International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

Optimized Forecasting Air Pollution Model Based On Multi-Objective Staked Feature Selection Approach Using Deep Featured Neural Classifier

Chitra Paulpandi[1*], Murukesh Chinnasamy[2]
[*1]Assistant Professor, Department of ECE,
Agni College of Technology, Thalambur, Chennai-103.
Affiliated to Anna University Tamil Nadu, India.
[*]Email: chitrapaulpandi09@gmail.com
[2]Associate Professor, Department of ECE,
Velammal Engineering College, Ambattur-Redhills Road, Chennai-66.
Affiliated to Anna University, Tamil Nadu, India.
Email: pcrmurukesh@gmail.com

Abstract—In recent days air pollution has been an essential issue affecting the environment nature leads to various natural causes. Especially the Covid-19 pandemic period has a variation environment changes due to vehicle controls and industrial facts at regular intervals. So air pollution has different scaling factors before and after the pandemic, period produces non-scaled data features. Many methodologies provide the differential solution to analyze the air quality measurements under various conditions to make warnings to avoid air pollution. By the impact of exiting forecasting, ML approaches do not provide the accuracy in precision levels because feature dependencies are non-relevant in high dimension nature. To create the best Air quality index, we need to improve the feature analysis and classification objectives to produce higher prediction performance. This paper proposes a new forecasting model based on the Multi-objective Staked Feature Selection Approach (MoSFS) using the Deep Featured Neural Classifier (DFNC) model to predict air pollution. Initially, the Successive Feature Defect Scaling Rate (SFDSR) was carried out Auto Regressive Integrated Moving Average (ARIMA) rate for finding variation dependencies. The multi-objective relational successive feature index was scaled using the Spider Herding Algorithm (SHA) to select the features based on these variations in feature limits. Then the chosen features get activated to logical activation function with Long Short Term Memory (LSTM) and trained with a Fuzzified Convolution Neural Network (F-CNN) to predict the class by variance. This resultant factor proves the performance of RMSE values attaining the best level to forecast the features and in precision rate produce higher performance in classification accuracy compared to the other system.

Keywords: air quality prediction, staked feature selection, deep learning classifier, CNN, ARIMA, LSTM, forecasting analysis.

I. INTRODUCTION

Air quality forecasting is an important step taken by the government as it is a significant concern as it affects human health. In today's developing world, there are many air pollutants, including Carbon Dioxide (CO_2), Nitrogen Dioxide (NO_2), vehicle smoke, burning of unnecessary materials like polythene, factories, and carbon monoxide. This pollution causes acute diseases like cancer, respiratory problems, and heart diseases. Thus, Air Quality Index (AQI) is used to know the air pollution through which it can be known whether the air pollution is high or low every day.

Air pollution is difficult to detect by government and private companies worldwide. Machine learning (ML) methods are used to identify AQI. However, these approaches are very challenging tasks for predicting AQI. This paper addresses the problems of detecting AQI and reducing air pollution before it becomes unfavorable.

To date, several computational models ranging from statistical and ML and Deep Learning (DL) have been compared to demonstrate the accuracy of predicting air quality standards. Pollution levels in some parts of the world remain uncontrollable due to various sources and causes. Many studies are being conducted in this area, but they have not yielded accurate results regarding pollution. Air pollution data is available on the Kaggle site, divided into two parts, training and testing. ML algorithms such as Support Vector Machine (SVM), Random Forests (RF), and Artificial Neural Networks (ANN) are used to predict AQI.

