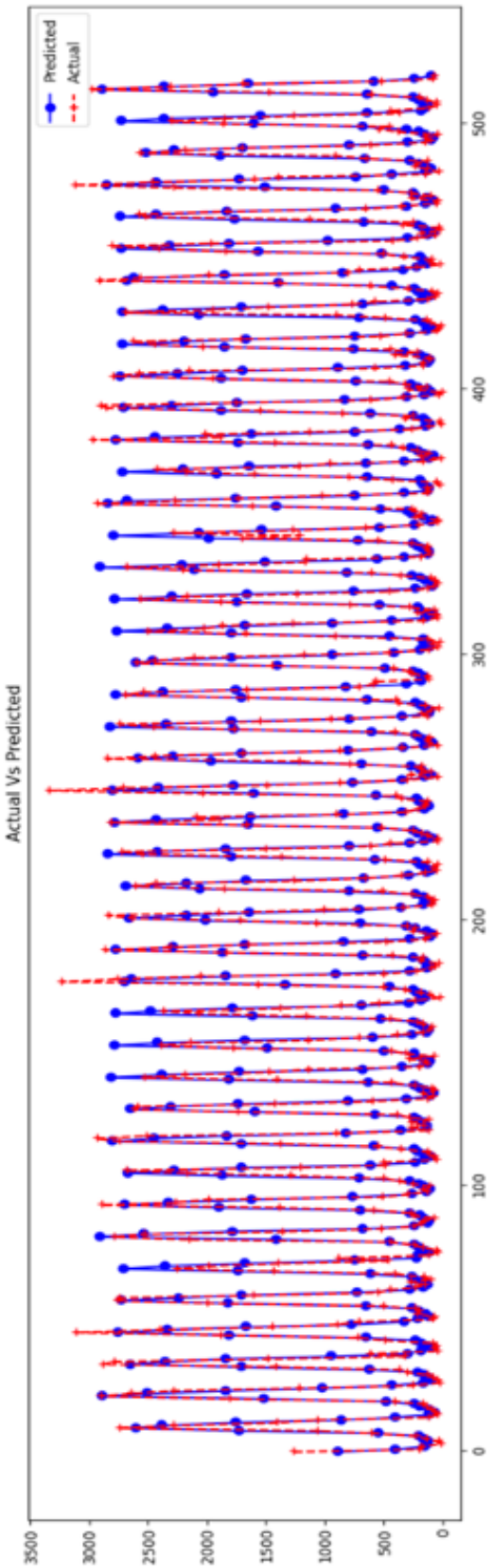

Figure 3.8: Prediction result of LSTM Model_3

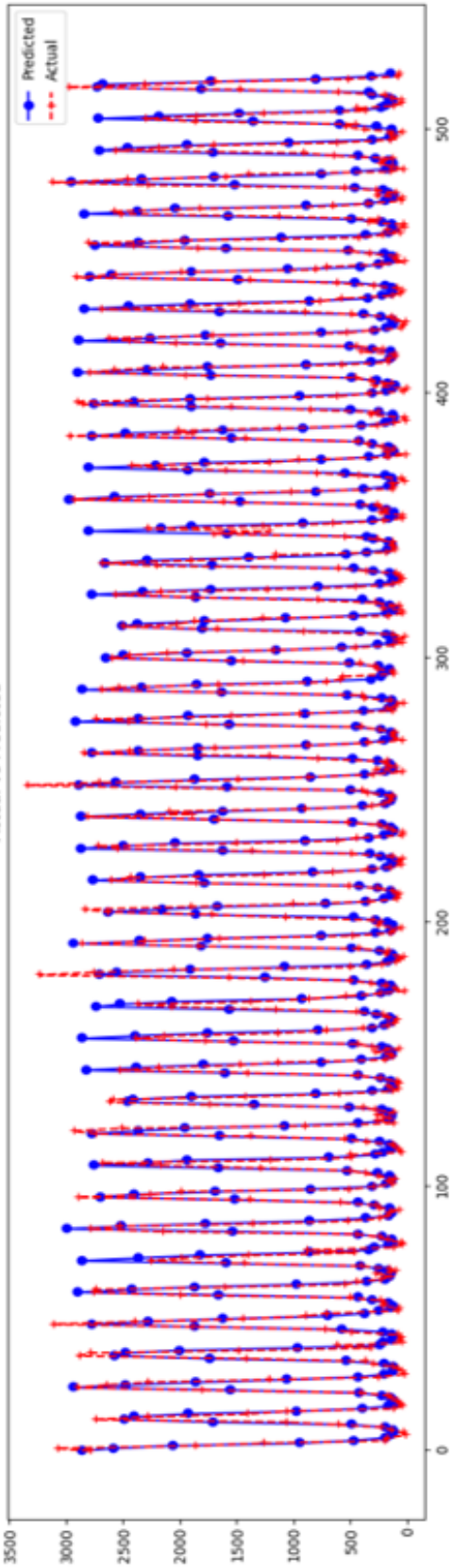

Figure 3.9: Prediction result of LSTM Model_4

Table 3.3: Performance of the models for the prediction of average monthly rainfall in India

LSTM Model	**Input Timesteps**	**No. of Epochs**	**Loss function**	**Optimizer**	**MAE**	**RMSE**
RNN (Kumar *et al.*, 2019)	20	500	MSE	SGD	279.6	261.7
LSTM (Kumar *et al.*, 2019)	20	500	MSE	SGD	169.35	251.63
LSTM Model_1	12	400	MSE	Adam	174.35	250.47
LSTM Model_2	24	500	MSE	Adam	170.60	246.36
LSTM Model_3	24	500	MSE	Adam	169.89	246.48
LSTM Model_4	**28**	500	MSE	Adam	**168.64**	**245.30**

performance of the models. To validate the effectiveness of the LSTM models in this study, a comparison was made with RNN and LSTM models from a previous study by Kumar *et al.* (2019).

Interestingly, all the LSTM models in this study outperformed the models proposed by Kumar *et al.* (2019) in terms of RMSE. This demonstrates that incorporating more neurons and additional timesteps can improve rainfall prediction accuracy. Overall, these findings highlight the potential of LSTM models for forecasting monthly average rainfall in India. The ability to accurately predict such meteorological patterns is crucial for various sectors, including agriculture, water resource management, and disaster preparedness. By leveraging advanced machine learning techniques like LSTM, decision-makers can make informed choices based on reliable predictions, ultimately contributing to better planning and resource allocation strategies.

Chapter 4

Prediction of monthly rainfall using Bidirectional LSTM [3]

4.1 Introduction

India, being a country with a predominantly agricultural economy, is highly dependent on rainfall for irrigation purposes. The primary goal of this chapter is to evaluate the predictive performance of the Bidirectional LSTM model compared to the Vanilla LSTM, Stacked LSTM, and a benchmark model from the literature. The objective is to estimate the average rainfall for India one month in advance, using previous months' rainfall as a predictor. Additionally, this study aims to determine the optimal number of previous timesteps, epochs, and cells in the LSTM layers that result in the lowest RMSE when forecasting the next month's rainfall. The performance of the proposed model is compared with Vanilla LSTM, Stacked LSTM, and the benchmark RNN and LSTM models by Kumar *et al.* (2019). The all-India monthly average rainfall from 1871 to 2016 was used to train and test the LSTM models.

4.2 Study area and data source

The all-India monthly average rainfall from the 1871 to 2016 dataset used in this chapter is presented and described in Chapter 3.

[3] The content of this article is published as a research article in: Zoremsanga, C., Hussain, J. (2024). "An Evaluation of Bidirectional Long Short-Term Memory Model for Estimating Monthly Rainfall in India". *Indian Journal of Science and Technology*, **17(18)**, 1828-1837. https://doi.org/10.17485/IJST/v17i18.2505

4.2.1 Data preprocessing

Training and testing data

To conduct this study, we partitioned the rainfall data into two sets: training and testing sets. This division aimed to ensure that the proposed models were trained on a substantial amount of data while having a separate set to evaluate their performance. To achieve this, 70% of the data were allotted for training the models, allowing the models to learn the underlying patterns using a significant portion of the dataset. The remaining 30% was reserved as the testing set, enabling us to assess how well the models could predict monthly rainfall based on what they had learned during training. This method helped us to thoroughly analyze and validate the effectiveness of the proposed models in predicting monthly rainfall patterns, ensuring that the results were reliable and accurate.

Normalization

The Min-Max normalization technique discussed in Chapter 3 is used for normalizing the training and testing data.

4.3 LSTM

The LSTM method used in this chapter is described in Chapter 3.

4.4 Stacked LSTM

The stacked LSTM model is an expanded version of the traditional LSTM structure, incorporating multiple layers of LSTM memory cells. This advanced deep learning technique has gained recognition for effectively addressing complex prediction challenges. In a stacked LSTM architecture, the output of one LSTM layer is fed as input to the next LSTM layer in the stack. This allows the network to learn increasingly abstract and complex representations of the input sequence as it progresses through the layers. Stacking multiple LSTM layers can help the network model learn intricate patterns and relationships in the data. In this architecture, depicted in Figure 4.1,

Figure 4.1: Architecture of Stacked LSTM

each LSTM layer generates a sequence of outputs instead of a single output for the underlying layer. Consequently, one output is produced for every input time step and vice versa, resulting in one output time step corresponding to all input time steps (Yu *et al.*, 2019).

4.5 Bidirectional LSTM

To overcome the limitation of traditional RNN, which considers only the past context, Schuster and Paliwal (1997) introduced the Bidirectional Recurrent Neural Network (BRNN). This particular architecture was trained in both directions simultaneously, utilizing separate layers for each direction. Building upon this concept, Graves and Schmidhuber (2005) combined the BRNN with the LSTM cell to create the Bidirectional LSTM. This sequence processing model consists of two LSTMs that process input in opposite directions. Doing so provides the network with more information and improves the overall context available to the algorithm (Yu *et al.*, 2019). The bidirectional aspect of an LSTM is practical as it enhances the models' performance by considering both past and future time steps. It allows for learning long-term dependencies within input data, which can result in improved accuracy when applied to tasks such as sentiment analysis, speech recognition, and named entity identification. Figure 4.2 illustrates the architecture of a Bidirectional LSTM model.

4.6 Model development

This study proposed a Bidirectional LSTM model to predict the average monthly rainfall in India. The performance of the proposed model is compared to that of Vanilla LSTM, Stacked LSTM, and a benchmark model found in previous research. A

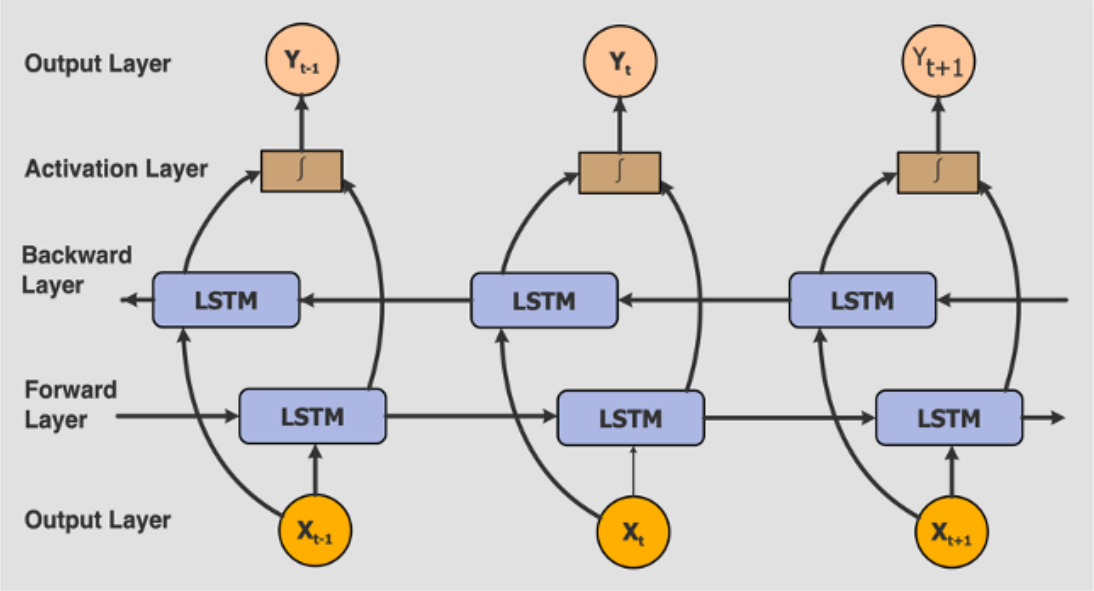

Figure 4.2: The Architecture of Bidirectional LSTM

grid search method is employed to determine the parameters for the model (including lead time, number of epochs, and number of LSTM cells). However, adjusting the parameters of a neural network can be a problem, and the parameters of the neural network also require fine-tuning. While it may learn the inherent connections within the data, it may also try to fit the noise in the data. As a result, although the network may perform exceptionally well on training data, its ability to accurately predict new data can be compromised (French *et al.*, 1992). To overcome this issue, we tested various numbers of epochs until reaching a point where the data fit well into the model and achieved optimal performance. We used performance metrics such as RMSE and MAE to assess and demonstrate the accuracy of the models.

Additionally, visual assessments were conducted by creating graphs that compared actual rainfall data with predictions made by each respective model. In conclusion, the study successfully predicted average monthly rainfall in India using different variants of LSTM models while considering their performances against existing benchmarks from prior literature. The diagram in Figure 4.3 illustrates the flowchart of the method used for forecasting the all-India monthly average rainfall.

Table 4.1: Summary of Models architecture under study

LSTM Model	**Input Timesteps**	**Model Architecture**	**Epochs**
RNN (Kumar *et al.*, 2019)	20	RNN (1) – Dense (1)	500
LSTM (Kumar *et al.*, 2019)	12	LSTM (1) – Dense (1)	500
Vanilla LSTM	12	LSTM (50) – Dense (1)	400
Stacked LSTM	28	LSTM (12) – LSTM (12) – Dense (12) – Dense (1)	500
Bidirectional LSTM	24	Bi-LSTM (80) – Dense (1)	150

In this research, the MSE was used as the loss function, while the Adam optimizer was utilized to reduce the loss function. The initial weights of the models were randomly selected for each execution. To deal with the algorithm's stochastic nature, we trained and tested each LSTM model 10 times using identical training and testing datasets. The final performance of the model was determined by averaging these ten results.

Table 4.1 provides the architecture of the Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, and the benchmark models from the literature investigated in this study. The Vanilla LSTM model was designed with a single hidden layer comprising 50 cells, followed by an output-dense layer. The Stacked LSTM model consists of two hidden layers, both consisting of 12 cells. It also includes a hidden dense layer with 12 neurons and an output dense layer. The Bidirectional LSTM model has a single hidden LSTM layer with 80 cells and an output-dense layer. These architectures are the final values after searching multiple hyperparameters using the grid search method for each category.

4.6.1 Performance Metrics

The effectiveness of the LSTM models is measured using two statistical metrics - MAE and RMSE, which are described in Chapter 3.

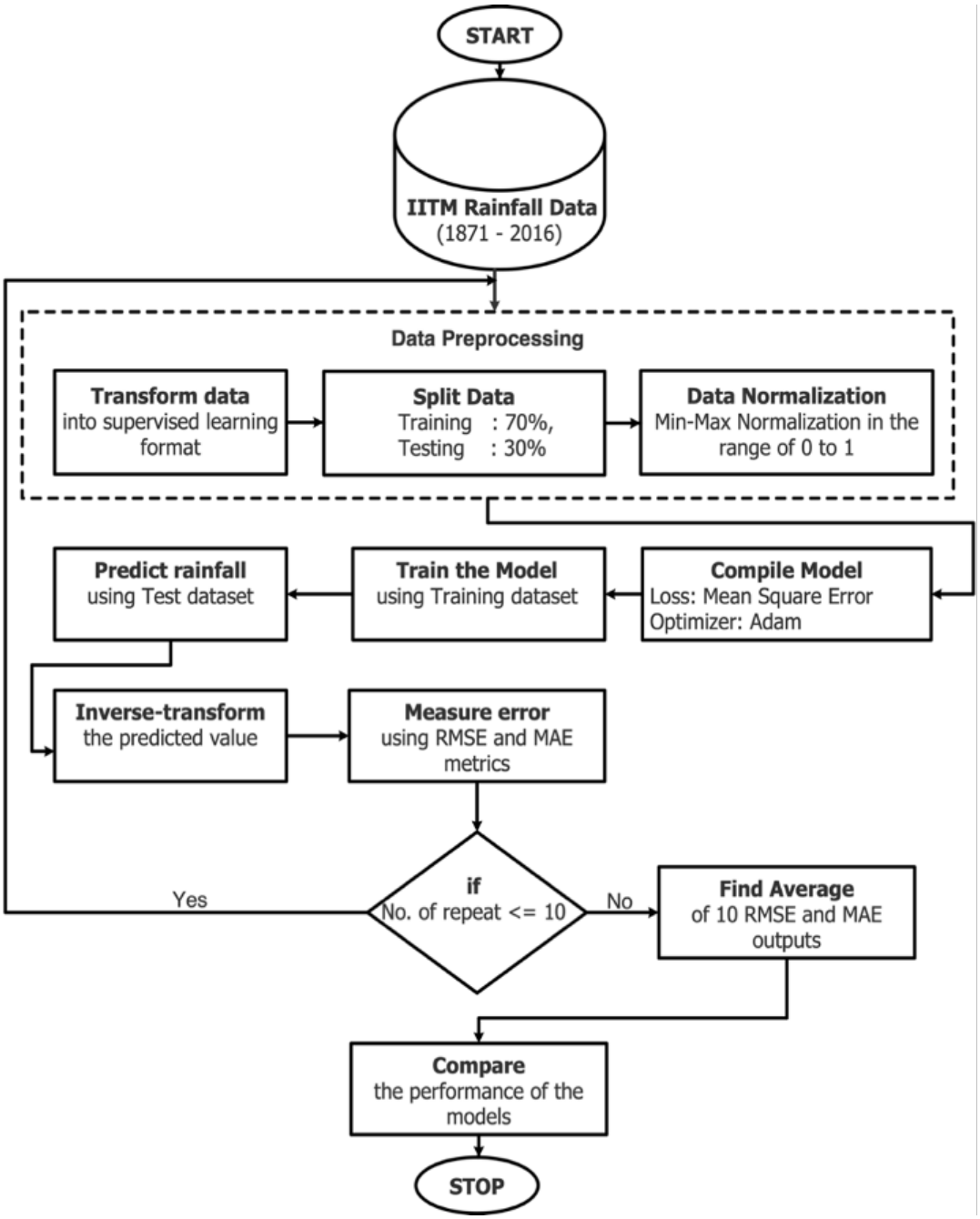

Figure 4.3: Flowchart for predicting monthly rainfall using three LSTM models

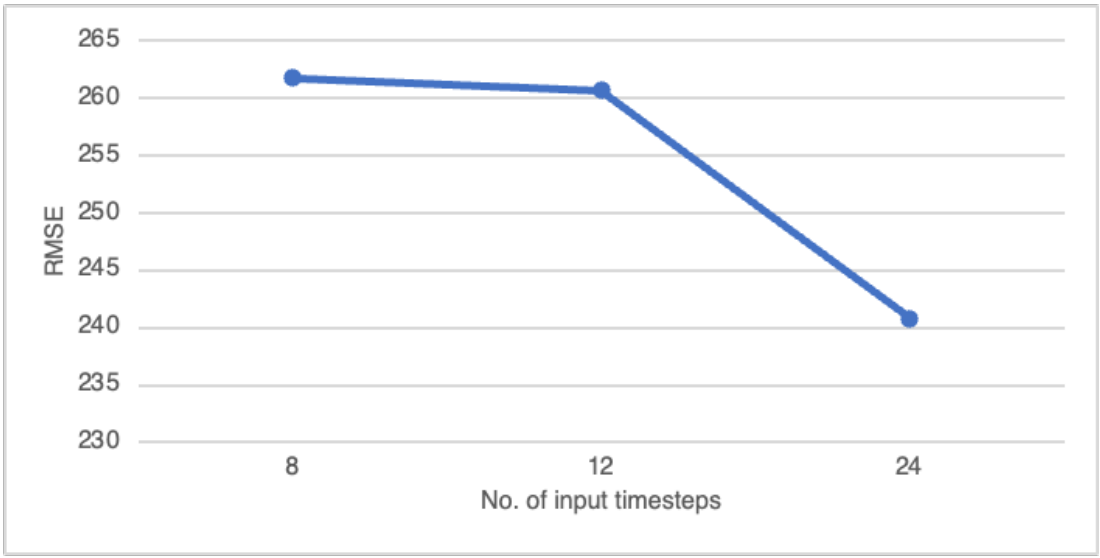

Figure 4.4: Input Timesteps vs RMSE

4.7 Results and discussion

This study focused on analyzing the prediction performance of the Bidirectional LSTM model. Various hyperparameters were considered in this study to achieve optimal results, including the number of previous timesteps, epochs, and cells. The performance of the Bidirectional LSTM model was then compared to that of Vanilla LSTM, Stacked LSTM, and a benchmark model found in the literature. Figure 4.4 illustrates the comparison of the Bidirectional LSTM model with 8, 12, and 24 timesteps. It is evident from the figure that increasing the input timestep tends to provide improved prediction accuracy.

The performance of the Bidirectional LSTM model was analyzed using an epoch ranging from 50 to 500, as illustrated in Figure 4.5. It was observed that the RMSE of the model gradually decreased with an increase in the number of epochs until it reached a minimum RMSE at 150 epochs. However, beyond this point, the RMSE started to increase again. This result suggests that the model may have been overfitting the data, causing a decline in its performance. As a result, an epoch of 150 was chosen for training the Bidirectional LSTM model. Choosing this epoch value made

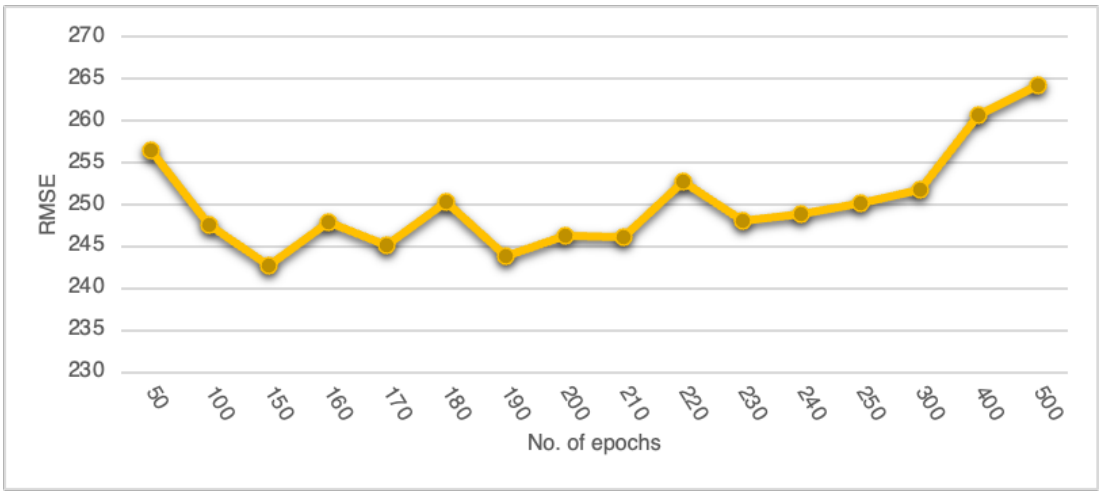

Figure 4.5: No. of epochs vs RMSE

it possible to strike a balance between capturing sufficient patterns in the data and avoiding overfitting. This approach aimed at achieving a reliable and generalized model that could effectively handle unseen data. The selection of an appropriate epoch plays a crucial role in training deep learning models and helps to avoid potential pitfalls such as overfitting or underfitting. Therefore, careful consideration is required when determining the optimal number of epochs for training machine learning models.

As the number of cells in the Bidirectional LSTM model was increased from 10 to 100, significant changes were observed in the performance of the models. These results are presented in Figure 4.6. One interesting finding was that by increasing the number of cells, the RMSE of the model decreased to 80 cells. This result implies that increasing the complexity and capacity of the model led to improved accuracy and reduced prediction errors. However, it is essential to note that beyond this point, when the number of cells exceeded 80, there was a noticeable increase in RMSE. This indicates that adding more cells resulted in diminishing returns and potentially introduced overfitting. As a result, it was discovered that configuring the Bidirectional LSTM model with 80 cells reached a balance between complexity and performance, offering optimal results for this particular study.

The final model, a Bidirectional LSTM, was developed using 24 input timesteps,

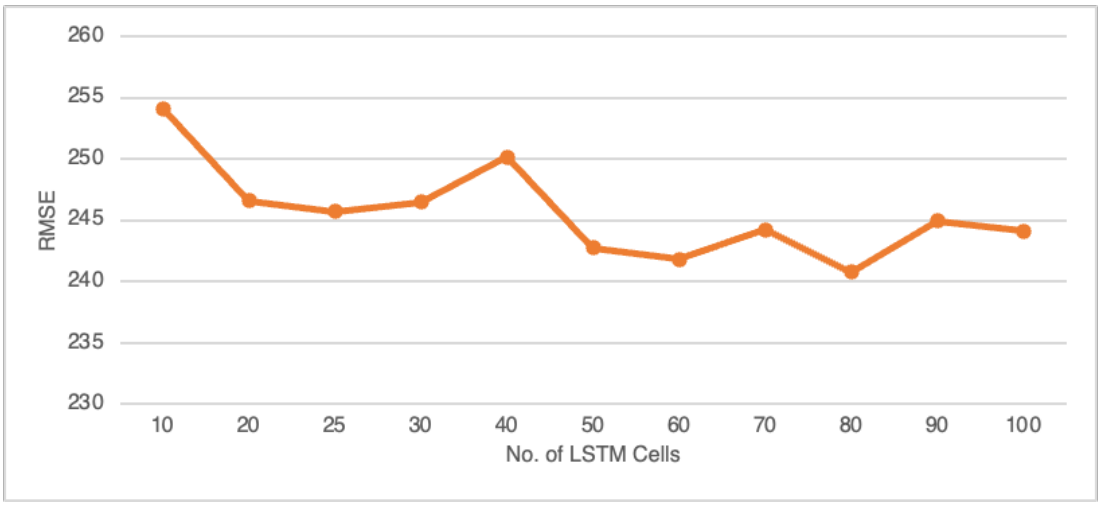

Figure 4.6: No. of LSTM cell vs. RMSE

80 LSTM cells, and trained for 150 epochs. The outcome of this training process resulted in an RMSE value of 240.79. To validate the performance of the proposed Bidirectional LSTM model, Bidirectional LSTM was compared against other models such as Vanilla LSTM, Stacked LSTM, and benchmark models like RNN (Kumar *et al.*, 2019) and LSTM (Kumar *et al.*, 2019). The comparison demonstrated that the proposed Bidirectional LSTM model outperformed the other models in terms of its overall performance. These findings are presented in Table 4.2 and also visually represented in Figure 4.7, both of which show the performance metrics (MAE and RMSE) for the all-India monthly rainfall dataset for each of the compared models. The results supported that the proposed Bidirectional LSTM model is superior to the compared models for accurately predicting average monthly rainfall patterns in India.

