

# Air Quality Prediction using Voronoi-Based Spatial Temporal Sequence Similarity with Conjugate Gradient Enabled Sparse Autoencoder Deep Learning

P. Shree Nandhini<sup>1</sup>, Dr. P. Amudha<sup>2</sup>

<sup>1</sup>Ph.D. Research Scholar

Department of Computer Science and Engineering, School of Engineering  
Avinashilingam Institute for Home Science and Higher Education for Women  
Coimbatore, India  
shreenandhini2016@gmail.com

<sup>2</sup>Professor

Department of Computer Science and Engineering, School of Engineering  
Avinashilingam Institute for Home Science and Higher Education for Women  
Coimbatore, India  
amudharul@gmail.com

**Abstract**—Air Quality Prediction (AQP) remains a difficult task because of multidimensional nonlinear spatiotemporal features. To solve this issue, an Improved Sparse Autoencoder with Deep Learning (ISAE-DL) and Enriched ISAE-DL (EISAE-DL) models were developed with the combination of concentric circle-based clustering, Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) followed by the ISAE for AQP. In EISAE-DL, concentric circle-based clustering uses Manhattan distance to efficiently split the locations into four regions using its center and cluster the spatially and temporally similar candidate locations. But it was considered a fixed structure and may struggle to find variations in several data points. Also, it accommodate clusters with regular and circular patterns, whereas irregular and non-circular cluster patterns were not handled. Similarly, the ANN inference was often offended or ignored because of complex meteorological characteristics. Hence, this paper proposes a Voronoi-based spatial-temporal sequence similarity with the Conjugate gradient-enabled SAE-DL (VCSAE-DL) model for effective AQP. First, a Voronoi clustering is performed by creating the Voronoi diagram for analogous candidate location clustering. Then, the resultant clusters of location data along with the PM2.5 and other meteorological data are given to the Improved ANN (IANN), and the target stations are given to the LSTM to capture the spatiotemporal relationship features and temporal features, respectively. Also, CNN is used to extract relationships between terrain and air quality features. These extracted features are fused in the merge layer and transferred to the ISAE for final prediction of air quality. Finally, the test outcomes demonstrate that the VCSAE-DL achieves better prediction performance compared to the existing AQP models.

**Keywords**-Air quality prediction, EISAE-DL, Concentric circle-based clustering, Voronoi-based clustering, Conjugate gradient.

## I. INTRODUCTION

People are starting to worry more about air quality due to the health risks posed by high levels of air contamination [1]. PM2.5, PM10, and Sulfur Dioxide (SO<sub>2</sub>) monitoring stations are among the many that have been set up in various parts of the world to keep the public informed about air quality. But the needs of the public cannot be met by the data available in real-time. Predicting the air quality in a specific area using historical data, such as meteorological data and historical air quality data, is the primary goal of AQP [2]. AQP has been proposed using several different approaches. At first, the prediction model is built using empirical assumptions, such as those used in air pollution dispersion models. Predictions of air quality are made using machine learning techniques like linear regression [3] and ANNs [4]. But such methods are imperfect due to a wide

variety of factors, including weather, pollutants, and traffic patterns. The sensors are the source of data for the air quality dataset. There is a lot of unexpected or erratic information in these data. Anomaly information degrades the performance of a Spatio-Temporal Deep Neural Network (ST-DNN) [5] since it is not handled in the network. To learn the distribution of the data over several dimensions, the ISAE-DL and EISAE-DL models [6] were built. In these models, a mixture of continuous and discrete characteristics was used. Then, the k-Nearest Neighbor Euclidean Distance (kNN-ED) and the kNN-Dynamic Time Wrapping Distance (kNN-DTWD) were used to find spatially and temporally related data. But the delay interval is important for long-term predictions.

To deal with the time delay at the location in the prediction, a concentric circle-based distance partition approach was proposed. In this method, similar sites in space

and time were clustered using the Manhattan distance. It discovered the center of the four groups of sites that are most similar to each other in terms of both space and time. Zero was used as the starting point for this center. From this central point, the rest of the distances between places with similar locations and times can be determined. The k-NN clustering, which uses Euclidean distance, and clustering based on concentric circles were two distinct approaches. In this case, a single centroid value was utilized to classify locations into distinct categories. Using standard centric and Manhattan distance measures, similar places were grouped. The PM<sub>2.5</sub> and meteorological data, as well as the results of the concentric circles and the information about the target stations, were fed into the ANN, the LSTM, and the CNN, and the combined results were then sent to the ISAE-based Feed Forward Neural Network (FFNN) to predict the air quality.

On the other hand, it was considered a fixed structure and may struggle to find variations in many data points. Also, it accommodate clusters with regular and circular patterns, whereas irregular and non-circular cluster patterns were not handled. Similarly, the ANN inference was often offended or ignored because of complex meteorological characteristics. Therefore, this manuscript proposes the VCSAE-DL model for effective AQP. First, a Voronoi clustering is performed by creating the Voronoi diagram for clustering analogous candidate places. Then, the resultant clusters of location data along with the PM<sub>2.5</sub>, other meteorological data, and target stations are given to the IANN based on the Conjugate Gradient-enabled Artificial Backpropagation neural Network (CG-ABN), LSTM, and CNN for feature extraction. These extracted features are fused in the merge layer and transferred to the ISAE for AQP.

