


Document Information

Analyzed document	main (4).pdf (D172932832)
Submitted	2023-08-18 17:39:00
Submitted by	
Submitter email	
Similarity	
Analysis address	
	vipinkumar@mgcub.ac.in
	1%
	vipinkumar.mgcub@analysis.urbund.com

Sources included in the report

	URL: https://www.sciencedirect.com/science/article/pii/S0959652619325892 . Fetched: 2023-08-18 17:39:00		3
	URL: https://www.sciencedirect.com/science/article/pii/S0959652621038373 . Fetched: 2023-08-18 17:39:00		2
	URL: https://www.sciencedirect.com/science/article/pii/S0959652622044638 . Fetched: 2023-08-18 17:39:00		1
	URL: http://www.sciencedirect.com/science/article/pii/S0959652623011940 . Fetched: 2023-08-18 17:39:00		1
	URL: https://www.sciencedirect.com/science/article/pii/S0959652622015086 . Fetched: 2023-08-18 17:39:00		3
	URL: https://arxiv.org/pdf/2112.13444 Fetched: 2021-12-30 06:01:01		1

Entire Document

Multi-view Stacked CNN-BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities A dissertation-II submitted to the Mahatma Gandhi Central University in partially fulfillment of the requirements for the award of the degree of MASTER OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING BY SUBHAM KUMAR DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY MAHATMA GANDHI CENTRAL UNIVERSITY, MOTIHARI, BIHAR - 845401, INDIA August 18, 2023

Multi-view Stacked CNN-BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities A dissertation-II submitted to the Mahatma Gandhi Central University in partially fulfillment of the requirements for the award of the degree of MASTER OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING BY SUBHAM KUMAR (MGCU2021CSIT4029) Under the Supervision of Dr. VIPIN KUMAR DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY MAHATMA GANDHI CENTRAL UNIVERSITY, MOTIHARI, BIHAR - 845401, INDIA August 18, 2023

i कं कंप्यूटर विज्ञान और सूचना प्रौद्योगिकी विभाग Department Of Computer Science And Information Technology महात्मा गाँधी केन्द्रीय विश्व विद्यालय, बिहार-८४५४०१ MAHATMA GANDHI CENTRAL UNIVERSITY, MOTIHARI, BIHAR - 845401, INDIA
DECLARATION This is to certify that the dissertation-II entitled " Multi-view Stacked CNN- BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities" is being submitted to the Department Of Computer Science And Information Technology, Mahatma Gandhi Central University, Motihari, Bihar - 845401, India in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science & Engineering, is a record of bonafide work carried out by me under the supervision of "Dr. Vipin Kumar, Department Of Computer Science And Information Technology, Mahatma Gandhi Central University, Motihari, Bihar - 845401, India." The matter embodied in the dissertation has not been submitted in part or full to any University or Institution for the award of any other degree or diploma. During the preparation of this work, I have not used any AI-based tool to write any part of this dissertation report. I take full responsibility for the submitted content including similarity.
Subham Kumar (MGCU2021CSIT4029) Department Of Computer Science And Information Technology Mahatma Gandhi Central University, Motihari, Bihar - 845401, India Email id: - Subh700454@gmail.com

ii कं कंप्यूटर विज्ञान और सूचना प्रौद्योगिकी विभाग Department Of Computer Science And Information Technology महात्मा गाँधी केन्द्रीय विश्व विद्यालय, बिहार-८४५४०१ MAHATMA GANDHI CENTRAL UNIVERSITY, MOTIHARI, BIHAR - 845401, INDIA
CERTIFICATE This is to certify that the dissertation-II entitled "Multi-view Stacked CNN- BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities" submitted by Subham Kumar to the Department Of Computer Science And Information Technology, Mahatma Gandhi Central University, Motihari, Bihar - 845401, India for the award of the degree of Master of Technology in Computer Science & Engineering, is a research work carried out by him under the supervision of "Dr. Vipin Kumar, Department Of Computer Science And Information Technology, Mahatma Gandhi Central University, Motihari, Bihar - 845401, India." Head of Department Prof. Vikas Pareek Department Of Computer Science And Information Technology Mahatma Gandhi Central University, Motihari, Bihar - 845401, India Supervisor Dr. Vipin Kumar Department Of Computer Science And Information Technology Mahatma Gandhi Central University, Motihari, Bihar - 845401, India

iii Abstract The presence of PM 2.5 is a significant concern for human well-being and ecosystems. The practical measure of PM 2.5 is the vital problem worldwide. These tiny particles can quickly enter the respiratory system and deeply infiltrate the lungs, leading to various health issues, including respiratory disorders, cardiovascular diseases, and premature death. The literature shows that hybrid deep learning (DL) models are performing better than stand-alone DL models of time series (i.e., CNN, RNN, GRU, LSTM and BiLSTM) to predict the PM 2.5 pollutant, but effective performance is not achieved yet. In this research, the author has proposed a hybrid stacked CNN Bidirectional-LSTM model architecture that utilises the multiple views of the data corresponding to seasonal repetitions to induce the multiple models, called Multi-view Stacked CNN Bidirectional-LSTM (MvS CNN-BiLSTM). The proposed model has been deployed over seventeen univariate time series (PM 2.5) data of highly polluted Indian cities and stand-alone DL models. The performances of the proposed model have been compared using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) measures. The average enhancement of the proposed model on all datasets has been achieved compared to stand-alone DL models as RMSE: 7.11% (CNN), 5.08% (RNN), 3.80% (GRU), 5.57% (LSTM) and 4.05% (BiLSTM) and MAPE: 27.16% (CNN), 28.52% (RNN), 26.22% (GRU), 27.22% (LSTM), 23.11% (BiLSTM). Moreover, the non-parametric statistical analysis (Friedman and Holm's) have been performed and proves that the proposed model MvS CNN-BiLSTM is performed as distinct and compelling over both performance measures.

iv Acknowledgements This M.Tech dissertation-II is the result of hard work, upon which many people have contributed and given their support. I have made this dissertation on the topic "Multi-view Stacked CNN-BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities." I have also tried my best in this dissertation to explain all the related detail. I would like to express my sincere gratitude towards my Supervisor Dr. Vipin Kumar, Department of CS & IT, for providing excellent guidance, encouragement, inspiration, and constant and timely support throughout this M.Tech dissertation work. He taught me how to pursue the right aim towards the work, and showed me different ways to approach the research problem. His wide knowledge and logical ways of thinking have been great value for me, and his understanding and guidance have provided the successful completion of the Dissertation work. First and foremost, I would like to express my gratitude to our beloved Dean of the Computational Sciences, Information and Communication Technology and Head of Department of Computer Science and Information Technology Prof. Vikas Pareek, for providing his kind support in various aspects. A special thanks to all the Respected Teachers Dr. Sunil Kumar Singh, and Mr. Shubham Kumar, of the Department of Computer Science and Information Technology. I am always grateful to the university, our Honble Vice chancellor Prof. Sanjay Srivastava for providing such a good research environment. Special thanks to Ph.D scholar, especially Ritika Singh, Surbhi Kumari, Ibrahim Momin, Naushad Ahmad and my friends Tej Prakash, Gajendra Patel, Abhijeet Kumar, Amod Kumar, Rana Kumar, Krishna Murari, Rajan Kumar, Suraj, Md. Aamir Sohail, Shahzeb Khan, and all my lovely juniors for their invaluable feedbacks, care, and moral support during this endeavor. Mother and Father, it is impossible to thank adequately for everything you have done, from loving me unconditionally to rising me in a stable household, where your persistent efforts and traditional values taught your children to celebrate and embrace life. I could not have asked for better parents or role-models. You showed me that anything is possible with faith, hard work and determination. Subham Kumar (MGCU2021CSIT4029) M.Tech(CSE)

vi List of Publications 1. Multi-view Stacked CNN-BiLSTM (MvS CNN-BiLSTM) for Urban PM 2.5 Concentration Prediction of India's Polluted Cities journal of cleaner production (Impact Factor: 11.1) Indexed by SCI (Submitted on 14 th Aug 2023) Authors - Subham Kumar and Vipin Kumar

vii Contents	DECLARATION i	CERTIFICATE ii	Abstract iii	Acknowledgements iv	List of Publications vi	List of Figures x	List of Tables xi	List of Abbreviations xi	List of Symbols xii
1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
1	2	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.1
3.1	3.1	3.1.1	3.1.1	3.1.1	3.1.1	3.1.1	3.1.1	3.1.1	3.1.1
3.1.2	3.1.2	3.1.2	3.1.2	3.1.2	3.1.2	3.1.2	3.1.2	3.1.2	3.1.2
3.1.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4	3.1.4
3.1.5	3.1.5	3.1.5	3.1.5	3.1.5	3.1.5	3.1.5	3.1.5	3.1.5	3.1.5
3.2	3.2	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1
3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1
3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2	3.2.2
3.2.3	3.2.3	3.2.3	3.2.3	3.2.3	3.2.3	3.2.3	3.2.3	3.2.3	3.2.3
3.2.4	3.2.4	3.2.4	3.2.4	3.2.4	3.2.4	3.2.4	3.2.4	3.2.4	3.2.4
3.2.5	3.2.5	3.2.5	3.2.5	3.2.5	3.2.5	3.2.5	3.2.5	3.2.5	3.2.5
3.2.6	3.2.6	3.2.6	3.2.6	3.2.6	3.2.6	3.2.6	3.2.6	3.2.6	3.2.6
3.2.7	3.2.7	3.2.7	3.2.7	3.2.7	3.2.7	3.2.7	3.2.7	3.2.7	3.2.7
3.2.8	3.2.8	3.2.8	3.2.8	3.2.8	3.2.8	3.2.8	3.2.8	3.2.8	3.2.8
3.2.9	3.2.9	3.2.9	3.2.9	3.2.9	3.2.9	3.2.9	3.2.9	3.2.9	3.2.9
3.2.10	3.2.10	3.2.10	3.2.10	3.2.10	3.2.10	3.2.10	3.2.10	3.2.10	3.2.10
3.2.11	3.2.11	3.2.11	3.2.11	3.2.11	3.2.11	3.2.11	3.2.11	3.2.11	3.2.11
3.2.12	3.2.12	3.2.12	3.2.12	3.2.12	3.2.12	3.2.12	3.2.12	3.2.12	3.2.12
3.2.13	3.2.13	3.2.13	3.2.13	3.2.13	3.2.13	3.2.13	3.2.13	3.2.13	3.2.13
3.2.14	3.2.14	3.2.14	3.2.14	3.2.14	3.2.14	3.2.14	3.2.14	3.2.14	3.2.14
3.2.15	3.2.15	3.2.15	3.2.15	3.2.15	3.2.15	3.2.15	3.2.15	3.2.15	3.2.15
3.2.16	3.2.16	3.2.16	3.2.16	3.2.16	3.2.16	3.2.16	3.2.16	3.2.16	3.2.16
3.2.17	3.2.17	3.2.17	3.2.17	3.2.17	3.2.17	3.2.17	3.2.17	3.2.17	3.2.17
3.2.18	3.2.18	3.2.18	3.2.18	3.2.18	3.2.18	3.2.18	3.2.18	3.2.18	3.2.18
3.2.19	3.2.19	3.2.19	3.2.19	3.2.19	3.2.19	3.2.19	3.2.19	3.2.19	3.2.19
3.2.20	3.2.20	3.2.20	3.2.20	3.2.20	3.2.20	3.2.20	3.2.20	3.2.20	3.2.20
3.2.21	3.2.21	3.2.21	3.2.21	3.2.21	3.2.21	3.2.21	3.2.21	3.2.21	3.2.21
3.2.22	3.2.22	3.2.22	3.2.22	3.2.22	3.2.22	3.2.22	3.2.22	3.2.22	3.2.22
3.2.23	3.2.23	3.2.23	3.2.23	3.2.23	3.2.23	3.2.23	3.2.23	3.2.23	3.2.23
3.2.24	3.2.24	3.2.24	3.2.24	3.2.24	3.2.24	3.2.24	3.2.24	3.2.24	3.2.24
3.2.25	3.2.25	3.2.25	3.2.25	3.2.25	3.2.25	3.2.25	3.2.25	3.2.25	3.2.25
3.2.26	3.2.26	3.2.26	3.2.26	3.2.26	3.2.26	3.2.26	3.2.26	3.2.26	3.2.26
3.2.27	3.2.27	3.2.27	3.2.27	3.2.27	3.2.27	3.2.27	3.2.27	3.2.27	3.2.27
3.2.28	3.2.28	3.2.28	3.2.28	3.2.28	3.2.28	3.2.28	3.2.28	3.2.28	3.2.28
3.2.29	3.2.29	3.2.29	3.2.29	3.2.29	3.2.29	3.2.29	3.2.29	3.2.29	3.2.29
3.2.30	3.2.30	3.2.30	3.2.30	3.2.30	3.2.30	3.2.30	3.2.30	3.2.30	3.2.30
3.2.31	3.2.31	3.2.31	3.2.31	3.2.31	3.2.31	3.2.31	3.2.31	3.2.31	3.2.31
3.2.32	3.2.32	3.2.32	3.2.32	3.2.32	3.2.32	3.2.32	3.2.32	3.2.32	3.2.32
3.2.33	3.2.33	3.2.33	3.2.33	3.2.33	3.2.33	3.2.33	3.2.33	3.2.33	3.2.33
3.2.34	3.2.34	3.2.34	3.2.34	3.2.34	3.2.34	3.2.34	3.2.34	3.2.34	3.2.34
3.2.35	3.2.35	3.2.35	3.2.35	3.2.35	3.2.35	3.2.35	3.2.35	3.2.35	3.2.35
3.2.36	3.2.36	3.2.36	3.2.36	3.2.36	3.2.36	3.2.36	3.2.36	3.2.36	3.2.36
3.2.37	3.2.37	3.2.37	3.2.37	3.2.37	3.2.37	3.2.37	3.2.37	3.2.37	3.2.37
3.2.38	3.2.38	3.2.38	3.2.38	3.2.38	3.2.38	3.2.38	3.2.38	3.2.38	3.2.38
3.2.39	3.2.39	3.2.39	3.2.39	3.2.39	3.2.39	3.2.39	3.2.39	3.2.39	3.2.39
3.2.40	3.2.40	3.2.40	3.2.40	3.2.40	3.2.40	3.2.40	3.2.40	3.2.40	3.2.40
3.2.41	3.2.41	3.2.41	3.2.41	3.2.41	3.2.41	3.2.41	3.2.41	3.2.41	3.2.41
3.2.42	3.2.42	3.2.42	3.2.42	3.2.42	3.2.42	3.2.42	3.2.42	3.2.42	3.2.42
3.2.43	3.2.43	3.2.43	3.2.43	3.2.43	3.2.43	3.2.43	3.2.43	3.2.43	3.2.43
3.2.44	3.2.44	3.2.44	3.2.44	3.2.44	3.2.44	3.2.44	3.2.44	3.2.44	3.2.44
3.2.45	3.2.45	3.2.45	3.2.45	3.2.45	3.2.45	3.2.45	3.2.45	3.2.45	3.2.45
3.2.46	3.2.46	3.2.46	3.2.46	3.2.46	3.2.46	3.2.46	3.2.46	3.2.46	3.2.46
3.2.47	3.2.47	3.2.47	3.2.47	3.2.47	3.2.47	3.2.47	3.2.47	3.2.47	3.2.47
3.2.48	3.2.48	3.2.48	3.2.48	3.2.48	3.2.48	3.2.48	3.2.48	3.2.48	3.2.48
3.2.49	3.2.49	3.2.49	3.2.49	3.2.49	3.2.49	3.2.49	3.2.49	3.2.49	3.2.49
3.2.50	3.2.50	3.2.50	3.2.50	3.2.50	3.2.50	3.2.50	3.2.50	3.2.50	3.2.50
3.2.51	3.2.51	3.2.51	3.2.51	3.2.51	3.2.51	3.2.51	3.2.51	3.2.51	3.2.51
3.2.52	3.2.52	3.2.52	3.2.52	3.2.52	3.2.52	3.2.52	3.2.52	3.2.52	3.2.52
3.2.53	3.2.53	3.2.53	3.2.53	3.2.53	3.2.53	3.2.53	3.2.53	3.2.53	3.2.53
3.2.54	3.2.54	3.2.54	3.2.54	3.2.54	3.2.54	3.2.54	3.2.54	3.2.54	3.2.54
3.2.55	3.2.55	3.2.55	3.2.55	3.2.55	3.2.55	3.2.55	3.2.55	3.2.55	3.2.55