4.8 Conclusions

Predicting rainfall is critical to weather forecasting, agriculture, and various other industries. A dependable and precise method for rainfall prediction is required to facilitate informed decision-making. This study proposed an approach employing Bidirectional LSTM to forecast the average monthly rainfall in India using the historical all-India average monthly rainfall data spanning from 1871 to 2016. To evaluate its

Table 4.2: Performance of the compared models and parameters of the models

Model	**Input Timesteps**	**No. of Epochs**	**Loss function**	**Optimizer**	**MAE**	**RMSE**
RNN (Kumar *et al.*, 2019)	**20**	**500**	**MSE**	**SGD**	**279.6**	**261.7**
LSTM (Kumar *et al.*, 2019)	20	500	MSE	SGD	169.35	251.63
Vanilla LSTM	12	400	MSE	Adam	174.35	250.47
Stacked LSTM	28	500	MSE	Adam	168.64	245.30
Bidirectional LSTM	**24**	150	MSE	Adam	**168.48**	**240.79**

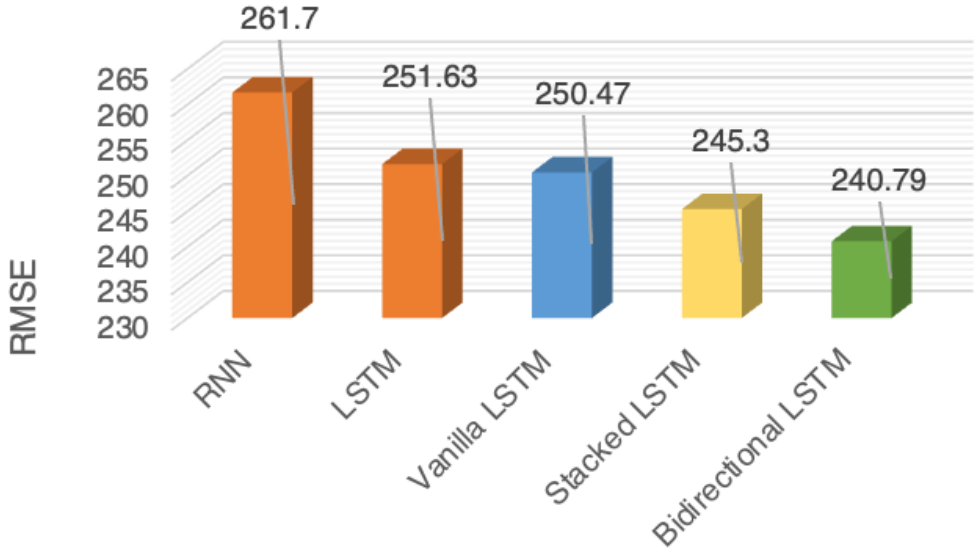

Figure 4.7: Performance of the compared Models

effectiveness, the performance of this proposed model was compared with Vanilla LSTM, Stacked LSTM, and two benchmark RNN and LSTM models from the existing literature. Among the models examined, the Bidirectional LSTM demonstrated superior performance, achieving the lowest RMSE value of 240.79, followed by the Stacked LSTM with an RMSE value of 245.3, and the Vanilla LSTM with an RMSE value of 250.47. These results surpassed those of the benchmark RNN, which had an RMSE of 261.7, and the benchmark LSTM model with an RMSE value of 251.63.

The study revealed that optimizing prediction accuracy requires considering factors such as increasing the number of input timesteps, augmenting the number of LSTM cells, and extending the training epochs while monitoring the models for potential overfitting. Thus, finding a balance between the input data size, the number of cells in the LSTM layers, and the training dataset's exposure to the LSTM model is essential. Furthermore, the research demonstrated that employing stacked LSTM layers and adopting a Bidirectional LSTM approach can significantly enhance model performance compared to single-cell LSTM and Vanilla LSTM models.

Chapter 5

PSO optimized Machine learning and Deep Learning models for prediction of monthly rainfall[4]

5.1 Introduction

Rainfall prediction involves forecasting the rainfall expected in a specific geographical area. Accurate rainfall predictions offer valuable insights to decision-makers, helping them allocate resources more efficiently, reduce risks, and improve socio-economic outcomes. This capability is crucial for farmers planning their crops, cities preparing for potential flood threats, and emergency responders gearing up for possible disasters. Machine learning and deep learning models come with various hyperparameters, encompassing factors like the optimal number of neurons or cells, input timesteps, epochs, learning rate, layer count, dropout rate, and more. Identifying the most effective combination of hyperparameters through grid search is resource-intensive, time-consuming, and doesn't guarantee the discovery of optimal settings. Recently, optimization algorithms like PSO have gained popularity. Nevertheless, most studies in the literature typically involve optimizing a single model using PSO with the parameters of the other models determined using trial-and-error, grid search, or literature-derived values.

Consequently, a gap exists in which all machine learning and deep learning model hyperparameters are optimized using PSO. In this study, multiple machine learning

[4]The content of this article is published as a research article in: Zoremsanga, C., and Hussain, J. (2024). "Hybrid Particle Swarm Optimized Models for Rainfall Prediction: A Case Study in India". *Pure and Applied Geophysics*, **181**, 2343–2357, doi: 10.1007/s00024-024-03528-7.

and deep learning models with PSO-optimized parameters were investigated and compared using the Aizawl monthly rainfall dataset and the all-India monthly average rainfall dataset. This research marks the first instance of examining the Mizoram state monthly rainfall dataset with machine learning and deep learning models, potentially establishing benchmark results. Additionally, to the author's knowledge, this study represents the first utilization of PSO-optimized SVR, ANN, RNN, LSTM, and BiLSTM models with the all-India monthly average rainfall dataset.

5.2 Study area and data source

In this chapter, the study utilized two primary datasets: the Aizawl Meteorological Data (AMD), acquired from the State Meteorological Centre, Directorate of Science and Technology, Government of Mizoram, India, and the all-India monthly rainfall dataset compiled by Kothawale and Rajeevan (2017). The all-India monthly average rainfall dataset used in this chapter is presented and described in Chapter 3.

The Aizawl monthly dataset was constructed by aggregating daily rainfall amounts for each month from 1985 to 2021. This dataset contains 444 months of rainfall data, ranging from a minimum of 0 mm to a maximum of 743 mm. The average rainfall amount in this dataset is 176.16 mm, with a standard deviation of 161. For a visual representation, Figure 5.1 illustrates the plot of Aizawl's monthly rainfall, while Figure 5.2 displays the location of the Aizawl meteorological station.

5.3 Methodology

5.3.1 Preparation of dataset

The Aizawl daily rainfall dataset was obtained from the State Meteorological Centre, Directorate of Science and Technology, Government of Mizoram, India. The initial dataset contained rainfall records spanning from 1960 to 2021. However, due to a significant number of missing values within the original dataset, this study considered data from the years 1985 to 2021 only. Additional rainfall data for the Aizawl Station

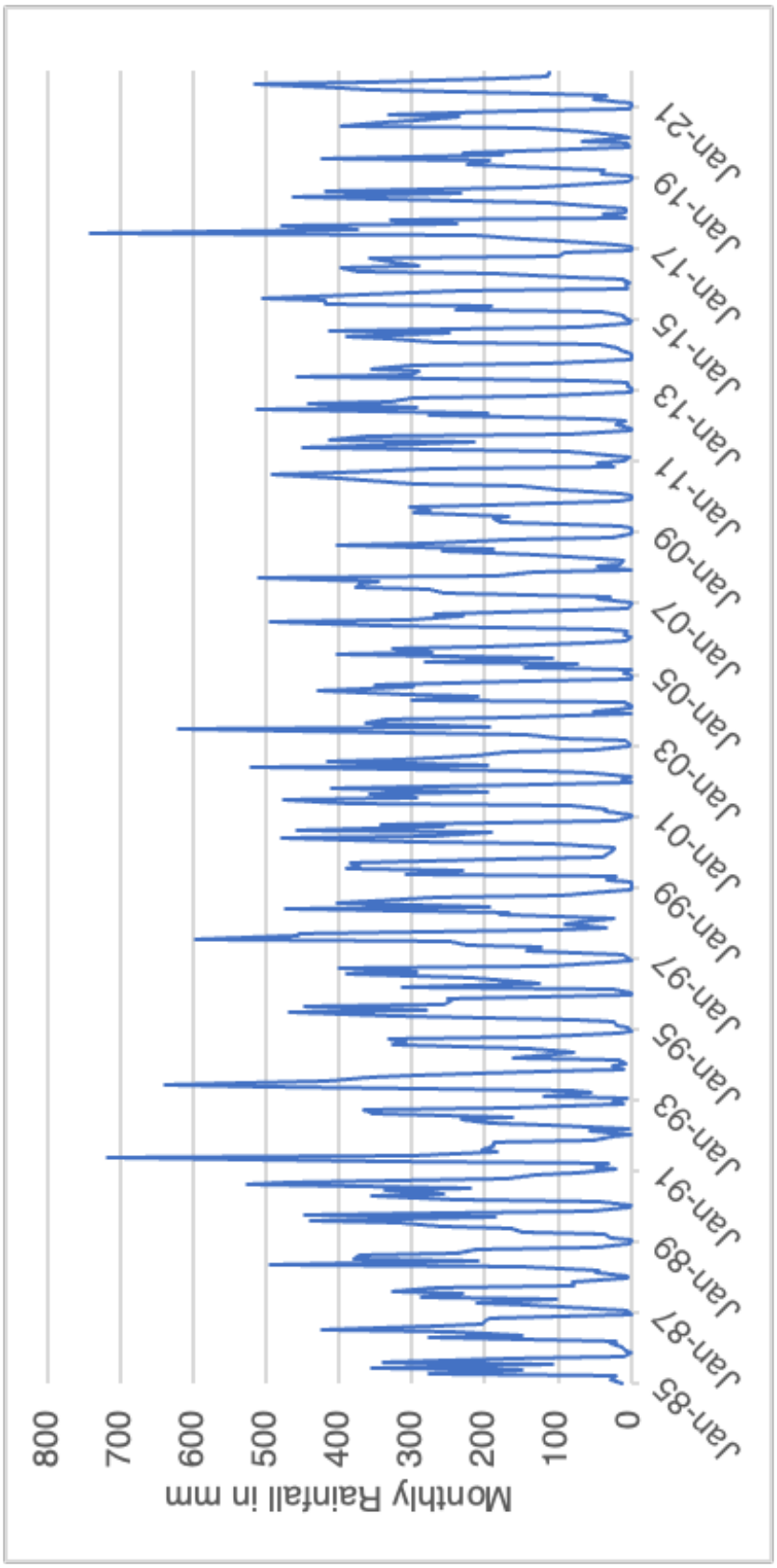

Figure 5.1: Monthly Rainfall in Aizawl from 1985 – 2021

Figure 5.2: Location of the Aizawl Meteorological Station

was acquired from the National Data Centre, Indian Meteorological Department (IMD), Pune, to validate the data and address the gaps in the rainfall records from 1985 to 2021. A comparative analysis of the two datasets was conducted to identify inconsistencies, and any missing data in the AMD was augmented with corresponding data from IMD.

Training and testing data

In this study, the rainfall dataset was divided into three distinct sets: training, validation, and testing datasets. Since there is no best method for dividing such datasets, we adopted well-known practices commonly found in the literature. In the context of the Aizawl monthly rainfall dataset, 379 months were allocated to the training set, 32 months were designated for the validation set, and an additional 32 months were reserved for the testing set. This division ensures that each segment of the dataset is representative and can be used for training, tuning hyperparameters, and evaluating model performance. Similarly, for the all-India monthly average rainfall dataset, the partitioning followed a distribution of 80% for training, 10% for validation, and another 10% for testing. This stratified data allocation facilitates robust model training, the validation of model generalization, and rigorous assessment of model performance.

Normalization of datasets

The Min-Max normalization technique discussed in Chapter 3 is used for normalizing the training and testing data.

5.3.2 Particle Swarm Optimization

PSO was introduced by Kennedy and Eberhart (1995) as an algorithm employing a population-based stochastic optimization approach to tackle optimization problems. This algorithm draws inspiration from the collaborative behaviours observed in birds and fish, where individuals cooperate to identify the most favourable food sources or migration paths. In PSO, a group of particles collectively explores a multi-dimensional search space to locate the optimal solution. To achieve this, each particle continually updates its position by considering its own best position and the best positions of neighbouring particles. If we assume there are P particles, we can express the position

of the *i*th particle at iteration *t* as x_i^t, and the velocity of the *i*th particle as v_i^t, which are calculated using equations (5.1) and (5.2).

$$v_i^t = \omega v_i^{t-1} + c_1 r_1 \left(Pbest_i^{t-1} - x_i^{t-1}\right) + c_2 r_2 \left(Gbest^{t-1} - x_i^{t-1}\right) \tag{5.1}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{5.2}$$

Here, r_1 and r_2 represent random numbers ranging from 0 to 1, while c_1 and c_2 are the acceleration constants, and ω is the inertia weight. The two coefficients, c_1 and c_2, play a pivotal role in determining the trade-off between fine-tuning an individual particle's search and recognizing the collective effort of the swarm. These parameters dictate the relative importance of both individual and group contributions in achieving the desired search results. $Pbest_i^{t-1}$ corresponds to the best position explored by the *i*th particle, while $Gbest^{t-1}$ represents the best position explored by all particles in the swarm, and these values are updated after each iteration.

This research employs the PSO algorithm to determine the optimal hyperparameters for the models under comparison. The fitness of these models is measured by computing the RMSE between their predicted values and the actual data. The PSO algorithm iterates 300 times for each model to identify the hyperparameters that yield the most accurate predictions.

5.3.3 SVR

SVR is a regression analysis technique commonly used in machine learning. Its primary objective is to minimize error by creating a hyperplane that maximizes the distance between it and the nearest data points, known as the margin. To account for potential errors, SVR sets a tolerance margin called epsilon. What sets SVR apart from other regression methods is its utilization of principles derived from SVM, which are primarily used for classification tasks. While SVM classify data into multiple classes by finding the hyperplane that maximizes the margin, SVR predicts continuous output variables by identifying the hyperplane that maximizes the margin within a

specified tolerance. SVR aims to minimize loss function while ensuring the margin remains equal to epsilon. The main parameters of SVR include kernel function, kernel coefficient, regularization parameter, and tolerance margin (Awad and Khanna, 2015).

The PSO algorithm is employed in this study to discover the most suitable hyperparameters for SVR, including the number of previous timesteps, regularization parameter C, and margin of tolerance epsilon. Each kernel function - linear, RBF, and Polynomial - is examined individually as a distinct model. Furthermore, due to the consistent output values of the SVR model, there is no need to repeat training and testing processes.

5.3.4 ANN

ANN, shown in Figure 5.3, is a type of machine-learning model and is designed to mimic the human brain. Artificial neural networks are composed of several layers of nodes, which include an input layer, hidden layers, and an output layer. Each node is interconnected with others and has a weight and threshold assigned to it. If a node's output surpasses the threshold value, it becomes activated and transfers data to the subsequent layer. Otherwise, it does not transmit any information. To improve their accuracy, neural networks rely on training data. Once these learning algorithms have been fine-tuned, they became a powerful tool in artificial intelligence and computer science. They can classify and cluster data quickly, making them valuable assets. In this figure, x_i is the input, and y_i is the output (Russell and Norvig, 2016). ANN with a many-to-one model is used in this study, with multiple inputs and a single output.

This chapter uses the PSO, SVR, and ANN methods explained above, as well as the RNN, LSTM, and BiLSTM methods.

5.3.5 RNN, LSTM, Stacked LSTM, and BiLSTM

The RNN, and LSTM models are described in Chapters 3. The Stacked LSTM and BiLSTM models are explained in Chapters 4.

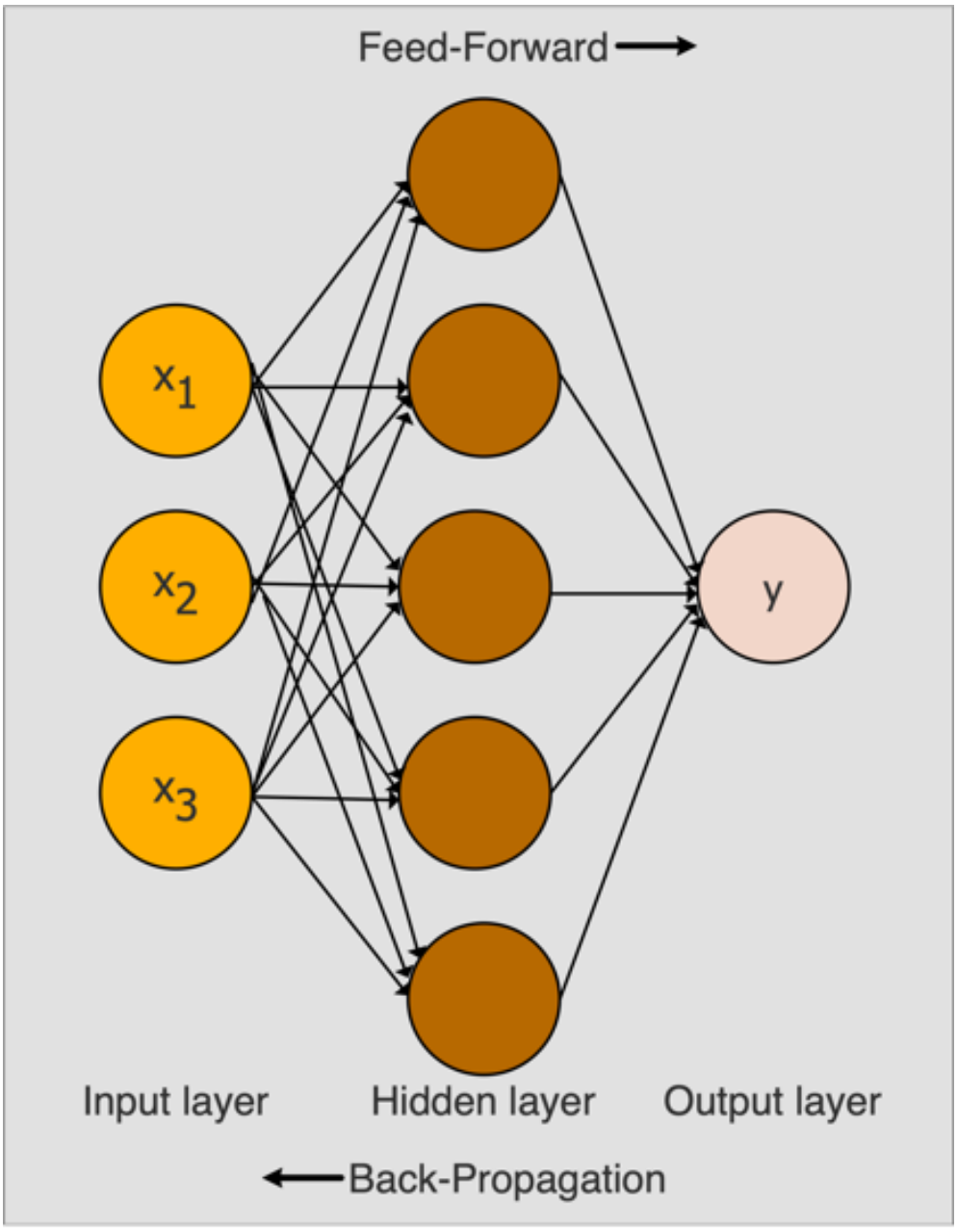

Figure 5.3: The basic architecture of ANN

Table 5.1: Hyperparameter configurations of the models and the PSO

Sl No.	Hyperparameter	Configuration
1	ANN Activation Function	Sigmoid
2	RNN, LSTM, BiLSTM Activation Function	Tanh
3	Optimizer	Adam
4	Learning Rate	0.001
5	Loss Function	MSE
6	ANN Neurons	1 to 150
7	RNN Units	1 to 150
8	LSTM Cells	1 to 150
9	Epochs	30 to 1000
10	Input Timesteps	1 to 30
11	SVR regularization parameter C	1 to 10
12	SVR epsilon	0 to 1
13	PSO number of particles	10
14	PSO number of generations	300
15	PSO particle acceleration constants c1	1.5
16	PSO particle acceleration constants c2	2.0
17	PSO fitness function	RMSE

5.3.6 Model development

This research compares machine learning and deep learning models, namely PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR. This comparison is based on the Aizawl monthly rainfall dataset and the all-India monthly average rainfall dataset. To ensure optimal performance of these models, parameters such as the number of neurons/cells, number of epochs, and number of previous timesteps for input are optimized using the PSO method for BiLSTM, LSTM, RNN, and ANN models. For the PSO-SVR model, optimization of hyperparameters involved tuning the number of previous timesteps along with regularization parameter C and tolerance margin epsilon. Several metrics, such as RMSE, MAE, and Coefficient of determination (R^2), are used to assess the performance of these compared models. The chosen loss function that needs to be minimized by these models is MSE. Furthermore, the Adam optimizer is employed to achieve this minimization objective. Figure 5.4 illustrates the step-by-step process showing how rainfall is predicted using these optimized PSO-based models.

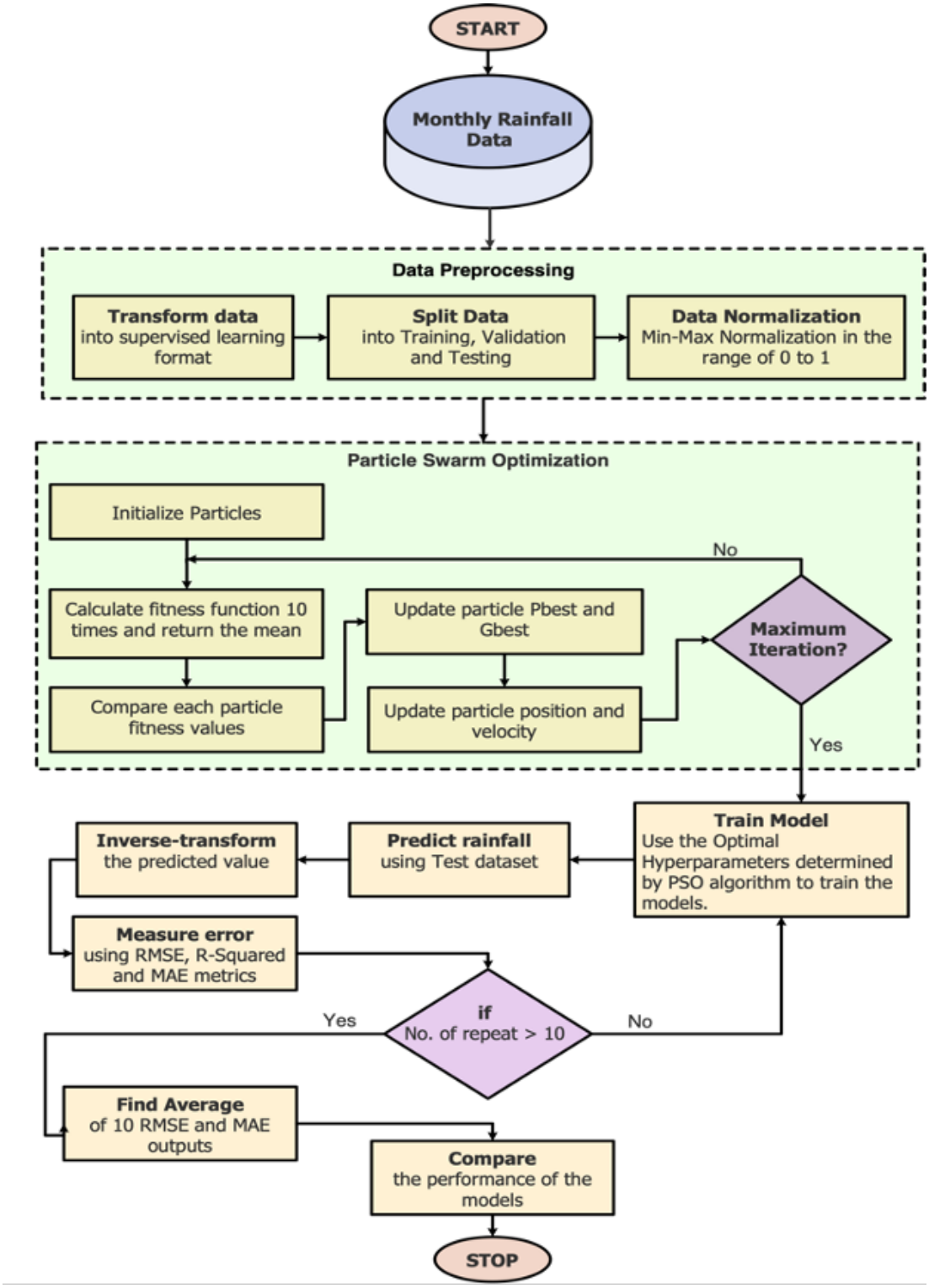

Figure 5.4: The flowchart for predicting rainfall using PSO optimized models

The collected rainfall data was pre-processed before being used in the models. The pre-processing involves cleaning the data and filling in any missing values by comparing information from various sources. After that, the data is transformed into a format suitable for supervised learning. Next, it is divided into training, validation, and testing datasets and normalized using the Min-Max Normalization method. The steps for optimizing and training the models are summarized below:

Step 1: The dataset is transformed into a supervised learning format and divided into training, validation, and testing datasets. Min-Max normalization is then applied to normalize the datasets in the range of 0 to 1.

Step 2: The hyperparameters of the models, such as the number of neurons/cells, number of epochs, and number of previous timesteps for the input shown in Table 5.1, are optimized using the PSO optimization method. Each hyperparameter's lower and upper bounds are set within a specified limit to find the best hyperparameter values. The number of neurons/units/cells ranges from 1 to 150, and epochs range from 30 to 1000. Additionally, the previous input timesteps are set in the range of 1 and 30.

For SVR, the hyperparameters include previous timesteps ranging from 1 to 30, regularization parameter C ranging from 1 to 10, and epsilon ranging from 0 to 1. Setting lower and upper bounds for these hyperparameters does not follow a specific method or rule; it depends on finding suitable values based on experimentation.

Furthermore, the number of particles is 10 for the PSO method used in this process. The optimization process would then run for 300 generations. Finally, the constants c1 and c2 representing particle acceleration are respectively assigned a value of 1.5 and 2.0.

Step 3: The hyperparameter values provided by the PSO algorithm are utilized to define and compile the models, employing the MSE loss function and Adam optimizer.

Step 4: The training dataset is utilized to train the model, which predicts the test dataset based on its training. The predicted values are then converted to their original scale through an inverse transformation. To evaluate the accuracy of the predictions, RMSE, MAE, and R^2 scores are calculated by comparing them with the actual values.

This process is repeated ten times, and the average metrics are computed. Finally, the mean RMSE serves as a fitness value for further analysis and decision-making purposes.

Step 5: The PSO algorithm is executed 300 times, updating the particles Pbest and Gbest during each iteration. Once the iteration process is finished, the PSO algorithm produces the optimal hyperparameters.

Step 6: The model undergoes a process of creation, training, and testing ten additional times using the optimal hyperparameters determined by the PSO algorithm. Subsequently, the predicted values are transformed back to their original state, and RMSE, MAE, and R^2 are recalculated. The average of these ten calculations represents the final evaluation of the model's performance.

Step 7: The process is repeated for the two datasets mentioned: the Aizawl monthly rainfall dataset and the all-India monthly rainfall dataset.

Step 9: Finally, the performance of PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR are compared for each dataset.

The model architectures and their corresponding names, which are compared in this study, are presented in Table 5.2. In addition to the model architecture mentioned in the table, experiments were conducted with models trained using 1 and 32 batch input and stateful LSTM. However, due to their poor performance and high memory requirements, these models were not experimented with for all datasets and, therefore, are not included in this chapter.