The residual sections are prepared as the following: Various research on AQP is the focus of Section II. The VCSAE-DL model for AQP is described in Section III. The effectiveness of its performance is shown in Section IV. Section V summarizes the article with future scope.

## II. LITERATURE SURVEY

To forecast PM<sub>2.5</sub> levels, a Spatial-Temporal Support Vector Regression (STSVR) has been presented in [7]. Spatial grouping was first used on the data to deal with the issue of spatial heterogeneity. Then, spatial autocorrelation variables were proposed to be part of input features by using the Gauss vector weight function. Finally, a sub-area of AQP depending on the regional SVR with space autocorrelation factors was built. However, the space-time clustering technique will be investigated to accurately identify space-time heterogeneity with spatial heterogeneity.

An integrated LSTM and CNN model using historical and current data on air pollution and weather conditions [5]

was presented for AQP. However, due to the noise, the overall prediction performance suffers when all locations are included. A sophisticated Spatial-Temporal Ensemble (STE) model was developed [8] for AQP. Using an ensemble technique and a partitioning approach based on weather patterns, this model dynamically trains and integrates numerous independent models. After that, spatial data was generated and analyzed using Granger causalities to determine where stations were associated with one another. The long- and short-term air quality dependencies were then trained by the LSTM. However, LSTM cannot retain data for extended periods.

To predict the air quality a hybrid learning approach was developed [9]. Air quality index raw data was initially processed using Wavelet Packet Decomposition (WPD) to create an air quality index low-frequency subseries. To provide accurate predictions for the subseries, a Modified Extreme Learning Machine (MELM) was trained with pigeon-inspired weight and threshold optimization. To further establish the predictions by frequency, Multidimensional Scaling and K-means (MSK) clustering schemes were used. The ultimate result of air quality index prediction was achieved by processing the subseries together using MELM. The accuracy of future predictions will be improved with the help of DL models.

To forecast the PM<sub>2.5</sub> absorptions at various places in Beijing [10], The Light Gradient Boosting Machine (LightGBM) approach was developed. When generating the dataset for AQP, it merged historical information with predictive information to extract temporal features that enhanced prediction accuracy. To better pinpoint features that could be crucial, multi-dimensional statistical features were considered. Also, high-dimensional time features were built by intensively mining the time association, and a sliding window was used for dimensionality reduction of training data to boost the model's prediction accuracy. But it was not suitable for large-scale datasets.

A complete prediction model using LSTM was developed to accurately monitor the AQP [11]. Particle concentration data from the existing monitoring system, data from neighboring stations, weather data, and data on gaseous pollutants in the air were integrated to create a more accurate picture of air quality. The information was translated into a supervised learning format after being normalized. Predictions of the air quality were then made using the LSTM. But it has a gradient vanishing issue that impacts the model efficiency.

A context prediction model was presented for AQP [12]. This model integrated information from the user's health profile and nearby pollution sources with an accurate air pollution prediction method utilizing LSTM-DNN, taking into account the principles of context-aware computing. To

further enhance the accuracy of AQPs, this model was integrated with a My Air Quality Index (MyAQI). Inaccuracies were introduced, however, when real-world data was used due to the possibility of disruptions or faulty measurement stations.

Predicting PM2.5 for measuring the air pollution level using a semi-supervised bidirectional LSTM was proposed in [13]. This procedure needs fine PM2.5 as input and then used Empirical Mode Decomposition (EMD) to decompose the data and extract the amplitude and frequency properties. Also, the bidirectional LSTM model was used for final predictions. However, certain unusual values of Mean Absolute Percentage Error (MAPE) were recorded as a result of the uneven numerical distribution of fine particulate matter.

The Neuro-fuzzy network method was developed [14] for predicting air quality. At first, the training data were partitioned into fuzzy clusters whose membership function was specified by means and variances. Then, a four-layer fuzzy neural network was built using the fuzzy rules extracted from the fuzzy clusters. Finally, the network was trained for AQP using the steepest descent backpropagation, genetic, and particle swarm optimization algorithms. But the performance depends on the membership function of the Neuro-fuzzy network.

A transfer learning-based bidirectional LSTM was proposed for predicting air quality [15]. This method combined sophisticated DL models and transfer learning mechanisms to relocate the data from historical locations to new locations for enhancing the performance of AQP. But it did not learn spatiotemporal relationships among various data.

### III. PROPOSED METHODOLOGY

This section briefly describes the proposed VCSAE-DL for predicting air quality. The place with the strongest spatio-temporal connection to the goal location is identified first, and then the location with the best air quality is predicted. To improve AQP, it is necessary to retrieve key elements by identifying temporal and spatial characteristics. Assume there is a set of locations  $L = \{l_1, l_2, \dots, l_n\}$  and a set of features  $F = \{f_1, f_2, \dots, f_m\}$  that are useful in identifying these places. The coordinates of a given location are based on its latitude and longitude data, which are given in the form of,

$$Ll_i = (l_i, p_i, q_i) \quad Ll_i \in Ll_i(1)$$

The coordinates for the latitude and longitude at point  $l_i$  are given by  $p_i$  and  $q_i$  respectively, in Eq. (1). Voronoi diagram is used to find the place closest to the target, while the top h relevant places are generated based on the data used to train the algorithm.