AQI shows the quality of air pollution, whether the air is clean or not. The major constituents of air pollutants are hybrids such as NO_2, CO_2, and CO. This paper analyzed the air quality measurements from various sensors, including $PM_{2.5}$, PM_{10}, O_3, CH_4, temperature, pressure and CO_2. Previous methods have made the detection of air pollution a challenging task. The Deep Learning (DL) algorithm is the best method to predict air pollution and gives more results than previous methods. Figure 1 shows the proposed model and process of air quality analysis.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

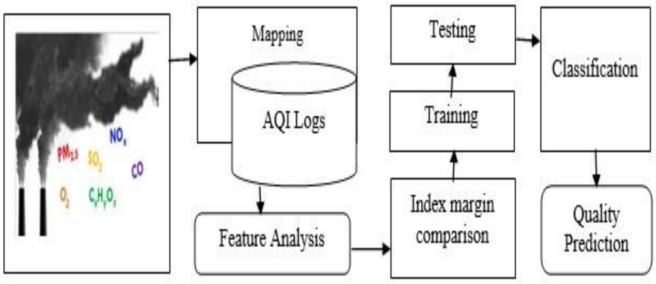

Figure 1. Process of Air quality analysis

The feature selection and classification are essential in the machine learning approach. The features get important for reducing classification burdens. All over the decision are carried out fuzzy rules. Fuzzy rules fix the condition rules based on logical redundancy. The feature index returns marginal weights and input to the neural network. The training samples are trained and tested in a hidden layer to categorize the result. The searching links remain the timing and neuron weightage class principles to progress the classification accuracy.

This paper is structured as follows: First it describes previous methods for air pollution prediction. Next explains the materials and methods in detail, followed by next section which describes the detailed simulation results and the simulation tool. Finally, the paper describes the conclusions.

II. RELATED WORK

Phuong et al. (2016), focuses on AQI during the covid-19 lockdown. The AQI prediction used DL-based LSTM, and Recurrent Neural Network (RNN) approaches to identify the AQI level. Similarly, X. L. L. et al. (2016) the author introduces DL-based techniques for AQI. Yet, these techniques challenge the process during classification.

Huang et al. (2018), concentrated on $PM_{2.5}$ pollution in smart cities. So the study expresses the CNN-LSTM technique to predict the AQI. Likewise, Thanongsak Xayasouk et al. (2020), describes an LSTM-DAE approach used to analyze the AQI. The study enhances prediction accuracy and precision. Yang et al. (2017), introduces Fuzzy logic for early air prediction and AQI assessment. Similarly, Munawar et al. (2017), presented Neuro-Fuzzy (NF) technique to predict AQI based on some factors like NO_2, CO_2, O_3 and $PM_{2.5}$. However, these methods didn't provide satisfactory results about AQI.

1D convnets and bidirectional GRU designed by Tao et al. (2019). The suggested method produces air forecasting based on such factors as temperature, speed, and wind. Similarly to Pardo et al. (2017), used a DL-based LSTM technique to find air forecasting based on NO2. However, these methods produce a high false rate during air prediction.

Ghaddar et al. (2018) novel focuses on dimensional, and classification features using the SVM technique. However, the study produces the challenging task of analyzing the best elements of air prediction.

Liu et al. (2017), concentrated on urban air quality index forecasting using SVR. The suggested method improves accuracy and reduces the false rate performance during classification.

ML-based LR technique was implemented by Aditya et al. (2018). The suggested method analysis vital features of $PM_{2.5}$ to identify the AQI. Zhili Zhao et al. (2020), introduces combing forward with the RNN method to predict air forecasting based on polluted factors NO_2, CO_2, and SO_2.

Zhang et al. (2019), focused on reducing the air dataset dimensionality by using light Gradient Boosting Machine (GBM). It improves the classification results and selects useful polluted air information. B. Liu et al. et al. (2019), introduces Sequence-to-Sequence (S2S) and RNN techniques for air forecasting. However, this study didn't provide better performance.

Chen et al. (2020) explored Adaptive Kalman Filtering (AKF) technique for reducing noise and removing unwanted data to change air dataset structure. P. W. Soh et al. (2018), aims to predict air forecasting using multiple classifiers like CNN, LSTM, and ANN methods. The study extracted polluted air information from the dataset.

