**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 32, No. 2, November 2023, pp. 1014∼1021
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v32.i2.pp1014-1021


# Predicting air quality in smart city using novel transfer learning based framework

**Shilpa Sonawani[1,2], Kailas Patil[2]**
[1]School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India
[2]Department of Computer Engineering, Vishwakarma University, Pune, India

**Article Info**

***Article history:***

Received Nov 10, 2022
Revised Jun 16, 2023
Accepted Jul 30, 2023

***Keywords:***

Air quality
Chaining approach
Deep learning
Multi-headed CNN-GRU
Smart city and safety
Transfer learning

**ABSTRACT**

Air quality is a matter of concern these days due to its adverse effect on human health. Multiple new air pollution monitoring and prediction stations are being developed in smart cities to tackle the issue. Recent advanced deep learning techniques show excellent performance for air quality predictions but need sufficient training data for model performance. The data insufficiency issue at a new station can be resolved using the proposed novel transfer learning-based framework to predict pollution concentration at the new station. The prediction ability at a new station can be significantly enhanced by this effective technology. The performance of the model is assessed on various stations in Delhi, India.



***Corresponding Author:***

Kailas Patil
Department of Computer Engineering, Vishwakarma University
Pune 411048, India
Email: kailas.patil@vupune.ac.in

## 1. INTRODUCTION

Increased number of diesel vehicles in urban areas [1]-[3] creating serious harm to the quality of the air and human health. Bad air quality is responsible for many cardiovascular, respiratory diseases [4] in rural areas and millions of premature deaths worldwide, according to the World Health Organization (WHO) [5]. Due to exposure to small particles of particulate matter (PM2.5), 91% of these premature deaths occur in low income countries. Better policies and monitoring strategies can control this air pollution by having efficient smart homes [6] and prioritizing pollution control by policymakers [7]. It is observed that even in the most polluted cities, there are insufficient air quality monitoring stations to fully cover the city [8] and hence it is highly recommended to increase this number.

This study focuses on the precise forecasting of PM2.5 pollutant concentration for the next hour because it is crucial to monitor this pollutant. Due to its small size, PM2.5 can easily enter the bloodstream, causing harmful effects on human health such as the development of cardiovascular and respiratory disorders [9]. Nitric oxide (NO), nitrogen oxides (NOx), nitrogen dioxide ($NO_2$), carbon monoxide (CO), ozone ($O_3$), PM2.5, particulate matter (PM10), and other pollutants are all found in the air. The air pollutant concentration prediction problem has time-series data.

The number of training samples needed to create the model and make predictions is typically lower at new air quality monitoring stations in smart city environments [10]. There are certain publications already published that can transmit knowledge from many sources. This type of strategy decreased the prediction error in industrial manufacturing for time-series data [11]. In another study, autoencoders are applied to many sources simultaneously, and predictions are made using target data and consensus regularization based on entropy [12]. Estimates of hospital admissions for respiratory disorders in the Brazilian cities of Campinas and Sao Paulo are made using artificial neural network (ANN) with ensemble technique [13] and particulate matter and weather variables. Predicting weekly pharmaceutical costs in the healthcare industry uses an ensemble approach using numerous sources [14]. For the new station for monitoring air quality, the ensemble approach [15] produced cumulative predictions utilising data from numerous sources in order to address the issue of data scarcity. Deep learning proved to work well for time series classification (TSC) problems [16], [17]. A hybrid deep learning model was proposed [18] for air quality prediction, but there is still a lot to be explored in this domain. Additionally, there is a dearth of research on the development of air pollution prediction models employing data from many source stations.

Studies have explored popular machine learning methods in this domain. Methods like ANN [19], multi-layer perceptron (MLP) [20], support vector machines (SVM) [21], genetic algorithms (GA) [22], manifold learning [23], and support vector regression (SVR) [24]. Although these models have achieved acceptable performance, they lack the ability to capture long term dependencies accurately in time-series data for air quality prediction. Deep learning models for time-series patterns can be well captured by advanced types of networks, such as the long short-term memory (LSTM) network [25], recurrent neural network (RNN), the encoder-decoder model [26], the bidirectional GRU model [27], gated recurrent unit (GRU) [28], stacked autoencoder (SAE) [29]. Also, LSTM and bidirectional long short-term memory (BDLSTM) architectures with combined layers have produced good results [30].