#### A. Voronoi Clustering

A multidimensional Voronoi diagram is a splitting of the location area  $\mathbb{R}^d$  into areas  $R_j$  containing the possessions: all centers  $c_j$  lie exactly in one area  $R_j$ , which comprises each point  $x \in \mathbb{R}^d$  that are nearer to  $c_j$  than any other center  $c_k, j \neq k$ :

$$R_j = \{x \in \mathbb{R}^d: \|x - c_j\| < \|x - c_k\|, \forall j \neq k\} \quad (2)$$

The centers  $c_j$  are called Voronoi points. The area related to a Voronoi point  $c_j$  is called a cell. Since a cell comprises each point, which are nearer to its centroid compared to another centroid, the cell boundaries locate accurately in the mid of 2 centroids. Fig. 1 shows the Voronoi diagram for a dataset, which comprises 6 independent clusters.

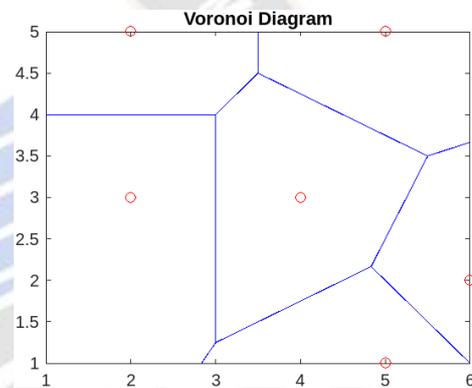


Figure 1. Voronoi Diagram for Six Clusters

Given a set of  $L \subseteq \mathbb{R}^d$  to be clustered, the clustering is begun with creating the Voronoi diagram for  $L$ . Consider the 2D plane. As a given variable, the method is provided a max threshold, which represents the highest value permitted for a cell that may remain merged into a developing cluster. Because cell volume approximates local location area density, merely that counts in deciding if the density is large sufficient for further clustering. Separate clusters can thus develop to any dimension given the local density is adequate. Once each cell of at most volume max has been considered, only comparatively huge cells are retained.

All points in the data Voronoi diagram form their cells. Following that, the cell volume is determined. This study calculates the appropriate cell region in the plane as the polygon region, yet in larger sizes  $d$ , it simply approximates the cell to be a hyperball with volume  $(\pi^{n/2} r^d) / \Gamma(d/2 + 1)$ , wherein  $\Gamma(z + 1) = z!$ . The radius  $r$  refers to the mean distance of cell angles from its centroid. This estimation is not more precise in low sizes, yet it

enhances when increasing the size. All cells are related to a class label.

Initially, the moment at which the minimum Voronoi cell is dealt with. Because it contains no identified adjacent cells, this cell is assigned the label 1. The cell examined successively may be a neighbor of the initial cell, as indicated by the distribution of an angle. When the latter cell volume falls below the max threshold, they are merged and the initial cell label is considered. Or else, the 2<sup>nd</sup> cell will become unrelated to the initial, wherein it obtains its label. But it is remaining promising that the 2 locations belong to a similar cluster and will eventually be merged.

In most cases, a cell contains numerous nearby cells with multiple labels and a few that haven't been annotated. The identified adjacent is analyzed based on their dimensions (i.e., label values). The cell under examination is fused to its adjacent with the last label, and its annotated adjacent considers a similar label. Cells are joined when their volume is less than the limit. There is no essential to continue by the residual adjacent cells when the maximum value has been achieved.

After each point has been considered, every cluster combination has been executed. There is no necessity to iterate the procedure. But the clustering post-processing is needed to create it feasible. Assume an illustration in Fig. 1. It is seen that the above-explained cell merging process can identify 6 clusters for different max threshold values. But it is addressed that the farthest cluster points can't be merged, since the clusters could develop together.

Thus, in this case, the heuristics considers the cluster marginal points. Centroids with a cell volume higher than the max threshold can be merged to the nearby adjacent label.

Algorithm for Voronoi Clustering:

- **function(S, max)**
- Create the Voronoi diagram for  $L = \{l_1, l_2, \dots, l_n\}$ ;
- Estimate the Voronoi cell volumes and rank the instances. Consider the resultant rank  $l_1, l_2, \dots, l_n$ ;
- **for**( $i = 1:n$ )
  - If**  $R_i$  volume related to  $s_i$  is maximum
  - Merge  $R_i$  with an adjacent cluster with the smallest class number;
  - If one exists, or else allocate a new class number to the cell;
  - else**
  - Allocate  $R_i$  to the nearest adjacent cluster;
  - end if**
  - end for**
- **end function**

Thus, the top  $h$  sites are chosen through a clustering method based on the Voronoi diagram, and these locations will be used to try and identify the target sequence. To predict future time series, the Temporal Relationship Extractor (TRE) takes attributes of the target historical site. The trends in the desired region are obtained by TRE LSTM, whereas LSTM models the behavior of historical time series. LSTM and IANN are used to collect data at low and high frequencies, respectively.