Fazziki et al. (2017), focused on roadside AQ control using agent-based traffic. Similarly, Gu et al. (2018), expresses a heuristic Recurrent Air Quality Predictor (RAQP) for AQ prediction. However, these techniques take much more time during classification.

Mokhtari et al. (2021), introduced DL-based ConvLSTM techniques for AQI prediction. Likewise, Yang et al. (2022), expressed LSTM and GRU techniques to predict the AQI based on polluted factors $PM_{2.5}$, NO_2, SO, and CO.

III. PROPOSED SYSTEM

The development progress of the AQI prediction models by intent the feature analysis and classification objectives to produce higher prediction performance. Figure 2 shows the proposed architecture of MoSFS-CNN. Initially, the Successive Feature Defect Scaling Rate (SFDSR) was carried out with an Autoregressive Integrated Moving Average (ARIMA) rate to find variation dependencies. Based on these feature limit variations, the multi-objective relational successive feature index was scaled using the spider Herding algorithm (SHA) to select the features. Then the chosen features get activated to logical activation function with LSTM and trained with a Fuzzified Convolution neural network (F-CNN) to predict the class by variance. This resultant factor proves the performance of RMSE values attaining the best level to forecast the features and in precision rate produce higher performance in classification accuracy compared to the other system.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

The intention of an adaptive deep neural network consists of three consecutive optimization models, i.e., LSTM, Fuzzy membership function and convolution layers. The features are analyzed by a spider hardening algorithm, which they despond on foraging search models for fitness evaluations. The selected features are selected by trained progress in the optimization process. The rest of the section defines the implementation progress of the proposed system.

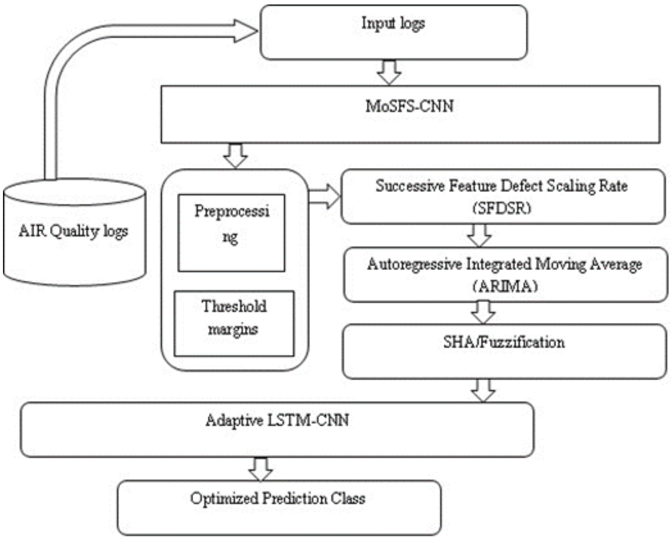

Figure 2. Proposed architecture of MoSFS-CNN

A. Data preprocessing

In this stage, the presence of a collective dataset is from Air quality features observation in various locations. It contains time series data at regular intervals holding scaling index values, which verifies the features of present cases, label counts, and missing values prediction. This can quickly treat it as an outlier training model and convert it to the normalized negative value. The second step is to create class labels before preparing using the data wrapping model. To make noise fewer data by verifying the threshold margins and scaling interval range from the features points.

B. Successive Feature Defect Scaling Rate (SFDSR)

Based on the preprocessing margins, the AQI features F' contains the variation relative features F'', Features to be integrated to choose the threshold range based on the class. This estimates the mean average feature weight depends on the feature defects rate.