x List of Tables	2.1 summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap.	7
	2.1 summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap.	8
	2.1 summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap.	9
	2.1 summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap.	10
4.1 Statistical exploratory data analyses (EDA) of 17 Time Series Datasets of polluted Indian cities based on PM 2.5 [79].		21
4.2 Parameter setting of traditional DL models and proposed MvS CNN- BiLSTM		33
5.1 RMSE Performance of traditional DL models and proposed models (MvS CNN-BiLSTM).		42
5.2 Percentage improvement of MvS CNN-BiLSTM respectively BiLSTM, CNN, GRU, LSTM and RNN on RMSE.		44
5.3 Average Rankings of the algorithms (Friedman) in a contest of RMSE.		45
5.4 MAPE Performance of traditional DL models and proposed models (MvS CNN-BiLSTM).		47
5.5 Improvement of MvS CNN-BiLSTM respectively BiLSTM, GRU, LSTM and RNN on MAPE.		49
5.6 Average rankings of the algorithms (Friedman) in the context of MAPE.		50
5.7 Overall average of RMSE and MAPE ranking of traditional models and proposed model (MvS CNN-BiLSTM).		52
xi List of Abbreviations	PM 2.5 Particulate Matter less then 2.5 mm AQI Air Quality Index ML Machine Learning DL Deep Learning SVM Support Vector Machine ARIMA Autoregressive Integrated Moving Average LR Linear Regression ANN Artificial Neural Network FL Fuzzy Logic LSTM Long Short Term Memory GRU Gated Recurrent Units CNN Convolutional Neural Network MVMT Multi-view multi-task RNN Recurrent Neural Network BiLSTM Bi-directional Long Short Term Memory MvS CNN-BiLSTM Multi-view Stacked CNN-BiLSTM RMSE Root Mean Square Error MAPE Mean Absolute Percentage Error MAE Mean Absolute Error CPCB Central Pollution Control Board	
xii List of Symbols	F Filter b Bias s Activation Function Multiplication represented W Weight t Time Step x i Missing Value D Dataset D c Dataset chunk D s No. of Data point available in chunk X Univariate time series X v v th -view of univariate time series L Lag L + lowest Lowest Positive Lag X T Tread X S Seasonal X R Reminder r L Auto correlation function	
xiii Dedicated to	Maa, and Papajee	
1 Chapter 1 Introduction	1.1 Introduction Air pollution has become a major global problem due to industrialisation and urbanisation. The rising levels of air pollutants, such as CO, SO ₂ , O ₃ , PM ₁₀ and PM _{2.5} . It has led to environmental issues like soil acidification, fog, haze and severe health problems such as heart attacks and lung diseases. The World Health Organization has revealed that contaminated air affects most of the global population, approximately 90% [1]. PM _{2.5} , or 2.5 micrometres or less aerodynamic diameter particulate matter, is an essential factor in calculating the Air Quality Index (AQI). AQI is a numerical scale used to communicate how polluted the air is and its potential health effects to the public. PM _{2.5} is a crucial air pollutant that can infiltrate the respiratory system and have detrimental health consequences. Time series data is a sequential data type collected regularly, with time as the index. It involves examining trends and patterns, with stationarity important, implying statistical properties' constancy over time. Forecasting based on historical patterns finds widespread applications in various fields, providing valuable decision-making and predictive modelling insights. Three principal methods, namely deterministic, statistical, and Machine learning (ML)/Deep Learning (DL), are widely utilised in predicting air quality. Deterministic methods simulate atmospheric chemistry's dispersion and transport processes, but they can be computationally expensive and less accurate due to limited actual	

Chapter 1. Introduction 2 observations. Statistical methods rely on historical data to forecast pollutant concentrations, but their linear assumptions may limit prediction performance. Researchers are incorporating non-linear machine learning models to surpass these constraints as alternative methods for predicting air quality. Machine learning and Deep Learning models such as Support Vector Machines (SVM) [2], Autoregressive-Integrated Moving Average (ARIMA) [3], Linear Regression (LR) [4], Artificial Neural Networks (ANNs) [5] and Fuzzy Logic (FL) [6] have been applied in air quality prediction studies. ANNs have been particularly popular, showing promising results in various applications. However, the rapid development of deep learning techniques has outperformed traditional ML models. Deep learning models, such as Long Short-Term Memory (LSTM) [7], Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN) [8], have shown improved prediction performance by capturing long-term dependencies and spatial features in air quality data. Multi-view learning [9], [10] has emerged as a potent methodology in machine learning and deep learning. It effectively utilises multiple perspectives or representations of data to enhance predictive performance, improve generalisation, and address intricate real-world problems. The technique has garnered significant attention due to its ability to manage varied and complementary information from multiple sources or modalities. This approach holds promise in its application and usability for ML/DL models across different domains, including Text and Image Analysis [11]–[14], Audio and Video Processing [13]–[16], and Environmental Monitoring [17]. Combining multi-view incorporation and hybrid deep learning models is a powerful technique in modern machine learning. This methodology improves predictive accuracy and feature extraction by integrating different data perspectives and utilising diverse neural network architectures. In time series analysis, the potential of multi-view learning has been shown by [18], which has utilised multivariate heterogeneous features as multiple views such as geographic location and time of day. The author has compared the proposed multi-view multi-task (MVMT) model with a single-view dataset setting and shows better performance as an outcome. In another research [19], a multi-view learning framework has been deployed for adaptive transfer learning to show the effectiveness of the inter-view usability of information transfer. In evaluating the proposed

Chapter 1. Introduction 3 model, the time series classification task has been performed to get the generalised evaluations. Several multi-view learning for time series approaches corresponding to deep learning with time series [20], [21], machine learning with time series [21], transfer learning [18], [20], etc. It has been observed from the literature that no research has been conducted yet based on a univariate time series dataset. Therefore, the potential of multi-view learning over univariate data may have an opportunity to perform time series prediction effectively. In this research, the author has proposed a hybrid multi-views stacked CNN-BiLSTM model framework. The hybrid CNN-BiLSTM architecture has been utilised to build the two-stack network. The seasonal characteristics of the univariate data have been utilised to generate the views corresponding to the required number of views (called partitions of the data). Then, a stacked CNN-BiLSTM model was deployed over each data view for the predictions. After this, predictions of view models are ensembled to get the final prediction based on their validation performance at each view. The proposed model has been evaluated over seventeen univariate datasets of PM 2.5 pollutants and compared their effectiveness based on various performance measures.

4 Chapter 2 Literature Review 2.1 Literature Review The authors used different deep learning architectures like GNN-LSTM Fully Connected (FC) network [22], Wavelet, ANFIS, PSO [23], LSTM Deep Feedforward Neural Network [24], Parallel multi-input 1DCNN-biLSTM [25], LGB algorithm [26], GOCI-based model, MAIAC-based model [27], MTCAN model [28], RNN [29], LSTM16 [30], CNN-LSTM [31], Conv LSTM [25], TL-BLSTM [32], CNN+LSTM [33], LSTM NN extended (LSTME) [34]. Autoencoder-based LSTM to predict air pollutant concentrations. They analysed and compared the performance of these models concerning traditional statistical methods and evaluated the impact of exogenous variables on the model's performance. The state-of-art models are compared to their proposed model with traditional DL models. Despite the advancement, it is lucid that additional comprehensive and varied datasets are mandatory to refine the acuteness of profound learning models. Certain studies' inability to incorporate exogenous variables limits the models' effectiveness in capturing external factors that could impact air quality. Future studies should tackle these deficiencies and investigate the potential of profound learning models for anticipating different air toxins and meteorological information in various urban improvement situations [28]. The research studies on air pollution forecasting using various deep learning models conducted by different authors have been summarised in the Table 2.1. The studies were executed in assorted regions worldwide and at different times. This

Chapter 2. Literature Review 5 investigation evaluated function measurements including Root Mean Squared Error (RMSE) [24], [26], [28]–[37], Mean Absolute Error (MAE) [22], [24], [32], [34]–[37], Mean Absolute Percentage Error (MAPE) [32], [34], and R-squared (R^2) values [24], [26], [27], [35], [38]. Different information sources were utilised, such as the US EPA [22], CPCB India [23], [28], [36], Ministry of the Environment, Ministry of Environmental Protection, ground-based observation stations, and air quality monitoring stations. It can be absorbed that RMSE & MAPE are the frequently used performance measures. The present study provides a summary (Table 2.1) of various investigations conducted on air quality based on different data sources employed for their models. These data sources are widely diverse, covering regions such as North America, Delhi (India), Santiago Chile, Shanghai China, Washington US, Sichuan Basin, Beijing (China); and others. The time intervals for data collection employed by these studies varied greatly, ranging from hourly to daily to 15-minute intervals. The data used in these investigations is predominantly collected from environmental monitoring agencies, government agencies, or ground-based observation stations. The broad range of data sources and collection intervals highlights the worldwide scope of air quality research and the plethora of techniques used to collect data. Moreover, the analyses have additionally brought to light the potential function of metropolitan woodlands in reducing PM 2.5 [39] concentration. It was found that AOD data could predict PM 2.5 concentration in resource-limited environments. The computation of air toxin levels affected by the Covid-19 crisis was also studied. Some research has delved into the efficacy of particular deep-learning model pairings in forecasting air pollutant concentrations [37]. The investigations have identified areas for improvement in air pollution forecasting, including enhancing data precision, considering alternative contaminants, and integrating dynamic parameters, as well as addressing issues related to computational cost, resource intensity, spatial resolution, and short-term prediction capability, with emphasis on the significance of continuous data and the effectiveness of LSTM models in capturing synoptic patterns [40]. Literature [13], [14], [41] of multi-views capability learning in various real-life

Chapter 2. Literature Review 6 tasks like object detections, image processing, and signal processing. The effectiveness of multi-view learning has been identified by [18] for multi-view multi-tasks with multivariate (MVMT) time series data, where the data was collected from distinct sources along with time stamps. The author has evaluated the proposed model against stand-alone DL models and showed the effectiveness of the MVMT model. Another researcher has utilised the multi-view learning method for adaptive transfer learning for time series data [19]. The generalised performance has been achieved for time series classification tasks with the help of a multi-view learning approach. In [42], the author proposed the calibrated multi-view time series forecasting model for multiple modality and structures data, which utilises the integration of the knowledge and their uncertainties in a dynamic environment. Compared with probabilistic forecasting models, it showed better performance by 25% (accuracy). The research gaps vary among the studies, reflecting areas where further investigation or improvement is needed. Some common research gaps include incorporating additional variables or pollutants into the models, exploring new data preprocessing techniques, enhancing historical data availability, improving model accuracy, and refining the selection of high-resolution variables. Other research gaps involve incorporating more comprehensive meteorological and traffic data, applying deep learning techniques, and addressing the potential for enhancing model performance by including exogenous variables. These research gaps highlight the ongoing efforts to enhance air quality forecasting models and provide valuable insights for future research directions.