B. Improved Artificial Neural Network

The CG-ABN structure is split into 4 different layers as shown in Fig. 2, which comprises weight initialization, the error's forward and backward propagation, weight, and bias update. The hidden layer comprises many neurons and all have an activation function as  $f(x) = Sigmoid(x)$ . The activation function obtains the total weighted input ( $\omega_{11} * x_1 + \omega_{21} * x_2 + \omega_{31} * x_3 + \dots + \omega_{n1} * x_n + 1 * b$ ) argument as follows:

$$f \sum_{i=1}^n W_{ij} X_i + b \tag{3}$$

In this IANN, forward propagation is executed initially. Then, backward propagation is prepared once the estimated result  $Y$  is compared with the observed result  $\hat{Y}$  in the computation of the gradient mistake at the final layer.

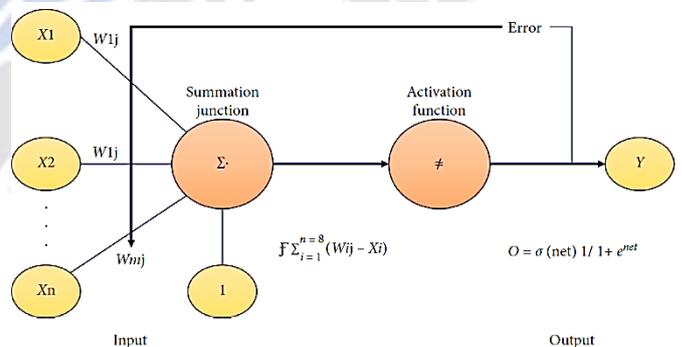


Figure 2. Architecture of CG-ABN-based IANN

The IANN is trained with air pollutants dataset utilizing the below schemes:

- Multi-Layer Perceptron (MLP)
- Bayesian normalized
- Scaled conjugate gradient

The IANN model comprises a single input, hidden, and output layer comprising  $n_{input}$ , which create  $L_{output}$ . The database has  $(X_i, Y_i)$ , wherein  $X_i$  denotes the input layer and  $Y_i$  denotes the estimated result. When  $n$  denotes the overall dimension of a whole set,

$$X_n = \{(x_i, y_i), \dots, (x_n, y_n)\} \tag{4}$$

The variables of feed-forward are represented as  $\theta$ . As learning the ANN contains the gradient of the error function  $E = (X, \theta)$  related to the weight  $w_{ij}^k$  at node  $j$  in layer  $k$  for node  $i$ , biases  $b_i^k$  at  $i$  in  $k$ , therefore according to training rate, gradient descent modifies the weights at all iterations  $t$  by

$$\theta_t = \theta_t - \alpha \frac{\partial E(X, \theta)}{\partial \theta_t} \tag{5}$$

In Eq. (5),  $\theta_t$  is the ANN variables at  $t$ . The Mean Square Error (MSE) in backpropagation is

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i) \tag{6}$$

In Eq. (6),  $\hat{y}_i$  refers to the estimated result and  $y_i$  denotes the observed result of  $x_i$ . The derivative of  $f(x)$  and sigmoid function is denoted by  $f'(x)$  and  $\sigma'(x)$ .  $b(k/i)$  in  $k^{th}$  at  $i$  is added to the weight as  $w_{o_i}^k$  with a result  $o_o^{k-1} = 1$  at node 0 in  $k - 1$ ; so,

$$w_{o_i}^k = b_i^k \tag{7}$$

$$\vec{W} = \begin{pmatrix} w_{00} & \dots \\ w_{0m} & \dots \\ w_{10} & \dots \\ w_{1m} & \dots \\ \vdots & \vdots \\ w_{n0} & \dots \\ w_{nm} & \dots \\ w_{n+1,0} & \dots \\ \vdots & \vdots \end{pmatrix} \tag{8}$$

Weight and bias ranges are set with arbitrary values; a modified weight is applied in consequent  $t$ . Eq. (8) is rewritten as:

$$\vec{W} = \begin{pmatrix} \vec{W}_{IH} \\ \vdots \\ \vec{W}_{HO} \end{pmatrix} \tag{9}$$

Eq. (9) demonstrates that  $\vec{W}_{IH}$  includes the weight from the input to hidden layer, whereas  $\vec{W}_{HO}$  denotes the weight defining from hidden to output layer.

$$\vec{W}_{IH} = \begin{pmatrix} w_{00} & w_{01} & \dots & w_{0m} \\ w_{10} & w_{11} & \dots & w_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,0} & w_{n,1} & \dots & w_{0m} \\ w_{00} & w_{11} & \dots & \vdots \end{pmatrix} \tag{10}$$

Eq. (10) includes the weights from the input to hidden layer. It has  $n \times m$  components and  $W_{a,b}$  denotes the weight among components.

$$\vec{W}_{HO} = \begin{pmatrix} w_{n+1,0} & w_{n+1,1} & \dots & w_{n+1,m} \\ w_{n+1,0} & w_{11} & \dots & w_{n+2,m} \\ \vdots & \ddots & \ddots & \vdots \\ w_{n+k,0} & w_{n+k,1} & \dots & w_{n+km} \\ w_{00} & w_{11} & \dots & \vdots \end{pmatrix} \tag{11}$$