This paper proposes a chaining model using a transfer learning-based approach to solve the training data insufficiency issue at a new station. The novel approach builds pre-trained models on nearby multiple source stations for air quality prediction. These models are chained together in an additive manner to perform prediction at the target station with performance improvement. The contributions of this work are summarized as follows:

- Proposed novel transfer learning-based approach to tackle the issue of insufficient training samples at new station.
- The proposed approach is evaluated on the datasets of multiple stations in Delhi, India, and the effectiveness of this model is determined using performance metrics.
- Proposed approach that gradually improves the model's capacity for forecasting the future.

## 2. BASE MODEL GENERATION

### 2.1. Multi-headed CNN-GRU model

The base model is used to build the best-performing model architecture, which can be used for knowledge transfer. The deep learning base model that is proposed is a multi-headed CNN-GRU architecture, which shows that a separate convolutional layer can be built for every attribute, then the output can be merged together, which is processed further by GRU layers. Model architecture selection for the base model is very important as it captures the trends and patterns in the data. This type of architecture is used in the natural language processing domain [31], signal processing [32], and performs well. Patil *et al.* [33] produce good results for multivariate time-series data. This model is a reliable choice for pre-trained model builders. The model architecture consists of the following layers.

#### 2.1.1. Convolutional layer

For feature extraction, the convolutional neural network performs well [34]. They may produce excellent results for time-series data and are widely used for picture data. CNN training is easier than multi-layer perceptron (MLP) training since fewer weights are used. As the weights are distributed among several heads, it increases the computational efficiency and learning capacity of the model while reducing the number of features for multivariate data. The prediction of air quality is a multivariate problem. For this particular model, each input series variable from the multivariate data is given its own distinct CNN. It is possible to define distinct input models and combine the results of each model into a lengthy vector. As a result of the multi-headed model's ability to learn from individual features across previous time steps [32], we can anticipate improved performance. These days, time series designs are gaining a lot of interest, and the majority of them use single headed structures.

#### 2.1.2. GRU for time-series forecasting

Simple recurrent neural networks (RNNs) in deep learning approaches struggle with gradient vanishing, also known as exploding gradients and gradients that are excessively large. They are unable to recall previous values. Other models, such as LSTM and GRU, are used to resolve this issue. The Forget gate, Update gate, and Output gate are the three gates that the LSTM network uses to learn long sequences. However, training takes a lot of time. The GRU model has two gates, the update gate and the reset gate, with a less complex architecture than LSTM but comparable performance. Update gate (z) maintains the information for the next level, and reset gate (r) maintains the previous state and new information combination [35]. The mathematical model of the GRU architecture is shown in (1) to (4).

$$z_{\mathrm{t}} = \sigma(wz \cdot [x(t), h(t-1)]) \tag{1}$$

$$r_{\mathrm{t}} = \sigma(wr \cdot [x(t), h(t-1)]) \tag{2}$$

$$\tilde{h}(t) = \sigma(wh \cdot [x(t), (r_{\mathrm{t}} \cdot h(t-1))]) \tag{3}$$

$$h(t) = (1 - z_{\mathrm{t}}) \cdot h(t-1) + z_{\mathrm{t}} \cdot \tilde{h}(t) \tag{4}$$

Where the activation function here is $\sigma$, input is x(t), previous output is h(t-1), weights of the reset gate, update gate, and candidate output are wr, wz and wh respectively.

The multi-headed CNN-GRU model architecture is shown in Figure 1. Pre-processing and transformations are carried out on raw pollutant concentration data to account for missing values. Min-max scaling is used to scale multiple feature data before they are transformed into supervised learning series. This information predicts the current step (t) and lagged observations (t-1). The features take into account the order from left to right and predict at the last right. You can apply convolution 1D to this reframed data in time steps, applying it to each attribute separately. Filters = 64, kernel = 2, strides = 1, and rectified linear unit (ReLu) is the activation function for each convolution. In this case, the filters are not predefined but rather learned over time. When it moves one cell forward across a 1D sample, a kernel of size 2 is equivalent to a window of size 2. Concatenation is done with all of CNN's 2D outputs. This output is sent to the 50-neuron GRU layer before being sent to the 20-neuron dense layer, which has an activation function for a single output. As it is more effective than the sigmoid and tanh functions, the ReLu activation function is utilized in this instance. Additionally, utilizing this function prevents simultaneous activation of all neurons, which is why deep learning algorithms favor it.