Chapter 2. Literature Review 7 TABLE 2.1: summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap. Ref. Data Performance Measures Model Research Gap [22] North America (Jan 21 -Sep 21) Hourly Data collected from US EPA MAE: 2.81 GNN-LSTM Fully Connected (FC) network Adding more variables and new data preprocessing technique for enhancing model. [23] Delhi India (18 -21) Daily Data collected from CPCB India. Correlation coefficients: [0.96, 0.98] (1 day), [0.86, 0.93] (2 days), [0.82, 0.91] (3 days) Wavelet, ANFIS, PSO Adding more historical data for enhance model. [24] Santiago Chile (05 - 19) Hourly Data provided by Ministry of the Environment RMSE 3.88, MAE 2.52, R^2 0.94 LSTM Deep Feedforward Neural Network Add more pollutants for enhance the model. [35] Shanghai china (2014-05-13 to 2020- 12-31) Hourly Data provided by Ministry of Environmental Protection RMSE 3.88, MAE 2.52, R^2 0.94 Parallel multi-input 1D-CNN- biLSTM model add factory data , creating a Smartphone Application for PM2.5 forecasting. [43] Delhi India (1 Jan 2018- 30 Nov 2019) 15 min interval data collected from CPCB India. R^2 = 0.75, RMSE = 25.13, MAE : 21.28 Cluster-based Graph Neural Network (SA- GNN) Add activation function, add more historical data

Chapter 2. Literature Review 8 TABLE 2.1: summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap. Ref. Data Performance Measures Model Research Gap [44] Washington, US (1st January to 31st January 2017) hourly data collected from US (EPA) RMSE = 0.043 Geo-LSTM Add more Data with more futures of Data. [40] Sichuan Basin (January 1, 2019, to December 31, 2019) Hourly data collected from China Environmental Monitoring Center $R^2 = 0.917$, RMSE = 7.4 data-driven spatial auto-correlation terms (DDW-RF) necessary to select more appropriate high-resolution variables. [45] Beijing, Tianjin, Dalian, and Yantai (1600 hour) China MAPE = 6.0819, RMSE = 11.8654, $R^2 = 0.9754$ multi-objective optimisation algorithm Add more pollutants, improving accuracy [46] shaanxi province (January 1, 2016, to December 31, 2020) daily data RMSE = 0.3997, MAPE = 0.14599, MAE = 0.2871 GBoost-MLP based on GARCH model apply deep learning model [47] India (January 2016 to December 2018) 15-minute interval RMSE = 19.89, 25.88 , $R^2 = 0.96$, 0.9 lstm apply more pollutants [26] Seoul South Korea (July 2018 to June 2021) Deliy Data bias = -0.25% to -0.10%, RMSE = 32.45%-33.23%, $R^2 = 0.83$ -0.86 LGB algorithm Add more data for enhance the model. [27] Seoul Korea (16 -19) Hourly Data collected from ground-based observation stations R^2 values: 0.61 and 0.78 GOCI-based model, MAIAC-based model Add more pollutants and Data for enhance the model.

Chapter 2. Literature Review 9 TABLE 2.1: summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap. Ref. Data Performance Measures Model Research Gap [28] Talcher India (02/02/2018 to 04/07/2020) per 15 min Data collected from CPCB india. RMSE values: 93% and 90% better than GRU MTCAN model add meteorological factors and traffic data for the enhanced model. [29] Sakarya Urbanization (01.08.2018 and 31.07.2020) Daliy Data provided from e Ministry of Environment RMSE: 2.8414.09 RNN Add more pollutants and Data for enhance model. [30] Istanbul Basaksehir (01.01.2021 and 09.02.2022) Hourly data taken from Ministry of Environment RMSE: 10.229478 LSTM16 add more pollutants and meteorological Data for enhance model. data. [31] Ulaanbaatar Mongolia (June 1, 2018, to April 30, 2020) Hourly Data from U.S. Embassy in Mongolia RMSE: 11.77 CNN-LSTM Add more Data and atmospheric Data for enhance the model. [38] Istanbul (15-19) Hourly Data collected from Kathane air quality monitoring station $R^2 = 0.98$ LSTM+LSTM Add more pollutants and Data for enhance the model. [25] Italian city (March 2004 to February 2005) Hourly Data collected from Kaggle 91% Prediction accuracy Conv.LSTM Deploy deep leading models for classification.

Chapter 2. Literature Review 10 TABLE 2.1: summary of recent state-of-art based on Data, Model proposed, Performance measures and Research gap. Ref. Data Performance Measures Model Research Gap [32] Guangdong China (3 years) Hourly Data collected from Guangdong province. RMSE: 8.652, MAE: 6.184, MAPE: 27.909 TL-BLSTM utilising transfer learning techniques and adding more data to enhance the model. [33] Shanghi (2015-2017) Daliy Data collected manually RMSE: 14.3 CNN+LSTM Add more data for enhance model. [34] Beijing China (Jan 2014 -may 2016) Hourly Data collected from Ministry of environmental protection. RMSE: 12.6, MAE: 5.46, MAPE: 11.93 LSTM NN extended(LSTM) Adding more pollutants to enhance the model. [37] Beijing China (01/01/2010-01/31/2010) Hourly Data collected from0 uci. RMSE: 77.38, MAE: 54.58 Deep Air Quality Forecasting Using Hybrid Deep Learning Framework Add more Data for model enhancing. [36] Kolkata India (Jan 16 - Feb 20) Daily Data collected from CPCB India. RMSE: 18.8, MAE: 15.88 Autoencoder based LSTM include exogenous variables for enhance the model.

11 Chapter 3 Basics Related Concepts 3.1 Deep learning models 3.1.1 Convolutional Neural Network (CNN) A CNN is an essential neural network utilised extensively for analysing time series and processing signals. Unlike standard fully connected neural networks, which process input as vectors, 1D CNNs extract local features using convolution operations using a sliding window technique. They're made to operate with one-dimensional signals like audio or time series sensor readings. In Figure 3.1 [48] 1D CNN, multiple filters are employed to extract various features from the input signal. These filters slide over the input, capturing local patterns and representations. The convolutional layer output is then downsampled using pooling layers, reducing the data dimension and preventing overfitting. The network may also include fully connected layers that perform tasks like classification or regression using the retrieved features. The simplicity of the 1D CNN [49] architecture lies in its effectiveness in extracting features from one-dimensional signals. The sliding window approach allows the network to focus on local details and extract relevant information effectively. Furthermore, the convolutional process reduces the number of parameters, resulting in a more computationally efficient network. Pooling layers help in generalisation by decreasing output complexity, resulting in higher performance on previously unknown data. 1D CNNs are versatile and practical in various domains since they can examine historical data and extract relevant characteristics. They've shown to

Chapter 3. Basics Related Concepts 12 be incredibly effective in applications such as speech recognition [50], [51], audio categorisation [52], [53], and sensor data processing [54], [55]. Overall, 1D CNNs are potent tools for collecting features from one-dimensional data, providing essential insights, and paving the way for signal processing and time series analysis advances. FIGURE 3.1: Traditional architecture of 1D CNN 1D CNN uses a set of learnable filters (or kernels) of length k to conduct the convolution operation. This operation involves computing the dot product between the filter and a k -length window sliding over the input sequence x with length L . Then a new sequence of feature maps is generated as an outcome.

Chapter 3. Basics Related Concepts 13 Denoting the set of filters as F and the output feature map at position i as h_i , we can express the output feature map h as follows: $h_i = s(\sum_{j=1}^k F_j \cdot x_{i+j-1} + b)$ (3.1) Where F_j represents the j th filter in the set of filters F , x_{i+j-1} is the value of the input sequence x at position $(i + j - 1)$, s is the activation function (such as ReLU or sigmoid), and b is the bias term. 3.1.2 Gated Recurrent Unit (GRU) The GRU is a specialised recurrent neural network architecture designed for managing sequential data input [56], and the unique proposal was put forth as a substitute for the well-liked LSTM network. These gating mechanisms selectively update the hidden state, allowing GRU to capture dependencies in sequential data effectively. GRU has found use in an array of areas, including speech recognition [57], [58], natural language processing [59], [60], and picture recognition [61], due to its gating mechanisms and ability to handle sequential dependencies. Its capabilities make it a powerful tool for modelling and understanding sequential data, providing valuable insights and improved performance in numerous tasks. GRU network based on gates and states are shown in Equation 3.2 to Equation 3.5: Update Gate : $z_t = s(W_z [h_{t-1}, x_t] + b_z)$ (3.2) Reset Gate : $r_t = s(W_r [h_{t-1}, x_t] + b_r)$ (3.3) Candidate activation : $\tilde{h}_t = \tanh(W_h [r_t \odot h_{t-1}, x_t] + b_h)$ (3.4) Hidden State : $h_t = (1 \odot z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$ (3.5)

Chapter 3. Basics Related Concepts 14 where, time step t , the hidden state is represented by h_t , the input is represented by x_t , and the update and reset gates are represented by z_t and r_t respectively. The refresh entryway directs the amount of the past covered state to keep for the present time step, while the reset entryway controls the amount of the past covered state to overlook. The candidate activation \tilde{h}_t represents new information that could be added to the hidden state. The sigmoid activation function s and element-wise multiplication represented by \odot are used. Weight matrices W_z , W_r , W_h and bias vectors b_z , b_r , b_h are also utilized. The GRU's update and reset gates allow it to learn when to update the hidden state and what information to forget, making it particularly useful for modelling sequential data with long-range dependencies. 3.1.3 Recurrent Neural Network (RNN) RNN is an ANN that utilises the outcome of the preceding measure to contribute to the present step. Forecasting the succeeding phrase in a statement is not a strong suit of RNNs because their Memory State, also recognised as the hidden layer, does not preserve any data about preceding words. It stores the previous input given to the network, which goes a long way in ensuring accurate predictions. The RNN technique employs identical parameters for every input, leading to it executing the same function on every hidden layer to obtain the results. Unlike other neural networks, it significantly reduces the complexity of parameters, making it a popular choice among researchers and developers. The elegance of RNN rests in its capacity to recall prior inputs, rendering it a valuable instrument in creating predictions that demand context. Ordinarily, profound learning has been exhibited to be a game-changer in AI and ML, and it will undoubtedly persist in being an essential instrument in the coming years. Recurrent Neural Network (RNN) model are input & output to hidden state can be written as Equation 3.6 & Equation 3.7 respectively: Input to Hidden State: $h_t = y(W_{hx} x_t + W_{hh} h_{t-1} + b_h)$ (3.6)

Chapter 3. Basics Related Concepts 15 Hidden State to Output: $y_t = W_{yh} h_t + b_y$ (3.7) where h_t is the hidden state at time step t , x_t is input at t step of time, y_t is output at t step of time, W_{hx} is a weight matrix that links the input to the concealed state, W_{hh} is the weighted matrix that connects the state that is hidden at time step $t - 1$ with the hidden value step t , W_{yh} is a weighted matrix that links the state that is hidden to the output, b_h and b_y are biased terms for both the state that is hidden and the output and $y(\cdot)$ is a function of activation applied to the concealed state element by element. 3.1.4 Long-Short-Term Memory (LSTM) LSTMs are a specialised type of RNN designed to handle sequential data. They excel at learning long-term dependencies, making them suitable for language translation, speech recognition, and time series forecasting. Utilising the memory cell and three gates facilitates the capability of LSTMs to learn intricate patterns in data by selectively retaining and discarding information. Deep LSTM networks, achieved by stacking LSTMs, are beneficial for tasks like speech recognition [62], [63] and natural language processing [64], [65]. Hochreiter and Schmidhuber [66] developed LSTMs to overcome the long-term dependency issue in traditional RNNs. LSTMs are widely used in processing [67], prediction [68], and classification [69], [70] of temporal data, and when combined with CNNs, they efficiently analyse images [71]–[73] and videos [74]–[77] by extracting spatial and temporal features. LSTMs are an effective instrument for analysing sequential data and can be incorporated with other neural network structures to accomplish more complex objectives. The LSTM architecture-related gates and states are shown in Equation 3.8 to Equation 3.13. Input Gate : $i_t = s(W_{xi} x_t + W_{hi} h_{t-1} + b_i)$ (3.8) Forget Gate : $f_t = s(W_{xf} x_t + W_{hf} h_{t-1} + b_f)$ (3.9)

Chapter 3. Basics Related Concepts 16 Candidate Hidden State : $g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$ (3.10) Cell State : $C_t = f_t C_{t-1} + i_t g_t$ (3.11) Output Gate : $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$ (3.12) Hidden State : $h_t = o_t \tanh(C_t)$ (3.13) where, h_t at a time step, is the concealed condition (also known as a hidden representation), x_t is the time step input, C_t is the condition of the cell at time step t , serving as the LSTM's long-term memory, σ Its sigmoid activation function, The tangent hyperbolic activation function is denoted by \tanh , represents element-by-element multiplication, W_{xi} , W_{hi} , W_{xj} , W_{hf} , W_{xg} , W_{hg} , W_{xo} , W_{ho} are the matrices of that will be taught throughout training and b_i , b_f , b_g , b_o are slanted terms. 3.1.5 Bi-directional Long Short Term Memory (BiLSTM) BiLSTM is a type of human stupidity designed to process sequential data in neither forward nor backward directions. The plan uses dual LSTM layers for forward and backward processing concurrently, enabling it to grasp the context from previous and upcoming time steps. Therefore, the network cannot identify any relationships in either direction, making it a disadvantage for speech recognition and language translation tasks. By providing a broader perspective of the input sequence, BiLSTMs can improve the performance in tasks requiring contextual understanding. These forms are remarkable for long-term dependencies and are commonly utilised in profound learning projects. BiLSTM network architecture gates and states are shown below from Equation 3.14 to Equation 3.19.