Each model in Table 5.2 consists of an output layer with a dense layer containing a single neuron. This information is not explicitly stated in the model architectures provided. The BiLSTM, LSTM, and RNN layers utilize the default tanh activation function, while the Dense layers use the sigmoid activation function. Separate models were created for SVR using different activation functions such as linear, RBF, and polynomial kernels. Additionally, various numbers of layers within each model architecture were studied as separate entities to reduce the search space dimension of the PSO algorithm. Regarding the monthly rainfall data for all-India, the PSO-optimized

Table 5.2: Model name and Model architecture under study

Sl No.	**Model Name**	**Model Architecture**
1	PSO-ANN I	Dense
2	PSO-ANN II	Dense - Dense
3	PSO-BiLSTM I	BiLSTM
4	PSO-BiLSTM II	BiLSTM-Dense
5	PSO-BiLSTM III	BiLSTM-BiLSTM
6	PSO-BiLSTM IV	BiLSTM-BiLSTM-Dense
7	PSO-BiLSTM V	BiLSTM-BiLSTM-BiLSTM-Dense
8	PSO-LSTM I	LSTM
9	PSO-LSTM II	LSTM-Dense
10	PSO-LSTM III	LSTM-LSTM
11	PSO-LSTM IV	LSTM-LSTM-Dense
12	PSO-RNN I	RNN
13	PSO-RNN II	RNN-Dense
14	PSO-SVR I	SVR (Linear kernel)
15	PSO-SVR II	SVR (RBF kernel)
16	PSO-SVR III	SVR (Polynomial kernel)
17	Kumar-RNN	RNN
18	Kumar-LSTM	LSTM
19	Grid-LSTM I	LSTM
20	Grid -LSTM II	LSTM-LSTM-Dense
21	Grid -BiLSTM	BiLSTM

models were compared to existing benchmark RNN and LSTM models by (Kumar *et al.*, 2019). Additionally, these models were compared to LSTM and BiLSTM models, whose parameters were determined through the grid search method.

5.3.7 Performance Metrics

Three statistical measures – MAE and RMSE, defined in Chapter 3 and coefficient of determination R^2, described below- were utilized to assess the effectiveness of the models under study. R^2 evaluates how well the predicted values align with actual values and ranges from 0 to 1. An R^2 value close to 1 signifies improved performance and Equation (5.3) shows the mathematical representations for calculating R^2.

$$R^2 = \left(\frac{n\left(\sum y_i \hat{y}_i\right) - \left(\sum y_i\right)\left(\sum \hat{y}\right)}{\sqrt{\left[n\left(\sum {y_i}^2 - \left(\sum y_i\right)^2\right)\right] * \left[n\left(\sum \hat{y}^2 - \left(\sum \hat{y}\right)^2\right)\right]}} \right)^2 \tag{5.3}$$

Where, y_i indicates the i^{th} observation of rainfall, $\hat{y}_i$ the i^{th} predicted rainfall, and n is the total number of observations.

5.4 Results and discussion

This research conducted a comparison between Particle swarm-optimized deep learning and machine learning models, PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR. The evaluation used the Aizawl monthly rainfall and all-India monthly average rainfall datasets. To optimize the performance of the BiLSTM, LSTM, RNN, and ANN models, parameters such as the number of neurons/cells used in each layer were optimized using the PSO technique. Additionally, the number of previous time steps for the input was optimized. For the PSO-SVR model, optimization includes determining the optimal values for parameters such as regularization parameter C and margin of tolerance epsilon and selecting an appropriate number of previous timesteps. This study used statistical metrics like RMSE, MAE, and R^2 to assess performance accuracy. The models employed the MSE loss function and the Adam optimizer during

training. Detailed results concerning prediction accuracy for each dataset are elaborated in subsequent sections.

5.4.1 Prediction using the Aizawl monthly rainfall dataset

The Aizawl monthly rainfall dataset was compiled by summing up the daily rainfall for each month between 1985 and 2021. This dataset includes 444 months of rainfall data, with the highest recorded rainfall value of 743 mm and the lowest 0 mm. Table 5.3 shows the prediction results of 16 models that were compared in terms of their ability to forecast Aizawl's monthly rainfall. These results are arranged in ascending order based on their RMSE values, while Figure 5.5 illustrates a plot depicting these RMSE values. Instead of providing plots for all compared models, only the plots for the best model from each variant are shown in the table and figure.

Table 5.3: Optimal hyperparameters and performance for Aizawl rainfall

Sl No.	**Model Name**	**Hidden Layer Architecture**	**Epoch**	**Time-steps**	**RMSE**	**MAE**	**R^2**
1	PSO-BiLSTM II	BiLSTM(3)-Dense(103)	54	13	76.6	60.53	0.74
2	PSO-BiLSTM I	BiLSTM(6)	69	13	77.84	64.26	0.73
3	PSO-BiLSTM IV	BiLSTM(11)-BiLSTM(28)-Dense(89)	56	13	78.19	63.01	0.72
4	PSO-BiLSTM III	BiLSTM(11)-BiLSTM(28)	56	13	78.35	63.13	0.72

Table 5.3: Optimal hyperparameters and performance for Aizawl rainfall

5	PSO-BiLSTM V	BiLSTM(1)-BiLSTM(7)-BiLSTM(13)-Dense(120)	57	13	79.15	64.07	0.72
6	PSO-LSTM II	LSTM(50)-Dense(148)	602	8	79.28	64.21	0.72
7	PSO-ANN I	Dense(9)	303	13	79.37	65.02	0.72
8	PSO-ANN II	Dense(5)-Dense(12)	264	16	80.21	66.98	0.71
9	PSO-LSTM IV	LSTM(17)-LSTM(51)-Dense(90)	346	8	81.06	66.39	0.70
10	PSO-SVR I	Linear Kernel, C=7.8, epsillon = 0.1	NA	14	81.5	67.77	0.70
11	PSO-SVR II	RBF Kernel, C=1.0, epsillon = 0.1	NA	14	82.14	63.88	0.70
12	PSO-LSTM III	LSTM(8)-LSTM(12)	601	8	83.35	68.55	0.69
13	PSO-LSTM I	LSTM(50)	602	8	83.55	68.25	0.69
14	PSO-RNN II	RNN(25)-Dense(73)	275	9	84.33	69.74	0.68
15	PSO-RNN I	RNN(25)	275	9	87.05	72.55	0.66

Table 5.3: Optimal hyperparameters and performance for Aizawl rainfall

16	PSO-SVR III	Poly Kernel, C=1.0, epsilon = 0.1	NA	13	110.29	86.99	0.45

The findings indicate that, among the various versions of BiLSTM models, PSO-BiLSTM II demonstrated the highest level of prediction accuracy. The PSO-BiLSTM II architecture comprises a BiLSTM Layer with 3 cells, followed by a Dense layer containing 103 neurons and an output Dense layer with 1 neuron. In this particular dataset, all the variations of PSO-BiLSTM models outperformed the other models being compared. Within the range of LSTM models tested, PSO-LSTM II, consisting of an LSTM layer with 50 neurons followed by a Dense layer comprising 148 neurons and an output Dense layer with 1 neuron - achieved the lowest RMSE score. However, both single-layered and double-layered configurations for PSO-LSTM performed worse than PSO-ANN and PSO-SVR models. The optimum number of previous timesteps for each PSO-LSTM model is also found to be 8 timesteps.

Among the ANN models, PSO-ANN I, consisting of a single Dense layer, yields better metrics than the model with 2 Dense layers. It also outperforms all the PSO-SVR and PSO-RNN models. For the SVR models, the linear kernel in the PSO-SVR model provides better predictions than both the RBF kernel and polynomial kernel. Additionally, both linear and RBF kernels in the PSO-SVR model perform better than the PSO-RNN models. Among all the compared PSO-optimized models, it is observed that the performance of the polynomial kernel in the PSO-SVR III model is the lowest.

The results demonstrate that the best number of previous timesteps returned by the PSO algorithm varies across different models. Specifically, for PSO-BiLSTM models, 13 previous timesteps were found to be the optimal timesteps, while PSO-LSTM models performed best with 8 timesteps. The optimal number for PSO-RNN models was 9, whereas, for PSO-ANN models, it ranged between 13 and 16. Lastly, the optimal

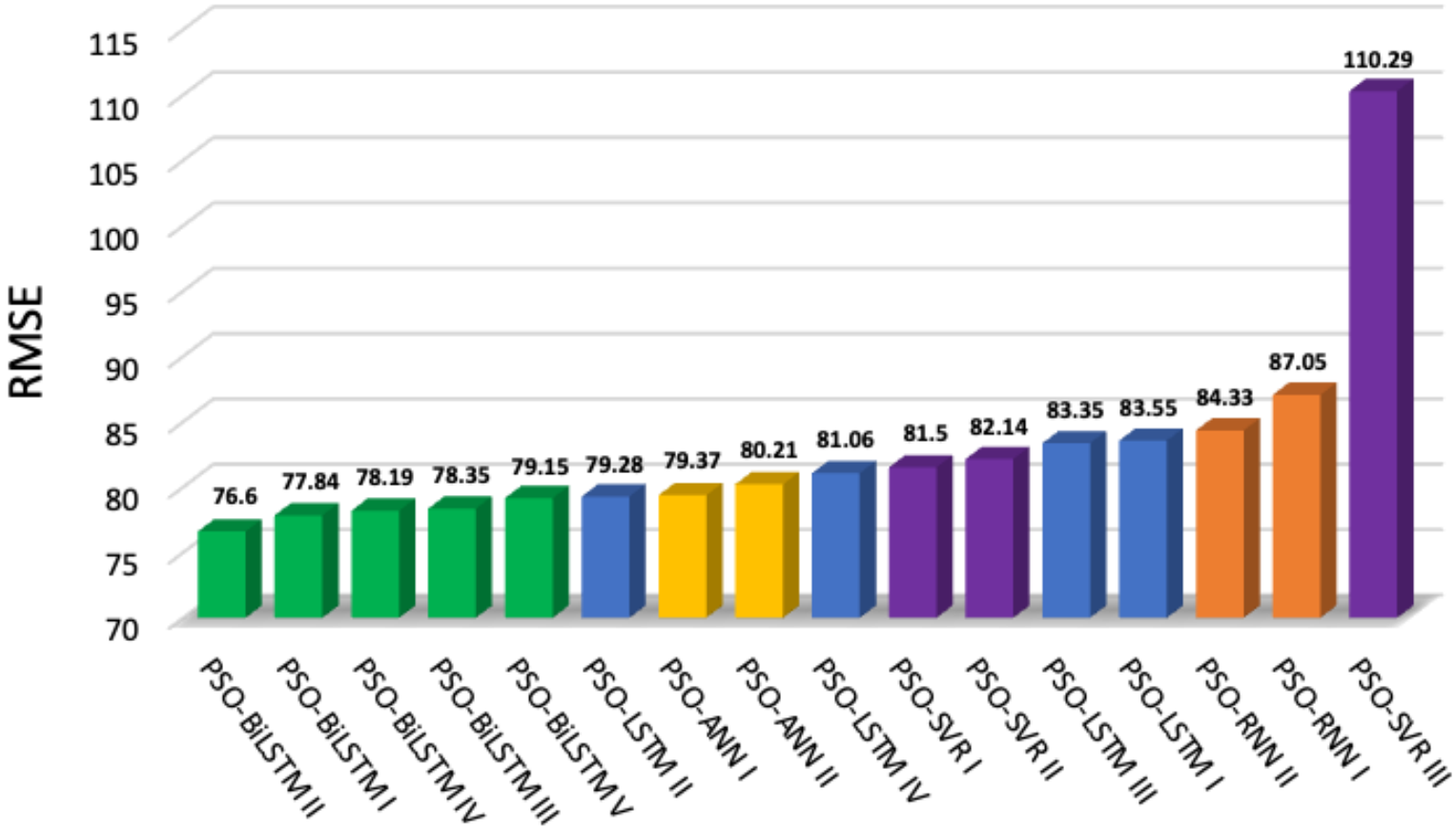

Figure 5.5: The plot of Aizawl monthly rainfall prediction

number of previous time steps for PSO-SVR models fell within the range of 13 to 14. These findings suggest that the appropriate number of prior time steps to input into a model depends on its specific type and cannot be universally fixed. In addition to achieving superior scores, the PSO-BiLSTM models also required fewer epochs (ranging from 54 to 69) compared to other models, which required significantly more epochs (between 264 and 602) to attain optimum parameter settings.

Figure 5.6 shows the behavior of the PSO algorithm for the PSO-BiLSTM II model in 3-dimensional space. In this figure, the dimensions for the number of cells in the LSTM layer, the number of neurons in the Dense layer, and the RMSE metrics are only shown in the figure due to the large dimension of the PSO search space. The range of the parameters for the Dense, RNN, and LSTM layer neurons, units, and cells are 1 to 150, 30 to 1000 epochs, and 1 to 30 previous input timesteps.

In the SVR models, the hyperparameters range from 1 to 30 for previous timesteps, from C=1 to C=10 for regularization parameter C, and from epsilon=0 to epsilon=1. The PSO is initialized with 10 particles and runs through 300 iterations. The particle acceleration constants c1 and c2 are 1.5 and 2.0, respectively. By examining Figure 10, it is apparent that the particles in the PSO are initially positioned randomly and then proceed toward the optimal hyperparameters for the entire population.

Figure 5.7, Figure 5.8, Figure 5.9, Figure 5.10, and Figure 5.11 display the plot of the RMSE for Aizawl Monthly Rainfall (AMR) prediction. The RMSE represents the difference between the actual rainfall value and the predicted value for various models, including PSO-BiLSTM II, PSO-LSTM II, PSO-ANN I, PSO-SVR I, and PSO-RNN II. Each model was repeated ten times to ensure accuracy. However, in the case of the PSO-SVR model, only one training and testing cycle was conducted, as each prediction produced identical results. The figures demonstrate that all models performed well in learning from the training data. Furthermore, a mean value was calculated for each model's predictions and plotted in Figure 5.12. Additionally, a Box and Whisker plot depicting RMSE values is shown in Figure 5.13. Upon analyzing these plots, it becomes evident that the PSO-LSTM model consistently provided stable predictions

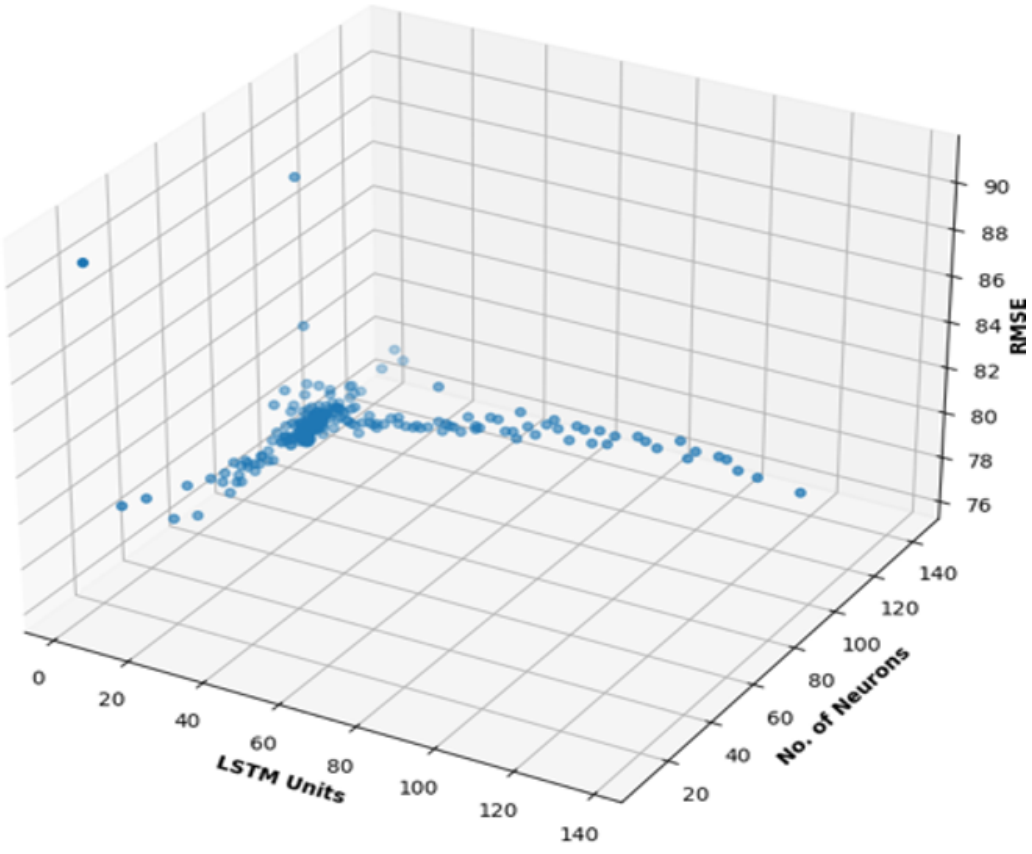

Figure 5.6: PSO behavior for PSO-BiLSTM II model

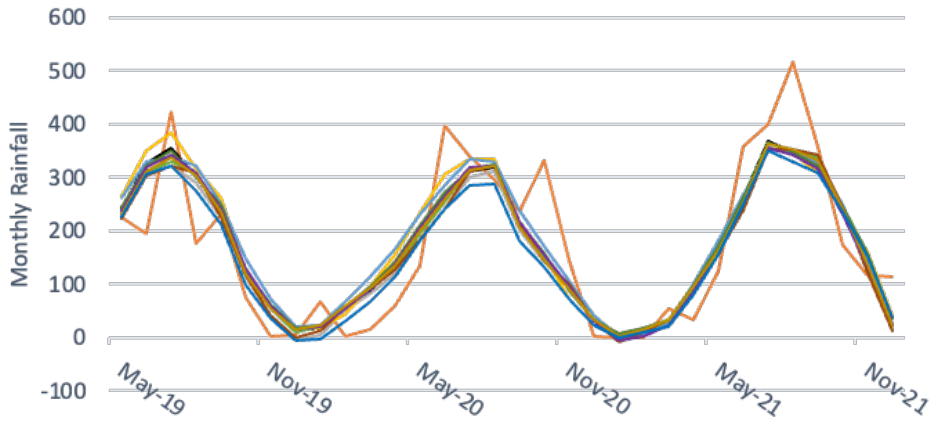

Figure 5.7: AMR - Actual vs Predicted for PSO-BiLSTM II

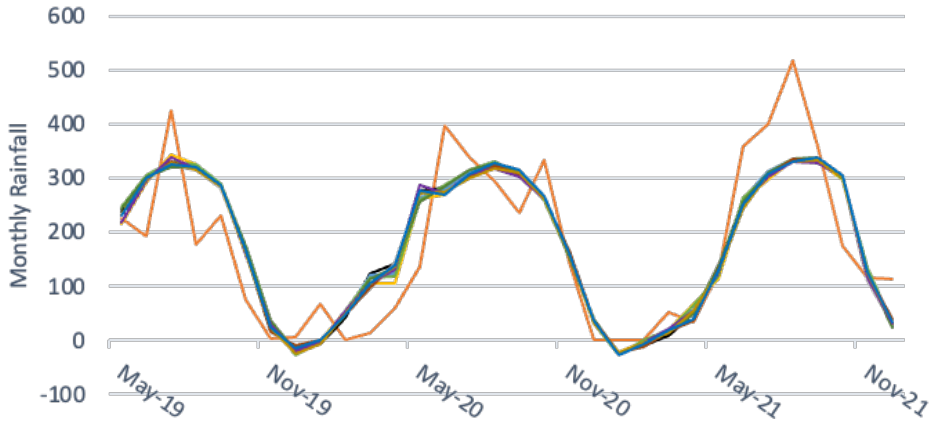

Figure 5.8: AMR - Actual vs Predicted for PSO-LSTM II

with minimal variability. In contrast, the predictions generated by PSO-RNN exhibited significant variability across different metrics

5.4.2 Prediction using all-India monthly average rainfall

The dataset utilized for predicting all-India monthly average rainfall was compiled by Kothawale and Rajeevan (2017). It covers a period spanning from 1871 to 2016. The measurements are expressed in units of tenths of millimeters. When examining this dataset, it is found that the minimum recorded rainfall value is 3 units, while the maximum value reaches as high as 3460 units. On average, the rainfall amounts

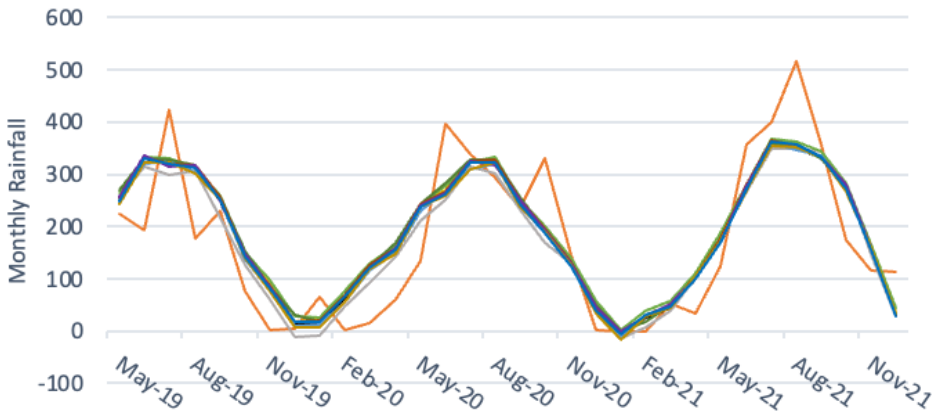

Figure 5.9: AMR - Actual vs Predicted for PSO-ANN I

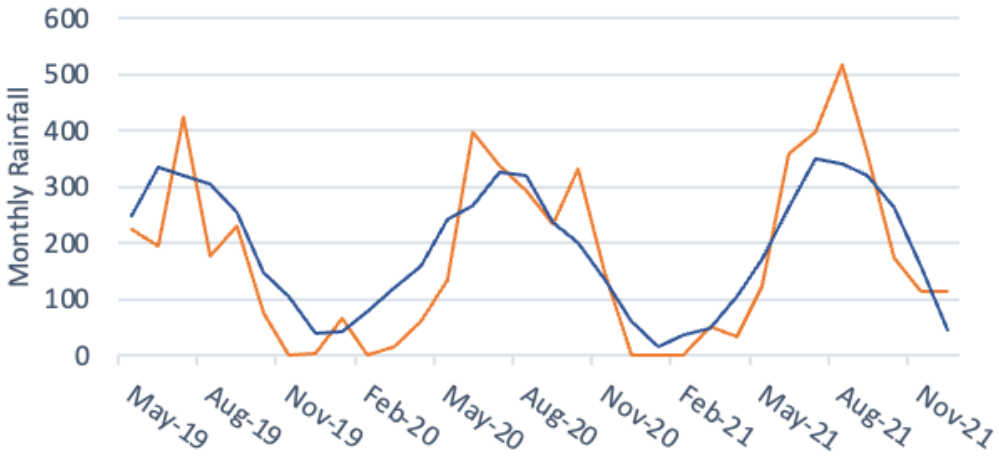

Figure 5.10: AMR - Actual vs Predicted for PSO-SVR I

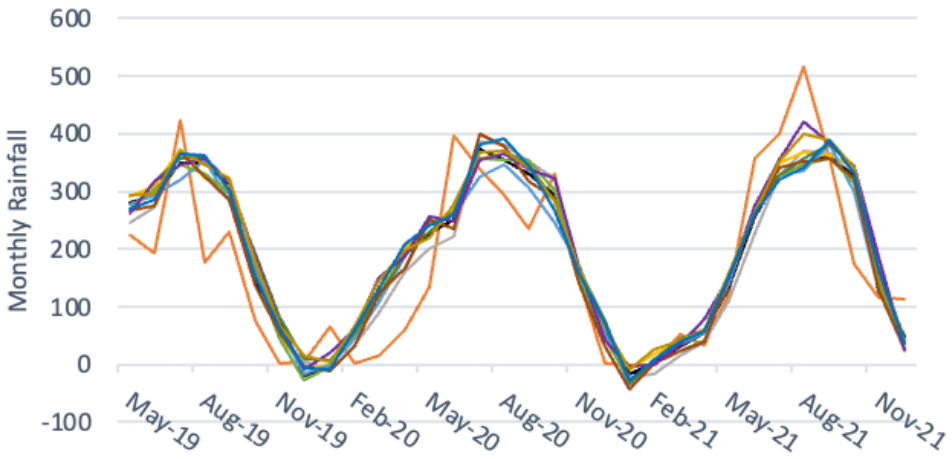

Figure 5.11: AMR - Actual vs Predicted for PSO-RNN II

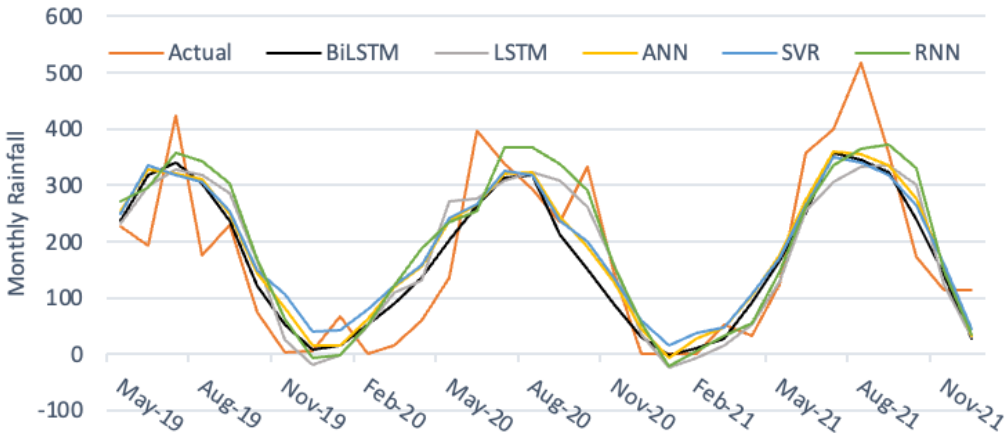

Figure 5.12: AMR – Mean of the Actual vs Predicted for all models

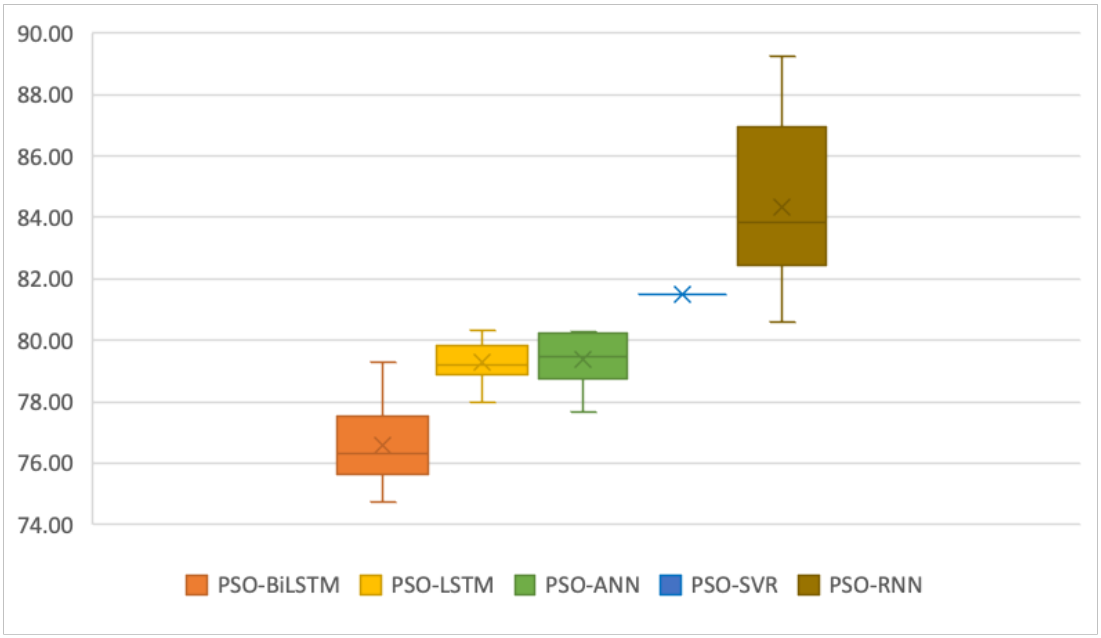

Figure 5.13: AMR - Box and Whisker plot of RMSE

to approximately 904.94 units, with a standard deviation of 951.71. The dataset was divided into three subsets to facilitate further analysis and model development: training, validation, and testing data. These subsets have been allocated in an 80:10:10 ratio, respectively, meaning that eighty percent of the dataset is utilized for training purposes while ten percent each is reserved for validation and testing. The dataset is normalized within a range from zero to one using the Min-Max normalization method. Once predictions are obtained using this normalized scale, these predicted values are transformed back into their original scale for accurate interpretation and comparison with actual observations through inverse transformation techniques.