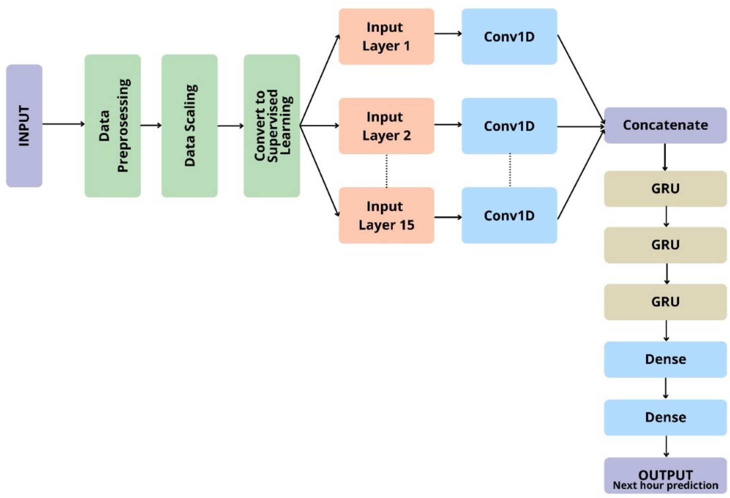


Figure 1. Multi-headed CNN-GRU model architecture

### 2.2. Model evaluation

Root mean square error (RMSE) [36], which measures the difference between actual and predicted values, is the metric employed in the study to assess the performance of the models. The more accurate the forecast, the lower the error value. Mean absolute error (MAE) [21], a loss function that captures prediction error during training, is another metric that is employed. The performance improves as the value decreases. The mean squared error (MSE) metric determines how closely the fitted line corresponds to the real data points. The dataset is better fitted with a smaller value. Another R-square metric is employed to determine model fit for data with values between 0 and 1. A larger value, close to 1, is better for the ability of the model to predict the trend. The mathematical expressions for these metrics are shown in (5) to (8).

$$RMSE_{(y',y)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y'_i - y_i)^2} \tag{5}$$

$$MAE_{(y',y)} = \frac{1}{n}\sum_{i=1}^{n}|(y'_i - y_i)| \tag{6}$$

$$MSE_{(y',y)} = \frac{1}{n}\sum_{i=1}^{n}(y'_i - y_i)^2 \tag{7}$$

$$R - Square = r^2 \tag{8}$$

Where n is the number of samples, $y_i$ are observed values, $y'_i$ are modeled values, and r is the correlation coefficient. The proposed multi-headed CNN-GRU model is evaluated with the other deep learning architectures to prove its performance [15], [33]. It performs best when compared with the other models: CNN, GRU, LSTM, bidirectional LSTM, multi-headed GRU, multi-headed CNN-LSTM, and multi-headed CNN-GRU. From the study [15], it is observed that the multi-headed CNN-GRU model is performing best and is hence used as a base model architecture for transfer learning.

### 2.3. Transfer learning

The model is trained using a transfer learning (TL) approach, and the model's weights are stored. Weights from the source dataset can be used to adjust the destination dataset. Different freezing techniques might be used during the process. Layers at the source station can be frozen, which prevents weight changes for those layers when training is completed at the target station. The various freezing procedures are employed to assess the performance of the model. The following steps are followed during experimentation:

i) A target dataset with hourly granularity is identified as $T_{hourly}$. The experiment would employ the remaining N datasets ($(S_i^{hourly}, i \in (1, N+1), S_i^{hourly} \neq T_{hourly})$ as source datasets.
ii) The base model architecture is trained on the first source dataset with the selected train and test parameter sizes, and the model loaded with the best weights is saved as .h5 file.
iii) Different freezing strategies are employed for transferring the patterns learned by the base model once the pre-trained model has been loaded from the .h5 file.
iv) The model's best weights are preserved by model checkpoint, loaded during testing, and the RMSE metric and goodness of fit are assessed for each freezing approach. For each test sample, the best weights generate a list of predictions, and the results are averaged to produce the final model RMSE score. The model is pre-trained using a single source, and the model that performs the best is saved.