Step 1: Initialize data source from AQI Input features
Step 2: For all Preprocessed features, Ps
Step 3: Create margin Cluster class (Mcls)
For each class l ←Mcls
Step 4: Create a Decision Tree support index for each class
Step 5: Compute For each Decision node, create a feature vector size(N)
If Fv(Pl) ∈ s, then choose the Feature variance
i = 1
Select Count features = count +1 on each class.
Select the marginal weights Fli= $\frac{\mathrm{Fv(pi+AmR)}}{\mathrm{Fv(marginal\ weight)}} \times \mathrm{Fv(NI)}$
End If
End for
Step 6: Estimate Intensive feature points (IrPl).
If max supportive limits for each class
IrPl = $\frac{\mathrm{Fv(l-1)(Lower\ limit)}}{\mathrm{Fv(l-1)(marginal\ weight)}} \times \mathrm{Fv(l-1)(NI)}$
Identify the Feature Limits IR = $^{\mathrm{CI}}/_{\mathrm{IrPl}}$
Return Ir as Feature vector→ Vs class
End if
End for

The above algorithm predicts the successive scaling of the Air quality ranges, which supports upcoming feature variations under the moving average limits. Also, the source of feature limits has Max-Min difference limits marginalized to get an average mean rate. The difference variation is selected from Euclidian distance estimation.

C. Autoregressive Integrated Moving Average (ARIMA)

This ARIMA model creates the average moving index structural rate to find the approximation of upcoming feature weights. It defines the Auto regressive successive rate of average terms at difference variance between current and moving feature variables. Let us consider the average moving parameter consideration from 'Vs' by assuming L as the Lag point ϕ_i is the autoregressive moving rate,

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t = \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t \qquad (1)$$

Based on the Normal distribution, ε_t the error rate is identically distributed from the average feature rate to vary feature terms with zero mean rates.

$$\left[\nabla^d \nabla_s^D Y_t - \mu\right] = \frac{\theta(B)\Theta(B^s)}{\phi(B)\Phi(B^s)} e_t \qquad (2)$$

Depending upon the Regressive model (p,d,q) is the current at the successive moving rate 'S' respectively at Difference D in varied as 'd', and Lag error difference is the 'q'. Let's consider,

$\emptyset(B) = 1 - \emptyset_1 B - \emptyset_2 B^2 -, \dots \dots . - \emptyset_p B^q$

$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_p B^q$

$\Phi(B) = 1 - \Phi_1 B - \Phi_2 {B_s}^2 - \cdots - \phi_p {B_s}^p$
$\Theta(B) = 1 - \Theta_1 B - \theta_2 B_s^2 - \cdots - \theta_Q B_s^Q$
$\omega_t = \nabla_s^D \nabla^d y_t$

The forecasting cycles be formed at the successive rate of the feature vectors by the consecutive difference at each feature variance limit. The ARIMA model discretely finds the different terms between the mutual differences; also, the variation of feature difference is formed based on the duty cycle terms. By finding the average moving rate between the current Features $Hf(A)$ and $Hf(B)$, the equivalence series of moving index reparative equivalent to relational time series features be considered as,

$$Hf(A) = \sum_i^n ni\ (Xi) == 1\ which\ is\ equivalent\ another\ input\ series P(Ai) \log 2p(A2),$$

Get Equivalence difference, between

$$Hf(B) = \sum_j^m mi(Xj) == 1\ which\ is\ equeivalent\ realtional\ input\ series P(Bi) \log 2p(B2)$$

The joint features from source on A and B state equivalent {A, B} = $\{a_1b_1, a_1b_2, \cdots, a_nb_m\}$, this generates joint distribution outside the total weightage among the two states {A, B}.

$$\begin{vmatrix} a_1b_1 & a_1b_2 & \ldots & a_nb_m \\ p(a_1b_1) & p(a_1b_2) & \ldots & p(a_nb_m) \end{vmatrix}$$

The joint feature data correlation between the behaviours is the entropy value A and B.

$$Hf(A,B) = 1 - \sum_{i=1}^n \sum_{j=1}^m p(a_ib_j) log_2(a_2b_j)$$

The independent feature from A and B is the Relation Feature (RF) is

$$RF\ (A,B) = H(A) + H\ (B) - H\ (Ab) real\ probability\ (0,1) joint\ relation$$

This returns the average moving index variation by getting the real probability of air quality difference in the upcoming strategy. Also, this progress the feature difference of supportive margin index at independent variables.