The results of the prediction for India Monthly Rainfall (IMR) are presented in Table 5.4, while Figure 5.14 displays the plot depicting the RMSE between the actual and predicted rainfall. The hidden layer architecture in the result table does not include the output Dense layer, which has one neuron. Based on these findings, it is evident that the PSO-BiLSTM IV model, comprising two BiLSTM layers with 77 and 53 LSTM cells, respectively, followed by a Dense layer with 112 neurons and an output Dense layer with one neuron, achieved the lowest RMSE score. Additionally, on average,

the PSO-BiLSTM models demonstrated superior prediction performance compared to other models under consideration.

The PSO-LSTM IV model, which comprises two LSTM layers containing 113 and 44 LSTM cells, respectively, followed by a Dense layer with 84 neurons and an output Dense layer with a single neuron, ranks as the second-best model for this dataset. It also has a lower RMSE score than the PSO-BiLSTM I, II, and III models. This finding indicates that stacking layers yields better results than using single LSTM or BiLSTM layers alone in the context of this dataset. Among all the compared models, PSO-SVR I with a linear kernel exhibits the poorest performance. These results also highlight that incorporating a Dense layer in conjunction with the output Dense layer enhances the accuracy of model predictions compared to a model with only the BiLSTM, LSTM, or RNN layers.

The Kumar-LSTM and Kumar-RNN are models developed by Kumar *et al.* (2019) to forecast monthly average rainfall across India. These models utilize a single LSTM and RNN layer, each with a single LSTM and RNN cell. On the other hand, the Grid-LSTM I, Grid-LSTM II, and Grid-BiLSTM models involve hyperparameters determined through the Grid search method. This study demonstrates that the PSO-optimized models, excluding the SVR model, outperformed both the benchmark models and those employing the grid search method regarding accuracy and performance.

Table 5.4: Optimal hyperparameters and performance for all-India rainfall

Sl No.	**Model Name**	**Hidden Layer Architecture**	**Epoch**	**Time-steps**	**RMSE**	**MAE**	**R^2**
1	PSO-BiLSTM IV	BiLSTM(77)-BiLSTM(53)-Dense(112)	87	27	225.12	166.02	0.94

Table 5.4: Optimal hyperparameters and performance for all-India rainfall

2	PSO-LSTM IV	LSTM(113)-LSTM(44)-Dense(84)	62	27	226.14	162.62	0.94
3	PSO-BiLSTM II	BiLSTM(6)-Dense(141)	207	28	229.21	166.29	0.94
4	PSO-BiLSTM I	BiLSTM(6)	207	28	229.67	165.78	0.94
5	PSO-BiLSTM III	BiLSTM(81)-BiLSTM(82)	97	24	230.31	168.97	0.94
6	PSO-LSTM II	LSTM(114)-Dense(37)	148	25	231.02	168.81	0.94
7	PSO-RNN II	RNN(15)-Dense(68)	180	27	232.97	170.29	0.94
8	PSO-RNN I	RNN(50)	180	26	235.65	170.96	0.94
9	PSO-ANN I	Dense(17)	508	28	240.21	176.13	0.93
10	PSO-SVR I	Linear Kernel, C=8.8, epsillon = 0.1	NA	28	279.37	213.16	0.91
11	Kumar-RNN	RNN (1)	500	20	261.7	279.6	0.92
12	Kumar-LSTM	LSTM (1)	500	12	251.63	169.35	0.93
13	Grid-LSTM I	LSTM (50)	400	12	250.47	174.35	0.93
14	Grid-LSTM II	LSTM (12)-LSTM (12)-Dense (12)	500	28	245.3	168.64	0.93
15	Grid-BiLSTM	BiLSTM(80)	150	24	240.79	168.48	0.93

Figure 5.15, Figure 5.16, Figure 5.17, Figure 5.18, and Figure 5.19 display the plot of the actual and predicted values of the models, which are repeated 10 times each. However, only the plot for the top-performing model in each variant is shown in these plots. Figure 5.20 shows the plot of the actual and mean values of all the best-performing models for each variation. Additionally, Figure 5.21 presents a Box and Whisker plot illustrating the RMSE. It can be observed from this graph that, compared to other models, the PSO-RNN model exhibits greater instability in its RMSE values.

5.5 Conclusions

Accurate rainfall prediction is of utmost importance when making well-informed decisions in various aspects of human activities. In this research, the performance of Particle Swarm optimized deep learning and machine learning models, namely PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR were compared using two datasets - Aizawl monthly rainfall and all-India monthly average rainfall datasets. This study marks the first time machine learning and deep learning models have been utilized to analyze Aizawl rainfall data. It is also the first instance where the models' predictive capabilities (PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN) have been compared for their effectiveness in forecasting rainfall patterns.

In the case of the Aizawl monthly rainfall dataset, it has been observed that all versions of PSO-BiLSTM models performed better than compared models, with PSO-SVR using a polynomial kernel showing the worst performance. The most effective model is the PSO-BiLSTM II model, which includes a BiLSTM Layer with 3 cells, followed by a Dense layer featuring 103 neurons. Stacking the BiLSTM, LSTM, RNN, and ANN models does not improve model prediction for this dataset. However, adding a Dense layer before the output Dense layer increased prediction accuracy for the BiLSTM, LSTM, and RNN models. It was observed that each model produced different optimal input timesteps when using the PSO algorithm. This finding suggests

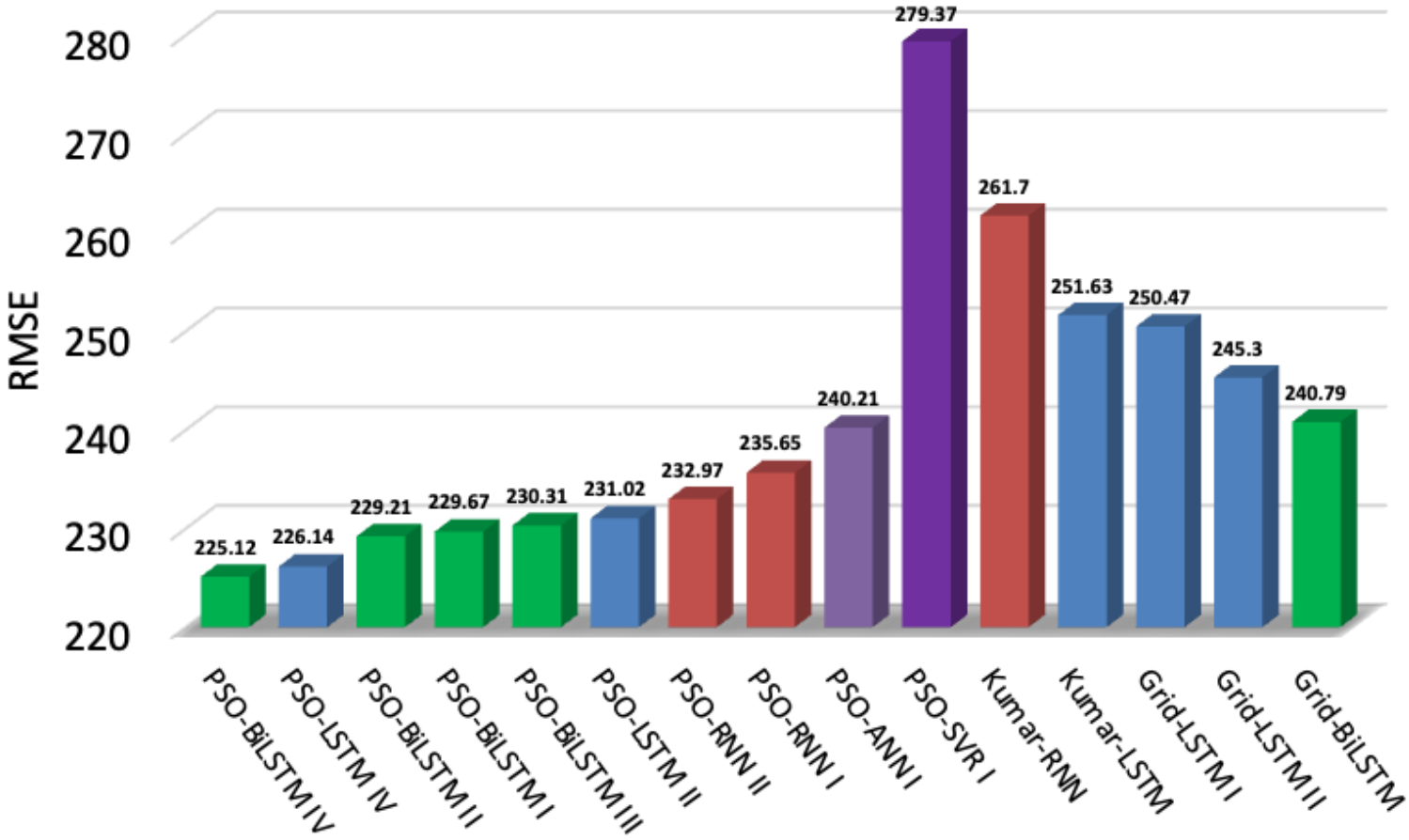

Figure 5.14: The plot of all-India average monthly rainfall prediction

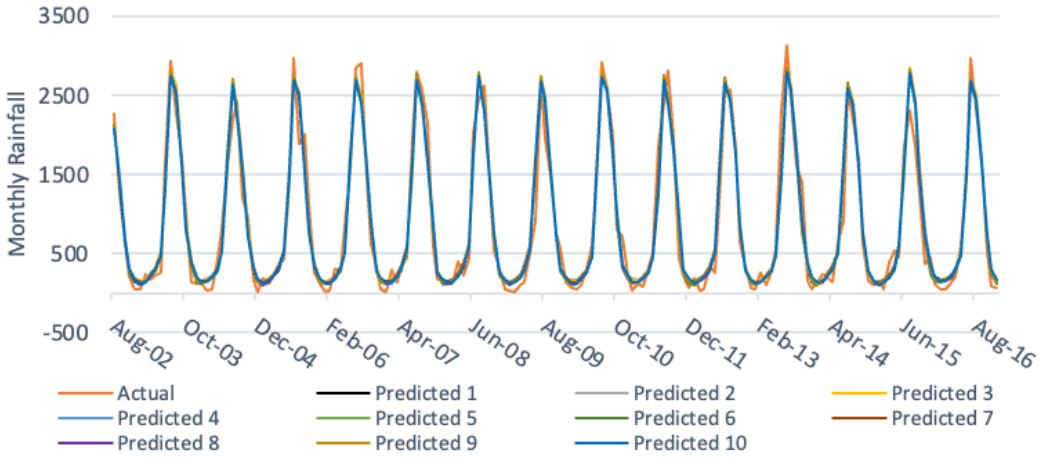

Figure 5.15: IMR - Actual vs Predicted for PSO-BiLSTM IV

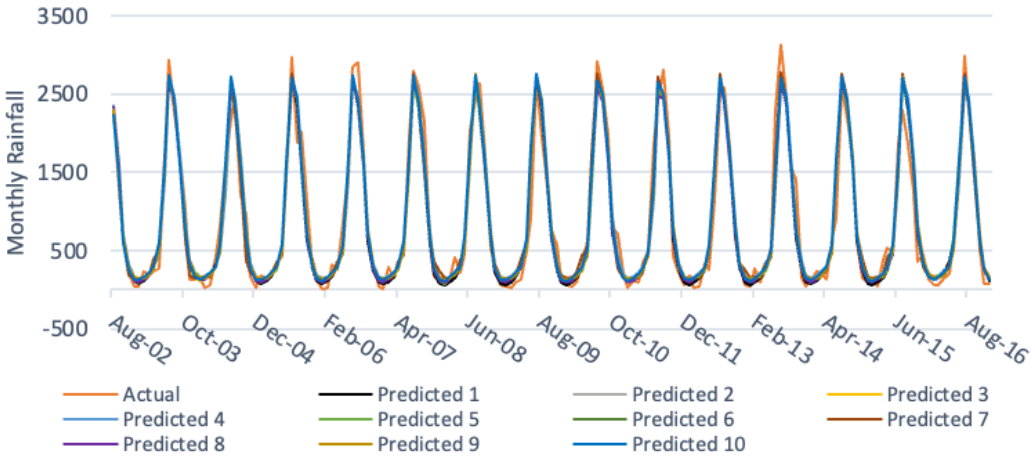

Figure 5.16: IMR - Actual vs Predicted for PSO-LSTM IV

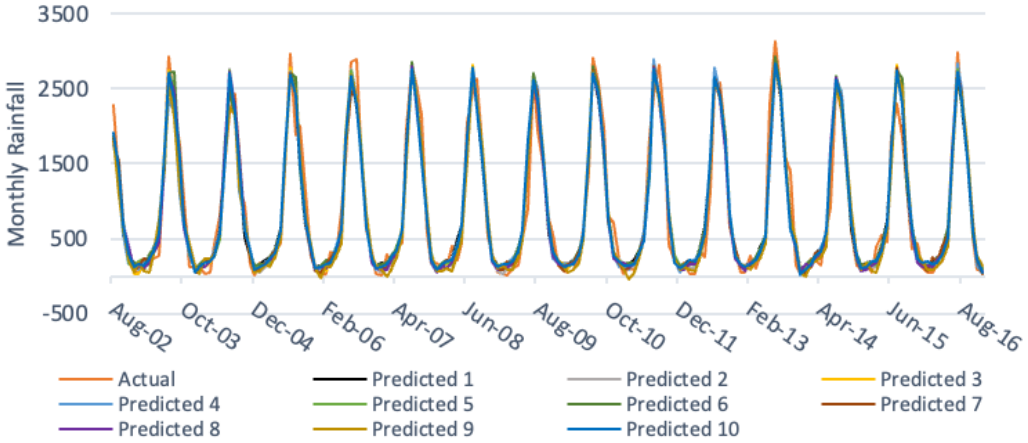

Figure 5.17: IMR - Actual vs Predicted for PSO-RNN II

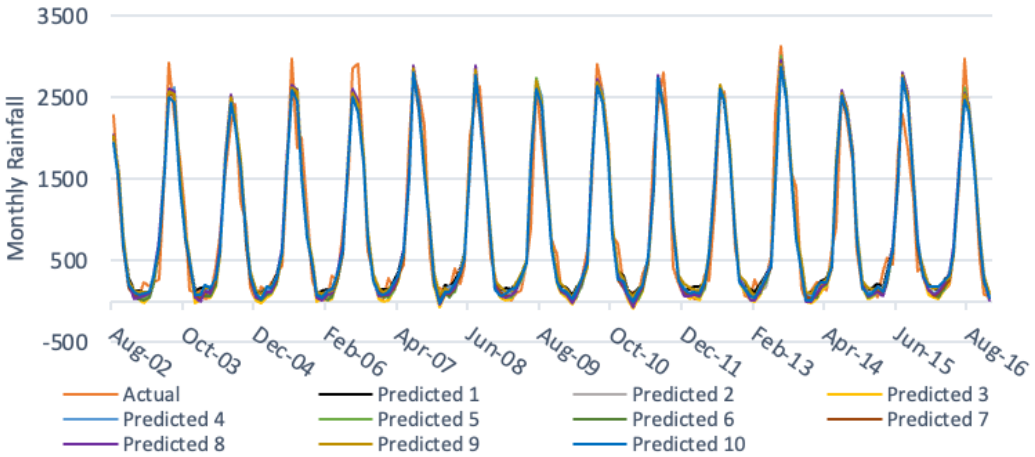

Figure 5.18: IMR - Actual vs Predicted for ANN I

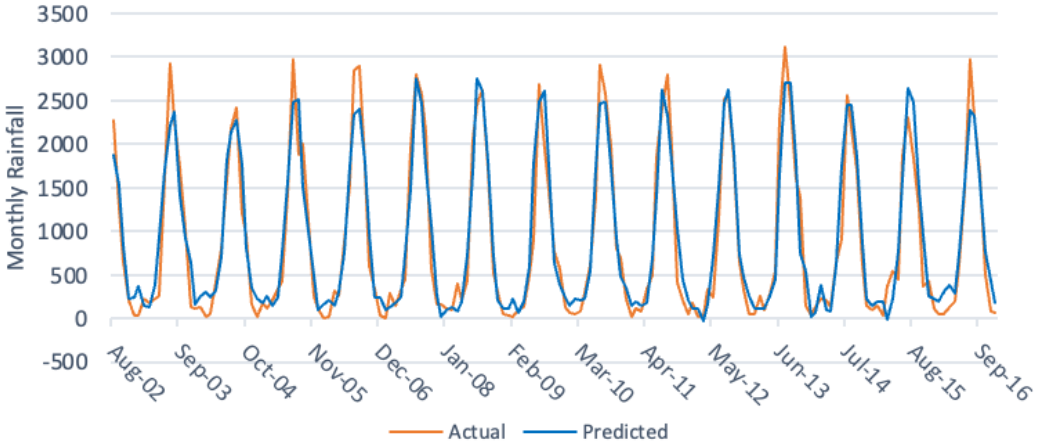

Figure 5.19: IMR - Actual vs Predicted for PSO-SVR I

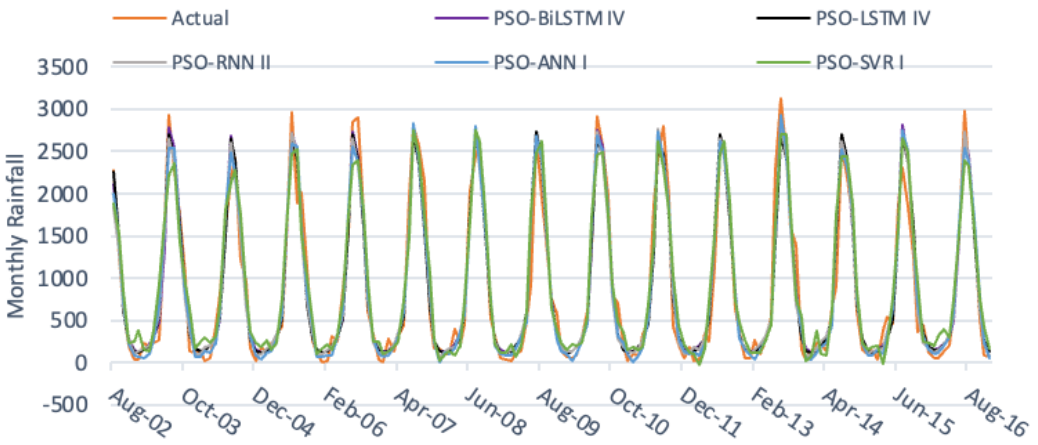

Figure 5.20: IMR - Actual vs Mean of all models

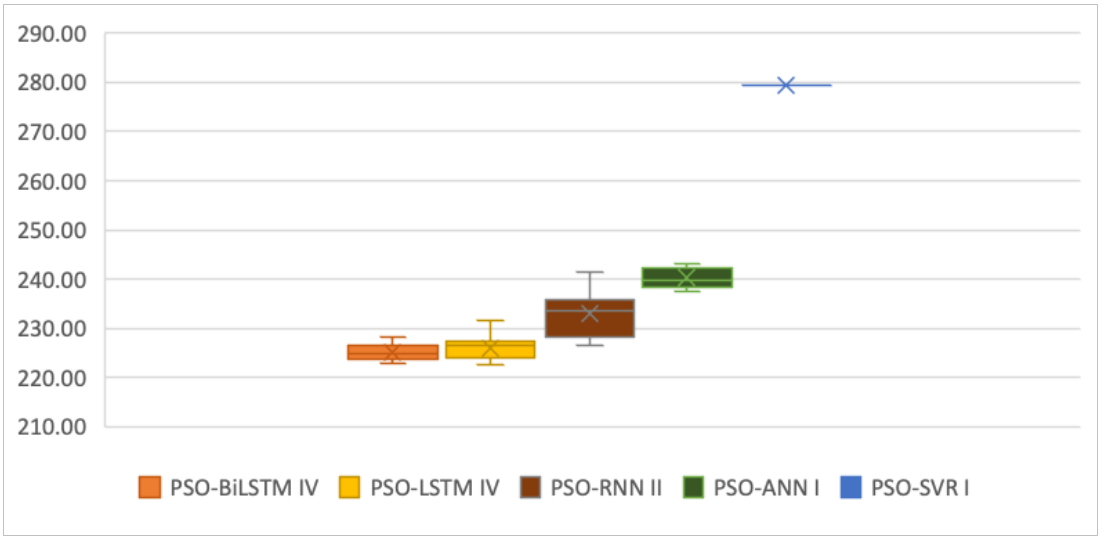

Figure 5.21: IMR - Box and Whisker plot of RMSE

that determining the best input timesteps depends on both the type of models used for prediction and the specific dataset utilized. Additionally, it was found that, compared to other models examined in this study, the BiLSTM model required fewer epochs to converge toward a solution.

In the all-India average monthly rainfall dataset, all the PSO-BiLSTM models achieved the lowest RMSE score compared to other models. Additionally, models with 2 layers of BiLSTM stacked with a Dense layer and 2 layers of LSTM stacked with a Dense layer performed better compared to other models. Similar trends were observed in the Aizawl monthly rainfall dataset. BiLSTM, LSTM, and RNN stacked with a Dense layer demonstrated superior prediction performance compared to models without a Dense layer. Among all the compared models in this dataset, it was found that the PSO-SVR model exhibited the worst performance. Furthermore, all PSO-optimized models outperformed those models with their hyperparameters determined through the Grid search methods and the benchmark models from existing literature.

This finding indicates that utilizing the PSO algorithm is reliable for identifying optimal hyperparameters. However, due to its stochastic nature, relying solely on one model prediction could lead to incorrect conclusions about its superiority over other models. Therefore, model training and testing were repeated 10 times in this study

as an extra cautionary measure. The findings from this investigation also demonstrate that the model's hyperparameter may vary based on the specific characteristics and scale of the dataset employed. Consequently, it is recommended that the most suitable hyperparameters for training a model be determined while using a new dataset.

Chapter 6

PSO Hybrid Deep Learning Models for The Prediction of Daily Rainfall Using Multivariate Data [5]

6.1 Introduction

Rainfall prediction plays a crucial role in various sectors and aspects of human society. Mizoram, situated in the north-eastern part of India, lies between the geographical coordinates of 21° 58' and 24° 35' N latitude and 92° 15' and 93° 29' E longitude. This state shares its borders with Tripura, Assam, and Manipur within India and a border spanning approximately 722 kilometres with the neighbouring countries of Bangladesh and Myanmar. Mizoram experiences significant yearly rainfall, averaging between 2,500 mm to 3,000 mm. Hills and rugged landscapes characterize the terrain in this region, with changing altitudes ranging from sea level to slightly over 2,000 m above sea level. Landslides happen frequently during the monsoon seasons in Mizoram, causing significant damage.

A recent analysis of meteorological data in Mizoram was carried out by Ralte and Sil (2021) to study drought patterns within the state. In addition, Saha, Das, *et al.* (2021) conducted a quantitative assessment of rainfall forecasts at the district level for Mizoram. Several researchers have also utilized machine learning and deep learning models to predict rainfall in various parts of India (Chakraverty and Gupta, 2008; Dash *et al.*, 2018; Gope *et al.*, 2016; Poornima and Pushpalatha, 2019). However, there has been no previous application of machine learning or deep learning models to analyze

[5] The content of this article is published as a research article in: Zoremsanga and J. Hussain, (2024). "Particle Swarm Optimized Deep Learning Models for Rainfall Prediction: A Case Study in Aizawl, Mizoram". *IEEE Access* (pp. 1–13). doi: 10.1109/ACCESS.2024.3390781.

meteorological data specifically for Mizoram.

This chapter presents an exhaustive comparison of twelve hybrid deep learning and machine learning models optimized using PSO. These models include BiLSTM, LSTM, RNN, ANN, and SVR. The main objective of this study is to predict the amount of rainfall using the five meteorological data observed in the previous days.

6.2 Study area and data source

This study utilized a dataset comprising 12,418 days of meteorological data obtained from the Aizawl Weather Station, which is under the supervision of the State Meteorological Centre, Directorate of Science and Technology, Govt. of Mizoram, India shown in Figure 5.2. The dataset includes five parameters: Minimum humidity, maximum humidity, minimum temperature, maximum temperature, and rainfall, recorded daily between 1985 and 2018. The study area experiences substantial annual rainfall ranging from 2,500 mm to 3,000 mm. The meteorological parameters collected by the State Meteorological Centre are limited to five; however, due to large missing values in the data set, the Minimum humidity was not considered for this study. The four meteorological parameters considered in this study are plotted in Figure 6.1, Figure 6.2, Figure 6.3, and Figure 6.4.