Chapter 3. Basics Related Concepts 17 FIGURE 3.2: Architecture of BiLSTM [78] • Forward LSTM: Input Gate : $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$ (3.14) Forget Gate : $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$ (3.15) Candidate Hidden State : $g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$ (3.16) Cell State : $c_t = f_t c_{t-1} + i_t g_t$ (3.17) Output Gate : $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$ (3.18) Hidden State : $h_t = o_t \tanh(c_t)$ (3.19) • Backward LSTM: The formulas for the rear LSTM are similar to those of the

Chapter 3. Basics Related Concepts 18 forward LSTM but with different weights and biases. The superscript "b" is used to denote the backward direction (e.g., W_{bxi} is the weight matrix for the input gate in the backward LSTM). The BiLSTM combines the hidden forward and backwards states to form the final output. The output y_t at time step t is typically computed using a combination of both forward and backwards hidden states. The approach for combining the two directions (e.g., concatenation, addition, etc.) depends on the specific task and model architecture. 3.2 Performance Measures 3.2.1 Root Mean Square Error (RMSE) The RMSE is hardly ever utilised as a standard for assessing the precision of a regression model. The computation of the root mean square of discrepancies between projected and factual figures can facilitate identifying the divergence between them while keeping the units of measurement consistent with the data. This provides valuable insight into the model's effectiveness. A lower RMSE digit suggests superior model performance, while an increased RMSE value indicates substandard model performance. The RMSE Equation 3.20 uses metrics for regression models by effectively evaluating the model's accuracy, as it quantifies the discrepancy between projected and actual values in the same unit as the data. By analysing the RMSE score, experts can evaluate the model's efficacy and implement necessary tweaks to enhance its precision. As a final point, the RMSE is a crucial statistic for evaluating the accuracy of regression models, and its usage is essential in industries that heavily depend on data science, finance, and engineering. $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ (3.20) where n represents the number of times that the summation iteration occurs, y_i denotes the actual value, and \hat{y}_i represents the forecast value.

Chapter 3. Basics Related Concepts 19 Mean Absolute Percentage Error (MAPE) A popular metric for judging the efficiency of regression models is MAPE. It measures the percentage of variance between predicted and actual results and averages these discrepancies. MAPE can be computed by finding the average of the absolute percentage errors between the expected and actual values, such as Equation 3.21. $MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}$ (3.21) where n represents the number of times that the summation iteration occurs, y_i denotes the Actual value, and \hat{y}_i represents the forecast value. MAPE determines the average percentage discrepancy between the projected and observed values. In business and finance, it is a common practice to assess the precision of forecasts or predictions using this technique. Industries utilise MAPE to compute the mean percentage deviation between anticipated and actual values, where a lower value suggests better model performance and a higher value suggests poorer performance. However, MAPE can cause division by zero errors or huge percentage errors when the actual numbers are close to zero.

20 Chapter 4 Methodology 4.1 Data: The dataset in this study was collected from the Indian government portal CPCB. These datasets contain univariate Time series hourly data of 17 Indian cities, with most cities providing approximately four years (see in Table 4.1). The table lists Bhiwadi, Jodhpur, Singrauli, Ankleshwar, Ludhiana, Durgapur, Yamuna Nagar, Charkhi Dadri, Jind, Kurukshetra, Sonipat, Dharuhera, Ambala, Hisar, Fatehabad, Bulandshahr, and Muzaffarnagar as the 17 cities from which data was collected shown on (Figure 4.1) with Name, Latitude and Longitude. The Table 4.1 also provides information about the number of data points available in datasets, the highest number of data points in Jodhpur, and the lowest number of data points in Durgapur. In Table 4.1, All the datasets are thoroughly analysed based on several components such as Count, Min, Mean, Std, 25%, Max, 75%, and 50%. The detailed description of the analysis is then summarised for ease of understanding.

Chapter 4. Methodology 21 TABLE 4.1: Statistical exploratory data analyses (EDA) of 17 Time Series Datasets of polluted Indian cities based on PM 2.5 [79].

D.No.	DataSet	Year	Sam-	ples	Mean	Std	Min	25%	50%	75%	Max
D1	BHIWADI	2017-2022	43394	108.03	79.76	0.02	55.22	97.32	135.36	999.99	
D2	JODHPUR	2015-2022	61409	84.31	56.18	0.18	53.25	84.31	93.42	999.99	
D3	SINGRAULI	2017-2022	43695	84.08	78.33	0.25	32.25	66	111.25	985	
D4	ANKLESHWAR	2019-2022	33535	58.47	35.83	0.51	32.75	58.47	72.24	977.39	
D5	LUDHIANA	2017-2022	49010	54.18	41.73	0.07	29.7	47.66	64.88	999.99	
D6	DURGAPUR	2020-2022	17434	71.67	46.20	0.33	37.47	62.05	98.03	565.41	
D7	YAMUNA_NAGAR	2019-2022	34299	77.86	52.31	0.1	43.8	69.91	94.28	930	
D8	CHARKHI_DADRI	2020-2022	24099	80.19	62.81	0.01	39.54	77.92	94.49	995.1	
D9	JIND	2019-2022	34145	81.21	71.20	0.2	38.99	61.45	98.25	845.6	
D10	KURUKSHETRA	2019-2022	34208	68.75	53.80	0.46	33.33	56.38	87.56	962.7	
D11	SONIPAT	2019-2022	34362	54.88	43.21	0.02	27.87	49.4	62.72	543.1	
D12	DHARUHERA	2019-2022	34265	78.86	59.21	0.02	40.9	70.32	92.85	838.9	
D13	AMBALA	2019-2022	34174	61.58	45.39	0.02	32.94	51.27	76.18	754.89	
D14	HISAR	2019-2022	34143	86.22	71.02	0.63	42.62	69.33	102.89	999.99	
D15	FATEHABAD	2019-2022	34160	63.01	60.46	0.07	32.63	49.01	72.5	999.99	
D16	BULANDSHAHR	2018-2022	39869	90.53	85.08	0.25	34	63.75	120.25	985	
D17	MUZAFFARNAGAR	2018-2022	38786	89.29	72.84	1	42.75	81.25	102.25	986	

Chapter 4. Methodology 22 FIGURE 4.1: Geographical representation of polluted cities of India with their Latitude, Longitude and Name. • Preprocessing: Data preprocessing is critical in the ML process as it ensures the data is clean, consistent, and ready for analysis. Techniques like missing

Chapter 4. Methodology 23 value imputation and min-max scaling facilitate data normalisation and im- prove the effectiveness of machine learning algorithms by allowing them to learn from the data and make accurate predictions. During the initial data preprocessing stage, missing values are replaced with the means. The dataset is divided into ten parts or chunks, and the mean is calculated for each chunk. Subsequently, the means of all the chunks are averaged together, as shown in Equation 4.1: $x_i = \frac{1}{D_c} \sum_{k=1}^{D_s} D_{c,i,k}$ (4.1) where $k = 10$ & x_i is missing value in time series and D_c represent chunks of Dataset. D_s is the number of available samples in the dataset chunks represented by D_c . Once the missing values have been attributed in the dataset using Equation 4.1, the next step is to apply min-max scaling Equation 4.2. This technique ensures that univariate data is scaled to fall within the range of $[0, 1]$. The scaling is achieved Equation 4.2: $x_{norm,i} = \frac{x_i - \min(x)}{\max(x) - \min(x)}$ (4.2) Where x_i is the i th data point and m is the mean of univariate data $\min(x)$ and $\max(x)$ denote the minimum values and maximum values in the univariate series.

Consistency is essential for comparing data across different datasets. Additionally, the scaling technique reduces the influence of outliers and enhances the data's robustness. • Decomposition and analysis: In Figures 4.2 to 4.4, all subplot has been plotted on 480 recent data points. Figures 4.2 to 4.4 presents the graphical representation of the dataset, showcasing different categories like trend & seasonal. In rows 1, and 4 the data is labelled as "Original." Rows 2, and 5 correspond to the "Seasonal" category, while rows 3, and 6 represent the "Trend" category. The last two row combines all three categories, displaying "Original," "Seasonal," and "Trend" data in a single dataset. Each subplot Chapter 4. Methodology 24 within the graph has a specific dataset name, indicated in the legend. The graph consists of three columns, each corresponding to a specific range of rows (1 to 3, and 3 to 6) from the dataset. The border colour remains consistent across these columns, indicating they belong to the same dataset. The legend provides clarity by associating the dataset name with its respective subplot, facilitating a comprehensive understanding of the data distribution and categories. Figures 4.2 to 4.4 illustrates the seasonal variations in the dataset with a frequency of every 24 hours. Each subplot within the graph contains 20 data points representing the seasonal pattern.

Additionally, the graph showcases various types of trends present in the data. These trends may include upward, downward, or fluctuating trends over time. The combination of seasonal variations and diverse trends provides valuable insights into the underlying patterns and behaviours of the dataset. 4.2 Method This research section will delve into the methodology of predicting PM 2.5 using various deep-learning models. Our data was obtained from the CPCB (Central Pollution Control Board), India [79] hourly univariate data collection. It focused on a single future PM 2.5 in this dataset. Then we processed the obtained data and implemented several deep-learning models to generate our forecasts. The primary objective of employing these is to enhance the model's accuracy of our predictions. 4.2.1 Model Development Let's $X = [x_1, x_2, x_3, \dots, x_m]$ is a univariate time series (see the Definition 4.2.1), where i th data point $x_i \in \mathbb{R}$ and m is the length of the time series which follow the ergodic properties (see the Definition 4.2.4). It means time series X should not have the following:

Chapter 4. Methodology 25 FIGURE 4.2: Decomposition of 17 data sets based on trend, seasonal, and original dataset.

Chapter 4. Methodology 26 FIGURE 4.3: Continuation of decomposition of 17 data sets based on trend, seasonal, and original dataset.

Chapter 4. Methodology 27 FIGURE 4.4: Continuation of decomposition of 17 data sets based on trend, seasonal, and original dataset.

Chapter 4. Methodology 28 • High-variability : The cowehy-distributed i.i.d X has $\bar{x} = d \times 1$ which means, high variability of marginal distribution function. • Lock of stability : The X distribution changes too much for marginal distribution functions such as variance approaches to infinity. • Absorbing state : The range of probability measure varies from 0 to 1. It is also assumed that the univariate time series X is at least weakly stationary or strictly stationary.

Definition 4.2.1 (Univariate time series). "A univariate time series $X = [x_1, x_2, \dots, x_m]$, $X \in \mathbb{R}^m$ is an ordered sequence of single dimensional vector, where x_i denotes the i th time stamp observation and length of series is m ." Definition 4.2.2 (Multi-view univariate time series). "A multi-view univariate time series is a set of a subset $V_k = \{x^{(k)}_v\}_{v=1}^m$ of univariate time series $X \in \mathbb{R}^m$, where (k total number of view) v th view from X is observed as $X^{(k)}_v \in \mathbb{R}^{m_v}$, $m_v \in \mathbb{N}$; $m, jX_j = \sum_{v=1}^K X^{(v)}_j$ and view univariate series denoted as $X^{(v)} = [x^{(v)}_1, x^{(v)}_2, \dots, x^{(v)}_{m_v}]$." Definition 4.2.3 (v th -Views of univariate time series). "A v th -view $X^{(v)} = [x^{(v)}_1, x^{(v)}_2, \dots, x^{(v)}_{m_v}]$ of univariate of time series X is $X^{(v)} \in \mathbb{R}^{m_v}$ & $X \in \mathbb{R}^m$, where $m_v \in \mathbb{N}$; m , and $m \geq m_v$ is the length of time series X and $X^{(v)}$." Definition 4.2.4 (Ergodic property of univariate time series). "A time series has the long-term time mean of a process is equivalent to the mean of all possible realisation of the time series." (a) Measuring strength of seasonality by decomposition of series X : the decomposition of time series X can be represented as $X = X_T + X_S + X_R$ (4.3) where Trend component : X_T , Seasonal component : X_S and Remainder component : X_R are denoted respectively. The strength of seasonality F_S can be defined as Equation 4.4:

Chapter 4. Methodology 29 $F_S = \max(0, 1 - \frac{\text{Var}(X_R)}{\text{Var}(X_S + X_R)})$ (4.4) Where $\text{Var}(X_R)$ and $\text{Var}(X_S + X_R)$ are the variance of Remainder and de-trended data over seasonal and Remainder, respectively and F_S close to 0 and 1 exhibits no seasonality and strong seasonality. (b) Identifying seasonal for lowest lag L : The coefficient of Auto Correlation Function (ACF) for lag L for time series X can be obtained as in Equation 4.5: $r_L = \frac{E((x_t - \bar{x})(x_{t+L} - \bar{x}))}{s^2}$ (4.5) where $\bar{x} = E(X_t)$ is constant mean and $s^2 = \frac{1}{m} \sum_{t=1}^m (X_t - \bar{x})^2$, $\bar{x} = \frac{1}{m} \sum_{t=1}^m X_t$ is the sample mean the coefficients r_L of auto correlation function has range $[-1, +1]$, where close to +1 exhibits the strong positive lag. Let's consider r_L is a set of ACF coefficients of lag that has a positive coefficient, which can be denoted as $r_L = \{r_L\}_{L=1}^J$ (4.6) J : Total no. of ACF coefficients. Then the lowest lag value will indicate the smallest time stamp that has season behaviour in the series X , which can be identified as $L_{\text{lowest}} = \arg\min_L r_L$ (4.7) (c) Seasonality based view generation($X^{(k)} \in \mathbb{R}^{m_v}$) : Let's consider $T_{L_{\text{lowest}}}$ is the time stamp of the length of seasonal with lowest L_{lowest} lag; then the series X can be partitioned with seasonal repetition for L_{lowest} lags, where k - no. of partition can be obtained corresponding to L_{lowest} .