### 2.4. Transfer learning using chaining approach

Transfer learning using the chaining approach can be used to transfer knowledge from multiple nearby source stations to a target station, in order to determine whether an additive effect in terms of RMSE improvement can be obtained. The idea can be described as chaining together the models trained on different source stations and then fine-tuning the consolidated model on the target dataset. The models chosen are in increasing order of their RMSE results on the target dataset. The best chained model with the lowest RMSE value is considered for model building at the new station. Multiple sources of data can be merged together and used for learning, but this novel approach considers only those stations in a chained additive manner, which can lead to a model with better learning capability at the newer station. The chaining algorithm is as shown in Algorithm 1.

**Algorithm 1** Chaining algorithm

**Input:** $N$ models $i$ pre-trained on source dataset $S_i^{hourly}$, a parameter $\eta$
**Output:** Weights $w_i$ for each model $i$
1: Sort the source stations for a given target station from best to worst. $T_{hourly}$
2: **for** each source station $S^{hourly}$ **do**
3: The best weights gained during training are preserved.
4: Another $S_{i+1}^{hourly}$ source dataset is loaded and the model object is initialized with $S_i^{hourly}$ saved weights. The best weights gained during training are preserved to .sav file.
5: The consolidated model pre-trained on multiple source stations is finetuned on the target dataset $T_{hourly}$ using a freezing strategy.
6: **end for**
7: **return** $w_i$ for each $i$

### 2.5. Data collection and pre-processing

Dataset [37] considered for evaluation to test for its performance is collected from the central pollution control board (CPCB), India's official online portal. The attributes are meteorological attributes: ambient temperature, relative humidity, solar radiation level, wind direction, wind speed, and air pollutant attributes: PM10, PM2.5, ozone, benzene, toluene, CO, NO, $NO_2$, NOX, $SO_2$, and $NH_3$. The raw dataset obtained for Delhi, India, from the CPCB website has a number of null values. Linear interpolation is used to fill in the missing values. Scaling is performed, and training and test sets are created. Data is converted to a supervised learning series [27].

## 3. RESULT AND DISCUSSION

Deep learning models used in experimentation use TensorFlow on the backend and the Keras framework. Models are using the Adam optimizer and mean squared error (MSE) as a loss function. The study was performed on five source stations in New Delhi, India. Here, 16 months of data are used for source stations, while the target has only 2 months of data. The RMSE value for the target station, Najafgarh, with 2 months of data without transfer learning is 11.48 using the base model (multi-headed CNN-GRU model). The poor fit of the model can be seen in Figure 2.

Chaining is performed on 5 stations, considering their prediction ability on Najafgarh station as in the single source TL of section 4. The results obtained after chaining model application are shown in Table 1 and Table 2. It is observed that stations Okhala and DCStadium chained together can improve the performance of the model using freezing strategy 2 as shown in Table 2 and reduce the RMSE tremendously from a 11.48 value to 3.82 using the chaining approach with 66.72% improvement in performance accuracy. The best fit of the model can be seen in Figure 3.

The best model obtained is obtained by using freezing strategy 2 and chaining together stations Okhala and DCStadium, as shown in Table 2. The best model is saved and finetuned on the 2 months of data at Najafgarh station, and the results obtained are as shown in Table 3. In the study, lower values of RMSE, MAE, and MSE show better performance of the model on the target station dataset using freezing strategy 2. Also, the higher R-square value of the model using freezing strategy 2 shows a better fit of the model to the data. The comparison of the performances before and after applying the chaining model approach is shown in Figure 4.

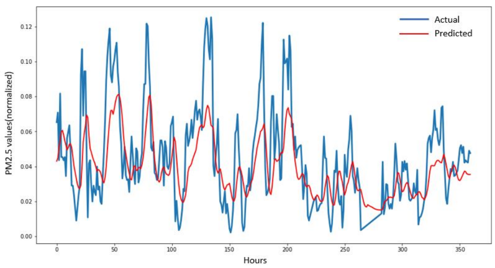


Figure 2. Base model poor fit on 2 months of Najafgarh monitoring station, Delhi, India

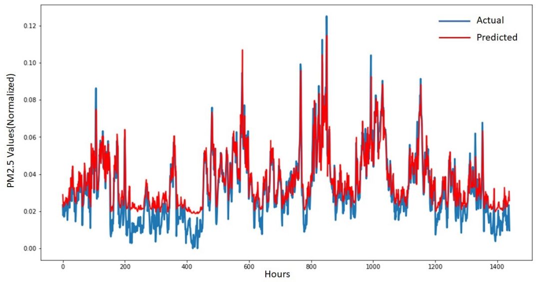


Figure 3. Chaining model fit on 2 months a of Najafgarh monitoring station, India