In this study, only one output neuron is used. Eq. (11) is rewritten as:

$$\vec{W}_{HO} = (w_{n+1,0} \quad w_{n+1,1} \quad \dots \quad w_{n+1,m}) \tag{12}$$

Corresponding to the original formulation  $J_{k-1}$ ,

$$a_i^k = b_i^k + \sum_{j=1}^{J_{k-1}} w_{ji}^k o_j^{k-1} = \sum_{j=0}^{J_{k-1}} w_{ji}^k o_j^{k-1} \tag{13}$$

After that, the error is determined by

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \tag{14}$$

After taking the derivative of Eq. (14),

$$\frac{\partial E(X, \theta)}{\partial w_{ji}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ji}^k} \left( \frac{1}{2} (\hat{y}_i - y_i)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ji}^k} \tag{15}$$

The backpropagation scheme is associated with a single I/O set and each I/O in  $X$  can be generated by fusing all gradients. For derivation,  $E$  is

$$E = \frac{1}{2} (\hat{y} - y)^2 \tag{16}$$

*Error Function Derivatives:* Once the chain principle is applied,

$$\frac{\partial E_d}{\partial w_{ji}^k} = \frac{\partial E}{\partial a_j^k} \frac{\partial a_j^k}{\partial w_{ij}^k} \tag{17}$$

In Eq. (17),  $a_j^k$  acts as activation of  $j$  in  $k$ .

$$\delta_j^k = \frac{\partial E}{\partial a_j^k} \tag{18}$$

The 2<sup>nd</sup> expression from Eq. (17) for  $a_j^k$  is

$$\frac{\partial a_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} \left( \sum_{i=0}^{J_{k-1}} w_{ij}^k o_i^{k-1} \right) = o_i^{k-1} \tag{19}$$

The partial derivative of  $E$  w.r.t  $w_{ij}^k$  is as:

$$\frac{\partial E}{\partial a_j^k} = \delta_j^k o_i^{k-1} \tag{20}$$

*Output Layer:* Backpropagation defines  $\varphi_1^m$ , wherein  $m$  denotes the last layer. 4-layer ANN possess  $m = 3$

for the last layer and  $m = 2$  for the 2<sup>nd</sup> to the final layer. Defining  $E$  in terms of  $a_1^m$  provides

$$E = \frac{1}{2}(\hat{y} - y)^2 = E \frac{1}{2}(g_0(a_1^m) - y)^2 \quad (21)$$

After, applying the partial derivative,

$$\delta_j^m(g_0(a_1^m) - y)g_0'(a_1^m) = (\hat{y} - y)g_0'(a_1^m) \quad (22)$$

The partial derivative of  $E$  w.r.t  $w_{i1}^m$  is

$$\frac{\partial E}{\partial w_{i1}^m} = \delta_1^m o_i^{m-1} = (\hat{y} - y)g_0'(a_1^m)o_i^{m-1} \quad (23)$$

Hidden Layer: The hidden layers'  $E$  is determined by

$$\delta_j^k = \frac{\partial E}{\partial a_j^k} = \sum_{l=1}^{J^{k+1}} \frac{\partial E}{\partial a_l^{k+1}} \frac{\partial a_l^{k+1}}{\partial a_j^k} \quad (24)$$

Error term  $\phi_l^{k+1}$  gives the following equation:

$$\delta_j^k = \sum_{l=1}^{J^{k+1}} \delta_l^{k+1} \frac{\partial a_l^{k+1}}{\partial a_j^k} \quad (25a)$$

$$a_l^{k+1} = \sum_{j=1}^{J^k} w_{jl}^{k+1} g(a_j^k) \quad (25b)$$

Recalling the description of  $a_l^{k+1}$ , where  $g(x)$  denotes the activation function.

$$\frac{\partial a_l^{k+1}}{\partial a_j^k} = w_{jl}^{k+1} g'(a_j^k) \quad (26)$$

Observe that

$$\delta_j^k = \sum_{l=1}^{J^{k+1}} \delta_l^{k+1} w_{jl}^{k+1} g'(a_j^k) = g'(a_j^k) \sum_{l=1}^{J^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (27)$$

The partial derivative of  $E$  w.r.t  $w_{ij}^k$  for  $1 \leq k < m$  is

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_i^k o_i^{k-1} = g'(a_j^k) o_i^{k-1} \sum_{l=1}^{J^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (28)$$

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_i^k o_i^{k-1} \quad (29)$$

For the partial derivative of  $E$  at the final layer,

$$\delta_l^m = g'(a_1^m)(\hat{y}_d - y_d) \quad (30a)$$

$$\delta_l^m = g'(a_j^k) \sum_{l=1}^{J^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \quad (31b)$$

For the hidden layers'  $E$ ,

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ij}^k} \left( \frac{1}{N} (\hat{y}_d - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ij}^k} \quad (32)$$

For merging the partial derivatives,

$$\Delta w_{ij}^k = a \frac{\partial E(X, \theta)}{\partial w_{ij}^k} \quad (33)$$

For updating the weights,

$$w_{ij}^{k+1} = w_{ij}^k + \lambda \Delta w_{ij}^k \quad (34)$$

Algorithm of IANN Model: Consider  $a$  as the training rate and weight initialization as  $w_{ij}^k$ , below procedures are utilized in the proposed IANN model.