D. Spider herding algorithm (SHA)

This stage analyzes the collective feature weight, which highly supports the maximum average index. The particular feature is communications via low-frequency vibrations with other elements in the genus are related spider population, feeding feature weights are search objects, The formed with neural layers, the spiders from max weightage features are selected from the centroid closest weight, even average index feature rate observed by maximum frequency of relative features get closer to the centroid value. However, the aligned weights are trained with decision tree classification with randomized aligned neurons by weightage of fitness value.

Algorithm: SHA

Step 1: Attain Vs ARMIA Margin $Rl\ (A,B)$
Step 2: To collect (Herding) marginalized features F🡨($Rl\ (A,B)$)
Step 3: For all F'🡪{Sum of Feature relative class}
Check all variance distance and difference D."
Vd🡪 $\sum_{i=0}^n F'(D''(Rl\ (A,B)))$
Return support values
Step 4: Create spider Layer Centroid (Vd) point in 8*8 layers
Yi (Vd..x-Init i+b)= 1-€i, $€i \geq 0, 1 \leq i \leq n$
Step 5: Compute fitness weight Ft'('x' features and 'y' variance)
Step 6: Compute the Fitness by Min Weight 'w'
$\min \frac{1}{2} \left||w|\right| + c \sum_{l=1}^n €i$
Step 7: compute the population limit Ci clans spider limit for each layer
Step 8: For all Min 🡪w to Process Ci at layer J at Ci
Train the Moving Index features at Max🡪ci, at each layer 'j.'
Choose the best case fitness at Maxnew, ci,j
$X_{new,ci,j}=X_{ci,j}+ά*(X_{best,ci}-X_{ci,j})*r$
Attain spider feed data weight if xci,j=xbest, ci, then
Xnew,ci = β*Center,ci
End For
Step 9: Assign the closest weight at centroid in the current layer to get Feature weight in the best case
For all xci,j and generate xnew,ci,j
$X_{center,ci,d} = \frac{1}{nci} \sum_{l=1}^{nci} xci,l,d$
End for
Step 10: For all best weights to form Cluster Clan ci choose amx weight
$X_{worst,ci}=X_{min}+(X_{max}-X_{min}+1).rand$
End For
Step 11: Update cluster clans Cls.
Step 12: Return best class clans 🡪Cls

The above algorithm produces analysis the feature weight which highly support maximum value index. After gaining autonomy over their structure, spiders can adequately represent the different environments of their offspring and support the maximum possible trait variation. Along with integrating their feature community, the air quality retains the best cases 'Cls' of match case results. In this case, the model operator performs a separate update to group the classes according to the index level.

E. Convoluted LSTM

The convoluted linear has 16 layers by constructing 12 consequent pooling layers with a gated projection of Fuzzy formation. The pooling layers are active by linearity characterization with 2 × 2 kernels for feature observation. Then 3×3 convoluted layers for feature extraction are defined by the logical activated ReLu function for fuzzified convolution. By constructing the convoluted linearity subscriptions, the features are carried out through Input gates by convoluted progress get trained to receive data. The processing gate conditionally varied the Logical level current feature index to the output gate. The LSTM principles are,

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

Let us $\widetilde{C}_t$ and $\widetilde{C}_{t-1}$ $is\ the\ input\ process$ based on current and moving index heuristically h_{t-1} at 'x' series of input data at each weight 'w' in corresponding states be constructed as,

$$\widetilde{C}_t \ = \ \tanh(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$C_t = f_t C_{t-1} + i_t \hat{C}_t$$

The process gate is convoluted to the pooling layer to choose the max supported heuristic index in best case index level b_t in ReLu activation. Then the relative feature weights are selected by the probability from 'C_t' to get maximum W_t to offset σ variation features to get Maximum support values.

$$f_t = \sigma(W_t \cdot Max[h_{t-1}, x_t] + b_t)$$

The output gate gets the maximum support index class required from $h_{t-1}\ and\ x_t$ from C_t and f_t to decision vectors,

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \tanh(C_t)$$

The features trained to produce class at best-optimized $b_o\ and\ W_o$ maximum support levels, respectively, the consecutive LSTM Bias.