Table 1. Performance of TL using chaining approach on target station as Najafgarh, Delhi, India using freezing strategy 1

| Chained stations | Freezing strategy 1 |
|---|---|
| Okhala | 11.42 |
| Okhala+AshokVihar | 9.33 |
| Okhala+AshokVihar+DCStadium | **8.96** |
| Okhala+AshokVihar+DCStadium +Dwarka | 10.38 |
| Okhala+AshokVihar+DCStadium +Dwarka+NehruNagar | 12.07 |

Table 2. Performance of TL using chaining approach on target station as Najafgarh, Delhi, India using freezing strategy 2

| Chained stations | Freezing strategy 2 |
|---|---|
| Okhala | 10.30 |
| Okhala+DCStadium | **7.86** |
| Okhala+DCStadium+AshokVihar | 10.86 |
| Okhala+DCStadium+AshokVihar +NehruNagar | 15.35 |
| Okhala+DCStadium+AshokVihar +NehruNagar+Dwarka | 10.48 |

Table 3. Performance of multiple sources TL using chaining approach on target station as Najafgarh, Delhi, India

| Error measures | Freezing strategy 1 | Freezing strategy 2 |
|---|---|---|
| RMSE | 4.47 | **3.82** |
| MAE | 3.65 | **3.33** |
| MSE | 0.94 | **0.92** |
| R-square | 0.90 | **0.94** |

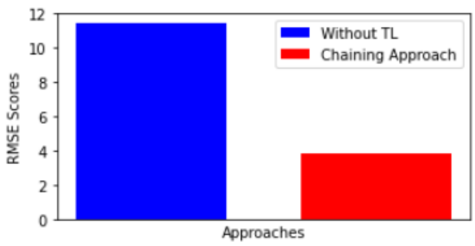


Figure 4. Comparison of performance of approaches

The incremental learning process involves upgrading a model's knowledge over time. With fewer data points in this process, learning might be quicker and more precise. Our proposed model, constructed at a new station, has to learn more quickly with the new air pollution data available without forgetting any previously learned information. This study suggests an incremental modeling strategy for future model learning. This can be done by first employing a chaining strategy to train the model on various source datasets of adjacent stations to the new station. At the target station, the model is saved and used for forecasting. The model is reloaded so that it can continue to learn from new data. To do this, the model can be trained using the previously saved base model architecture after being fed data in batches. The best model can be saved using a model checkpoint. The model with the best weights is saved after each iteration and used for future learning, allowing the model to develop its own patterns over time and produce improved predictions.

## 4. CONCLUSION

In a smart city, multiple air quality monitoring stations capturing similar pollution patterns and trends can be identified and chained together for forecasting pollution levels at newer stations with training data insufficiency problems. A transfer learning-based chaining model approach is proposed for this purpose. Transfer learning is executed using two freezing strategies to evaluate its effect on knowledge transfer at the new station. The chaining model showed 66.72% improvement in its performance. The results obtained from this work are encouraging and will hopefully fuel future research in the domain of pre-trained models for air pollution forecasting. Transfer learning also helps in incremental model learning and pattern capturing at the new station, making it ready to be self-sufficient over time. The model can be tested on a huge and increasing number of datasets.

## REFERENCES


[1] P. Jiang, C. Li, R. Li, and H. Yang, "An innovative hybrid air pollution early-warning system based on pollutants forecasting and Extenics evaluation," *Knowledge-Based Systems*, vol. 164, pp. 174–192, Jan. 2019, doi: 10.1016/j.knosys.2018.10.036.

[2] I. H. Fong, T. Li, S. Fong, R. K. Wong, and A. J. Tallón-Ballesteros, "Predicting concentration levels of air pollutants by transfer learning and recurrent neural network," *Knowledge-Based Systems*, vol. 192, p. 105622, Mar. 2020, doi: 10.1016/j.knosys.2020.105622.

[3] S. Sonawani and K. Patil, "Air quality measurement, prediction and warning using transfer learning based IOT system for ambient assisted living," *International Journal of Pervasive Computing and Communications*, Jan. 2023, doi: 10.1108/IJPCC-07-2022-0271.

[4] J. Ma, J. C. P. Cheng, C. Lin, Y. Tan, and J. Zhang, "Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques," *Atmospheric Environment*, vol. 214, p. 116885, Oct. 2019, doi: 10.1016/j.atmosenv.2019.116885.