6.3 Methodology

6.3.1 Data preprocessing

Training and testing data

The meteorological data collected in this study has been divided into three sets: the training, validation, and testing datasets. The training dataset consists of a total of 11,322 days of data from the years 1985 to 2015. The validation dataset, on the other hand, includes 731 days from 2016 to 2017. To evaluate the performance of the proposed model, the study utilized data from rainy seasons for testing purposes. The

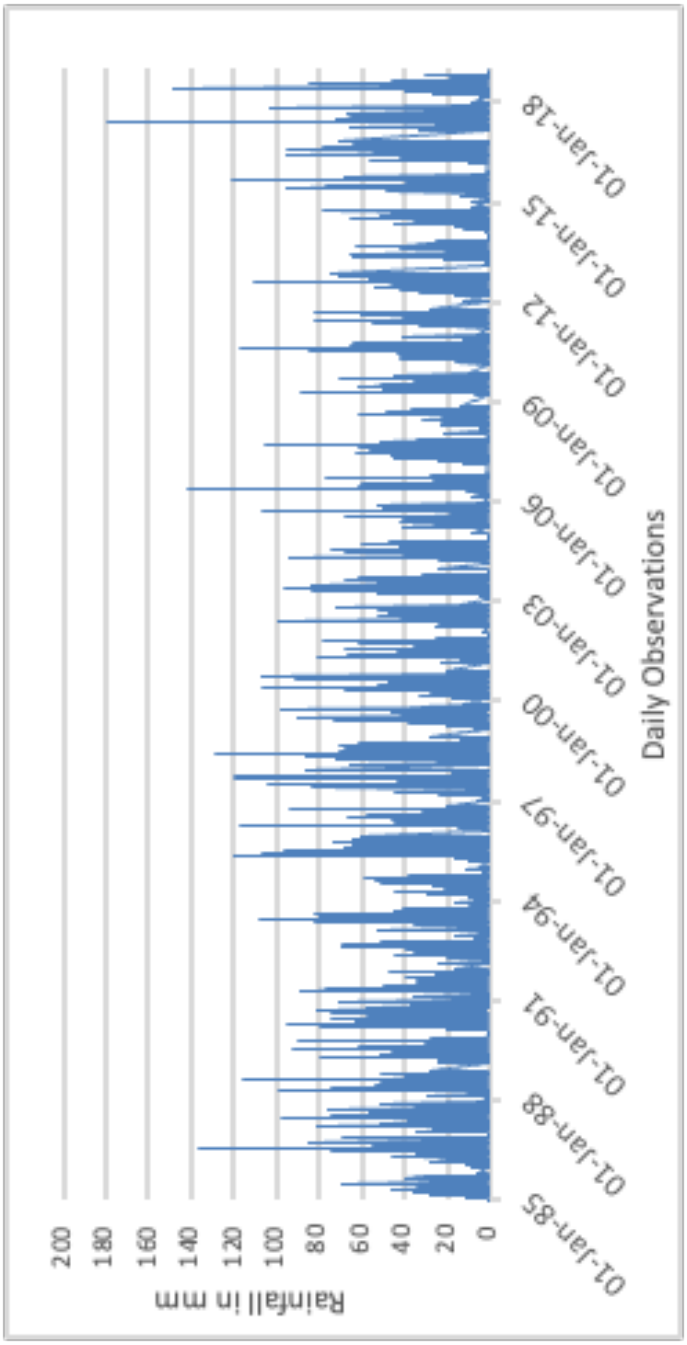

Figure 6.1: Daily Rainfall in Aizawl from 1985 to 2018

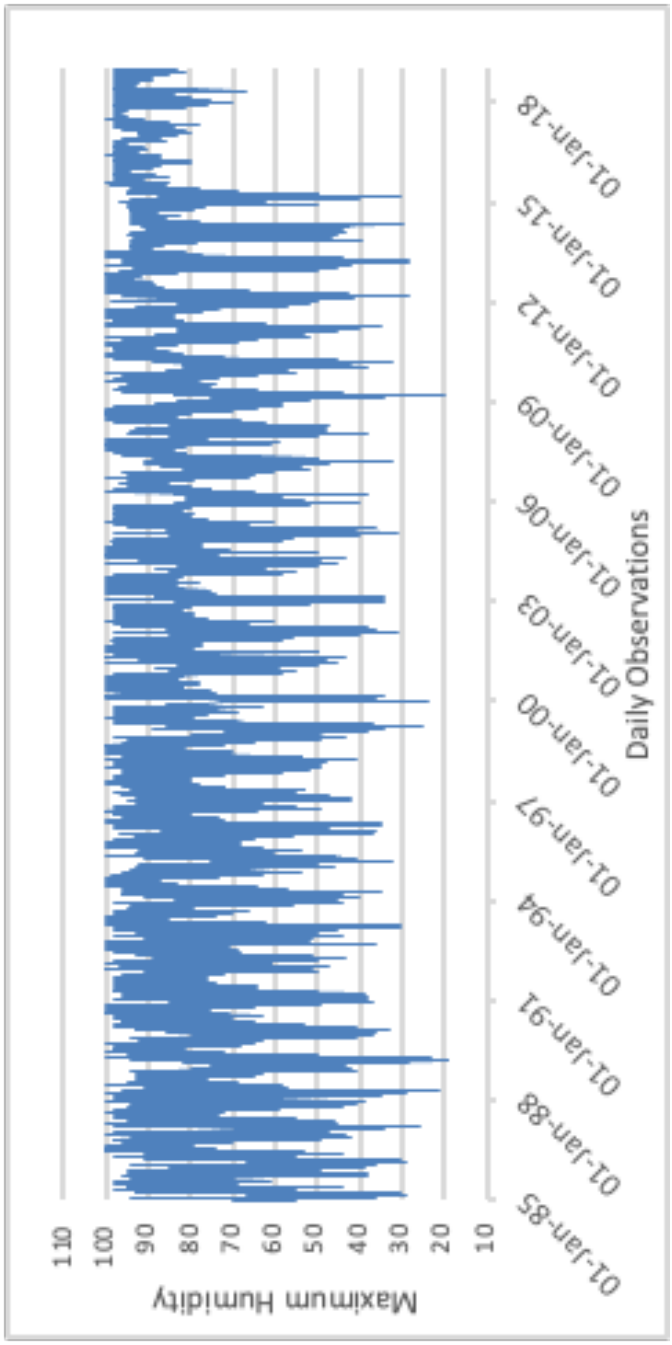

Figure 6.2: Daily Maximum Humidity in Aizawl from 1985 to 2018

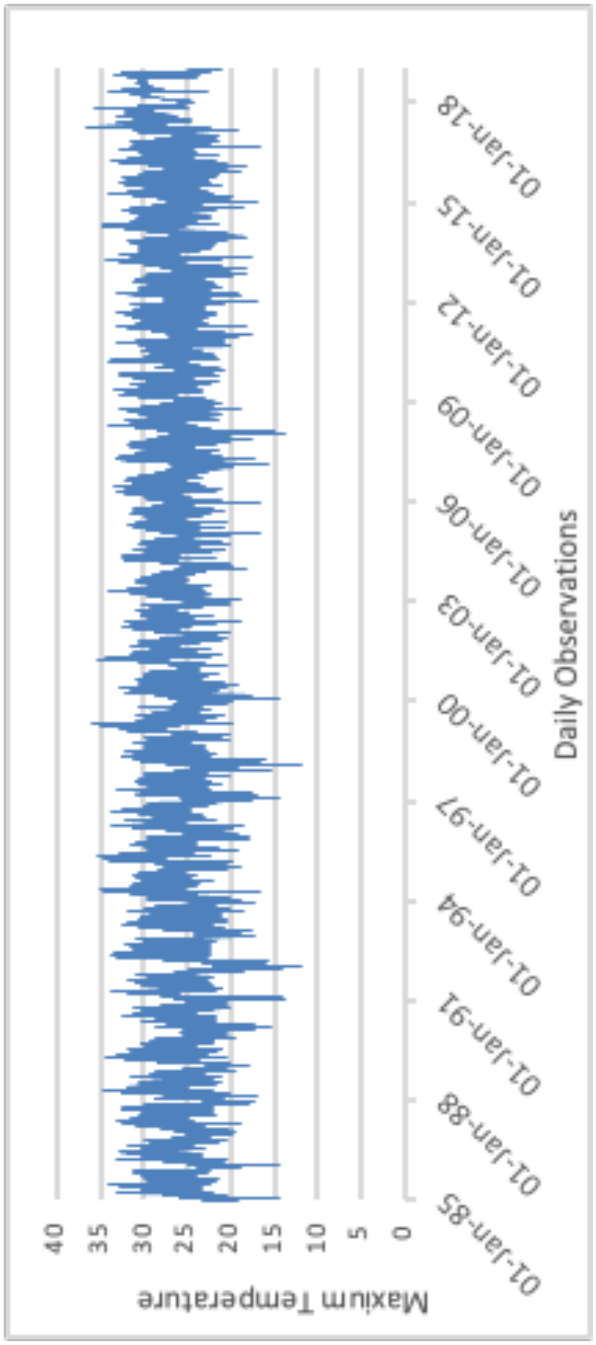

Figure 6.3: Daily Maximum Temperature in Aizawl from 1985 to 2018

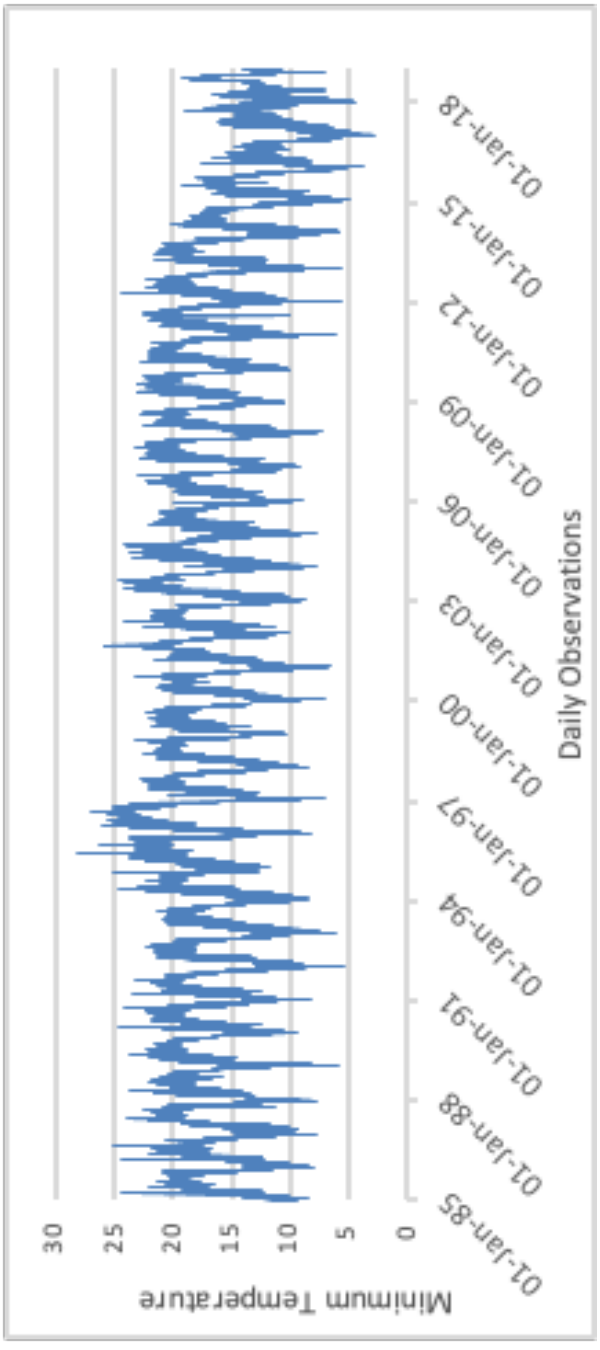

Figure 6.4: Daily Minimum Temperature in Aizawl from 1985 to 2018

testing dataset comprises 183 days of meteorological data ranging from 1st April 2018 to 30th September 2018.

Normalization of datasets

The Min-Max normalization technique discussed in Chapter 3 is used for normalizing the training and testing data.

6.3.2 Performance Metrics

The performance of the models under study was evaluated using three statistical measures: MAE and RMSE, defined in Chapter 3 and R^2 defined in Chapter 5.

6.3.3 PSO, SVR, ANN, RNN, LSTM, Stacked LSTM, and BiLSTM

The RNN and LSTM models are described in Chapters 3. The Stacked LSTM and BiLSTM models are explained in Chapters 4. The PSO, SVR and, ANN models are described in Chapters 5

6.3.4 Model development

The main objective of this study is to conduct a comparison between PSO-optimized hybrid deep learning models and machine learning models, specifically PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR. The data for this study was collected from the Aizawl Weather Station maintained by the State Meteorological Centre in Mizoram, India. The dataset has 12,418 days of meteorological information, including Maximum Humidity, Minimum Temperature, Maximum Temperature, and Rainfall, spanning from 1985 to 2018. Several parameters are considered to optimize the performance of these models using particle swarm optimization (PSO). For the ANN, optimization includes determining the optimal number of neurons. For the LSTM and RNN layers, optimization involves finding the optimal number of cells and other factors like the number of epochs and dropout rate. For the SVR model, the optimal number of previous timesteps for input and the values for regularization

parameter C and margin of tolerance, epsilon, are optimized. To evaluate how well each model performs compared to other models under consideration, metrics like RMSE, MAE, and R^2 are used. The models under study aim to minimize the MSE loss function utilizing the Adam optimizer.

The meteorological data collected is pre-processed, including cleansing the data and filling in missing information. Afterward, the data is converted into a format appropriate for supervised learning. The subsequent steps involve splitting the data into training, validation, and testing datasets and normalizing it using the Min-Max Normalization technique. The optimization and training of the models then follow a predefined set of procedures outlined in Figure 6.5. These steps are described in detail as given below:

Step 1: The dataset is imported, pre-processed, and converted into a format suitable for supervised learning. Then, it is split into training, validation, and testing datasets. Furthermore, the dataset is normalized using the Min-Max normalization technique.

Step 2: Secondly, the PSO method is used to optimize the hyperparameters of the models, including the number of neurons in the ANN layer, the number of cells in the LSTM and RNN layer, the number of epochs, dropout rate, and the number of timesteps for the input, The PSO also optimized the SVR regularization parameter C and the SVR margin of tolerance, epsilon. Each hyperparameter's lower and upper values are limited to a particular value. The number of neurons in the ANN layer and the number of cells in the LSTM or RNN layer are within the values of 1 to 150, the number of epochs within 30 to 600, and the number of input timesteps from 1 to 30. For the SVR model, the regularization parameter C is from 1 to 10, and the epsilon is from 0 to 1. The PSO is initialized with 10 particles, and the algorithm is repeated 300 times. The particle acceleration constants c1 and c2 are 1.5 and 2.0, respectively.

Step 3: Each model is defined using the hyperparameter values produced by the PSO algorithm, and the models are compiled using the MSE loss function and Adam optimizer.

Step 4: The training, validation, and testing datasets are used to train, validate, and

test each model. Then, the models' outputs are reversed and converted to their original scale. Performance metrics such as RMSE, MAE, and R^2 are then computed to measure the accuracy between predicted and actual values. This process is repeated ten times to obtain an average performance score. The mean RMSE is subsequently employed as the fitness value for the PSO algorithm.

Step 5: The optimal hyperparameters of the models are obtained by calculating and updating the best positions, Pbest and Gbest, for each particle. This process is repeated 300 times. After completing these 300 iterations, the PSO algorithm generates the optimal hyperparameters for the models.

Step 9: The performance of the PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR are compared to determine the best model.

6.4 Results and discussion

This chapter compares the prediction performance of PSO-optimized hybrid deep learning and machine learning models. The study's primary objective is to make accurate predictions regarding the rainfall expected for the next day. To achieve this, meteorological data obtained from the Aizawl Weather Station in Mizoram, India, is used for training and testing. The optimization process involves determining the ideal number of neurons in the ANN layer and cells in both LSTM and RNN layers, as well as epochs, dropout rate, and previous timesteps for input using the PSO method. For the SVR model, parameters such as previous timesteps, regularization parameter C, and tolerance margin epsilon are optimized using the PSO method. A range of metrics, including RMSE, MAE, and $R^{2,}$ are used in this study to assess the performance of these models effectively.

Table 6.1 presents the descriptive statistics for the Aizawl meteorological dataset. The data reveals that the daily rainfall ranges from 0 mm to 180.0 mm, with an average of 5.8 mm and a standard deviation of 13.3 mm. The Maximum Humidity values vary between 19.0 and 100.0, with a mean value of 80.2 and a standard deviation of

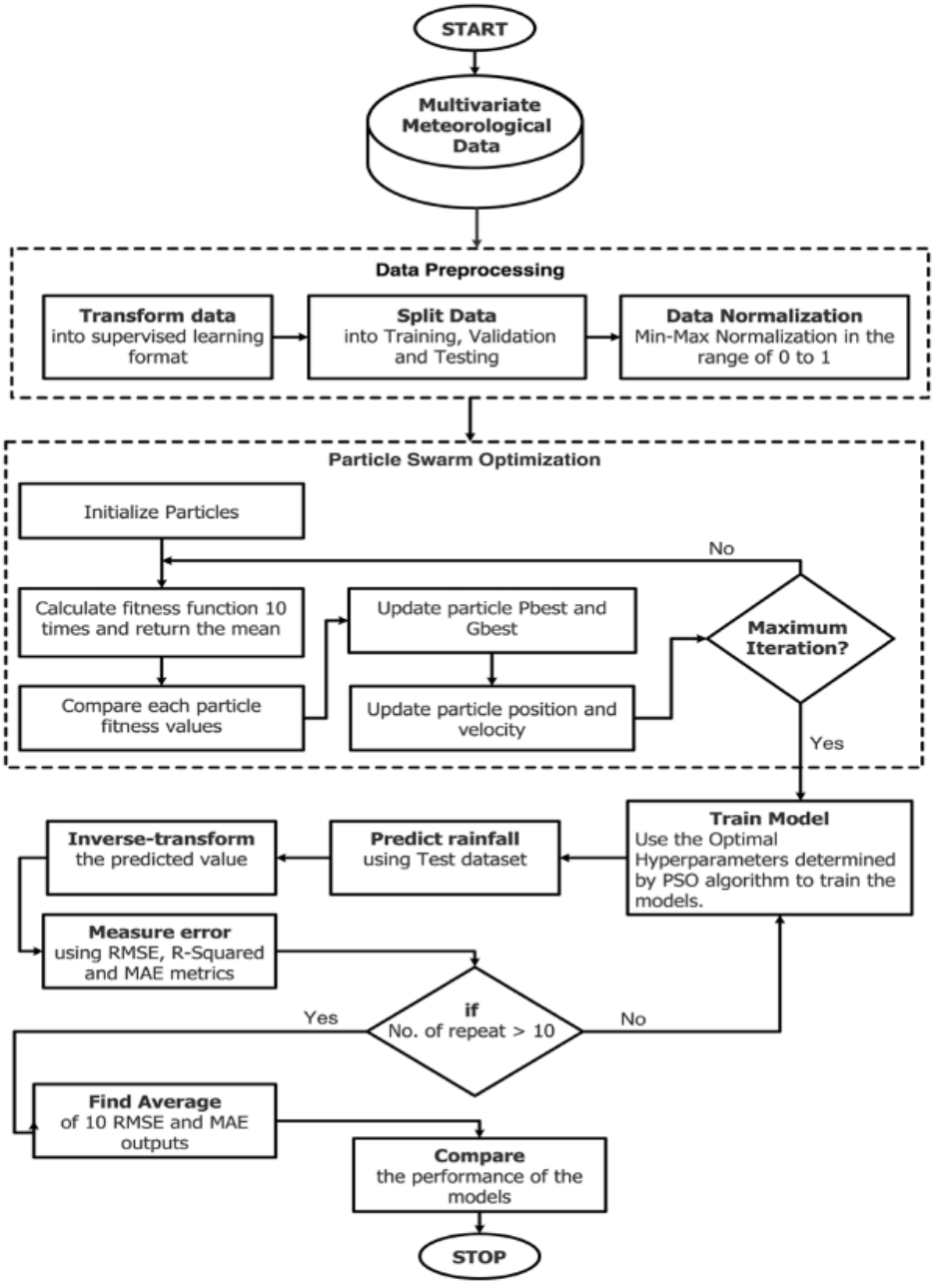

Figure 6.5: Flowchart for predicting Aizawl rainfall using PSO models

Table 6.1: Descriptive Statistics of Aizawl meteorological data

Parameters	Mean	Standard Deviation	Minimum	Maximum
Max. Humidity	80.2	15.7	19.0	100.0
Max. Temperature (°C)	26.7	3.1	11.8	36.5
Min. Temperature (°C)	16.7	4.0	2.7	28.3
Rainfall (mm)	5.8	13.3	0.0	180.0

Table 6.2: Architecture of the model under study

Sl No.	Model Name	Model Architecture
1	PSO-ANN I	ANN(9)
2	PSO-ANN II	ANN(10) - ANN(10)
3	PSO-BiLSTM	BiLSTM(123)
4	PSO-BiLSTM-ANN I	BiLSTM(105) - ANN(116)
5	PSO-BiLSTM-ANN II	BiLSTM(17) - ANN(38) - Dropout(0.4)
6	PSO-BiLSTM-ANN III	BiLSTM(92) - BiLSTM(100) - ANN(13)
7	PSO-LSTM	LSTM(47)
8	PSO-LSTM-ANN I	LSTM(105) - ANN(116)
9	PSO-LSTM-ANN II	LSTM(46) - ANN(85) - Dropout(0.6)
10	PSO-LSTM-ANN III	LSTM(139) - LSTM(66) - ANN(132)
11	PSO-RNN-ANN	RNN(130) - ANN(137)
12	PSO-SVR	Linear Kernel, C=4.5, epsilon = 0.1

15.7. Additionally, the Maximum Temperature spans from a minimum of 11.8°C to a maximum of 36°C, with an average value of approximately 26.7°C and a standard deviation of 3.1°C. Lastly, the Minimum Temperature ranges from as low as 2.7°C to as high as 28.3 °C, with an average temperature of 16.7°C and a standard Deviation of 4°C.

Table 6.2 presents the hidden layer structure of the models studied in this research. In each model, the output layer is a single neuron ANN layer, except for the SVR model, which generates a single value. The value inside the parenthesis indicates the number of neurons in the ANN layer, the number of cells in the LSTM or RNN layer, and the Dropout rate.

The optimal hyperparameters returned by the PSO method are displayed in Table 6.3. In the initial configuration of the PSO method, both the LSTM cells and ANN

Table 6.3: Optimal hyperparameters of the models

Sl No.	Model Name	Epochs	Timesteps	Dropout Rate
1	PSO-ANN I	303	4	-
2	PSO-ANN II	51	4	-
3	PSO-BiLSTM	110	1	-
4	PSO-BiLSTM-ANN I	110	1	-
5	PSO-BiLSTM-ANN II	35	1	0.4
6	PSO-BiLSTM-ANN III	69	1	-
7	PSO-LSTM	53	1	-
8	PSO-LSTM-ANN I	100	1	-
9	PSO-LSTM-ANN II	66	4	0.6
10	PSO-LSTM-ANN III	58	1	-
11	PSO-RNN-ANN	79	4	-
12	PSO-SVR	1	1	-

neurons ranged from 1 to 150. The number of epochs spanned from 30 to 600, while the input timestep varied between 1 and 30, and the dropout rate ranged from 0 to 1. As most models, except SVR, possess stochastic characteristics, training, and testing were performed ten times. For the SVR model, C (regularization parameter) was set within a range of 1 to 10 and epsilon between a range of 0 and 1. The PSO is initialized using 10 particles, and the algorithm iterates 300 times. Particle acceleration constants c1 and c2 were initialized at values of 1.5 and 2.0, respectively. Thus, each BiLSTM, LSTM, RNN, and ANN model variant is trained and tested 3000 times, and the SVR model is trained and tested 300 times.

The PSO method provided in Table 6.3 reveals that the ideal number of previous timesteps for the LSTM and BiLSTM models is 1 day, while for the ANN and RNN models, it is 4 days. The epoch ranging from 35 to 303 shows that each model achieves better results within a smaller epoch. The best-performing model converges at 35 epochs. Additionally, it was discovered that the number of LSTM cells in the hidden LSTM layer of the BiLSTM model is comparatively smaller than in other models.

The performance of the models being compared, as measured by RMSE, MAE, and R^2, can be found in Table 6.4 and the models are listed in ascending order based on their RMSE values. Figure 6.6 displays a plot of the RMSE, and Figure 6.7 to

Table 6.4: Performance of the model under study

Sl No.	Model Name	RMSE	MAE	R^2
1	PSO-BiLSTM-ANN II	16.17	9.06	0.05
2	PSO-RNN-ANN	16.18	9.37	0.05
3	PSO-LSTM-ANN II	16.27	9.58	0.04
4	PSO-LSTM-ANN III	16.27	9.75	0.04
5	PSO-BiLSTM-ANN I	16.29	9.41	0.03
6	PSO-BiLSTM-ANN III	16.29	9.48	0.03
7	PSO-LSTM-ANN I	16.3	9.74	0.02
8	PSO-ANN I	16.33	8.87	0.03
9	PSO-LSTM	16.34	9.05	0.02
10	PSO-ANN II	16.35	8.81	0.03
11	PSO-BiLSTM	16.38	8.86	0.02
12	PSO-SVR	17.28	12.81	-0.1