Chapter 4. Methodology 30 Figure Item (c) is the graphical representation of seasonal of series X corresponding to L_{lowest} lag. The length of time stamp of i th partition m_v (i.e. a view of X) can be obtained as Equation 4.8: $m_v = m - T_{L_{\text{lowest}}} + 1$ (4.8) where m is the length of time stamp of X , and $T_{L_{\text{lowest}}}$ is the length of time stamp of lowest lag. The total no. of possible partition ranges as $1, m - T_{L_{\text{lowest}}} + 1$ lets a view $X^{(k)} \in \mathbb{R}^{m_v}$ denote as : $X^{(k)}_v = [x^{(k)}_1, x^{(k)}_2, \dots, x^{(k)}_{m_v}]$ (4.9) for k - no. of partition with m_v length of series, is the v th view of the k - no. of view of X (see the Definition 4.2.3), where $X = [x^{(k)}_v]_{k=1}^K$ and $m = \sum_{k=1}^K m_v$, from Equation 4.8 (see the Definition 4.2.2). (d) Stacked CNN-BiLSTM : 1D CNN layer for forward propagation is expressed as Equation 4.10 : $x^{(k,v)}(i,l) = b^{(k,v)}(i,l) + L^{(l*1)} \hat{a}_{j=1}^{\text{conv1D}} W^{(k,v)}(j,l^{*1})$, $S^{(k,v)}(j,l^{*1})$ (4.10) $L^{(l*1)}$: Number of neurons weight at (l^{*1}) layer. $x^{(k,v)}(i,l)$: i th input at l -layer of v th view $X^{(k)}_v$ for partition k , i.e., $x^{(k,v)}(i,l) \in X^{(k)}_v \in \mathbb{R}^{m_v}$; $b^{(k,v)}(i,l)$: defined as bias at l -layer. $S^{(k,v)}(j,l^{*1})$: output of j th neurons at layer (l^{*1}) . $W^{(k,v)}(j,l^{*1})$: Kernel from the j th neuron at layer (l^{*1}) . Conv1D(.) : perform "in-valid" 1D convolution without zero padding.

Chapter 4. Methodology 31 And intermediate output $\hat{x}^{(k,v)}(i,l)$ can be written as Equation 4.11 : $\hat{x}^{(k,v)}(i,l) = f(x^{(k,v)}(i,l))$ (4.11) In back-propagation, the weight and bias are updated as follows : $W^{(k,v)}(i,l^{*1})(t+1) = W^{(k,v)}(i,l)(t) * \eta \nabla E \nabla W^{(k,v)}(i,l^{*1})$ (4.12) and $b^{(k,v)}(i,l)(t+1) = b^{(k,v)}(i,l)(t) * \eta \nabla E \nabla b^{(k,v)}(i,l)$ (4.13) where $\nabla E \nabla W^{(k,v)}(i,l^{*1}) = \text{conv1D}(S^{(k,v)}(i,l), D^{(k,v)}(i,l+1))$ and $\nabla E \nabla b^{(k,v)}(i,l) = \hat{a}_{j=1}^{\text{conv1D}} W^{(k,v)}(j,l^{*1})$. Then, the final output of the 1DCNN may be written as Equation 4.14. $\hat{x}^{(k,v)}(i,l) = F(\hat{x}^{(k,v)}(i,l))$ (4.14) where $F(\cdot)$ is activation function at last layer of the network. Now, the input gate of BiLSTM can have $\hat{x}^{(k,v)}(i)$ output of 1DCNN as input of BiLSTM shown in Equation 4.15. $i_f(t) = s W_f \hat{x}^{(k,v)}(i) + W_{fh} i_h(t^{*1}) + W_{fc} i_c(t^{*1}) + b_{fi}$ (4.15) Then, the final output gate can have output $q^{(k,v)}(t,1)$ for the first stack $\hat{y}^{(k,v)}(t) = q^{(k,v)}(t, \text{stack}) = s W_{\hat{x}} \hat{x}^{(k,v)}(i) + W_{ho} \hat{x}^{(k,v)}(i) + W_{hc} i_c(t) + b_{ho}$ (4.16)

Chapter 4. Methodology 32 The stacked CNN-BiLSTM include repeating the same architecture shown from Equation 4.10 to Equation 4.16. Then the second stack output $q(k, v)(t, 2)$ can be obtained as Equation 4.16 which will corresponding to v th view of x for k -partition. (e) Multi-view Stacked CNN-BiLSTM Prediction : The prediction of v th view $X \times 2 \times X$ for k -partitioning is $q(k, v)(t, 2)$, so weighted ensemble of all view output for time t is denoted as in Equation 4.17. $\hat{y}_k(t) = \sum_{v=1}^V w_v \hat{y}_k(v, t) w_v$ (4.17) where w_v is the weight of view, i.e., the performance of stacked CNN-BiLSTM over validation set of X over v th view $X \times v$. (f) Finding best views of multi-view for learning of stacked CNN-BiLSTM : Let a set of final prediction of k -partition of X is denoted as $\hat{y}_k(t, L + \text{lowest } k = L + \text{lowest and } M_k(\cdot))$ function to measure the performance of k -partition (for minimization). Then, the best views k best can be exhibits as Equation 4.18 $k_{\text{best}} = \text{argmin}_{t \in T} \sum_{k=L+\text{lowest}}^{L+\text{highest}} M_k(\hat{y}_k(t, o))$ (4.18) where $L + \text{lowest}$ and $L + \text{highest}$ are minimum and maximum of $r + L$ ACF coefficient. Proposed Multi-view Stacked CNN-BiLSTM (MvS CNN-BiLSTM) The model consists of various layers, starting with a 1D Convolutional layer with 128 filters, followed by a MaxPooling layer for downsampling. Next, a Bidirectional LSTM layer with 64 units and ReLU activation is applied, returning only the last output. A Reshape layer is introduced to reshape the output, and another 1D Convolutional layer with 32 filters is utilised, followed by another MaxPooling layer. Another Bidirectional LSTM layer with 32 units and ReLU activation is added, returning only the last output. To prevent overfitting, a Dropout layer with a rate

Chapter 4. Methodology 33 of 0.2 is introduced before the final Dense layer, which contains a single neuron for regression predictions. The model is given the name 'stacked CNN-BiLSTM'. Implamentation setup and parameters settings The various Python packages are utilised to implement the proposed and baseline models. These assortments incorporate Keras (v2.12.0), TensorFlow (v2.12.0) for Keras backend, scikit-learn (v1.2.2) for model building and performance analysis, Pandas (v1.5.3) & NumPy (v1.23.5) for exploratory data analysis, and matplotlib (v3.7.1), seaborn (v0.12.2) & Plotly (v5.14.1) for representing the results and plotting graphs. Additionally, It have utilized https://app.diagrams.net/ for creating flowcharts. Two unique systems were employed for the experiments: a Windows-based computer with an Intel® Core™ i7-11800H @ 2.30GHz 8 GB RAM and a Kali Linux machine with an Intel® Core™ i7-4600U @ 2.2GHz 16 GB RAM running JupyterLab (v4.0.1).

TABLE 4.2: Parameter setting of traditional DL models and proposed MvS CNN-BiLSTM Hyperparameters Values Batch Size 66 Optimizer Adam Loss function Mean Squared Error Epoch 250 with early stopping activation function ReLU training size 0.8

Chapter 4. Methodology 34 Algorithm 1: Generation of multiple views from univariate time series X . Input: $X = [x_1, x_2, x_3, \dots, x_m]$, where $x_i \in \mathbb{R}^m$ and follows the ergodic properties. $K \in \mathbb{N}^+$ // no. of partition of series X and ($v = 1, 2, 3, 4, \dots$) Output: $X_{k,v} = [x(1,v), x(2,v), x(3,v), \dots, x(m,v)]$, where $x_i \in X_{k,v} \& X_{k,v} \in X$, $m \> m // v$ th k -no. of partition of X series. 1 Initialization: $k \in \mathbb{N}^+ + 2$ indentifying seasonal for lower lag L using ACF coefficient $r_L : r_L = E(x(t-m)(x(t+m)-m)) / s^2$ // see Equation 4.5 3 Finding set of positive coefficient of ACF $r_L + : r_L + = fr_L \< 0; 0g$ jr all $j (L = 1)$ // see Equation 4.6 4 obtaining the lowest lag $L + \text{lowest} : L + \text{lowest} = \text{argmin}_{L \in \mathbb{R}^+} r_L + //$ see Equation 4.7 5 Generation of v th view $X_{k,v} \times 2 \times X$ for k -partition : $m_v = m / L + \text{lowest } k //$ see Equation 4.8 6 $X_{k,v} = [x(1,v), x(2,v), x(3,v), \dots, x(m_v,v)]$ 7 Return ($X_{k,v}$) 8 End

Chapter 4. Methodology 35 FIGURE 4.5: Flow diagram of proposed Multi-view Stacked CNN- BiLSTM (MvS CNN-BiLSTM)

Chapter 4. Methodology 36 FIGURE 4.6: Sesisonality based view $X \times 2 \times X$ generation. FIGURE 4.7: Architecture of Stacked CNN-BiLSTM

Chapter 4. Methodology 37 Algorithm 2: Stacked CNN-BiLSTM Model. Input: $X_{k,v} = [x(1,v), x(2,v), x(3,v), \dots, x(m_v,v)]$ // from Algorithm 1 Output: $\hat{y}_k(v, 0)$ // output at second stack of CNN-BiLSTM 1 Initialization: stack = 1 2 if stack 2 then 3 1DCNN layer for forward propagation: $x(k, v)(i, l) = b(k, v)(i, l) + \sum_{j=1}^L \text{conv1D } W(k, v)(j, (l-1)) \cdot S(k, v)(j, (l-1))$ // see Equation 4.10 4 outcome of 1DCNN from last layer using activation function $F(\cdot) : \hat{x}(k, v)(i, l) = F(\hat{x}(k, v)(i, l))$ // see Equation 4.14 5 The output of the previous step-2 is the input of the BiLSTM input Gate (forward) : $if_t = s W f \hat{x}(k, v)(i, l) + W f h i h f(t-1) + W f c i c f(t-1) + b f i$ // see Equation 4.15 6 Then output of BiLSTM $q(k, v)(t, \text{stack})$ can be recived as : $q(k, v)(t, \text{stack}) = s W \hat{x}(k, v) \text{out} \hat{x}(k, v) \text{out} + W h \text{out } h t + W h \text{out } c t + b \text{out}$ 7 Repeat step-2 to 6 for stack=2; then the final output can be written as : $\hat{y}_k(v, t) = q(k, v)(t, \text{stack}) = s W \hat{x}(k, v) \text{out} \hat{x}(k, v) \text{out} + W h \text{out } h t + W h \text{out } c t + b \text{out}$ // see Equation 4.16 8 Return ($\hat{y}_k(v, 0)$) 9 End

Chapter 4. Methodology 38 Algorithm 3: Stacked CNN-BiLSTM Model. Input: $X = [x_1, x_2, x_3, \dots, x_m]$, $x_i \in \mathbb{R}^m$ // Time series data (univariate). 1, $K \in \mathbb{N}^+ //$ no. of partition of series X 2 . Output: $\hat{y}_k(t)$ // Find weight aggregation of all output of model induced from views $X \times v$ 3 Initialization: $K \in \mathbb{N}^+ , \hat{y}_k(v, t) = \text{ffg}$ 4 Checking ergodicity of univariate time series X after preprocessing. 5 Find the strength of seasonality of X as $F_s : F_s = \max(0, 1 - \text{Var}(X \times R) / \text{Var}(X \times S + X \times R))$ // see Equation 4.4 6 if ($F_s \> 0$) then 7 for $v = 1$ to k : do 8 Generating the v th view for k - partition from Algorithm 1 i.e. $X_{k,v} //$ Algorithm 1 9 Deployment of CNN-BiLSTM model Algorithm 2 for the input series $X_{k,v}$, then output recieved i.e. $\hat{y}_k(v, 0)$ // Algorithm 2 10 Update the set of prediction of model induced with v th view $X_{k,v}$ as: $\hat{y}_k(v, t) \leftarrow \hat{y}_k(v, t) \cup \{\hat{y}_k(v, 0)\}$ 11 Weighted aggregation of view-wise model prediction to get the find output $\hat{y}_k(t)$ as: $\hat{y}_k(t) = \sum_{v=1}^V \hat{y}_k(v, t) w_v //$ see Equation 4.17 12 Return ($\hat{y}_k(t)$) 13 End

39 Chapter 5 Results and Analysis 5.1 Results and Analysis Figures 5.1 and 5.2 depicts a series of subplots that illustrate different datasets, each showcasing the comparison between the predicted and actual PM 2.5 values on the test dataset. The x-axis indicates the quantity of data points, while the PM 2.5 values are represented by the y-axis. Every subplot is associated with a specific dataset name, signifying the MvS CNN-BiLSTM model's evaluation performance on various datasets, which may reflect different periods or locations where PM 2.5 concentrations were monitored and documented. In each subplot, the test dataset's samples are represented by data points. For each data point, there are two values plotted: the actual PM 2.5 value (ground truth) and the corresponding predicted PM 2.5 value generated by the MvS CNN-BiLSTM model. The plot allows us to visually compare how well the model's predictions match the actual values for each dataset. In Figure 5.1, the "prediction mimicking test" refers to comparing the model's predicted values with the actual values in the test dataset to assess the accuracy and effectiveness of the predictive model.