F. Adaptive Fuzzification

The fuzzy creates a conditional membership function for choosing marginal values logically to bound air quality moving average features. It establishes a fuzzy set class to rank the variations of the element. This optimizes LSTM regressive gated outputs at maximum supported values.

$$\hat{X} = fuzzification\ \left(x_{ij} \middle| cx_{ij}\right)$$

Let assume, i, j, are component indices x in input matrix X, in addition to input fuzzy membership function centre cx.

$$x_{ij} = possibility\ \left(x_{ij} \middle| \widehat{MF_{ij}}\right) = \ max_{x \epsilon X}(\widehat{MF_{ij}} \delta\left(x - x_{ij}\right))$$

$$Where\ \delta\left(x - x_{ij}\right)\ is\ the\ kronecket\ delta\ function.$$

Each fuzzy convolutional layer has two processing stages: the fuzzy convolution stage and the pooling stage. The fuzzy convolution stage is applying a fuzzy convolution filter to the original 1D data, as shown in equation (4), where the calculation of the fuzzy convolution filter Wμ is done by Equation (5), where W is the original convolution.

$$x_i = b_i + \sum_{a=0}^{m-1} W_\mu\ x_{(i+a)}$$

$$W_\mu = fuzzification\ (W)$$

This stage downscales the input to the subsequent blurring convolutional layer to keep the translation of the input unchanged. In this work, ReLu functional layer will be used for the pooling layer. The rectified linear unit is a commonly used activation function in deep learning models. If the process receives any negative input, the function returns 0, but for any positive value xx, it returns that value.

$$f(x) = \max\ (0, x)$$

Where f(x) is the activation function of the convolutional layer, feature reduction is primarily achieved by autoencoders, which aim to learn efficient data encodings in an unproven way and are viewed as a type of artificial neural network.

$$\Phi: X \rightarrow F$$

$$\Psi: F \ \rightarrow \ X$$

$$\Phi, \Psi = \ arg \min_{\Phi, \Psi} \|X - (\Phi, \Psi)\ X\|^2$$

Let us assume a single hidden layer in which an autoencoder stage results in the input $x\ \epsilon\ R^d = X$, in addition, maps it to $h\ \epsilon\ R^p = F$

Consider a single hidden layer in simple, the encoder stage of an autoencoder ensues the input $x\ \epsilon\ R^d = X$ in maps it to $h\ \epsilon\ R^p = F$

$$h = \ \sigma(Wx + b)$$

The fully connected layer of the FCNN is working as a classifier with input features being the crisp value $z_i$fetched

$$z_i = defuzz\ (x_i) \ = \frac{\sum C_y x_i}{\sum x_i}$$

$$\widehat{y_i} = \ W_{fc} z_i$$

Where the C_y- defuzzification membership function centre.
$\widehat{y_i}$ - Classifier output and
W_{fc}- Fully connected layer weight matrix.