[5] "Ambient (outdoor) air pollution." https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health (accessed Aug. 03, 2020).

[6] A. Miles, A. Zaslavsky, and C. Browne, "IoT-based decision support system for monitoring and mitigating atmospheric pollution in smart cities," *Journal of Decision Systems*, vol. 27, no. sup1, pp. 56–67, May 2018, doi: 10.1080/12460125.2018.1468696.

[7] G. Beig *et al*., "India's Maiden air quality forecasting framework for megacities of divergent environments: The SAFAR-project," *Environmental Modelling & Software*, vol. 145, p. 105204, Nov. 2021, doi: 10.1016/j.envsoft.2021.105204.

[8] X. Hu *et al*., "Estimating ground-level PM2.5 concentrations in the southeastern U.S. using geographically weighted regression," *Environmental Research*, vol. 121, pp. 1–10, Feb. 2013, doi: 10.1016/j.envres.2012.11.003.

[9] Y. Chen *et al*., "Local characteristics of and exposure to fine particulate matter (PM2.5) in four indian megacities," *Atmospheric Environment: X*, vol. 5, p. 100052, Jan. 2020, doi: 10.1016/j.aeaoa.2019.100052.

[10] H. M. K. K. M. B. Herath and M. Mittal, "Adoption of artificial intelligence in smart cities: A comprehensive review," *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100076, Apr. 2022, doi: 10.1016/j.jjimei.2022.100076.

[11] W. Zellinger *et al*., "Multi-source transfer learning of time series in cyclical manufacturing," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 777–787, Mar. 2020, doi: 10.1007/s10845-019-01499-4.

[12] F. Zhuang, X. Cheng, S. J. Pan, W. Yu, Q. He, and Z. Shi, "Transfer learning with multiple sources via consensus regularized autoencoders," in *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2014*, Berlin: Springer, 2014, pp. 417–431.

[13] R. Medrano, V. B. Remiro, and J. L. Aznarte, "SOCAIRE: forecasting and monitoring urban air quality in Madrid," *Environmental Modelling & Software*, vol. 143, p. 105084, Sep. 2021, doi: 10.1016/j.envsoft.2021.105084.

[14] S. Kaushik, A. Choudhury, N. Dasgupta, S. Natarajan, L. A. Pickett, and V. Dutt, "Ensemble of multi-headed machine learning architectures for time-series forecasting of healthcare expenditures," in *Applications of Machine Learning. Algorithms for Intelligent Systems*, Singapore: Springer, 2020, pp. 199–216.

[15] A. Dhole, I. Ambekar, G. Gunjan, and S. Sonawani, "An ensemble approach to multi-source transfer learning for air quality prediction," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Feb. 2021, pp. 70–77, doi: 10.1109/ICCCIS51004.2021.9397138.

[16] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.06995.

[17] J. C. B. Gamboa, "Deep learning for time-series analysis," Jan. 2017, [Online]. Available: http://arxiv.org/abs/1701.01887.

[18] S. Bhanja and A. Das, "A hybrid deep learning model for air quality time series prediction," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 22, no. 3, pp. 1611–1618, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1611-1618.