- Perform the forward stage: For all pairs of input and outputs  $(\vec{x}_d, y_d)$ , save the results  $(\hat{y}_d, a_j^k)$  and  $(o_j^k)$  for all nodes  $j$  in  $k$  by arranged from layer zero, input to the final layer ( $m$ ).
- Perform the backward stage: For all  $(\vec{x}_d, y_d)$ , save the results  $\left(\frac{\partial E_d}{\partial w_{ij}^k}\right)$  for  $w_{ij}^k$  linking  $i$  in  $(k - 1)$ , to  $j$  in  $k$  by arranged from  $m$ , output to the initial layer. Compare  $E$  for the last layer using Eq. (4). Back-propagate  $E$  for  $\delta_l^k$ , processing backward from  $k = m - 1$ , via constantly utilizing Eq. (5). Compare the partial derivatives of a separate error  $E_d$  regarding  $w_{ij}^k$  utilizing Eq. (3).
- Combine the individual gradients: Merge the separate gradients for all I/O pairs  $\left(\frac{\partial E_d}{\partial w_{ij}^k}\right)$  to obtain the overall gradient  $\left(\frac{\partial E(X, \theta)}{\partial w_{ij}^k}\right)$ , for the whole collection of  $X = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$  by using Eq. (6).
- Modify the weights: Based on  $a$  and  $\frac{\partial E(X, \theta)}{\partial w_{ij}^k}$  and utilizing Eq. (7), the weights are updated using Eq. (29).

In this study,  $a$  is set to 0.2 and the momentum coefficient is 0.5. The network is trained for 500 epochs.

Moreover, the extracted features are fed to the FFNN-based ISAE for the final prediction of air quality. Fig.3 depicts the entire architecture of the VCSAE-DL.

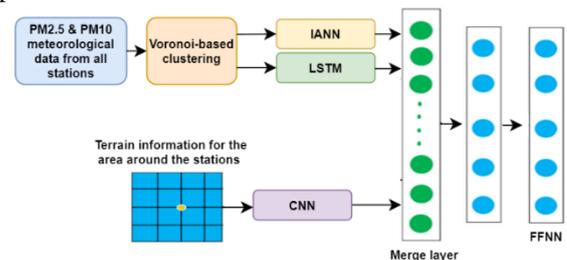


Figure 3. Overall Architecture of VCSAE-DL

IV. RESULTS AND DISCUSSION

Here, the efficiency of VCSAE-DL is evaluated and compared to that of ST-SVR [7], ISAE-DL [6], and EISAE-DL [6]. Experiments are conducted using a dataset based on the prediction of PM2.5 and PM10, which are both widely reported air pollutants for predicting air quality.

A. Accuracy and Precision

It is the ratio of true forecasts amid an overall data tested.

$$Accuracy = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \tag{35}$$

Precision is determined by using the TP and FP rates.

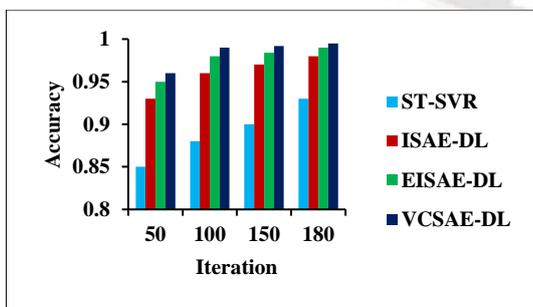
$$Precision = \frac{TP}{TP+FP} \tag{36}$$

Accuracy and precision are compared in Table 1 for existing and proposed models.

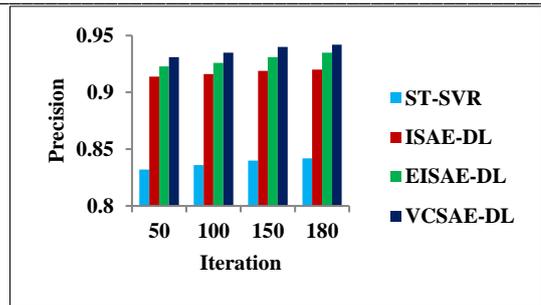
Table 1. Evaluation of VCSAE-DL in terms of Accuracy and Precision

Iteration	Accuracy				Precision			
	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL
50	0.85	0.93	0.95	0.96	0.832	0.914	0.923	0.931
100	0.88	0.96	0.98	0.99	0.836	0.916	0.926	0.935
150	0.90	0.97	0.984	0.992	0.84	0.919	0.931	0.94
180	0.93	0.98	0.99	0.995	0.842	0.92	0.935	0.942

The accuracy (precision) of various AQP models is depicted in Fig. 4a (b) for varying iteration counts. When using 100 iterations, VCSAE-DL's accuracy (precision) in predicting air quality is higher than that of ST-SVR, ISADE-DL, and EISAE-DL by 12.5% (11.84%), 3.13% (2.07%) and 1.02% (0.97%), respectively. Thus, it demonstrates that the proposed VCSAE-DL outperforms other existing models for AQP.