Chapter 5. Results and Analysis 40 FIGURE 5.1: PM 2.5 predictions of proposed models MvS CNN-BiLSTM

Chapter 5. Results and Analysis 41 FIGURE 5.2: Continuation of PM 2.5 predictions of proposed models MvS CNN-BiLSTM

Chapter 5. Results and Analysis 42 TABLE 5.1: RMSE Performance of traditional DL models and proposed models (MvS CNN- BiLSTM). DataSet BiLSTM CNN GRU LSTM RNN MvS CNN-BiLSTM Best-view AMBALA 33.382 36.239 33.012 33.136 33.455 33.272 4 ANKLESHWAR 22.916 23.724 22.975 23.359 22.807 22.081 6 BHIWADI 28.043 25.168 25.521 26.512 25.974 24.959 10 BULANDSHAHR 16.155 16.894 14.807 16.001 14.923 16.265 7 DCHARKHI_DADRI 30.071 32.709 32.299 32.629 33.836 31.064 3 DHARUHERA 25.751 26.948 26.057 26.108 25.489 24.983 3 DURGAPUR 10.426 9.719 11.375 11.994 10.664 8.906 6 FATEHABAD 31.501 31.251 32.169 31.045 30.946 30.86 10 HISAR 21.677 22.286 20.875 21.371 21.617 20.155 6 JIND 26.619 26.29 25.043 24.813 24.364 24.408 10 JODHPUR 27.749 27.617 27.787 27.519 27.951 27.482 7 KURUKSHETRA 16.475 16.352 15.239 17.359 15.414 15.921 5 LUDHIANA 14.573 17.013 13.817 17.668 14.936 14.315 9 MUZAFFARNAGAR 20.885 23.182 21.206 20.412 23.638 19.859 10 SINGRAULI 33.995 36.569 32.993 33.142 35.572 33.834 2 SONIPAT 13.644 13.951 14.025 13.773 13.755 12.75 5 YAMUNA_NAGAR 31.985 36.433 32.413 32.699 35.591 31.676 8

Chapter 5. Results and Analysis 43 Table 5.1 exhibits the RMSE performance of different conventional DL models and the proposed MvS CNN-BiLSTM model on 17 datasets. The graph empha- sises the random-performing model (i.e., the model with unpredictable RMSE) for each dataset using strikethrough text. The leftmost column lists the names of the evaluated datasets, while the next columns represent the RMSE values of different traditional DL models on each dataset. The proposed MvS CNN-BiLSTM model's RMSE values are displayed in the "MvS CNN-BiLSTM" column. The "Best-view" column identifies the DL model that achieved the lowest RMSE on each dataset, highlighted using bold text. The table thoroughly compares various DL models' RMSE performance across multiple datasets, with the MvS CNN-BiLSTM model demonstrating better RMSE values in its respective column and the bolded values indicating the best-performing model for each dataset. Table 5.1 provides valuable insights into the predictive capabilities of traditional DL models and the proposed MvS CNN-BiLSTM model. The MvS CNN-BiLSTM model for better performance over several traditional deep learning models on di- verse datasets, as evidenced by the lower RMSE values. The highlighted values help identify the best model for each dataset, highlighting the effectiveness of the proposed MvS CNN-BiLSTM approach over traditional DL models.

Chapter 5. Results and Analysis 44 TABLE 5.2: Percentage improvement of MvS CNN-BiLSTM respec- tively BiLSTM, CNN, GRU, LSTM and RNN on RMSE. Dataset BiLSTM CNN GRU LSTM RNN AMBALA 0.33% 8.19% -0.79 % -0.41% 0.55% ANKLESHWAR 3.64% 6.93% 3.89% 5.47% 3.18% BHIWADI 11.00% 0.83% 2.20% 5.86% 3.91% BULANDSHAHR -0.68% 3.72% -9.85% -1.65% -8.99% CHARKHI_DADRI -3.3% 5.03% 3.82% 4.80% 8.19% DHARUHERA 2.98% 7.29% 4.12% 4.31% 1.99% DURGAPUR 14.58% 8.37% 21.71% 25.75% 16.49% FATEHABAD 2.03% 1.25% 4.07% 0.60% 0.28% HISAR 7.02% 9.56% 3.45% 5.69% 6.76% JIND 8.31% 7.16% 2.54% 1.63% -0.18% JODHPUR 0.96% 0.49% 1.10% 0.13% 1.68% KURUKSHETRA 3.36% 2.64% -4.48% 8.28% -3.29% LUDHIANA 1.77% 15.86% -3.6% 18.98% 4.16% MUZAFFARNAGAR 4.91% 14.33% 6.35% 2.71% 15.99% SINGRAULI 0.47% 7.48% -2.55% -2.09% 4.89% SONIPAT 6.55% 8.61% 9.09% 7.43% 7.31% YAMUNA_NAGAR 0.97% 13.06% 2.27% 3.13% 11.00% Positive Avg 4.05% 7.11% 3.80% 5.57% 5.08% Table 5.2 illustrates the extent % of improvement from conventional deep learn- ing (DL) models to the proposed MvS CNN-BiLSTM model across 17 datasets. The chart showcases the growth rate of every deep learning model, namely BiLSTM, CNN, GRU, LSTM, and RNN, concerning the MvS CNN-BiLSTM model. Negative values indicate that the corresponding DL model outperformed the MvS CNN- BiLSTM model for a particular dataset. The table consists of a pair of columns that are connected. The right column denotes the percentage improvement of the MvS CNN-BLSTM model compared to traditional DL models for each dataset evaluated, as shown in the left columns. A positive percentage value denotes that the MvS

Chapter 5. Results and Analysis 45 CNN-BiLSTM model performed better than the DL model. TABLE 5.3: Average Rankings of the algorithms (Friedman) in a contest of RMSE. Algorithm Ranking p Holm BiLSTM 3.7059 0.002485 0.016667 CNN 4.8235 0.000002 0.01 GRU 3.1765 0.027801 0.05 LSTM 3.8235 0.001335 0.0125 RNN 3.7059 0.002485 0.025 MvS CNN-BiLSTM 1.7647 * * Table 5.3 exhibits the average rankings of various algorithms about the RMSE metric, thereby providing insights into their relative performance. The table comprises four columns: algorithm, Ranking, p-value, and Holm. The names of several algorithms, including BiLSTM, CNN, GRU, LSTM, RNN, and the recently suggested MvS CNN-BiLSTM, are being logged in the algorithm column while the assessment process is underway. The standing column exhibits the mean ranks of every algorithm relying on their performance in terms of RMSE, where a lesser rank indicates superior performance, with 1 being the best rank and higher numbers indicating weaker performance. Each algorithm's comparison is assigned a significance level in the p-value column, with a small p-value indicating a statistically significant difference in performance between the compared algorithms. The Holm column controls the family-wise error rate by indicating adjusted significance thresholds utilised in multiple statistical tests. The proposed MvS CNN-BiLSTM algorithm accomplished an average ranking of 1.7647, the finest among all compared models concerning RMSE. The MvS CNN-BiLSTM algorithm was not used in comparing traditional shallow learning models (BiLSTM, CNN, GRU, LSTM, and RNN). Furthermore, the minute p-values for every algorithmic comparison imply a statistically significant variance in performance amongst all the algorithms, signifying that the performance of each algorithm is discernible from others based on the RMSE metric.

Chapter 5. Results and Analysis 46 The Holm column likely indicates the adjusted significance thresholds (critical values) used in the Holm method for multiple comparisons, a statistical procedure used to control the family-wise error rate when performing multiple hypothesis tests. The MvS CNN-BiLSTM algorithm is the superior choice based on the table's data, performing better than other algorithms in RMSE.

Chapter 5. Results and Analysis 47 TABLE 5.4: MAPE Performance of traditional DL models and proposed models (MvS CNN-BiLSTM). DataSet BiLSTM CNN GRU LSTM RNN MvS CNN-BiLSTM Best-view AMBALA 54.468 49.153 76.839 58.619 53.375 18.73 3 ANKLESHWAR 18.86 19.968 19.015 19.576 20.72 16.265 10 BHIWADI 134.448 118.741 123.123 127.481 124.503 20.98 2 BULANDSHAHR 44.602 47.066 26.254 43.721 27.813 22.469 3 CHARKHI_DADRI 50.328 101.385 102.063 112.912 115.673 26.728 10 DHARUHERA 35.638 39.358 39.557 35.942 40.34 17.365 3 DURGAPUR 45.511 37.529 56.324 57.457 48.626 19.321 6 FATEHABAD 19.737 19.929 19.475 18.663 17.724 12.178 5 HISAR 17.971 18.422 17.988 17.983 21.475 15.082 6 JIND 25.945 30.822 27.344 24.596 33.464 16.618 3 JODHPUR 40.188 40.36 41.584 41.578 43.36 28.278 7 KURUKSHETRA 13.713 13.392 12.405 13.986 13.312 11.248 2 LUDHIANA 37.61 47.14 34.504 51.036 41.197 15.706 9 MUZAFFARNAGAR 24.162 27.253 23.37 26.08 27.061 16.571 5 SINGRAULI 48.642 67.143 39.738 34.372 68.525 28.545 7 SONIPAT 43.301 48.32 49.409 43.187 45.67 13.541 7 YAMUNA_NAGAR 65.601 63.698 64.637 63.404 70.02 28.29 10

Chapter 5. Results and Analysis 48 Table 5.4 showcases MAPE results of the MvS CNN-BiLSTM model and traditional DL models across 17 datasets. The DL models analysed in the table include BiLSTM, CNN, GRU, LSTM, and RNN. The MAPE values for each DL model are compared with those of the MvS CNN-BiLSTM model. The minimum MAPE values obtained by each model for each dataset and the corresponding best view are presented in bold. To expound the Table 5.4, the initial column enumerates the names of the datasets assessed in the investigation. The succeeding columns indicate the MAPE values for the different traditional DL models compared to each dataset's proposed MvS CNN-BiLSTM model. The table's interpretation involves the presentation of the corresponding MAPE value for each DL model to their respective columns. The "Best-view" column, on the other hand, specifies the best dataset for which the MvS CNN-BiLSTM model obtained the minimum MAPE value. In conclusion, Table 5.4 compares the MAPE performance of the MvS CNN-BiLSTM model and traditional DL models across multiple datasets. The values in bold highlight the MvS CNN-BiLSTM model's superior performance over traditional DL models for different datasets. The table demonstrates the MvS CNN-BiLSTM model's ability to reduce MAPE for air quality prediction in various locations and datasets.

Chapter 5. Results and Analysis 49

TABLE 5.5: Improvement of MvS CNN-BiLSTM respectively BiLSTM, GRU, LSTM and RNN on MAPE. Dataset BiLSTM CNN GRU LSTM RNN AMBALA 35.74% 30.42% 58.11% 39.89% 34.64% ANKLESHWAR 2.59% 3.70% 2.75% 3.31% 4.45% BHIWADI 113.47% 97.76% 102.14% 106.50% 103.52% BULANDSHAHR 22.13% 24.60% 3.78% 21.25% 5.34% CHARKHI_DADRI 23.60% 74.66% 75.34% 86.18% 88.94% DHARUHERA 18.27% 21.99% 22.19% 18.58% 22.98% DURGAPUR 26.19% 18.21% 37.00% 38.14% 29.30% FATEHABAD 7.56% 7.75% 7.30% 6.48% 5.55% HISAR 2.89% 3.34% 2.91% 2.90% 6.39% JIND 9.33% 14.20% 10.73% 7.98% 16.85% JODHPUR 11.91% 12.08% 13.31% 13.30% 15.08% KURUKSHETRA 2.46% 2.14% 1.16% 2.74% 2.06% LUDHIANA 21.90% 31.43% 18.80% 35.33% 25.49% MUZAFFARNAGAR 7.59% 10.68% 6.80% 9.51% 10.49% SINGRAULI 20.10% 38.60% 11.19% 5.83% 39.98% SONIPAT 29.76% 34.78% 35.87% 29.65% 32.13% YAMUNA_NAGAR 37.31% 35.41% 36.35% 35.11% 41.73% Positive Avg 23.11% 27.16% 26.22% 27.22% 28.52%

Table 5.5 displays the percentage improvement in MAPE attained by the proposed MvS CNN-BiLSTM model compared to various traditional DL models (BiLSTM, GRU, LSTM, RNN) for all 17 datasets. The first column of the table lists the names of the datasets evaluated in the study. The succeeding columns represent the percentage improvement in MAPE achieved by the MvS CNN-BiLSTM model over each traditional DL model on each dataset. The percentage values in the table indicate the reduction in MAPE attained by the MvS CNN-BiLSTM model relative to each traditional DL model, with negative values implying cases where the traditional DL model outperformed the MvS CNN-BiLSTM model.