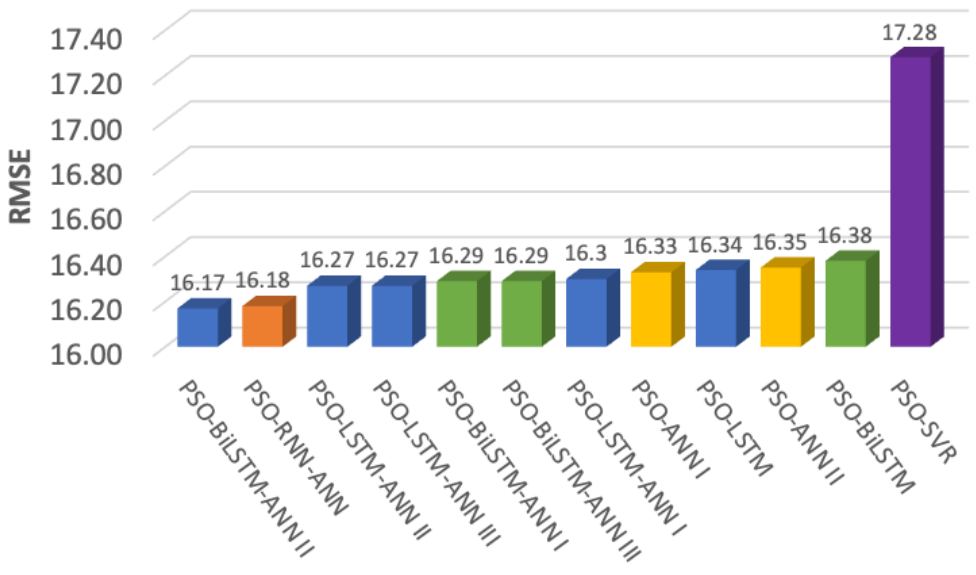

Figure 6.6: The plot of performance of the compared models in RMSE

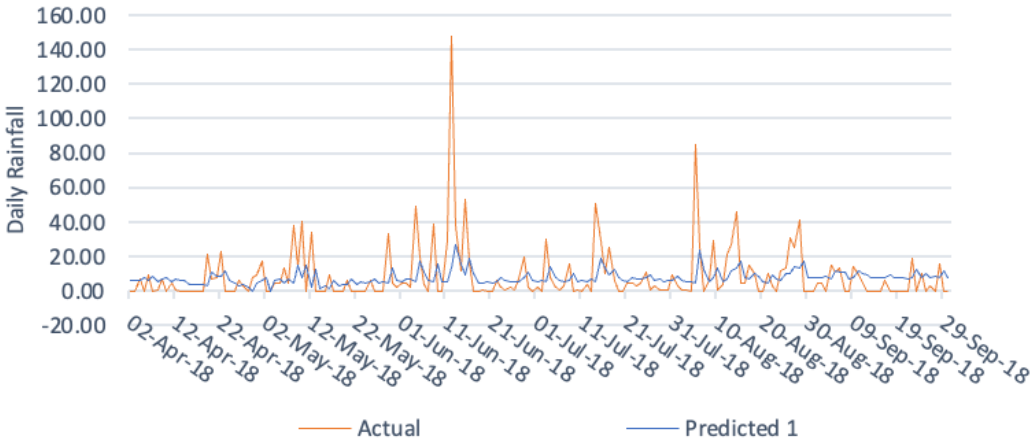

Figure 6.7: Prediction result of PSO-BiLSTM-ANN II

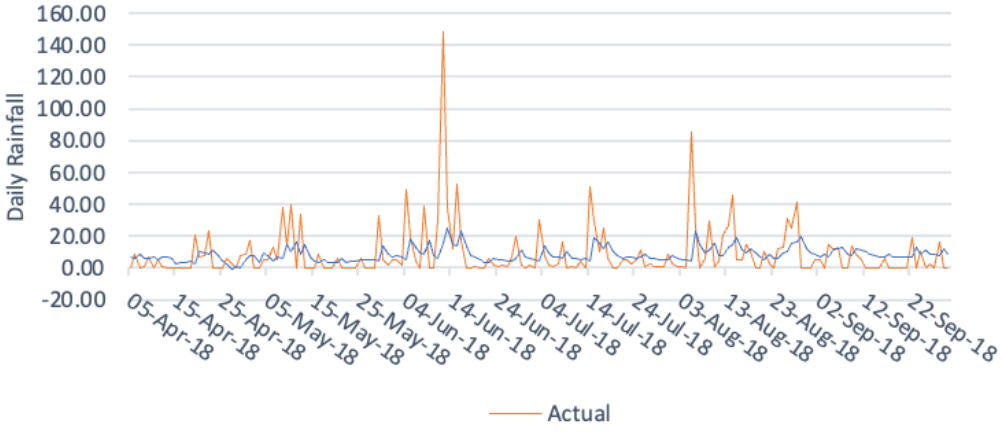

Figure 6.8: Prediction results of PSO-RNN-ANN

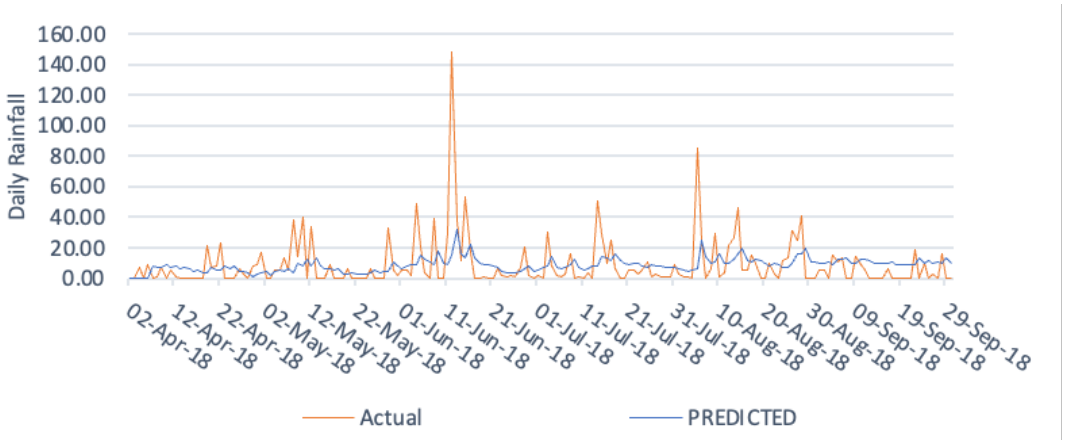

Figure 6.9: Prediction results of PSO-LSTM-ANN II

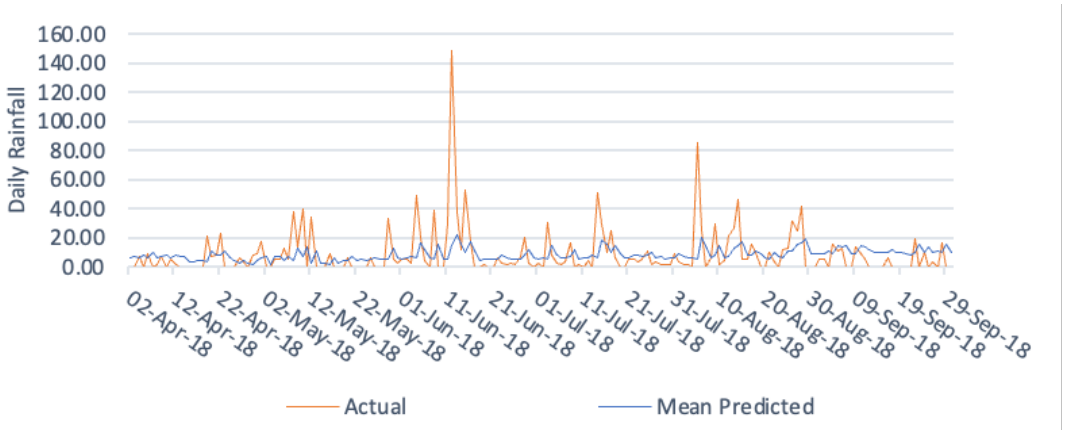

Figure 6.10: Prediction results of PSO-ANN I

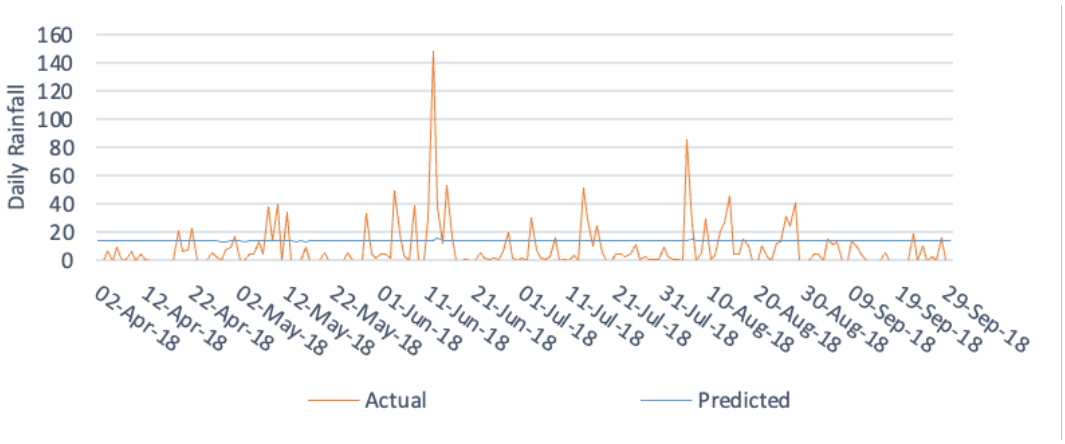

Figure 6.11: Prediction results of PSO-SVR

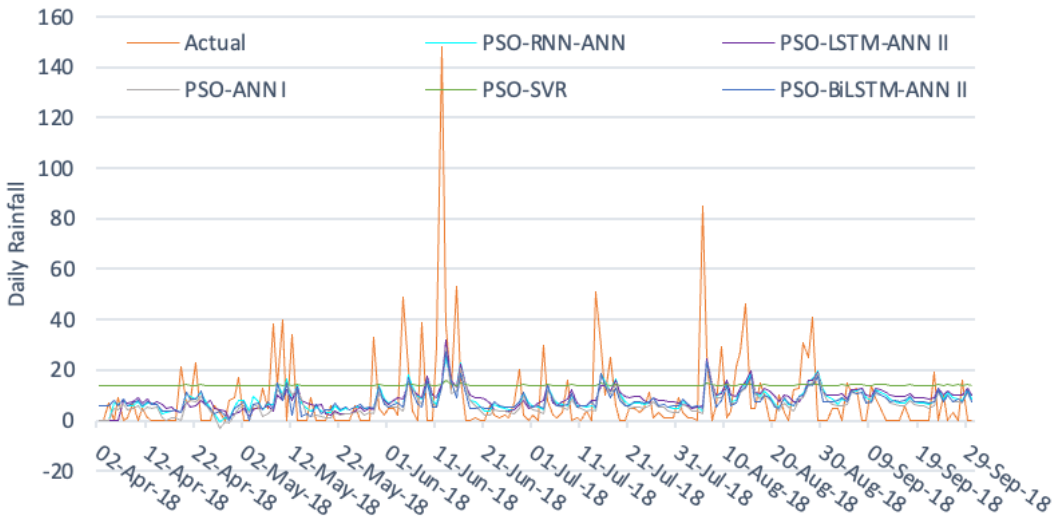

Figure 6.12: Comparison of the best model for each variant

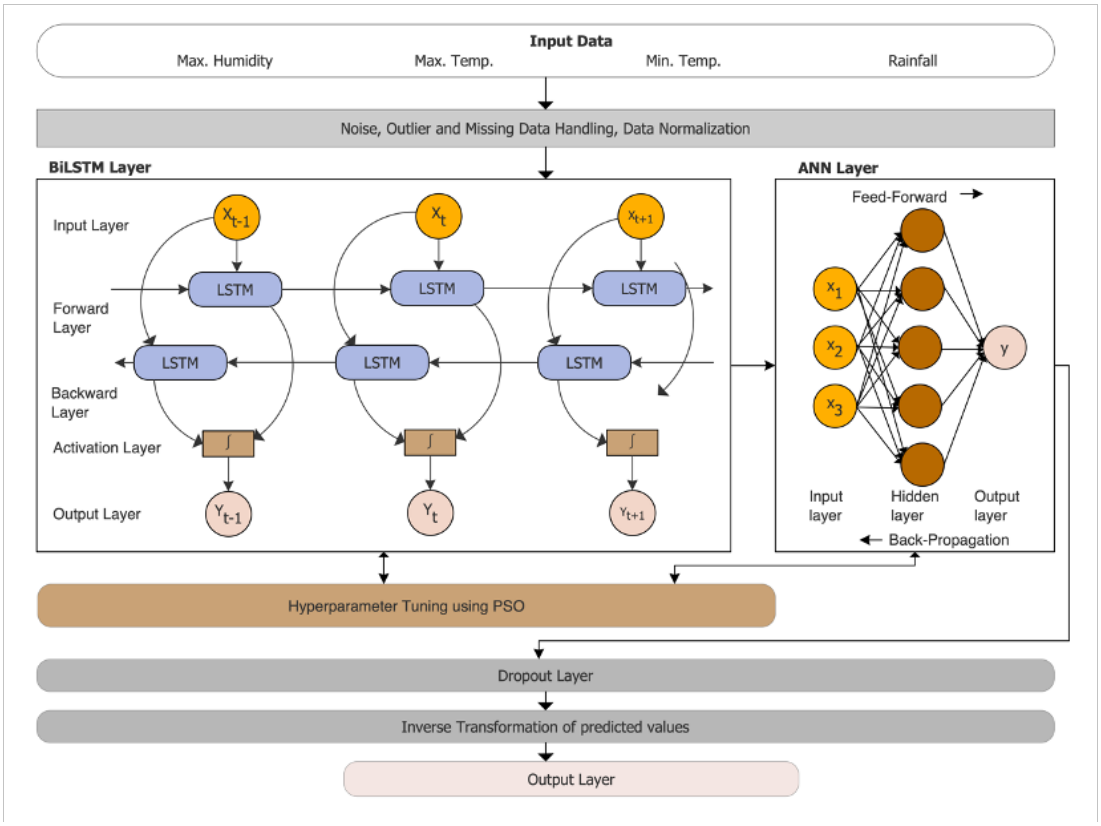

Figure 6.13: The structure of the PSO-BiLSTM-ANN II model

Figure 6.11 shows the actual and predicted rainfall for each variant's best model. Figure 6.12 illustrates a comparison between each variant's best-performing model predictions and the actual rainfall value. The structure of the PSO-BiLSTM-ANN II is also illustrated in Figure 6.13. After examining both the table and figures, it is evident that two models have the lowest RMSE with very similar performance: the PSO-BiLSTM-ANN II model with an RMSE of 16.17 and the PSO-RNN-ANN model with an RMSE of 16.18. Considering the performance across all compared models, it can be concluded that BiLSTM models are superior prediction models on average. On the other hand, it should be noted that the PSO-SVR model fails to accurately learn from patterns within the data set and demonstrates lower performance than other studied models. The hybrid PSO-BiLSTM-ANN model with a single BiLSTM and ANN hidden layer followed by a Dropout layer improved the RMSE result by 6.42%. Furthermore, introducing a dropout layer has shown improvement in overall model performance. Additionally, the hybrid models have contributed positively towards enhancing prediction accuracy. However, it is essential to acknowledge that none of these predictive models fit perfectly due to limitations such as insufficient weather parameter data collected by weather stations.

6.5 Conclusions

This study evaluated the predictive skills of twelve hybrid deep learning and machine learning models, namely BiLSTM, LSTM, RNN, ANN, and SVR, optimized using the PSO method. The models aim to predict the quantity of rainfall one day ahead. The meteorological data from the Aizawl Weather Station in Mizoram, India, is used to train, validate and test the models. The compared models converge at a few epochs, indicating the efficiency of the PSO model in avoiding overfitting. The hybrid models among the studied models are found to have better performance in comparison to the standalone models. Also, the BiLSTM models, on average, have better results, demonstrating that it is the most suitable model among the models under study. The

results also validate the feasibility of PSO-optimized hybrid deep learning models in predicting rainfall and demonstrate the advantage of the BiLSTM model in comparison to SVR, ANN, RNN, and LSTM. The hybrid PSO-BiLSTM-ANN model with a single BiLSTM and single ANN hidden layer followed by a Dropout layer improved the RMSE by 6.42%. In addition to these findings, this study provides a benchmark model for rainfall prediction using the Aizawl weather station dataset.

Chapter 7

Summary and Conclusions

This chapter summarizes the study of rainfall prediction in Mizoram using LSTM and PSO, presented in the previous chapters and provides a concluding remark.

Chapter 1 commences with an introduction to the context and rationale behind the study, accompanied by a statement of the problem and its objectives. This chapter also defines the scope and limitations of the study while highlighting its significance. Accurate rainfall prediction ensures sustainable resource management, disaster readiness, and climate adaptation strategies. By anticipating changes in rainfall patterns, decision-makers gain valuable insights that aid them in making informed decisions, allocating resources effectively, and mitigating the consequences of severe weather occurrences.

Accurate rainfall forecasting continues to be difficult, mainly because of the intricate nature of atmospheric processes and the dynamic interplay between different meteorological factors. The traditional regression-based approaches in this regard tend to oversimplify the underlying processes and fail to consider non-linear relationships. Additionally, these methods often struggle to handle missing data, outliers, and noisy datasets, resulting in less-than-optimal predictions. These limitations have sparked interest in utilizing advanced machine-learning techniques for rainfall prediction, particularly given the increasing availability of large-scale meteorological datasets. Many factors influence rainfall patterns, including temperature, humidity, wind patterns, geographical features, and global climate phenomena like El Ni˜no and La Ni˜na. These factors operate across various temporal scales, leading to intricate patterns that are difficult to capture using traditional methods

Traditional machine learning models have a limitation in capturing dependencies within data, as they typically consider only the current input during processing. Deep

learning models, such as RNN, address this issue by capturing dependencies within input data. However, RNNs face the vanishing gradient problem, limiting their ability to capture long-term dependencies. LSTM networks solve this problem by maintaining long-term dependencies. Despite their advantages, both RNN and LSTM transmit knowledge only in the forward direction. The BiLSTM model, introduced improves performance by capturing past and future hidden states through a bidirectional network, enabling the model to learn long-term dependencies more effectively.

The motivation behind investigating LSTM networks in rainfall prediction arises from their capacity to comprehend and anticipate rainfall variations through capturing temporal relationships and patterns. LSTM has proven successful in various fields, such as natural language processing, speech recognition, and financial prediction. Utilizing LSTM for rainfall prediction can enhance accuracy by modeling the complex dependencies inherent in meteorological time series data. LSTM networks are practical tools for capturing time-dependent patterns in data. However, their effectiveness is significantly impacted by hyperparameters such as the number of hidden units, hidden layers, and epochs. Determining the optimal combination of these hyperparameters through trial and error or grid search can be both costly and time-consuming. These methods can also get stuck in local optima and do not guarantee the discovery of the ideal hyperparameters.

PSO is an optimization technique known for efficiently searching optimal hyperparameters within a given parameter space. By combining LSTM with PSO, automatic fine-tuning of hyperparameters becomes feasible, improving predictive accuracy without manual adjustment requirements. Despite the success of PSO in optimizing proposed models, existing studies typically optimize only the proposed models while the hyperparameters of compared models are often determined through grid search, trial-and-error, or existing literature. This inconsistency indicates a gap where all models, both proposed and compared, should be uniformly optimized using PSO and tested on the same dataset, highlighting an area needing further investigation. Therefore, our objective is to examine if the PSO-optimized LSTM

model can effectively forecast the amount of rainfall by utilizing the meteorological dataset encompassing the all-India monthly averages and the Mizoram rainfall datasets.

Mizoram, situated in the north-eastern part of India, lies between the geographical coordinates of 21° 58' and 24° 35' N latitude and 92° 15' and 93° 29' E longitude. This state shares its borders with Tripura, Assam, and Manipur within India and a border spanning approximately 722 kilometres with the neighbouring countries of Bangladesh and Myanmar. Mizoram experiences significant yearly rainfall, averaging between 2,500 mm to 3,000 mm. Mizoram is directly influenced by the South-West Monsoon, which typically brings an adequate amount of rainfall. The rainy season, or summer monsoon, usually begins in April, with heavy rains occurring from May to September, and extends until late October. In contrast, the winter season, from November to February, is generally dry with minimal rainfall.

More than 70% of Mizoram's population relies on agriculture for their livelihood, with the majority practicing shifting cultivation. Given the state's topography and the significant rainfall it receives, Mizoram is highly vulnerable to abnormal climate variability and long-term climate changes. Climate-related hazards can severely impact all sectors in Mizoram, as its socio-economic conditions are relatively underdeveloped compared to other states in India. Both urban and rural populations heavily depend on agriculture and allied sectors, which are highly susceptible to climate variability and long-term changes. This vulnerability can lead to numerous issues, such as alterations in the timing of field preparation, sowing, harvesting, and overall yield. Additionally, climate-related hazards like landslides and flash floods, which are prevalent in Mizoram, can severely affect other developmental sectors. However, there has been no previous application of machine learning or deep learning models to analyze meteorological data specifically for Mizoram. This research aims to bridge the gap by exploiting the power of LSTM and the PSO optimization technique to improve rainfall prediction in Mizoram and contribute to the broader field of meteorology.

Chapter 2 presents an extensive examination of literature concerning rainfall prediction since 2015. The literature is categorized based on the specific datasets

utilized for training and testing the predictive models. These datasets encompass various weather parameters, radar images, and satellite images. This chapter thoroughly studies 45 papers from renowned publishers to gain insights into rainfall prediction. The classification of these papers focuses on the types of data implemented by the authors in their research. Furthermore, deep learning methods employed in these studies, the specific input data adopted for predicting rainfall, and metrics used to assess model performance are investigated extensively throughout this chapter. Additionally, both temporal and spatial distribution patterns of rainfall predictions are analyzed within this comprehensive exploration.

The research conducted in this chapter reveals a noticeable upward trend in the number of scientific papers focusing on rainfall prediction using deep learning techniques. Specifically, in 2019, there was a significant increase, representing 51% of all publications. This growth indicates a growing interest in applying deep learning methods for predicting rainfall patterns. It is worth noting that LSTM and ConvLSTM emerged as the most commonly employed methodologies. Authors often utilize AE and SAE to identify predictor variables when dealing with large input datasets. In cases where satellite or radar images are used as inputs, convolutional layers are frequently employed to extract relevant features from the data. Deep learning models have been found to outperform traditional machine learning models regarding accuracy in rainfall prediction tasks due to their ability to capture temporal and spatial information inherent within the input data. Geographically, China leads in researching this subject, followed by India and the USA. Amongst the software tools utilized by researchers, MATLAB takes precedence as the most widely used option, followed closely by Tensorflow and Keras. Regarding evaluation metrics for assessing performance, RMSE (Root Mean Square Error) is widely adopted among researchers studying rainfall prediction using deep learning methods; however, MAE, CSI, and FAR are also used regularly within published works.

To summarize, the findings of this study indicate that deep learning techniques outperformed traditional machine learning models and shallow neural network

architecture in the area of rainfall prediction. Therefore, deep learning methods are considered more favourable for this task.

Chapter 3 analyzes four LSTM models for predicting the monthly average rainfall in India from 1871 to 2016. The first model, Model_1, consists of a single hidden layer with fifty cells in an LSTM architecture. It also includes a Dense output layer with just one neuron. Model_2, conversely, comprises a hidden LSTM layer and a hidden Dense layer. The hidden LSTM layer comprises ten cells, while the hidden Dense layer comprises ten neurons. Model_2 has a Dense output layer with one neuron, similar to the previous models. Model_3 has two stacked LSTM layers and a dense output layer. These stacked LSTM layers contain ten cells in each layer, whereas the dense output only possesses one neuron. Lastly, Model_4 is constructed using two LSTMs as its hidden layers - each consisting of twelve cells, one Hidden Dense Layer featuring twelve neurons, and an output Dense layer with a single neuron.

The grid search method is used to evaluate the number of input timesteps, LSTM cells, and epochs required for each model. The models were compiled using the Adam optimization algorithm, and MSE was used as the loss function. To account for variability in results, we repeated each model ten times using the same training and evaluation dataset. The average of these ten outputs was then calculated to determine the final performance of each model. The number of input time steps is searched in the range of 1 to 30, the number of LSTM cells in the range of 1 to 100, and the number of epochs in the range of 50 to 1000. The evaluation of the LSTM models' performance was carried out by utilizing statistical metrics such as MAE and RMSE. These parameters serve as indicators of the accuracy of the models, with lower values representing superior performance. To determine the final MAE and RMSE values for a particular model, the average of ten MAE and ten RMSE outputs was computed.

The LSTM Model_4 outperformed the other models, achieving the lowest RMSE value of 245.30. This indicates that stacked LSTM models have a strong ability to forecast rainfall patterns in India accurately. Additionally, increasing the timesteps further improved the performance of these models. To assess the effectiveness of our

LSTM models, we compared them to RNN and LSTM models from a previous study by Kumar *et al.* (2019). In this study, all the proposed LSTM models outperformed the LSTM and RNN models proposed by Kumar *et al.* (2019) in terms of RMSE. This shows that incorporating more neurons and additional timesteps can significantly enhance rainfall prediction accuracy.

In Chapter 4, the predictive performance of the Bidirectional LSTM model is evaluated with the Vanilla LSTM, Stacked LSTM, and a benchmark model found in the existing literature. The main goal is to estimate the average rainfall for India one month ahead by utilizing the previous month's rainfall as a predictor. Furthermore, this study seeks to determine the optimal number of prior months' rainfall, epochs, and cells that produce the lowest RMSE when forecasting one-month rainfall into the future. To assess its effectiveness, our proposed model's performance is compared to Vanilla LSTM, Stacked LSTM, and both RNN and LSTM models proposed by Kumar *et al.* (2019).

A grid search method is employed to determine the parameters for the model (including lead time, number of epochs, and number of LSTM cells). In this research, the MSE was used as the loss function, while the Adam optimizer was utilized to reduce the loss function. The initial weights of the models were randomly selected for each execution. To deal with the algorithm's stochastic nature, we trained and tested each LSTM model 10 times using identical training and testing datasets. The final performance of the model was determined by averaging these ten results. The performance of the Bidirectional LSTM model was analyzed using an epoch ranging from 50 to 500.

It was observed that the RMSE of the model gradually decreased with an increase in the number of epochs until it reached a minimum RMSE at 150 epochs. However, beyond this point, the RMSE started to increase again. This result suggests that the model may have been overfitting the data, causing a decline in its performance. As a result, an epoch of 150 was chosen for training the Bidirectional LSTM model. Increasing the number of cells, the RMSE of the model decreased to 80 cells. This

result implies that increasing the complexity and capacity of the model led to improved accuracy and reduced prediction errors. However, it is essential to note that beyond this point, when the number of cells exceeded 80, there was a noticeable increase in RMSE.

Out of all the models analyzed, the Bidirectional LSTM achieved an RMSE value of 240.79, the lowest among them. The Stacked LSTM followed with an RMSE value of 245.3, while the Vanilla LSTM had a slightly higher RMSE value of 250.47. In comparison, the benchmark RNN and LSTM models performed less effectively with RMSE values of 261.7 and 251.63, respectively. The findings indicated that achieving optimal prediction accuracy requires considering various factors. These factors include increasing the number of input timesteps, adding more LSTM cells, and extending the training epochs while closely monitoring for overfitting. It is crucial to strike a balance between the size of the input data, the number of LSTM cells in each layer, and how extensively the training dataset is exposed to the LSTM model. Additionally, employing stacked LSTM layers and utilizing a Bidirectional LSTM approach significantly improved model performance compared to single-cell LSTM or Vanilla LSTM models.

Chapter 5 of the study compared various PSO-optimized deep learning and machine learning models. These models included PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN, and PSO-SVR. The comparison was conducted using the Aizawl monthly rainfall dataset and the all-India monthly average rainfall dataset. This research is significant as it marks the first time machine learning and deep learning models have been employed to analyze rainfall data of Mizoram. Furthermore, this study also presents the first instance where the predictive capabilities of the models mentioned above (PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN) were compared in terms of their effectiveness in predicting rainfall patterns.

To ensure optimal performance of these models, parameters such as the number of neurons/cells, number of epochs, and number of previous timesteps for input are optimized using the PSO method for BiLSTM, LSTM, RNN, and ANN models. For

the PSO-SVR model, optimization of hyperparameters involved tuning the number of previous timesteps along with regularization parameter C and tolerance margin epsilon. Several metrics, such as RMSE, MAE, and Coefficient of determination (R^2), are used to assess the performance of these compared models. The chosen loss function that needs to be minimized by these models is MSE. Furthermore, the Adam optimizer is employed to achieve this minimization objective.

In the Aizawl monthly rainfall dataset, all variants of PSO-BiLSTM models outperformed the other compared models. Among the studied models, PSO-SVR with a polynomial kernel showed the lowest performance. The best model was the PSO-BiLSTM II model, which consists of a BiLSTM Layer comprising 3 cells, followed by a Dense layer composed of 103 neurons and an output Dense layer containing 1 neuron. Stacking the BiLSTM, LSTM, RNN, and ANN models did not improve this dataset's prediction accuracy. However, incorporating an additional Dense layer before the output Dense layer increased prediction accuracy for both BiLSTM, LSTM, and RNN models. It was also observed that each model required different optimal input timesteps when utilizing the PSO algorithm. This indicates that determining optimal input timesteps depends on the specific dataset and the type of predictive models employed. Additionally, it was discovered that compared to other examined models in this study, the BiLSTM models demonstrated faster convergence toward a solution by requiring fewer epochs.

According to the all-India monthly average rainfall dataset, the PSO-BiLSTM models consistently achieved the lowest RMSE score. Similar results were observed in the Aizawl monthly rainfall dataset, where BiLSTM, LSTM, and RNN stacked with a Dense layer exhibited better prediction performance compared to models without a Dense layer. The PSO-SVR model performed poorly among all the compared models in this dataset. Additionally, all PSO-optimized models outperformed those that utilized hyperparameters determined through the grid search methods or benchmarked from existing literature. This demonstrates that the Particle Swarm Optimization (PSO) algorithm is reliable for identifying optimal hyperparameters. However, due

to its stochastic nature, relying on only one model prediction from PSO could lead to incorrect conclusions about its superiority over other models. To mitigate this issue and ensure accurate evaluation, this study repeated both model training and testing processes ten times. If computational resources allow for it, repeating these iterations more than ten times is recommended.

Chapter 6 presents an exhaustive comparison of twelve hybrid deep learning and machine learning models optimized using PSO. These models include BiLSTM, LSTM, RNN, ANN, and SVR. This study's main objective is to predict the amount of rainfall expected for the next day. To achieve this, meteorological data from 1985 to 2018, collected by the Aizawl Weather Station in Mizoram, India, is utilized. This dataset comprises 12,418 days of weather parameters such as maximum humidity, minimum temperature, maximum temperature, and previous rainfall amounts to forecast future precipitation patterns accurately.