(a)



(b)

Figure 4. Comparison of ST-SVR, ISAE-DL, EISAE-DL and VCSAE-DL in terms of (a) Accuracy (b) Precision

B. Sensitivity and Specificity

Sensitivity is the proportion of predicted true values. It is calculated as:

$$Sensitivity = \frac{TP}{TP+FN} \tag{37}$$

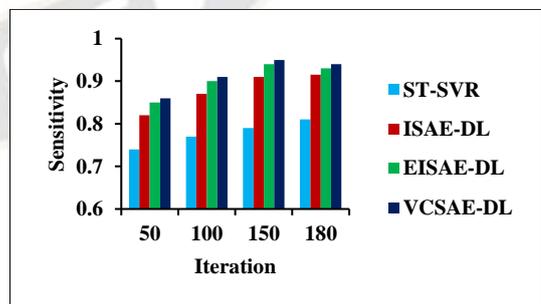
Specificity is the percentage of predicted negative values. It is calculated as:

$$Specificity = \frac{TN}{TN+FP} \tag{38}$$

The sensitivity and specificity of existing and proposed models are compared in Table 2.

Table 2. Comparison of VCSAE-DL in terms of Sensitivity and Specificity

Iteration	Sensitivity				Specificity			
	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL
50	0.74	0.82	0.85	0.86	0.75	0.81	0.83	0.845
100	0.77	0.87	0.90	0.91	0.77	0.83	0.86	0.87
150	0.79	0.91	0.94	0.95	0.79	0.87	0.888	0.90
180	0.81	0.915	0.93	0.94	0.80	0.89	0.91	0.92



(a)

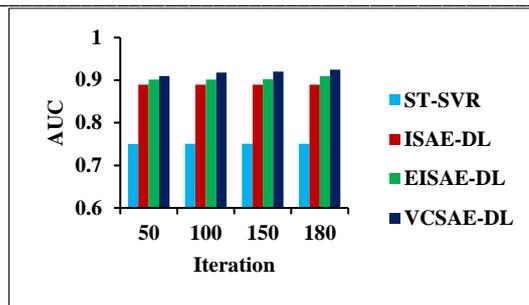
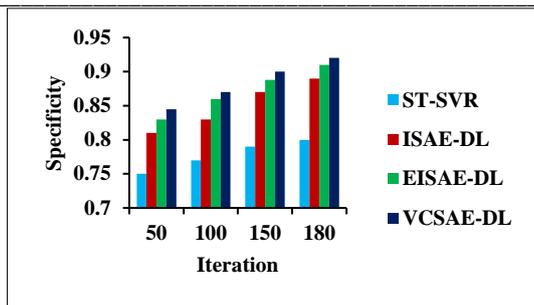


Figure 5. Comparison of ST-SVR, ISAE-DL, EISAE-DL and VCSAE-DL in terms of (a) Sensitivity (b) Specificity

Fig.5a (b) compares the sensitivity (specificity) of various AQP models over iteration counts. Compared to ST-SVR, ISADE-DL, and EISAE-DL, the VCSAE-DL had a higher sensitivity (specificity) by 18.18% (12.99%), 4.6% (4.82%), and 1.11% (1.16%), respectively for predicting air quality. Thus, it demonstrates that the proposed VCSAE-DL is more accurate in predicting air quality than other existing models.

C. Area Under Curve (AUC) and Matthew’s Correlation Coefficient (MCC)

AUC is calculated as:

$$AUC = \frac{Sensitivity+Specificity}{2} \tag{39}$$

MCC is the relationship between the forecasted and observed values.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{40}$$

The AUC and MCC comparisons of existing and proposed models are shown in Table 3.

Table 3. Comparison of VCSAE-DL in terms of AUC and MCC

Iteration	AUC				MCC			
	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL	ST-SVR	ISAE-DL	EISAE-DL	VCSAE-DL
50	0.7502	0.8892	0.9012	0.910	0.7603	0.9404	0.9802	0.981
100	0.7504	0.8894	0.9015	0.918	0.7605	0.9408	0.9804	0.9813
150	0.7508	0.8896	0.9018	0.920	0.7608	0.941	0.9806	0.9819
180	0.751	0.8897	0.91	0.925	0.7701	0.9414	0.9809	0.982

The AUC (MCC) for various AQP models at varying iteration counts is displayed in Fig. 6a (b).VCSAE-DL outperforms other existing AQP models regarding AUC (MCC) for predicting air quality by 22.33% (29.03%), 3.22% (4.3%), and 1.83% (0.09%), respectively, when the number of iterations is set to 100. Thus, it demonstrates that compared to existing AQP methodologies, the suggested VCSAE-DL has superior MCC and AUC.