Chapter 5. Results and Analysis 50

TABLE 5.6: Average rankings of the algorithms (Friedman) in the context of MAPE. Algorithm Ranking p Holm BiLSTM 3.4706 0.000118 0.05 CNN 4.1765 0.000001 0.0125 GRU 3.7647 0.000016 0.025 LSTM 3.8824 0.000007 0.016667 RNN 4.7059 0 0.01 MvS CNN-BiLSTM 1 * *

Table 5.6 showcases the average rankings of multiple algorithms in the context of the MAPE metric, providing insight into the relative performance of these algorithms. The primary row comprises the appellation of the systems being measured, particularly BiLSTM, CNN, GRU, LSTM, RNN, and the purpose of MvS CNN-BiLSTM. The subsequent column displays the average rankings of each algorithm according to their MAPE performance, with a lower ranking indicating superior performance and a greater number indicating inferior performance. The third column denotes the p-values, the significance level for comparing each algorithm. A statistically significant difference in performance between the compared algorithms is present if the p-value is low. The fourth column, "Holm," will likely denote the critical values or adjusted significance thresholds used for multiple comparisons. The Holm method manages the family-wise error rate when conducting multiple statistical tests. The MvS CNN-BiLSTM algorithm is better regarding MAPE, surpassing all other models, including BiLSTM, CNN, GRU, LSTM, and RNN, with an average rank of 1. It is thus improbable to assume that any other compared algorithm outperforms the MvS CNN-BiLSTM algorithm regarding MAPE. Furthermore, the small p-values for every algorithmic comparison indicate statistically significant efficacy disparities among all algorithms. This implies that the algorithms' performance is distinguishable from one another based on the MAPE metric. The "Holm" column likely denotes the adjusted significance thresholds (critical values) used

Chapter 5. Results and Analysis 51

in the Holm method for multiple comparisons. The Holm method is a statistical technique employed for controlling family-wise error rates during the execution of multiple hypothesis tests. In summary, Table 5.6 shows that the proposed MvS CNN-BiLSTM algorithm is the best among all the compared algorithms in terms of MAPE. Moreover, statistically meaningful performance distinctions exist among the MAPE metric-based algorithms. The Figures 5.3 and 5.4 contains 17 subplots, each displaying a scatter plot for multiple datasets, and compares the grey Stacked CNN-BiLSTM model with the black Proposed MvS CNN-BiLSTM model. The plot includes a blue reference line representing the optimal scenario in which predicted and actual values are aligned. Additionally, each subplot displays two R^2 values corresponding to the grey and black superimposed plots. A heightened R^2 value suggests a more powerful correlation between anticipated and actual values, making it a valuable statistical gauge of fitness in provided Figure; there are two potential interpretations: If the grey Stacked CNN-BiLSTM has a more excellent R^2 value than the black Proposed MvS CNN-BiLSTM, The analysis shows that the grey model had superior accuracy and a more resilient correlation between projected and actual values than the black model, leading to a superior overall fit. The same R^2 value of the black Proposed MvS CNN-BiLSTM and the grey Stacked CNN-BiLSTM makes it irrelevant which model performed better in predicting actual values and has a similar goodness-of-fit and correlation with actual values. Based on the information you provided, it appears that for the Figures 5.3 and 5.4 Singrauli, Fatehabad, Charkhi_Dadri, and Bulandshahr, the grey Stacked CNN-BiLSTM model has higher R^2 values compared to the black Proposed MvS CNN-BiLSTM model. However, you mentioned that the R^2 values for the black model are very close to the grey model. In this case, since the R^2 values for the black model are very close to the grey model, it suggests that both models have similar predictive performance for those datasets. The difference in their R^2 values might be slight, indicating that they are both relatively accurate in predicting the actual values. It has concluded that the association of multi-view leading with stacked CNN-BiLSTM is performing better than the stacked CNN-BiLSTM along with the traditional model over various measures.

Chapter 5. Results and Analysis 52 TABLE 5.7: Overall average of RMSE and MAPE ranking of traditional models and proposed model (MvS CNN-BiLSTM). Algorithm RMSE_Ranking MAPE_Ranking Average_Ranking BiLSTM 3.7059 3.4706 3.58825 CNN 4.8235 4.1765 4.5 GRU 3.1765 3.7647 3.4706 LSTM 3.8235 3.8824 3.85295 RNN 3.7059 4.7059 4.2059 MvS CNN-BiLSTM 1.7647 1.138235 In Table 5.7, the mean rankings of various algorithms are presented according to their performance on two performance metrics, namely RMSE and MAPE. The table also includes an additional column for the average ranking across both metrics. The initial column suggests the average positions of each formula based on their RMSE performance. In contrast, the second column displays the average positions of each formula based on their MAPE performance. The third column signifies the mean rank of each algorithm, factoring in performance on both metrics and providing a comprehensive measure of performance. In Table 5.7 show average ranking on both performance measures and concluded the best algorithm. The underperformance of conventional deep learning structures such as BiLSTM, CNN, GRU, LSTM, and RNN, despite their higher ranking on average, is a surprising observation compared to the MvS CNN-BiLSTM method. Thus, according to Figure 5.5, the mean ranking over the two performance metrics, the MvS CNN-BiLSTM algorithm is regarded as the optimal algorithm out of all the models. The bar graph provided is illegible and cannot be used to support the previous analysis, showing that the MvS CNN- BiLSTM algorithm has no ranking or comparison to the traditional deep learning models (BiLSTM, CNN, GRU, LSTM, and RNN) on average for the given task. The CNN and RNN algorithms were rated the worst on average, suggesting their performance was not as great as other models.

Chapter 5. Results and Analysis 53 FIGURE 5.3: Correlation of Actual values and Predictions of proposed models (Stacked CNN-BiLSTM and MvS CNN-BiLSTM) using Scatter plot.

Chapter 5. Results and Analysis 54 FIGURE 5.4: Continuation of Correlation of Actual values and Pre- dictions of proposed models (Stacked CNN-BiLSTM and MvS CNN- BiLSTM) using Scatter plot.

Chapter 5. Results and Analysis 55 FIGURE 5.5: Overall average Friedman ranking of traditional models and proposed MvS CNN-BiLSTM model.

56 Chapter 6 Conclusion 6.1 Conclusion In this study, a hybrid MvS CNN-BiLSTM model has been proposed. The model utilises stacked 1D CNN & BiLSTM as new architecture to view the univariate data PM 2.5 pollutant. Additionally, a novel multi-view approach has been proposed for univariate time series, which exploits the seasonal characteristics of the data to construct the views corresponding to the lowest lag. The seventeen datasets (PM 2.5) of highly polluted cities of India have been used to deploy the proposed MvS CNN-BiLSTM and stand-alone DL models. The evaluation of the models has been performed using RMSE and MAPE. The results have shown that the proposed model has improved the performed than corresponding stand-alone DL models over RMSE: 7.11% (CNN), 5.08% (RNN), 3.80% (GRU), 5.57% (LSTM) and 4.05% (BiLSTM) and MAPE: 27.16% (CNN), 28.52% (RNN), 26.22% (GRU), 27.22% (LSTM), 23.11% (BiLSTM). Moreover, the enhanced performance of the proposed model has also been validated using non-parametric statistical methods, i.e., Friedman ranking and Holm's procedure. The statistical analysis of results concludes that the pro- posed model performance is better and distinct from stand-alone DL models. Future research: The proposed model has utilised two-stacked CNN-BiLSTM for this research, where multiple combinations of stand-alone DL models may be investigated along with the stacking of the hybrid for better performance. In a multi- view approach, apart from seasonal characteristics, trends and the remainder may be utilised to generate the views of the dataset, yielding enhanced performance.

Chapter 6. Conclusion 57 Moreover feature set partitioning method of multi-view learning may be utilised for multivariate time series data (single source or multi-source data). Data availability Data will be available on CPCB (India) Webportal. The Central Pollution Control Board (CPCB) in India maintains a web portal that offers access to various environ- mental datasets, including air and water quality, emission inventories, and pollution monitoring data, aimed at promoting environmental awareness and research in the country. Acknowledgments We extend our heartfelt appreciation to the Central Pollution Control Board (CPCB), India, for generously providing invaluable environmental data, which significantly enhanced our research and played a pivotal role in completing this study.

58 References [1] C. Zhou, G. Wei, H. Zheng, et al., "Effects of potential recirculation on air qual- ity in coastal cities in the yangtze river delta," *Science of the total environment*, vol. 651, pp. 12–23, 2019. [2] K.-P. Lin, P.-F. Pai, and S.-L. Yang, "Forecasting concentrations of air pollutants by logarithm support vector regression with immune algorithms," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5318–5327, 2011. [3] S. Kumari and S. K. Singh, "Machine learning-based time series models for effective co2 emission prediction in india," *Environmental Science and Pollution Research*, pp. 1–16, 2022. [4] S. Kumari and S. K. Singh, "Deep learning-based time series models for gdp and ict growth prediction in india," in *2022 International Conference on Comput- ing, Communication, and Intelligent Systems (ICCCIS)*, IEEE, 2022, pp. 250–256. [5] O. Taylan, "Modelling and analysis of ozone concentration by artificial intelli- gent techniques for estimating air quality," *Atmospheric environment*, vol. 150, pp. 356–365, 2017. [6] B. Wang and Z. Chen, "A model-based fuzzy set-owa approach for integrated air pollution risk assessment," *Stochastic Environmental Research and Risk As- sessment*, vol. 29, pp. 1413–1426, 2015. [7] E. Kristiani, H. Lin, J.-R. Lin, Y.-H. Chuang, C.-Y. Huang, and C.-T. Yang, "Short-term prediction of pm2. 5 using lstm deep learning methods," *Sustain- ability*, vol. 14, no. 4, p. 2068, 2022. [8] Y. A. Ayturan, Z. C. Ayturan, and H. O. Altun, "Air pollution modelling with deep learning: A review," *International Journal of Environmental Pollution and Environmental Modelling*, vol. 1, no. 3, pp. 58–62, 2018.

References 59 [9] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017. [10] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013. [11] X. Yang, S. Feng, D. Wang, and Y. Zhang, "Image-text multimodal emotion classification via multi-view attentional network," *IEEE Transactions on Multi- media*, vol. 23, pp. 4014–4026, 2020. [12] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for im- age clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1501–1511, 2017. [13] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu, "Deep multi-view learning methods: A review," *Neurocomputing*, vol. 448, pp. 106–129, 2021, ISSN : 0925-2312. DOI : <https://doi.org/10.1016/j.neucom.2021.03.090>. [14] A. Kumar and J. Yadav, "A review of feature set partitioning methods for multi-view ensemble learning," *Information Fusion*, vol. 100, p. 101 959, 2023, ISSN : 1566-2535. DOI : <https://doi.org/10.1016/j.inffus.2023.101959>. [15] E. Garcia-Ceja, C. E. Galván-Tejada, and R. Brena, "Multi-view stacking for activity recognition with sound and accelerometer data," *Information Fusion*, vol. 40, pp. 45–56, 2018. [16] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Al- buquerque, "A comprehensive survey of multi-view video summarization," *Pattern Recognition*, vol. 109, p. 107 567, 2021. [17] X. Huang, D. Wen, J. Li, and R. Qin, "Multi-level monitoring of subtle urban changes for the megacities of china using high-resolution multi-view satellite imagery," *Remote sensing of environment*, vol. 196, pp. 56–75, 2017. [18] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "A multi-view multi-task learning framework for multi-variate time series forecasting," *IEEE Transac- tions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7665–7680, 2023. DOI : 10.1109/TKDE.2022.3218803.

References 60 [19] D. Zhan, S. Yi, D. Xu, et al., "Adaptive transfer learning of multi-view time series classification.," *arXiv: Learning*, 2019. [20] S. D. Bhattacharjee, W. J. Tolone, A. Mahabal, et al., "Multi-view, generative, transfer learning for distributed time series classification," 2019. [21] S. D. Bhattacharjee, W. J. Tolone, A. Mahabal, M. Elshambakey, I. Cho, and G. Djorgovski, "View-adaptive weighted deep transfer learning for distributed time-series classification," 2019. [22] J. Li, J. Crooks, J. Murdock, et al., "A nested machine learning approach to short-term pm2. 5 prediction in metropolitan areas using pm2. 5 data from dif- ferent sensor networks," *Science of The Total Environment*, vol. 873, p. 162 336, 2023. [23] D. Pruthi and Y. Liu, "Low-cost nature-inspired deep learning system for pm2. 5 forecast over delhi, india," *Environment International*, vol. 166, p. 107 373, 2022. [24] C. Menares, P. Perez, S. Parraguez, and Z. L. Fleming, "Forecasting pm2. 5 lev- els in santiago de chile using deep learning neural networks," *Urban Climate*, vol. 38, p. 100 906, 2021. [25] L. Zhu, Z. J. B. M. Husny, N. A. Samsudin, H. Xu, and C. Han, "Deep learning method for minimizing water pollution and air pollution in urban environ- ment," *Urban Climate*, vol. 49, p. 101 486, 2023. [26] B.-Y. Kim, Y.-K. Lim, and J. W. Cha, "Short-term prediction of particulate mat- ter (pm10 and pm2. 5) in seoul, south korea using tree- based machine learning algorithms," *Atmospheric Pollution Research*, vol. 13, no. 10, p. 101 547, 2022. [27] A. Lee, S. Jeong, J. Joo, C.-R. Park, J. Kim, and S. Kim, "Potential role of urban forest in removing pm2. 5: A case study in seoul by deep learning with satellite data," *Urban Climate*, vol. 36, p. 100 795, 2021. [28] K. K. R. Samal, K. S. Babu, and S. K. Das, "

85%

MATCHING BLOCK 1/11

W

Multi-directional temporal convo- lutional artificial neural network for pm2. 5 forecasting with missing values: A deep learning approach," *Urban Climate*,

vol. 36, p. 100 800, 2021.