Several parameters are considered to optimize the performance of these models using particle swarm optimization (PSO). For the ANN, optimization includes determining the optimal number of neurons. For the LSTM and RNN layers, optimization involves finding the optimal number of cells and other factors like the number of epochs and dropout rate. For the SVR model, the optimal number of previous timesteps for input and the values for regularization parameter C and margin of tolerance, epsilon, are optimized. To evaluate how well each model performs compared to other models under consideration, metrics like RMSE, MAE, and R^2 are used. The models under study aim to minimize the MSE loss function utilizing the Adam optimizer.

The ranges of boundary conditions of the PSO method are- the LSTM cells and ANN neurons ranging from 1 to 150. The number of epochs spanned from 30 to 600, while the input timestep varied between 1 and 30, and the dropout rate ranged from 0 to 1. As most models, except SVR, possess stochastic characteristics, training, and testing were performed ten times. For the SVR model, C (regularization parameter) was set within a range of 1 to 10 and epsilon between a range of 0 and 1. The PSO is

initialized using 10 particles, and the algorithm iterates 300 times. Particle acceleration constants c1 and c2 were initialized at values of 1.5 and 2.0, respectively. Thus, each BiLSTM, LSTM, RNN, and ANN model variant is trained and tested 3000 times, and the SVR model is trained and tested 300 times.

The PSO-BiLSTM-ANN II model with an RMSE of 16.17 and the PSO-RNN-ANN model with an RMSE of 16.18 has the best performance among the compared models. Considering the performance across all compared models, it can be found that BiLSTM models are superior prediction models on average. On the other hand, it should be noted that the PSO-SVR model fails to accurately learn from patterns within the data set and demonstrates lower performance than other studied models. The hybrid PSO-BiLSTM-ANN model with a single BiLSTM and ANN hidden layer followed by a Dropout layer improved the RMSE result by 6.42%. Furthermore, introducing a dropout layer has shown improvement in overall model performance. Additionally, the hybrid models have contributed positively towards enhancing prediction accuracy. However, it is essential to acknowledge that none of these predictive models fit perfectly due to limitations such as insufficient weather parameter data.

The studied models showed convergence at a remarkably low number of epochs, which indicates that the PSO model is efficient in avoiding overfitting the model. Hybrid models demonstrated superior predictive abilities compared to the standalone models. Also, the BiLSTM models generally outperformed the compared model, proving that it is the most suitable model among the models under study. These experimental results demonstrate the viability of PSO-optimized hybrid deep learning models in rainfall prediction and show the advantage of the BiLSTM model over other deep learning models. This research provides a foundational benchmark for future studies involving the Aizawl weather station data, and future work could be done to include more data from other weather stations.

Bibliography

Abayomi-Alli, O. O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., and Misra, S. (2022). "Data augmentation and deep learning methods in sound classification: A systematic review". *Electronics*, **11(22)**, 3795.

Abbas, A. M. (2021). "Social network analysis using deep learning: Applications and schemes". *Social Network Analysis and Mining*, **11(1)**, 106.

Abbot, J., and Marohasy, J. (2012). "Application of artificial neural networks to rainfall forecasting in queensland, australia". *Advances in Atmospheric Sciences*, **29**, 717–730.

Abbot, J., and Marohasy, J. (2014). "Input selection and optimisation for monthly rainfall forecasting in queensland, australia, using artificial neural networks". *Atmospheric Research*, **138**, 166–178.

Abbot, J., and Marohasy, J. (2017). "Skilful rainfall forecasts from artificial neural networks with long duration series and single-month optimization". *Atmospheric Research*, **197**, 289–299.

Abdel-Nasser, M., and Mahmoud, K. (2019). "Accurate photovoltaic power forecasting models using deep lstm-rnn". *Neural computing and applications*, **31**, 2727–2740.

Affonso, C., Rossi, A. L. D., Vieira, F. H. A., de Leon Ferreira, A. C. P., et al. (2017). "Deep learning for biological image classification". *Expert systems with applications*, **85**, 114–122.

Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., and Hickey, J. (2019). "Machine learning for precipitation nowcasting from radar images". *arXiv preprint arXiv:1912.12132*.

Ahmed, A., Bader, M., Shahin, I., Nassif, A. B., Werghi, N., and Basel, M. (2023). "Arabic mispronunciation recognition system using lstm network". *Information*, **14(7)**, 413.

Akbari Asanjan, A., Yang, T., Hsu, K., Sorooshian, S., Lin, J., and 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.

Al Hwaitat, A. K., Al-Sayyed, R. M., Salah, I. K., Manaseer, S., Al-Bdour, H. S., and Shukri, S. E. (2022). "Frequencies wave sound particle swarm optimisation (fpso)". *Journal of Experimental and Theoretical Artificial Intelligence*, **34(5)**, 749–780.

Aslan, M. F., Durdu, A., and Sabanci, K. (2023). "Goal distance-based uav path planning approach, path optimization and learning-based path estimation: Gdrrt*, pso-gdrrt* and bilstm-pso-gdrrt". *Applied Soft Computing*, **137**, 110156.

Awad, M., and Khanna, R. (2015). *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*. Apress Berkeley, CA.

Ayzel, G., Scheffer, T., and Heistermann, M. (2020). "Rainnet v1. 0: A convolutional neural network for radar-based precipitation nowcasting". *Geoscientific Model Development*, **13(6)**, 2631–2644.

Balaji, T., Annavarapu, C. S. R., and Bablani, A. (2021). "Machine learning algorithms for social media analysis: A survey". *Computer Science Review*, **40**, 100395.

Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., and Akanbi, L. A. (2022). "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting". *Machine Learning with Applications*, **7**, 100204.

Bathla, G., Rani, R., and Aggarwal, H. (2023). "Stocks of year 2020: Prediction of high variations in stock prices using lstm". *Multimedia Tools and Applications*, **82(7)**, 9727–9743.

Bhaskar, S., and Thasleema, T. (2023). "Lstm model for visual speech recognition through facial expressions". *Multimedia Tools and Applications*, **82(4)**, 5455–5472.

Bouktif, S., Fiaz, A., Ouni, A., and Serhani, M. A. (2018). “Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches”. *Energies*, **11(7)**, 1636.

Bromberg, C. L., Gazen, C., Hickey, J. J., Burge, J., Barrington, L., and Agrawal, S. (2019). “Machine learning for precipitation nowcasting from radar images”. *Proceedings of the Machine Learning and the Physical Sciences Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada*, 1–4.

Broni-Bedaiko, C., Katsriku, F. A., Unemi, T., Atsumi, M., Abdulai, J.-D., Shinomiya, N., and Owusu, E. (2019). “El niño-southern oscillation forecasting using complex networks analysis of lstm neural networks”. *Artificial Life and Robotics*, **24**, 445–451.

Buyukyildiz, M., and Tezel, G. (2017). “Utilization of pso algorithm in estimation of water level change of lake beysehir”. *Theoretical and Applied Climatology*, **128**, 181–191.

Chakraverty, S., and Gupta, P. (2008). “Comparison of neural network configurations in the long-range forecast of southwest monsoon rainfall over india”. *Neural Computing and Applications*, **17**, 187–192.

Chattopadhyay, S., and Chattopadhyay, G. (2010). “Univariate modelling of summer-monsoon rainfall time series: Comparison between arima and arnn”. *Comptes Rendus Geoscience*, **342(2)**, 100–107.

Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., and Liu, Y. (2021). “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities”. *ACM Computing Surveys (CSUR)*, **54(4)**, 1–40.

Chen, L., Cao, Y., Ma, L., and Zhang, J. (2020). “A deep learning-based methodology for precipitation nowcasting with radar”. *Earth and Space Science*, **7(2)**, e2019EA000812.

Choo, S., and Kim, W. (2023). “A study on the evaluation of tokenizer performance in natural language processing”. *Applied Artificial Intelligence*, **37(1)**, 2175112.

Daouad, M., Allah, F. A., and Dadi, E. W. (2023). “An automatic speech recognition system for isolated amazigh word using 1d and 2d cnn-lstm architecture”. *International Journal of Speech Technology*, **26(3)**, 775–787.

Dash, Y., Mishra, S. K., and Panigrahi, B. K. (2018). “Rainfall prediction for the kerala state of india using artificial intelligence approaches”. *Computers and Electrical Engineering*, **70**, 66–73.

Dimitrovski, I., Kitanovski, I., Kocev, D., and Simidjievski, N. (2023). “Current trends in deep learning for earth observation: An open-source benchmark arena for image classification”. *ISPRS Journal of Photogrammetry and Remote Sensing*, **197**, 18–35.

Dorbe, N., Jaundalders, A., Kadikis, R., and Nesenbergs, K. (2018). “Fcn and lstm based computer vision system for recognition of vehicle type, license plate number, and registration country”. *Automatic Control and Computer Sciences*, **52**, 146–154.

Dotse, S.-Q., Larbi, I., Limantol, A. M., and De Silva, L. C. (2024). “A review of the application of hybrid machine learning models to improve rainfall prediction”. *Modeling Earth Systems and Environment*, **10(1)**, 19–44.

Du, B., Huang, S., Guo, J., Tang, H., Wang, L., and Zhou, S. (2022). “Interval forecasting for urban water demand using pso optimized kde distribution and lstm neural networks”. *Applied Soft Computing*, **122**, 108875.

Du, J., Liu, Y., and Liu, Z. (2018). “Study of precipitation forecast based on deep belief networks”. *Algorithms*, **11(9)**, 132.

Duong, T. A., Bui, M. D., and Rutschmann, P. (2018). A comparative study of three different models to predict monthly rainfall in ca mau, vietnam. In G. Zenz (Ed.), *Wasserbau-symposium graz 2018. wasserwirtschaft – innovation aus tradition. tagungsband. beiträge zum 19. gemeinschafts-symposium der*

wasserbau-institute tu münchen, tu graz und eth zürich (pp. 425–432). Verlag der Technischen Universität Graz.

Fang, Z., Ma, X., Pan, H., Yang, G., and Arce, G. R. (2023). "Movement forecasting of financial time series based on adaptive lstm-bn network". *Expert Systems with Applications*, **213**, 119207.

Feng, S., Wang, Y., Liu, L., Wang, D., and Yu, G. (2019). "Attention based hierarchical lstm network for context-aware microblog sentiment classification". *World Wide Web*, **22**, 59–81.

French, M. N., Krajewski, W. F., and Cuykendall, R. R. (1992). "Rainfall forecasting in space and time using a neural network". *Journal of hydrology*, **137(1-4)**, 1–31.

Geng, Y. (2023). "Design of english teaching speech recognition system based on lstm network and feature extraction". *Soft Computing*, **27(17)**, 1–11.

Gope, S., Sarkar, S., Mitra, P., and Ghosh, S. (2016). "Early prediction of extreme rainfall events: A deep learning approach". *Advances in Data Mining. Applications and Theoretical Aspects: 16th Industrial Conference, ICDM 2016, New York, NY, USA, July 13-17, 2016. Proceedings 16*, 154–167.

Graves, A., and Schmidhuber, J. (2005). "Framewise phoneme classification with bidirectional lstm and other neural network architectures". *Neural networks*, **18(5-6)**, 602–610.

Gu, F., Chung, M.-H., Chignell, M., Valaee, S., Zhou, B., and Liu, X. (2021). "A survey on deep learning for human activity recognition". *ACM Computing Surveys (CSUR)*, **54(8)**, 1–34.

Hafeez, G., Alimgeer, K. S., and Khan, I. (2020). "Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid". *Applied Energy*, **269**, 114915.

Han, C., and Fu, X. (2023). "Challenge and opportunity: Deep learning-based stock price prediction by using bi-directional lstm model". *Frontiers in Business, Economics and Management*, **8(2)**, 51–54.

Haq, D. Z., Novitasari, D. C. R., Hamid, A., Ulinnuha, N., Farida, Y., Nugraheni, R. D., Nariswari, R., Rohayani, H., Pramulya, R., Widjayanto, A., et al. (2021). "Long short-term memory algorithm for rainfall prediction based on el-nino and iod data". *Procedia Computer Science*, **179**, 829–837.

Hassan, M. M., Rony, M. A. T., Khan, M. A. R., Hassan, M. M., Yasmin, F., Nag, A., Zarin, T. H., Bairagi, A. K., Alshathri, S., and El-Shafai, W. (2023). "Machine learning-based rainfall prediction: Unveiling insights and forecasting for improved preparedness". *IEEE Access*, **11**, 132196–132222.

He, Q.-Q., Wu, C., and Si, Y.-W. (2022). "Lstm with particle swam optimization for sales forecasting". *Electronic Commerce Research and Applications*, **51**, 101118.

He, Y.-L., Chen, L., Gao, Y., Ma, J.-H., Xu, Y., and Zhu, Q.-X. (2022). "Novel double-layer bidirectional lstm network with improved attention mechanism for predicting energy consumption". *ISA transactions*, **127**, 350–360.

Hernández, E., Sanchez-Anguix, V., Julian, V., Palanca, J., and Duque, N. (2016). "Rainfall prediction: A deep learning approach". *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings 11*, 151–162.

Hewage, P., Trovati, M., Pereira, E., and Behera, A. (2021). "Deep learning-based effective fine-grained weather forecasting model". *Pattern Analysis and Applications*, **24(1)**, 343–366.

Heye, A., Venkatesan, K., and Cain, J. (2017). "Precipitation nowcasting: Leveraging deep recurrent convolutional neural networks". *Proceedings of the Cray User Group (CUG)*, **2017**.

Hochreiter, S., and Schmidhuber, J. (1997). "Long short-term memory". *Neural computation*, **9(8)**, 1735–1780.

İnik, Ö. (2023). "Cnn hyper-parameter optimization for environmental sound classification". *Applied Acoustics*, **202**, 109168.

Jaouedi, N., Boujnah, N., and Bouhlel, M. S. (2020). “A new hybrid deep learning model for human action recognition”. *Journal of King Saud University-Computer and Information Sciences*, **32(4)**, 447–453.

Jeong, C.-S., Ryu, K.-H., Lee, J.-Y., and Jung, K.-D. (2020). “Deep learning-based tourism recommendation system using social network analysis”. *International Journal of Internet, Broadcasting and Communication*, **12(2)**, 113–119.

Ji, X., Yang, B., and Tang, Q. (2020). “Acoustic seabed classification based on multibeam echosounder backscatter data using the pso-bp-adaboost algorithm: A case study from jiaozhou bay, china”. *IEEE Journal of Oceanic Engineering*, **46(2)**, 509–519.

Jin, Z., Yang, Y., and Liu, Y. (2020). “Stock closing price prediction based on sentiment analysis and lstm”. *Neural Computing and Applications*, **32**, 9713–9729.

Kennedy, J., and Eberhart, R. (1995). “Particle swarm optimization”. *Proceedings of ICNN’95-international conference on neural networks*, **4**, 1942–1948.

Khamparia, A., Gupta, D., Nguyen, N. G., Khanna, A., Pandey, B., and Tiwari, P. (2019). “Sound classification using convolutional neural network and tensor deep stacking network”. *IEEE Access*, **7**, 7717–7727.

Khurana, D., Koli, A., Khatter, K., and Singh, S. (2023). “Natural language processing: State of the art, current trends and challenges”. *Multimedia tools and applications*, **82(3)**, 3713–3744.

Kidd, C., Levizzani, V., Turk, J., and Ferraro, R. (2009). “Satellite precipitation measurements for water resource monitoring 1”. *JAWRA Journal of the American Water Resources Association*, **45(3)**, 567–579.

Kim, S., Hong, S., Joh, M., and Song, S.-k. (2017). “Deeprain: Convlstm network for precipitation prediction using multichannel radar data”. *arXiv preprint arXiv:1711.02316*.

Kim, T.-Y., and Cho, S.-B. (2021). “Optimizing cnn-lstm neural networks with pso for anomalous query access control”. *Neurocomputing*, **456**, 666–677.

Kothawale, D., and Rajeevan, M. (2017). *Monthly, seasonal and annual rainfall time series for all-india, homogeneous regions and meteorological subdivisions: 1871-2016* (tech. rep. No. RR-138). Indian Institute of Tropical Meteorology.

Krishnapriya, S., and Karuna, Y. (2023). "Pre-trained deep learning models for brain mri image classification". *Frontiers in Human Neuroscience*, **17**, 1150120.

Kumar, A., Islam, T., Sekimoto, Y., Mattmann, C., and Wilson, B. (2020). "Convcast: An embedded convolutional lstm based architecture for precipitation nowcasting using satellite data". *Plos one*, **15(3)**, e0230114.

Kumar, D., Singh, A., Samui, P., and Jha, R. K. (2019). "Forecasting monthly precipitation using sequential modelling". *Hydrological sciences journal*, **64(6)**, 690–700.

Lallianthanga, R., Lalmalsawma, S., Hrahsel, L., and Lalthanpuia. *Climate profile of mizoram: A tabular and graphical representation of meteorological data of eight district of mizoram*. Mizoram State Climate Change Cell, Mizoram Science, Technology and Innovation Council (MISTIC), Directorate of Science and Technology, Government of Mizoram. 2018.

Latif, S. D., Hazrin, N. A. B., Koo, C. H., Ng, J. L., Chaplot, B., Huang, Y. F., El-Shafie, A., and Ahmed, A. N. (2023). "Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches". *Alexandria Engineering Journal*, **82**, 16–25.

Li, K., Ma, Z., Robinson, D., Lin, W., and Li, Z. (2020). "A data-driven strategy to forecast next-day electricity usage and peak electricity demand of a building portfolio using cluster analysis, cubist regression models and particle swarm optimization". *Journal of Cleaner Production*, **273**, 123115.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., and Benediktsson, J. A. (2019). "Deep learning for hyperspectral image classification: An overview". *IEEE Transactions on Geoscience and Remote Sensing*, **57(9)**, 6690–6709.

Li, Z., Yu, H., Xu, J., Liu, J., and Mo, Y. (2023). “Stock market analysis and prediction using lstm: A case study on technology stocks”. *Innovations in Applied Engineering and Technology*, **2(1)**, 1–6.

Liu, Z., and Yeh, W.-C. (2024). “Simplified swarm optimisation for cnn hyperparameters: A sound classification approach”. *International Journal of Web and Grid Services*, **20(1)**, 93–113.

Long, T., Wang, S., Cao, W., Zhou, H., and Fernandez, C. (2023). “An improved variable forgetting factor recursive least square-double extend kalman filtering based on global mean particle swarm optimization algorithm for collaborative state of energy and state of health estimation of lithium-ion batteries”. *Electrochimica acta*, **450**, 142270.

Ma, T., Wang, C., Wang, J., Cheng, J., and Chen, X. (2019). “Particle-swarm optimization of ensemble neural networks with negative correlation learning for forecasting short-term wind speed of wind farms in western china”. *Information Sciences*, **505**, 157–182.

Machová, K., Mikula, M., Gao, X., and Mach, M. (2020). “Lexicon-based sentiment analysis using the particle swarm optimization”. *Electronics*, **9(8)**, 1317.

Mahadevaswamy, U., and Swathi, P. (2023). “Sentiment analysis using bidirectional lstm network”. *Procedia Computer Science*, **218**, 45–56.

Mandal, S., Singh, G. K., and Pal, A. (2019). “Pso-based text summarization approach using sentiment analysis”. *Computing, Communication and Signal Processing: Proceedings of ICCASP 2018*, 845–854.

Manokij, F., Sarinnapakorn, K., and Vateekul, P. (2019). “Forecasting thailand’s precipitation with cascading model of cnn and gru”. *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1–6.

Markuna, S., Kumar, P., Ali, R., Vishwkarma, D. K., Kushwaha, K. S., Kumar, R., Singh, V. K., Chaudhary, S., and Kuriqi, A. (2023). “Application of innovative

machine learning techniques for long-term rainfall prediction". *Pure and Applied Geophysics*, **180(1)**, 335–363.

Mekanik, F., Imteaz, M., Gato-Trinidad, S., and Elmahdi, A. (2013). "Multiple regression and artificial neural network for long-term rainfall forecasting using large scale climate modes". *Journal of Hydrology*, **503**, 11–21.

Meyer, H., Kühnlein, M., Appelhans, T., and Nauss, T. (2016). "Comparison of four machine learning algorithms for their applicability in satellite-based optical rainfall retrievals". *Atmospheric research*, **169**, 424–433.

Miao, Q., Pan, B., Wang, H., Hsu, K., and Sorooshian, S. (2019). "Improving monsoon precipitation prediction using combined convolutional and long short term memory neural network". *Water*, **11(5)**, 977.

Mithoo, P., and Kumar, M. (2023). "Social network analysis for crime rate detection using spizella swarm optimization based bilstm classifier". *Knowledge-Based Systems*, **269**, 110450.

Mozaffari, S., Javadi, S., Moghaddam, H. K., and Randhir, T. O. (2022). "Forecasting groundwater levels using a hybrid of support vector regression and particle swarm optimization". *Water Resources Management*, **36(6)**, 1955–1972.

Ni, L., Wang, D., Singh, V. P., Wu, J., Wang, Y., Tao, Y., and Zhang, J. (2020). "Streamflow and rainfall forecasting by two long short-term memory-based models". *Journal of Hydrology*, **583**, 124296.

Ojo, O. S., and Ogunjo, S. T. (2022). "Machine learning models for prediction of rainfall over nigeria". *Scientific African*, **16**, e01246.

Olah, C. (2015). *Understanding lstm networks*. Retrieved August 27, 2015, from https: //colah.github.io/posts/2015-08-Understanding-LSTMs/

Oswalt Manoj, S., and Ananth, J. (2020). "Mapreduce and optimized deep network for rainfall prediction in agriculture". *The Computer Journal*, **63(6)**, 900–912.

Ouyang, Q., and Lu, W. (2018). "Monthly rainfall forecasting using echo state networks coupled with data preprocessing methods". *Water resources management*, **32**, 659–674.

Pal, L., Ojha, C. S. P., and Dimri, A. (2021). “Characterizing rainfall occurrence in india: Natural variability and recent trends”. *Journal of Hydrology*, **603**, 126979.

Pham, B. T., Le, L. M., Le, T.-T., Bui, K.-T. T., Le, V. M., Ly, H.-B., and Prakash, I. (2020). “Development of advanced artificial intelligence models for daily rainfall prediction”. *Atmospheric Research*, **237**, 104845.

Piczak, K. J. (2015). “Environmental sound classification with convolutional neural networks”. *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*, 1–6.

Poornima, S., and Pushpalatha, M. (2019). “Prediction of rainfall using intensified lstm based recurrent neural network with weighted linear units”. *Atmosphere*, **10(11)**, 668.

Primartha, R., Tama, B. A., Arliansyah, A., and Miraswan, K. J. (2019). “Decision tree combined with pso-based feature selection for sentiment analysis”. *Journal of Physics: Conference Series*, **1196(1)**, 012018.

Pudashine, J., Guyot, A., Petitjean, F., Pauwels, V. R., Uijlenhoet, R., Seed, A., Prakash, M., and Walker, J. P. (2020). “Deep learning for an improved prediction of rainfall retrievals from commercial microwave links”. *Water Resources Research*, **56(7)**, e2019WR026255.

Qing-dao-er-ji, R., Su, Y. L., and Liu, W. W. (2020). “Research on the lstm mongolian and chinese machine translation based on morpheme encoding”. *Neural Computing and Applications*, **32**, 41–49.

Qiu, M., Zhao, P., Zhang, K., Huang, J., Shi, X., Wang, X., and Chu, W. (2017). “A short-term rainfall prediction model using multi-task convolutional neural networks”. *2017 IEEE international conference on data mining (ICDM)*, 395–404.

Ralte, V., and Sil, B. S. (2021). “Drought analysis for the region in and around of mizoram state, india”. *International Conference on Hydraulics, Water Resources and Coastal Engineering*, 369–376.

Raza, A., Mehmood, A., Ullah, S., Ahmad, M., Choi, G. S., and On, B.-W. (2019). "Heartbeat sound signal classification using deep learning". *Sensors*, **19(21)**, 4819.

Ridwan, W. M., Sapitang, M., Aziz, A., Kushiar, K. F., Ahmed, A. N., and El-Shafie, A. (2021). "Rainfall forecasting model using machine learning methods: Case study terengganu, malaysia". *Ain Shams Engineering Journal*, **12(2)**, 1651–1663.

Ruma, J. F., Adnan, M. S. G., Dewan, A., and Rahman, R. M. (2023). "Particle swarm optimization based lstm networks for water level forecasting: A case study on bangladesh river network". *Results in Engineering*, **17**, 100951.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning representations by back-propagating errors". *nature*, **323(6088)**, 533–536.

Russell, S. J., and Norvig, P. (2016). *Artificial intelligence: A modern approach*. Pearson.

Saha, M., Mitra, P., and Nanjundiah, R. S. (2016a). "Autoencoder-based identification of predictors of indian monsoon". *Meteorology and Atmospheric Physics*, **128**, 613–628.

Saha, M., Mitra, P., and Nanjundiah, R. S. (2016b). "Predictor discovery for early-late indian summer monsoon using stacked autoencoder". *Procedia Computer Science*, **80**, 565–576.

Saha, M., Mitra, P., and Nanjundiah, R. S. (2017). "Deep learning for predicting the monsoon over the homogeneous regions of india". *Journal of earth system science*, **126**, 1–18.

Saha, M., Santara, A., Mitra, P., Chakraborty, A., and Nanjundiah, R. S. (2021). "Prediction of the indian summer monsoon using a stacked autoencoder and ensemble regression model". *International Journal of Forecasting*, **37(1)**, 58–71.

Saha, S., Das, B., Chakraborty, D., Chawdhury, S., Lalliansanga, S., and Saithantluanga, H. (2021). “Quantitative accuracy assessment of district level rainfall forecast in mizoram”. *J. Agric. Phys*, **21**, 332–340.

Sahoo, M., and Yadav, R. K. (2021). “Teleconnection of atlantic nino with summer monsoon rainfall over northeast india”. *Global and Planetary Change*, **203**, 103550.

Sahu, H. P., and Kashyap, R. (2023). “Fine_denseiganet: Automatic medical image classification in chest ct scan using hybrid deep learning framework”. *International Journal of Image and Graphics*, 2550004.

Salari, S., Moghaddasi, M., Mohammadi Ghaleni, M., and Akbari, M. (2021). “Groundwater level prediction in golpayegan aquifer using anfis and pso combination”. *Iranian Journal of Soil and Water Research*, **52(3)**, 721–732.

Samsi, S., Mattioli, C. J., and Veillette, M. S. (2019). “Distributed deep learning for precipitation nowcasting”. *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–7.

Sato, R., Kashima, H., and Yamamoto, T. (2018). “Short-term precipitation prediction with skip-connected prednet”. *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, 373–382.

Schuster, M., and Paliwal, K. K. (1997). “Bidirectional recurrent neural networks”. *IEEE transactions on Signal Processing*, **45(11)**, 2673–2681.

Sengoz, C., Ramanna, S., Kehler, S., Goomer, R., and Pries, P. (2023). “Machine learning approaches to improve north american precipitation forecasts”. *IEEE Access*, **11**, 97664–97681.