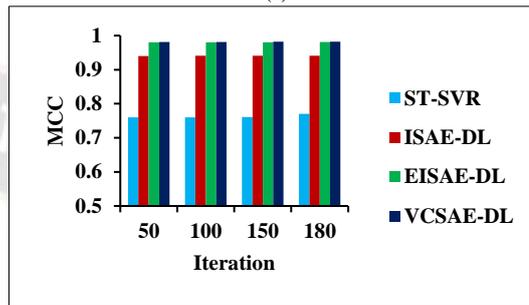


Figure 6. Comparison of ST-SVR, ISAE-DL, EISAE-DL and VCSAE-DL in terms of (a) AUC (b) MCC

V. CONCLUSION

In this paper, the VCSAE-DL model is proposed, which adopts the Voronoi clustering to cluster the locations which are geographically and temporally similar. The resultant clustered locations along with the PM2.5, target stations, and other meteorological data are given to the IANN, LSTM, and CNN followed by the ISAE-based FFNN to predict air quality. Finally, the experimental results demonstrated that the proposed VCSAE-DL achieved the highest performance in contrast with the existing AQP models.

REFERENCES

- [1] W. Mao, W. Wang, L. Jiao, S. Zhao and A. Liu, “Modeling air quality prediction using a deep learning approach: Method optimization and evaluation,”Sustainable Cities and Society, vol. 65, p. 102567, 2021, <https://doi.org/10.1016/j.scs.2020.102567>.
- [2] J. Ma, J. C. 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, 2019, <https://doi.org/10.1016/j.atmosenv.2019.116885>.
- [3] A. D. Syafei, A. Fujiwara and J. Zhang, “Prediction model of air pollutant levels using linear model with component analysis,”International Journal of Environmental Science and Development, vol. 6, no. 7, p. 519, 2015, <https://doi.org/10.7763/IJESD.2015.V6.648>.
- [4] J. J. Palop, L. Mucke and E. D. Roberson, “Quantifying biomarkers of cognitive dysfunction and neuronal network hyperexcitability in mouse models of Alzheimer’s disease: depletion of calcium-dependent proteins and inhibitory hippocampal remodeling,” Alzheimer’s disease and

- Frontotemporal dementia: methods and protocols, pp. 245-262, 2011, [https://doi.org/10.1007/978-1-60761-744-0\\_17](https://doi.org/10.1007/978-1-60761-744-0_17).
- [5] P. W. Soh, J. W. Chang and J. W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186-38199, 2018, <https://doi.org/10.1109/ACCESS.2018.2849820>.
- [6] P. ShreeNandhini, P. Amudha and S. Sivakumari, "Comparative analysis of air quality prediction using artificial intelligence techniques," *ECS Transactions*, vol. 107, no. 1, p. 6059, 2022, <https://doi.org/10.1149/10701.6059ecst>.
- [7] W. Yang, M. Deng, F. Xu and H. Wang, "Prediction of hourly PM<sub>2.5</sub> using a space-time support vector regression model," *Atmospheric Environment*, vol. 181, pp. 12-19, 2018, <https://doi.org/10.1016/j.atmosenv.2018.03.015>.
- [8] J. Wang and G. Song, "A deep spatial-temporal ensemble model for air quality prediction," *Neurocomputing*, vol. 314, pp. 198-206, 2018, <https://doi.org/10.1016/j.neucom.2018.06.049>.
- [9] F. Jiang, J. He and T. Tian, "A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction," *Applied Soft Computing*, vol. 85, p. 105827, 2019, <https://doi.org/10.1016/j.asoc.2019.105827>.
- [10] Y. Zhang, Y. Wang, M. Gao, Q. Ma, J. Zhao, R. Zhang and L. Huang, "A predictive data feature exploration-based air quality prediction approach," *IEEE Access*, vol. 7, pp. 30732-30743, 2019, <https://doi.org/10.1109/ACCESS.2019.2897754>.
- [11] D. Seng, Q. Zhang, X. Zhang, G. Chen and X. Chen, "Spatiotemporal prediction of air quality based on LSTM neural network," *Alexandria Engineering Journal*, vol. 60, no. 2, 2021, <https://doi.org/10.1016/j.aej.2020.12.009>.
- [12] D. Schürholz, S. Kubler and A. Zaslavsky, "Artificial intelligence-enabled context-aware air quality prediction for smart cities," *Journal of Cleaner Production*, vol. 271, p. 121941, 2020, <https://doi.org/10.1016/j.jclepro.2020.121941>.
- [13] L. Zhang, P. Liu, L. Zhao, G. Wang, W. Zhang and J. Liu, "Air quality predictions with a semi-supervised bidirectional LSTM neural network," *Atmospheric Pollution Research*, vol. 12, no. 1, pp. 328-339, 2021, <https://doi.org/10.1016/j.apr.2020.09.003>.
- [14] Y. C. Lin, S. J. Lee, C. S. Ouyang and C. H. Wu, "Air quality prediction by neuro-fuzzy modeling approach," *Applied soft computing*, vol. 86, p. 105898, 2020, <https://doi.org/10.1016/j.asoc.2019.105898>.
- [15] J. Ma, Z. Li, J. C. Cheng, Y. Ding, C. Lin and Z. Xu, "Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network," *Science of The Total Environment*, vol. 705, p. 135771, 2020, <https://doi.org/10.1016/j.scitotenv.2019.135771>.