References 61 [29] G. Kurnaz and A. S. Demir, "Prediction of so2 and pm10 air pollutants using a deep learning-based recurrent neural network: Case of industrial city sakarya," Urban Climate, vol. 41, p. 101 051, 2022. [30] B. Das, Ö. O. Dursun, and S. Toraman, "

100% MATCHING BLOCK 2/11

W

Prediction of air pollutants for air quality using deep learning methods in a metropolitan city," Urban Climate,

vol. 46, p. 101 291, 2022. [31] N. Natsagdorj, H. Zhou, et al., "Prediction of pm2. 5 concentration in ulaan- baatar with deep learning models," Urban Climate, vol. 47, p. 101 357, 2023. [32] J. Ma, J. C. Cheng, C. Lin, Y. Tan, and J. Zhang, "Improving air quality predic- tion accuracy at larger temporal resolutions using deep learning and transfer learning techniques," Atmospheric Environment, vol. 214, p. 116 885, 2019. [33] D. Qin, J. Yu, G. Zou, R. Yong, Q. Zhao, and B. Zhang, "A novel combined prediction scheme based on cnn and lstm for urban pm 2.5 concentration," Ieee Access, vol. 7, pp. 20 050–20 059, 2019. [34] X. Li, L. Peng,

92% MATCHING BLOCK 3/11

W

X. Yao, et al., "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation,"

Environmental pollution, vol. 231, pp. 997–1004, 2017. [35] M. Zhu and J. Xie, "Investigation of nearby monitoring station for hourly pm2. 5 forecasting using parallel multi-input 1d-cnn-bilstm," Expert Systems with Applications, vol. 211, p. 118 707, 2023. [36] P. Nath, P. Saha, A. I. Middya, and S. Roy, "Long-term time-series pollution forecast using statistical and deep learning methods," Neural Computing and Applications, pp. 1–20, 2021. [37] S. Du, T. Li, Y. Yang, and S.-J. Horng, "

100% MATCHING BLOCK 4/11

W

Deep air quality forecasting using hybrid deep learning framework," IEEE

Transactions on Knowledge and Data Engineering, vol. 33, no. 6, pp. 2412–2424, 2019. [38] B. Eren, İ. Aksangür, and C. Erden, "Predicting next hour fine particulate mat- ter (pm2. 5) in the istanbul metropolitan city using deep learning algorithms with time windowing strategy," Urban Climate, vol. 48, p. 101 418, 2023.

References 62 [39] V. Kumar and N. Ahmad, "Deep learning for air quality prediction after covid- 19 pandemic based on pollutant and meteorological data," Available at SSRN 4292346, 2022. [40] Y. Zhang, S. Zhai, J. Huang, et al., "

85% MATCHING BLOCK 5/11

W

Estimating high-resolution pm2.5 con- centration in the sichuan basin using a random forest model with data-driven spatial autocorrelation terms,"

Journal of Cleaner Production, vol. 380, p. 134 890, 2022, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2022.134890>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652622044638>.

[41] R. Zhang, F. Nie, X. Li, and X. Wei, "Feature selection with multi-view data: A survey," Information Fusion, vol. 50, pp. 158–167, 2019, ISSN : 1566-2535. DOI : <https://doi.org/10.1016/j.inffus.2018.11.019>. [42] H. Kamarathi, L. Kong, A. Rodríguez, C. Zhang, and B. A. Prakash, "Ca- mul: Calibrated and accurate multi-view time-series forecasting," 2022. arXiv: 2109.07438 [cs.LG]. [43] S. Mandal and M. Thakur, "

81% MATCHING BLOCK 6/11

W

A city-based pm2.5 forecasting framework using spatially attentive cluster-based graph neural network model,"

Journal of Cleaner Production, vol. 405, p. 137 036, 2023, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2023.137036>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652623011940>. [44] J. Ma, Y. Ding, J. C. Cheng, F. Jiang, and Z. Wan, "

84%

MATCHING BLOCK 7/11

W

A temporal-spatial interpolation and extrapolation method based on geographic long short-term memory neural network for pm2.5,"

Journal of Cleaner Production, vol. 237, p. 117 729, 2019, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2019.117729>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652619325892>. [45] Z. Tian and M. Gai, "

100%

MATCHING BLOCK 8/11

W

New pm2.5 forecasting system based on combined neural network and an improved multi-objective optimization algorithm: Taking the economic belt surrounding the bohai sea as an example," Journal of Cleaner Production,

vol. 375, p. 134 048, 2022, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2022.134048>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652622036204>. [46] H. Dai, G. Huang, H. Zeng, and F. Zhou, "

100%

MATCHING BLOCK 9/11

W

Pm2.5 volatility prediction by xgboost-mlp based on garch models,"

Journal of Cleaner Production, vol. 356, p. 131 898, 2022, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2022.131898>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652622015086>. [47] A. Aggarwal and D. Toshniwal, "

100%

MATCHING BLOCK 10/11

W

A hybrid deep learning framework for urban air quality forecasting," Journal of Cleaner Production,

vol. 329, p. 129 660, 2021, ISSN : 0959-6526. DOI : <https://doi.org/10.1016/j.jclepro.2021.129660>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652621038373>. [48] E. Chaerun Nisa and Y.-D. Kuan, "Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms," Sustainability, vol. 13, no. 2, p. 744, 2021. [49] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," Mechanical systems and signal processing, vol. 151, p. 107 398, 2021. [50] A.-L. Rusnac and O. Grigore, "Cnn architectures and feature extraction methods for eeg imaginary speech recognition," Sensors, vol. 22, no. 13, p. 4679, 2022. [51] D. Wang, X. Wang, and S. Lv, "End-to-end mandarin speech recognition combining cnn and blstm," Symmetry, vol. 11, no. 5, p. 644, 2019. [52] M. Ashraf, F. Abid, M. Atif, and S. Bashir, "The role of cnn and rnn in the classification of audio music genres," 2022. [53] H. Hu, C.-H. H. Yang, X. Xia, et al., "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," arXiv preprint arXiv:2007.08389, 2020.

References 64 [54] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (cnn) in vegetation remote sensing," *ISPRS journal of photogrammetry and remote sensing*, vol. 173, pp. 24–49, 2021. [55] X. Sun, L. Liu, C. Li, J. Yin, J. Zhao, and W. Si, "Classification for remote sensing data with improved cnn-svm method," *IEEE Access*, vol. 7, pp. 164 507– 164 516, 2019. [56] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. [57] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, 2019. [58] Y. Yuan, C. Tian, and X. Lu, "Auxiliary loss multimodal gru model in audio- visual speech recognition," *IEEE Access*, vol. 6, pp. 5573–5583, 2018. [59] S. Cascianelli, G. Costante, T. A. Ciarfuglia, P. Valigi, and M. L. Fravolini, "Full-gru natural language video description for service robotics applications," *IEEE robotics and automation letters*, vol. 3, no. 2, pp. 841–848, 2018. [60] D. Wang, J. Su, and H. Yu, "Feature extraction and analysis of natural language processing for deep learning english language," *IEEE Access*, vol. 8, pp. 46 335–46 345, 2020. [61] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K.-H. Park, and J. Kim, "An integrated mediapipe-optimized gru model for indian sign language recognition," *Scientific Reports*, vol. 12, no. 1, p. 11 964, 2022. [62] H. Soltan, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to- word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016. [63] J. Jo, J. Kung, and Y. Lee, "Approximate lstm computing for energy-efficient speech recognition," *Electronics*, vol. 9, no. 12, p. 2004, 2020.

References 65 [64] S. Wang and J. Jiang, "Learning natural language inference with lstm," *arXiv preprint arXiv:1512.08849*, 2015. [65] M. K. Nammous and K. Saeed, "Natural language processing: Speaker, language, and gender identification with lstm," *Advanced Computing and Systems for Security: Volume Eight*, pp. 143–156, 2019. [66] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [67] S. O. Sahin and S. S. Kozat, "Nonuniformly sampled data processing using lstm networks,"

100%
MATCHING BLOCK 11/11
W

IEEE transactions on neural networks and learning systems, vol. 30, no. 5, pp. 1452–1461, 2018. [68]

F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000. [69] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015. [70] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE access*, vol. 6, pp. 1662–1669, 2017. [71] T.-H. S. Li, P.-H. Kuo, T.-N. Tsai, and P.-C. Luan, "Cnn and lstm based facial expression analysis model for a humanoid robot," *IEEE Access*, vol. 7, pp. 93 998–94 011, 2019. [72] G. B. Rajendran, U. M. Kumarasamy, C. Zarro, P. B. Divakarachari, and S. L. Ullo, "Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an lstm classifier on hybrid pre-processing remote-sensing images," *Remote Sensing*, vol. 12, no. 24, p. 4135, 2020. [73] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images," *Informatics in medicine unlocked*, vol. 20, p. 100 412, 2020. [74] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE access*, vol. 6, pp. 1155–1166, 2017.

References 66 [75] J. Li, Y. Zhong, J. Han, G. Ouyang, X. Li, and H. Liu, "Classifying asd children with lstm based on raw videos," *Neurocomputing*, vol. 390, pp. 226–238, 2020. [76] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017. [77] Y. Bin, Y. Yang, F. Shen, N. Xie, H. T. Shen, and X. Li, "Describing video with attention-based bidirectional lstm," *IEEE transactions on cybernetics*, vol. 49, no. 7, pp. 2631–2641, 2018. [78] A. Ghourabi, "A security model based on lightgbm and transformer to protect healthcare systems from cyberattacks," *IEEE Access*, vol. 10, pp. 48 890–48 903, Jan. 2022. DOI : 10.1109/ACCESS.2022.3172432. [79] P. Bhawan and E. A. Nagar, "Central pollution control board," 2020.

Hit and source - focused comparison, Side by Side

Submitted text

As student entered the text in the submitted document.

Matching text

As the text appears in the source.

1/11	SUBMITTED TEXT	19 WORDS	85% MATCHING TEXT	19 WORDS
	Multi-directional temporal convo- lutional artificial neural network for pm2. 5 forecasting with missing values: A deep learning approach," Urban Climate,		Multi-directional temporal convolutional artificial neural network for PM2.5 forecasting with missing values: A deep learning approach Urban Climate,	
	W https://www.sciencedirect.com/science/article/pii/S0959652619325892 .			
2/11	SUBMITTED TEXT	16 WORDS	100% MATCHING TEXT	16 WORDS
	Prediction of air pollutants for air quality using deep learning methods in a metropolitan city," Urban Climate,		Prediction of air pollutants for air quality using deep learning methods in a metropolitan city 2022, Urban Climate	
	W https://www.sciencedirect.com/science/article/pii/S0959652621038373 .			
3/11	SUBMITTED TEXT	17 WORDS	92% MATCHING TEXT	17 WORDS
	X. Yao, et al., "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation,"		X. et al. Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation	
	W https://www.sciencedirect.com/science/article/pii/S0959652619325892 .			
4/11	SUBMITTED TEXT	10 WORDS	100% MATCHING TEXT	10 WORDS
	Deep air quality forecasting using hybrid deep learning framework," IEEE		Deep air quality forecasting using hybrid deep learning framework IEEE	
	W https://www.sciencedirect.com/science/article/pii/S0959652621038373 .			
5/11	SUBMITTED TEXT	19 WORDS	85% MATCHING TEXT	19 WORDS
	Estimating high-resolution pm2.5 con- centration in the sichuan basin using a random forest model with data-driven spatial autocorrelation terms,"		Estimating high-resolution PM2.5 concentration in the Sichuan Basin using a random forest model with data-driven spatial autocorrelation terms -	
	W https://www.sciencedirect.com/science/article/pii/S0959652622044638 .			
6/11	SUBMITTED TEXT	14 WORDS	81% MATCHING TEXT	14 WORDS
	A city-based pm2.5 forecasting framework us- ing spatially attentive cluster-based graph neural network model,"		A city-based PM2.5 forecasting framework using Spatially Attentive Cluster-based Graph Neural Network model -	
	W http://www.sciencedirect.com/science/article/pii/S0959652623011940 .			

7/11	SUBMITTED TEXT	16 WORDS	84% MATCHING TEXT	16 WORDS
A temporal-spatial interpolation and extrapolation method based on geographic long short-term memory neural network for pm2.5,"		A temporal-spatial interpolation and extrapolation method based on geographic Long Short-Term Memory neural network for PM2.5 -		
W https://www.sciencedirect.com/science/article/pii/S0959652619325892 .				
8/11	SUBMITTED TEXT	28 WORDS	100% MATCHING TEXT	28 WORDS
New pm2.5 forecasting system based on combined neural network and an improved multi-objective optimization algorithm: Taking the economic belt surrounding the bohai sea as an example," Journal of Cleaner Production,		New PM2.5 forecasting system based on combined neural network and an improved multi-objective optimization algorithm: Taking the economic belt surrounding the Bohai Sea as an example Journal of Cleaner Production,		
W https://www.sciencedirect.com/science/article/pii/S0959652622015086 .				
9/11	SUBMITTED TEXT	9 WORDS	100% MATCHING TEXT	9 WORDS
Pm2.5 volatility prediction by xgboost-mlp based on garch models,"		PM2.5 volatility prediction by XGBoost-MLP based on GARCH models -		
W https://www.sciencedirect.com/science/article/pii/S0959652622015086 .				
10/11	SUBMITTED TEXT	14 WORDS	100% MATCHING TEXT	14 WORDS
A hybrid deep learning framework for urban air quality forecasting," Journal of Cleaner Production,		A hybrid deep learning framework for urban air quality forecasting Journal of Cleaner Production,		
W https://www.sciencedirect.com/science/article/pii/S0959652622015086 .				
11/11	SUBMITTED TEXT	15 WORDS	100% MATCHING TEXT	15 WORDS
IEEE transactions on neural networks and learning systems, vol. 30, no. 5, pp. 1452–1461, 2018. [68]		IEEE transactions on neural networks and learning systems, vol. 31, no. 11, pp. 4688–4698, 2020. [39]		
W https://arxiv.org/pdf/2112.13444				