Sharma, O., Trivedi, D., Pattnaik, S., Hazra, V., and Puhan, N. B. (2023). “Improvement in district scale heavy rainfall prediction over complex terrain of north east india using deep learning”. *IEEE Transactions on Geoscience and Remote Sensing*, **61**, 1–8.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). "Convolutional lstm network: A machine learning approach for precipitation nowcasting". *Advances in neural information processing systems*, **28**.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. (2017). "Deep learning for precipitation nowcasting: A benchmark and a new model". *Advances in neural information processing systems*, **30**.

Shirzadi, N., Nizami, A., Khazen, M., and Nik-Bakht, M. (2021). "Medium-term regional electricity load forecasting through machine learning and deep learning". *Designs*, **5(2)**, 27.

Singh, P., Gnanaseelan, C., and Chowdary, J. (2017). "North-east monsoon rainfall extremes over the southern peninsular india and their association with el niño". *Dynamics of Atmospheres and Oceans*, **80**, 1–11.

Singh, S., Sarkar, S., and Mitra, P. (2017). "Leveraging convolutions in recurrent neural networks for doppler weather radar echo prediction". *Advances in Neural Networks-ISNN 2017: 14th International Symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21–26, 2017, Proceedings, Part II 14*, 310–317.

Song, S., Huang, H., and Ruan, T. (2019). "Abstractive text summarization using lstm-cnn based deep learning". *Multimedia Tools and Applications*, **78(1)**, 857–875.

Souto, Y. M., Porto, F., Moura, A. M., and Bezerra, E. (2018). "A spatiotemporal ensemble approach to rainfall forecasting". *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Suddle, M. K., and Bashir, M. (2022). "Metaheuristics based long short term memory optimization for sentiment analysis". *Applied Soft Computing*, **131**, 109794.

Tan, Q., Liu, N., and Hu, X. (2019). "Deep representation learning for social network analysis". *Frontiers in big Data*, **2**, 2.

Tao, Y., Gao, X., Hsu, K., Sorooshian, S., and Ihler, A. (2016). “A deep neural network modeling framework to reduce bias in satellite precipitation products”. *Journal of Hydrometeorology*, **17(3)**, 931–945.

Tian, L., Li, X., Ye, Y., Xie, P., and Li, Y. (2019). “A generative adversarial gated recurrent unit model for precipitation nowcasting”. *IEEE Geoscience and Remote Sensing Letters*, **17(4)**, 601–605.

Tran, Q.-K., and Song, S.-k. (2019). “Computer vision in precipitation nowcasting: Applying image quality assessment metrics for training deep neural networks”. *Atmosphere*, **10(5)**, 244.

Usmani, S., and Shamsi, J. A. (2023). “Lstm based stock prediction using weighted and categorized financial news”. *Plos one*, **18(3)**, e0282234.

Viswanath, S., Saha, M., Mitra, P., and Nanjundiah, R. S. (2019). “Deep learning based lstm and seqtoseq models to detect monsoon spells of india”. *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part II 19*, 204–218.

Wan, S., Qi, L., Xu, X., Tong, C., and Gu, Z. (2020). “Deep learning models for real-time human activity recognition with smartphones”. *Mobile Networks and Applications*, **25(2)**, 743–755.

Wang, L., Xu, Y., Cheng, J., Xia, H., Yin, J., and Wu, J. (2018). “Human action recognition by learning spatio-temporal features with deep neural networks”. *IEEE access*, **6**, 17913–17922.

Wang, W., Liang, D., Chen, Q., Iwamoto, Y., Han, X.-H., Zhang, Q., Hu, H., Lin, L., and Chen, Y.-W. (2020). Medical image classification using deep learning. In Y.-W. Chen and L. C. Jain (Eds.), *Deep learning in healthcare: Paradigms and applications* (pp. 33–51). Springer International Publishing.

Wang, Z., Kim, S., and Joe, I. (2023). “An improved lstm-based failure classification model for financial companies using natural language processing”. *Applied Sciences*, **13(13)**, 7884.

Weesakul, U., Kaewprapha, P., Boonyuen, K., and Mark, O. (2018). “Deep learning neural network: A machine learning approach for monthly rainfall forecast, case study in eastern region of thailand”. *Engineering and Applied Science Research*, **45(3)**, 203–211.

Wei, G., Chen, W., and Dongzhou, N. (2024). “Enhancing sustainable development through sentiment analysis of public digital resources: A pso-lstm approach”. *Journal of the Knowledge Economy*, 1–20.

Wu, H., Yang, Q., Liu, J., and Wang, G. (2020). “A spatiotemporal deep fusion model for merging satellite and gauge precipitation in china”. *Journal of Hydrology*, **584**, 124664.

Wu, L., Zhang, Q., Chen, C.-H., Guo, K., and Wang, D. (2020). “Deep learning techniques for community detection in social networks”. *IEEE Access*, **8**, 96016–96026.

Yan, Q., Ji, F., Miao, K., Wu, Q., Xia, Y., and Li, T. (2020). “Convolutional residual-attention: A deep learning approach for precipitation nowcasting”. *Advances in Meteorology*, **2020**, 1–12.

Yazici, I., Beyca, O. F., and Delen, D. (2022). “Deep-learning-based short-term electricity load forecasting: A real case application”. *Engineering Applications of Artificial Intelligence*, **109**, 104645.

Yen, M.-H., Liu, D.-W., Hsin, Y.-C., Lin, C.-E., and Chen, C.-C. (2019). “Application of the deep learning for the prediction of rainfall in southern taiwan”. *Scientific reports*, **9(1)**, 12774.

Ying, W., Zhang, L., and Deng, H. (2020). “Sichuan dialect speech recognition with deep lstm network”. *Frontiers of Computer Science*, **14(2)**, 378–387.

Yonekura, K., Hattori, H., and Suzuki, T. (2018). “Short-term local weather forecast using dense weather station by deep neural network”. *2018 IEEE international conference on big data (big data)*, 1683–1690.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). “A review of recurrent neural networks: Lstm cells and network architectures”. *Neural computation*, **31(7)**, 1235–1270.

Yuan, Q., Dai, Y., and Li, G. (2023). “Exploration of english speech translation recognition based on the lstm rnn algorithm”. *Neural Computing and Applications*, **35(36)**, 24961–24970.

Zerouali, B., Santos, C. A. G., de Farias, C. A. S., Muniz, R. S., Difi, S., Abda, Z., Chettih, M., Heddam, S., Anwar, S. A., and Elbeltagi, A. (2023). “Artificial intelligent systems optimized by metaheuristic algorithms and teleconnection indices for rainfall modeling: The case of a humid region in the mediterranean basin”. *Heliyon*, **9(4)**, 2405–8440.

Zhai, Z., Zhang, X., Fang, F., and Yao, L. (2023). “Text classification of chinese news based on multi-scale cnn and lstm hybrid model”. *Multimedia Tools and Applications*, **82(14)**, 20975–20988.

Zhang, C.-J., Wang, H.-Y., Zeng, J., Ma, L.-M., and Guan, L. (2020). “Tiny-rainnet: A deep convolutional neural network with bi-directional long short-term memory model for short-term rainfall prediction”. *Meteorological Applications*, **27(5)**, e1956.

Zhang, C.-J., Zeng, J., Wang, H.-Y., Ma, L.-M., and Chu, H. (2020). “Correction model for rainfall forecasts using the lstm with multiple meteorological factors”. *Meteorological Applications*, **27(1)**, e1852.

Zhang, J., Xie, Y., Wu, Q., and Xia, Y. (2019). “Medical image classification using synergic deep learning”. *Medical image analysis*, **54**, 10–19.

Zhang, L., Lim, C. P., and Yu, Y. (2021). “Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization”. *Knowledge-based systems*, **220**, 106918.

Zhang, L., Lim, C. P., Yu, Y., and Jiang, M. (2022). “Sound classification using evolving ensemble models and particle swarm optimization”. *Applied soft computing*, **116**, 108322.

Zhang, P., Jia, Y., Zhang, L., Gao, J., and Leung, H. (2018). “A deep belief network based precipitation forecast approach using multiple environmental factors”. *Intelligent Data Analysis*, **22(4)**, 843–866.

Zhang, P., Zhang, L., Leung, H., and Wang, J. (2017). “A deep-learning based precipitation forecasting approach using multiple environmental factors”. *2017 IEEE international congress on big data (bigdata congress)*, 193–200.

Zhou, K., Zheng, Y., Li, B., Dong, W., and Zhang, X. (2019). “Forecasting different types of convective weather: A deep learning approach”. *Journal of Meteorological Research*, **33**, 797–809.

Zulfiqar, M., Kamran, M., Rasheed, M., Alquthami, T., and Milyani, A. (2022). “Hyperparameter optimization of support vector machine using adaptive differential evolution for electricity load forecasting”. *Energy Reports*, **8**, 13333–13352.

BIO-DATA OF THE CANDIDATE

Personal Information:

Name : C ZOREMSANGA

Father's name : C Zaihmingthanga

Mother's name : KC Lalmuani

Date of Birth : 01.10.1988

Nationality : Indian

Gender : Male

Marital Status : Married

Present Address : F/C-40/4, Falkland Veng, Aizawl, Mizoram.

Email : zoremsanga@gmail.com

Academic Records:

X^{th} : M.B.S.E. (2003)

XII^{th} : M.B.S.E. (2005)

BE(Comp. Sc. and Engg.) : Dibrugarh University (2009)

M.Tech. : BITS Pilani (2015)

(C ZOREMSANGA)

LIST OF PUBLICATIONS

1. **Zoremsanga, C.**, and Hussain, J. (2024). "Particle Swarm Optimized Deep Learning Models for Rainfall Prediction: A Case Study in Aizawl, Mizoram", *IEEE Access*, **12**, 57172-57184, 2024.
   DoI: 10.1109/ACCESS.2024.3390781.

2. **Zoremsanga, C.**, and Hussain, J. (2024). "An Evaluation of Bidirectional Long Short-Term Memory Model for Estimating Monthly Rainfall in India", *Indian Journal of Science and Technology*, **17(18)**, 1828-1837, 2024.
   DoI: 10.17485/IJST/v17i18.2505.

3. **Zoremsanga, C.**, and Hussain, J. (2024). "Hybrid Particle Swarm Optimized Models for Rainfall Prediction: A Case Study in India", *Pure and Applied Geophysics*, **181**, 2343–2357,
   DoI: 10.1007/s00024-024-03528-7.

4. Hussain, J., and **Zoremsanga, C.** (2021). "A Survey of Rainfall Prediction Using Deep Learning", *2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, 2021, pp. 1-10.
   DoI: https://doi.org/10.1109/ICECIE52348.2021.9664730.

5. **Zoremsanga, C.**, and Hussain, J. (2023) "A Comparative Study of Long Short-Term Memory for Rainfall Prediction in India", In S. N. Singh, S. Mahanta, and Y. J. Singh (Eds.), *Proceedings of the NIELIT's International Conference on Communication, Electronics and Digital Technology, Lecture Notes in Networks and Systems*, 2023, pp. 547–558.
   DoI: https://doi.org/10.1007/978-981-99-1699-3_38.

CONFERENCES ATTENDED WITH PRESENTATION

1. IEEE 3rd International Conference on Electrical, Control and Instrumentation engineering, 2021 (ICECIE 2021), held during 27th -28th November 2021 at Kuala Lumpur, Malaysia.

2. NIELIT's International Conference on Communication, Electronics and Digital Technologies-NICE-DT'23, held during 10th - 11th February, 2023 at New Delhi, India.

PARTICULARS OF THE CANDIDATE

NAME OF CANDIDATE : C ZOREMSANGA

DEGREE : DOCTOR OF PHILOSOPHY

DEPARTMENT : MATHEMATICS AND COMPUTER SCIENCE

TITLE OF THESIS : RAINFALL PREDICTION IN MIZORAM USING LONG SHORT-TERM MEMORY NETWORKS (LSTM) AND PARTICLE SWARM OPTIMIZATION (PSO) TECHNIQUE

DATE OF ADMISSION : 28.08.2018

APPROVAL OF RESEARCH PROPOSAL:

1. DRC : 15.04.2019
2. BOS : 24.04.2019
3. SCHOOL BOARD : 08.05.2019

MZU REGISTRATION NO. : 1801172

Ph. D. REGISTRATION NO. : MZU/Ph. D./1268 of 28.08.2018

EXTENSION. : NA

(PROF. M. SUNDARARAJAN)
(Head of Department)
Dept. of Maths. and Comp. Sc.,
Mizoram University, Mizoram.

ABSTRACT

RAINFALL PREDICTION IN MIZORAM USING LONG SHORT-TERM MEMORY NETWORKS (LSTM) AND PARTICLE SWARM OPTIMIZATION (PSO) TECHNIQUE

AN ABSTRACT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

C ZOREMSANGA

MZU REGISTRATION NO.: 1801172

Ph.D. REGISTRATION NO.: MZU/Ph.D./1268 of 28.08.2018

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

SCHOOL OF PHYSICAL SCIENCES

JUNE, 2024

ABSTRACT

RAINFALL PREDICTION IN MIZORAM USING LONG SHORT-TERM MEMORY NETWORKS (LSTM) AND PARTICLE SWARM OPTIMIZATION (PSO) TECHNIQUE

BY

C ZOREMSANGA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

Supervisor: Prof. Jamal Hussain

Submitted

In partial fulfillment of the requirement of the Degree of Doctor of Philosophy in Computer Science of Mizoram University, Aizawl

ABSTRACT

Rainfall is a fundamental meteorological parameter critical in shaping ecosystems, water resources, agricultural productivity, and regional development. Accurate rainfall prediction is essential for sustainable resource management, disaster preparedness, and climate adaptation strategies. This research aims to develop and evaluate a rainfall prediction model using the all-India monthly average rainfall dataset and the Weather parameters collected by the Aizawl Weather Station. By combining Long Short-Term Memory (LSTM) neural networks with the Particle Swarm Optimization (PSO) technique, the study seeks to create a hybrid approach that captures temporal dependencies effectively and optimizes model parameters for enhanced prediction accuracy. This study addresses the gap in the literature and contributes to the advancement of rainfall prediction techniques. The main objectives of the study are as follows-

1. To analyze and preprocess the weather data collected from the Aizawl weather station and Data Supply Portal maintained by the National Data Center, IMD, Pune.
2. To analyze the input features of the weather data and network parameters for the LSTM networks using the PSO technique.
3. To predict rainfall using the LSTM network with the network parameters determined by PSO.
4. To compare the prediction result of LSTM and PSO with other neural network models such as ANN, SVM and RBF.

Chapter 1 commences with an introduction to the context and rationale behind the study, accompanied by a statement of the problem and its objectives. This chapter also defines the scope and limitations of the study while highlighting its

significance. Accurate rainfall prediction ensures sustainable resource management, disaster readiness, and climate adaptation strategies. By anticipating changes in rainfall patterns, decision-makers gain valuable insights that aid them in making informed decisions, allocating resources effectively, and mitigating the consequences of severe weather occurrences.

Accurately forecasting rainfall continues to be difficult, mainly because of the intricate nature of atmospheric processes and the dynamic interplay between different meteorological factors. The traditional regression-based approaches in this regard tend to oversimplify the underlying processes and fail to consider non-linear relationships. Additionally, these methods often struggle to handle missing data, outliers, and noisy datasets, resulting in less-than-optimal predictions. These limitations have sparked interest in utilizing advanced machine-learning techniques for rainfall prediction, particularly given the increasing availability of large-scale meteorological datasets.Many factors influence rainfall patterns, including temperature, humidity, wind patterns, geographical features, and global climate phenomena like El Ni˜no and La Ni˜na. These factors operate across various temporal scales, leading to intricate patterns that are difficult to capture using traditional methods

Traditional machine learning models have a limitation in capturing dependencies within data, as they typically consider only the current input during processing. Deep learning models, such as Recurrent Neural Network (RNN), address this issue by capturing dependencies within input data. However, RNNs face the vanishing gradient problem, limiting their ability to capture long-term dependencies. LSTM networks solve this problem by maintaining long-term dependencies. Despite their advantages, both RNN and LSTM transmit knowledge only in the forward direction. The Bidirectional Long Short-Term Memory (BiLSTM) model, introduced improves performance by capturing past and future hidden states through a bidirectional network, enabling the model to learn long-term dependencies more effectively.

The motivation behind investigating LSTM networks in rainfall prediction arises from their capacity to comprehend and anticipate rainfall variations through capturing

temporal relationships and patterns. LSTM has proven successful in various fields, such as natural language processing, speech recognition, and financial prediction. Utilizing LSTM for rainfall prediction can enhance accuracy by modeling the complex dependencies inherent in meteorological time series data. LSTM networks are practical tools for capturing time-dependent patterns in data. However, their effectiveness is significantly impacted by hyperparameters such as the number of hidden units, hidden layers, and epochs. Determining the optimal combination of these hyperparameters through trial and error or grid search can be both costly and time-consuming. These methods can also get stuck in local optima and do not guarantee the discovery of the ideal hyperparameters.

PSO is an optimization technique known for efficiently searching optimal hyperparameters within a given parameter space. By combining LSTM with PSO, automatic fine-tuning of hyperparameters becomes feasible, improving predictive accuracy without manual adjustment requirements. Despite the success of PSO in optimizing proposed models, existing studies typically optimize only the proposed models while the hyperparameters of compared models are often determined through grid search, trial-and-error, or existing literature. This inconsistency indicates a gap where all models, both proposed and compared, should be uniformly optimized using PSO and tested on the same dataset, highlighting an area needing further investigation. Therefore, our objective is to examine if the PSO-optimized LSTM model can effectively forecast the amount of rainfall by utilizing the meteorological dataset encompassing the all-India monthly averages and the Mizoram rainfall datasets.

Mizoram, situated in the north-eastern part of India, lies between the geographical coordinates of 21° 58’ and 24° 35’ N latitude and 92° 15’ and 93° 29’ E longitude. This state shares its borders with Tripura, Assam, and Manipur within India and a border spanning approximately 722 kilometres with the neighbouring countries of Bangladesh and Myanmar. Mizoram experiences significant yearly rainfall, averaging between 2,500 mm to 3,000 mm. Mizoram is directly influenced by the South-West Monsoon, which typically brings an adequate amount of rainfall. The rainy season, or summer

monsoon, usually begins in April, with heavy rains occurring from May to September, and extends until late October. In contrast, the winter season, from November to February, is generally dry with minimal rainfall.

More than 70% of Mizoram's population relies on agriculture for their livelihood, with the majority practicing shifting cultivation. Given the state's topography and the significant rainfall it receives, Mizoram is highly vulnerable to abnormal climate variability and long-term climate changes. Climate-related hazards can severely impact all sectors in Mizoram, as its socio-economic conditions are relatively underdeveloped compared to other states in India. Both urban and rural populations heavily depend on agriculture and allied sectors, which are highly susceptible to climate variability and long-term changes. This vulnerability can lead to numerous issues, such as alterations in the timing of field preparation, sowing, harvesting, and overall yield. Additionally, climate-related hazards like landslides and flash floods, which are prevalent in Mizoram, can severely affect other developmental sectors. However, there has been no previous application of machine learning or deep learning models to analyze meteorological data specifically for Mizoram. This research aims to bridge the gap by exploiting the power of LSTM and the PSO optimization technique to improve rainfall prediction in Mizoram and contribute to the broader field of meteorology.

Chapter 2 presents an extensive examination of literature concerning rainfall prediction since 2015. The literature is categorized based on the specific datasets utilized for training and testing the predictive models. These datasets encompass various weather parameters, radar images, and satellite images. This chapter thoroughly studies 45 papers from renowned publishers to gain insights into rainfall prediction. The classification of these papers focuses on the types of data implemented by the authors in their research. Furthermore, deep learning methods employed in these studies, the specific input data adopted for predicting rainfall, and metrics used to assess model performance are investigated extensively throughout this chapter. Additionally, both temporal and spatial distribution patterns of rainfall predictions are analyzed within this comprehensive exploration.

The research conducted in this chapter reveals a noticeable upward trend in the number of scientific papers focusing on rainfall prediction using deep learning techniques. Specifically, in 2019, there was a significant increase, representing 51% of all publications. This growth indicates a growing interest in applying deep learning methods for predicting rainfall patterns. It is worth noting that LSTM and Convolutional Long Short-Term Memory (ConvLSTM) emerged as the most commonly employed methodologies. Authors often utilize Auto Encoder (AE) and Stacked Auto Encoder (SAE) to identify predictor variables when dealing with large input datasets. In cases where satellite or radar images are used as inputs, convolutional layers are frequently employed to extract relevant features from the data. Deep learning models have been found to outperform traditional machine learning models regarding accuracy in rainfall prediction tasks due to their ability to capture temporal and spatial information inherent within the input data. Geographically, China leads the research on this subject, followed by India and the United States of America (USA). Among the software utilized by researchers, MATLAB takes precedence as the most widely used option, followed closely by Tensorflow and Keras. Regarding evaluation metrics for assessing performance, RMSE (Root Mean Square Error) is widely adopted among researchers studying rainfall prediction using deep learning methods; however, Mean Absolute Error (MAE), Critical Success Index (CSI), and False Alarm Rate (FAR) are also used regularly within published works.

To summarize, the findings of this study indicate that deep learning techniques outperformed traditional machine learning models and shallow neural network architecture in the area of rainfall prediction. Therefore, deep learning methods are considered more favorable for this task.

Chapter 3 analyzes four LSTM models for predicting the monthly average rainfall in India from 1871 to 2016. The first model, LSTM Model 1, consists of a single hidden layer with fifty cells in an LSTM architecture. It also includes a Dense output layer with just one neuron. LSTM Model 2, conversely, comprises a hidden LSTM layer and a hidden Dense layer. The hidden LSTM layer consists of ten cells, while the

hidden Dense layer comprises ten neurons. Similar to previous models, LSTM Model 2 has a Dense output layer with one neuron. LSTM Model 3 has two stacked LSTM layers and a dense output layer. These stacked LSTM layers contain ten cells in each layer, whereas the dense output only possesses one neuron. Lastly, LSTM Model 4 is constructed using two LSTMs as its hidden layers - each consisting of twelve cells, one Hidden Dense Layer featuring twelve neurons, and an output Dense layer with a single neuron.

The LSTM Model 4 outperformed the other models, achieving the lowest RMSE value of 245.30. This indicates that stacked LSTM models have a strong ability to forecast rainfall patterns in India accurately. Additionally, increasing the timesteps further improved the performance of these models. To assess the effectiveness of our LSTM models, we compared them to RNN and LSTM models from a previous study. In this study, all the proposed LSTM models outperformed the LSTM and RNN models found in the literature in terms of Root Mean Squared Error (RMSE). This shows that incorporating more neurons and additional timesteps can significantly enhance rainfall prediction accuracy.

In Chapter 4, the predictive performance of the Bidirectional LSTM model is evaluated with the Vanilla LSTM, Stacked LSTM, and a benchmark model found in the existing literature. The main goal is to estimate the average rainfall for India one month ahead by utilizing the previous month's rainfall as a predictor. Furthermore, this study seeks to determine the optimal number of previous months' rainfall, epochs, and cells that produce the lowest RMSE when forecasting one-month rainfall into the future. To assess its effectiveness, our proposed model's performance is compared to Vanilla LSTM, Stacked LSTM, and both RNN and LSTM models from existing studies.

Out of all the models analyzed, the Bidirectional LSTM achieved an RMSE value of 240.79, the lowest among them. The Stacked LSTM followed with an RMSE value of 245.3, while the Vanilla LSTM had a slightly higher RMSE value of 250.47. In comparison, the benchmark RNN and LSTM models performed less effectively with RMSE values of 261.7 and 251.63, respectively. The findings indicated that

achieving optimal prediction accuracy requires considering various factors. These factors include increasing the number of input timesteps, adding more LSTM cells, and extending the training epochs while closely monitoring for over-fitting. It is crucial to strike a balance between the size of the input data, the number of LSTM cells in each layer, and how extensively the training dataset is exposed to the LSTM model. Additionally, employing stacked LSTM layers and utilizing a Bidirectional LSTM approach significantly improved model performance compared to single-cell LSTM or Vanilla LSTM models.

Chapter 5 of the study compared various PSO-optimized deep learning and machine learning models. These models included PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-Artificial Neural Network (ANN), and PSO-Support Vector Regression (SVR). The comparison was conducted using the Aizawl monthly rainfall dataset and the all-India monthly average rainfall dataset. This research is significant as it marks the first time machine learning and deep learning models have been employed to analyze Aizawl's rainfall data. Furthermore, this study also presents the first instance where the predictive capabilities of the models mentioned above (PSO-BiLSTM, PSO-LSTM, PSO-RNN, PSO-ANN) were compared in terms of their effectiveness in predicting rainfall patterns.

In the Aizawl monthly rainfall dataset, all versions of PSO-BiLSTM models outperformed the other compared models. Among the studied models, PSO-SVR with a polynomial kernel showed the lowest performance. The best model was the PSO-BiLSTM II model, which consists of a BiLSTM Layer comprising 3 cells, followed by a Dense layer composed of 103 neurons and an output Dense layer containing 1 neuron. Stacking the BiLSTM, LSTM, RNN, and ANN models did not improve this dataset's prediction accuracy. However, incorporating an additional Dense layer before the output Dense layer increased prediction accuracy for both BiLSTM, LSTM, and RNN models. It was also observed that each model required different optimal input timesteps when utilizing the PSO algorithm. This indicates that determining optimal input timesteps depends on the specific dataset and the type of

predictive models employed. Additionally, it was discovered that compared to other examined models in this study, the BiLSTM models demonstrated faster convergence toward a solution by requiring fewer epochs.

According to the all-India monthly average rainfall dataset, the PSO-BiLSTM models consistently achieved the lowest RMSE score. Similar results were observed in the Aizawl monthly rainfall dataset, where BiLSTM, LSTM, and RNN stacked with a Dense layer exhibited better prediction performance compared to models without a Dense layer. The PSO-SVR model performed poorly among all the compared models in this dataset. Additionally, all PSO-optimized models outperformed those that utilized hyper-parameters determined through the grid search method or benchmarked from existing literature. This demonstrates that the Particle Swarm Optimization (PSO) algorithm is reliable for identifying optimal hyperparameters. However, due to its stochastic nature, relying on only one model prediction from PSO could lead to incorrect conclusions about its superiority over other models. To mitigate this issue and ensure accurate evaluation, this study repeated both model training and testing processes ten times. If computational resources allow for it, repeating these iterations more than ten times is recommended.

Chapter 6 presents an exhaustive comparison of twelve hybrid deep learning and machine learning models optimized using PSO. These models include BiLSTM, LSTM, RNN, ANN, and SVR. This study's main objective is to predict the amount of rainfall expected for the next day. To achieve this, meteorological data from 1985 to 2018, collected by the Aizawl Weather Station in Mizoram, India, is utilized. This dataset comprises 12,418 days of weather parameters such as maximum humidity, minimum temperature, maximum temperature, and previous rainfall amounts to forecast future precipitation patterns accurately.

The studied models showed convergence at a remarkably low number of epochs, which indicates that the PSO model is efficient in avoiding overfitting the model. Hybrid models demonstrated superior predictive abilities compared to the standalone models. Also, the BiLSTM models generally outperformed the compared model,

proving that they were the most suitable model among the models under study. These experimental results demonstrate the viability of PSO-optimized hybrid deep learning models in rainfall prediction and show the advantage of the BiLSTM model over other deep learning models. This research provides a foundational benchmark for future studies involving the Aizawl weather station data, and future work could be done to include more data from other weather stations.

Chapter 7 is Summary and Conclusion

Keywords: Machine learning, Deep learning, Neural network, Particle Swarm Optimization, LSTM, Stacked LSTM, Bidirectional LSTM, Rainfall prediction.