[19] V. Yadav and S. Nath, "Novel hybrid model for daily prediction of PM10 using principal component analysis and artificial neural network," *International Journal of Environmental Science and Technology*, vol. 16, no. 6, pp. 2839–2848, Jun. 2019, doi: 10.1007/s13762-018-1999-x.
[20] M. Gardner, "Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London," *Atmospheric Environment*, vol. 33, no. 5, pp. 709–719, Feb. 1999, doi: 10.1016/S1352-2310(98)00230-1.
[21] R. Chuentawat and Y. Kan-ngan, "The comparison of PM2.5 forecasting methods in the form of multivariate and univariate time series based on support vector machine and genetic algorithm," in *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Jul. 2018, pp. 572–575, doi: 10.1109/ECTICon.2018.8619867.
[22] Z. Wang and Z. Long, "Pm2.5 prediction based on neural network," in *2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Sep. 2018, pp. 44–47, doi: 10.1109/ICICTA.2018.00018.
[23] J. Xie, "Deep neural network for PM2.5 pollution forecasting based on manifold learning," in *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, Aug. 2017, pp. 236–240, doi: 10.1109/SDPC.2017.52.
[24] A. S. Sánchez, P. J. G. Nieto, P. R. Fernández, J. J. C. Díaz, and F. J. Iglesias-Rodríguez, "Application of an SVM-based regression model to the air quality study at local scale in the Avilés urban area (Spain)," *Mathematical and Computer Modelling*, vol. 54, no. 5–6, pp. 1453–1466, Sep. 2011, doi: 10.1016/j.mcm.2011.04.017.
[25] Q. Wu and H. Lin, "A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors," *Science of The Total Environment*, vol. 683, pp. 808–821, Sep. 2019, doi: 10.1016/j.scitotenv.2019.05.288.
[26] L. Yan, Y. Wu, L. Yan, and M. Zhou, "Encoder-decoder model for forecast of PM2.5 concentration per hour," in *2018 1st International Cognitive Cities Conference (IC3)*, Aug. 2018, pp. 45–50, doi: 10.1109/IC3.2018.00020.
[27] S. Sonawani, K. Patil, and P. Chumchu, "NO2 pollutant concentration forecasting for air quality monitoring by using an optimised deep learning bidirectional GRU model," *International Journal of Computational Science and Engineering*, vol. 24, no. 1, p. 64, 2021, doi: 10.1504/IJCSE.2021.113652.
[28] V. Athira, P. Geetha, R. Vinayakumar, and K. P. Soman, "DeepAirNet: applying recurrent networks for air quality prediction," *Procedia Computer Science*, vol. 132, pp. 1394–1403, 2018, doi: 10.1016/j.procs.2018.05.068.
[29] X. Li, L. Peng, Y. Hu, J. Shao, and T. Chi, "Deep learning architecture for air quality predictions," *Environmental Science and Pollution Research*, vol. 23, no. 22, pp. 22408–22417, Nov. 2016, doi: 10.1007/s11356-016-7812-9.
[30] R. Xu, D. Wang, J. Li, H. Wan, S. Shen, and X. Guo, "A hybrid deep learning model for air quality prediction based on the time–frequency domain relationship," *Atmosphere*, vol. 14, no. 2, p. 405, Feb. 2023, doi: 10.3390/atmos14020405.
[31] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, Mar. 2017, doi: 10.1049/iet-its.2016.0208.
[32] S. Kaushik, A. Choudhury, V. Dutt, N. Dasgupta, S. Natarajan, and L. A. Pickett, "Evaluating single- and multi-headed neural architectures for time-series forecasting of healthcare expenditures," in *Computational Intelligence*, De Gruyter, 2020, pp. 159–176.
[33] K. Patil, S. Sonawani, and P. Chumchu, "Multi-headed CNN-GRU model for particulate matter (PM2.5) prediction in smart cities," *International Journal of Environment and Waste Management*, vol. 1, no. 1, p. 1, 2023, doi: 10.1504/IJEWM.2023.10039181.
[34] S. Dara and P. Tumma, "Feature extraction by using deep learning: a survey," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Mar. 2018, pp. 1795–1801, doi: 10.1109/ICECA.2018.8474912.
[35] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air pollution forecasting using a deep learning model based on 1D Convnets and bidirectional GRU," *IEEE Access*, vol. 7, pp. 76690–76698, 2019, doi: 10.1109/ACCESS.2019.2921578.
[36] A. S. Sayegh, S. Munir, and T. M. Habeebullah, "Comparing the performance of statistical models for predicting PM10 concentrations," *Aerosol and Air Quality Research*, vol. 14, no. 3, pp. 653–665, 2014, doi: 10.4209/aaqr.2013.07.0259.
[37] S. Sonawani and K. Patil, "Delhi multi-site air-quality data set," 2021, doi: 10.17632/bzhzr9b64v.1.

## BIOGRAPHIES OF AUTHORS

**Shilpa Sonawani** is a Ph.D. student of Vishwakarma University, currently working as assistantprofessor in School of Computer Engineering and Technology, MITWPU, Pune, India. Her research interests include machine learning and data analysis. She can be contacted at email: shilpa.sonawani-024@vupune.ac.in.

**Kailas Patil** is an alumnus of National University of Singapore, currently working at Vishwakarma University, Pune, India. His research interests include cyber security, ubiquitous computing, internet of things, and embedded systems. He has been a reviewer of many international journals. He can be contacted at email: kailas.patil@vupune.ac.in.