The output error evaluation is done through cross entropy which refers to the loss function as revealed in equation 9, where y represents target ^ y denotes classifier output N notates the number of samples.

$$E = \ -\frac{1}{N} \sum_{n=1}^{N} [y_n \log(\widehat{y}_n) + \ (1 - \ y_n) \log(1 - \ \widehat{y_n})\ ]$$

The conventional backpropagation learning algorithm is extensively utilized along cross-entropy loss function for training model parameters. The weight update as presented in equation10

$$W_{fc}(k + 1) = W_{fc}(k) - a_{fc} \frac{\partial E}{\partial W_{fc}}$$

The updating of defuzzification membership function centre $C_y(k)$ is performed through equation (11) where a_{cy}denotes updating centre $y_{k+1}\ and\ \widehat{y_{k+1}}$ are output target and model's actual output, respectively,

$$C_y(k + 1) = \ C_y(k) + a_{cy} \nabla C_y$$

The fuzzification gets a membership to the targeted level to return feature supportive of deciding on the convoluted layer

G. Fuzzified Convolution neural network (F-CNN)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

This work uses CNNs to extract features from an air quality feature index. The convolutional layer is trained with threshold margins based on ARIMA, LSTM and Fuzzification. To generate an actual scaling index of the feature set, be trained into the feature margins. Next, selective features are developed based on the fuzzy inference system. Finally, the CNN model trains internal information and external communication. The purpose of using network model training data is to continuously adjust the parameters used in the network to achieve model optimization. Figure 3 shows the Illustration of adaptive LSTM-F-CNN.

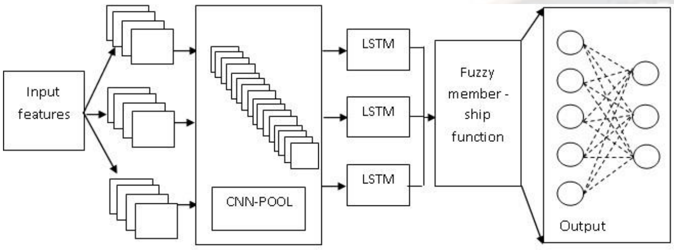

Figure 3. Illustration of adaptive LSTM-F-CNN

CNN has two stages forward transfer and reverses transfer. For the forward transfer stage, a 3*3 convolution kernel is used to process the convolution layer's internal information and accident information. A dependent parameter is added to the convolution of the upper layer feature map to obtain the lower layer output as a function of the activation function.

Algorithm: Adaptive (F-CNN)

Step 1: Construct an F-CNN layer with a pooled feed-forward layer with softmax activated
LSTM depending on the mean-variance rate.

Step 2: Train the feature weights to each pooled layer with attained margin class rate and choose the
selective LSTM Gated result

$\Phi: X \rightarrow F(ft)$

$$\Psi: F \rightarrow X\,(Ot)$$

Step 3: Attain the Pooling spread features to select the margin dependencies to
get the importance of extracted features

Step 4: Apply the convoluted cross-layer margins at each step

$$x_i = b_i + \sum_{a=0}^{m-1} W_\mu\ x_{(i+a)}$$

Step 5: Compute ReLu activation function on convoluted input features fuzzification$f(x) = \max(0, x)$

Step 6: Selective feature weights difference vector in argument feature array is
calculated by Minimum threshold margin on non-active class

$$\Phi, \Psi = arg \min_{\Phi,\Psi} \|X - (\Phi \circ \Psi)\, X\|^2$$

Step 7: To marginalize defuzzification based on Upper and Lower bound difference vector to regret weight

$\hat{y}_i = W_{fc} z_i \leftarrow (\Phi, \Psi)$

Step 8: Construct the F-CNN Layer from LSTM optimized Feature weights on getting
training threshold margins to get optimized output.

$$C_y(k+1) = C_y(k) + a_{cy} \nabla C_y \rightarrow (ft, Ot)$$

Step 9: Select the Max-Margin Threshold rate. If the class for all feature get a supportive margin,
the feature will attain the class by training.

Step 10: Return Output class by category

The results are classified as a class by AQI prediction levels. This gets categorized based on the fully connected layers to provide outputs, and the LSTM gates optimize the results to get Membership conditions. The logical activation prefers the augmented feature substances from the AQ to predict the importance of the feature class to predict the result. It attains higher performance which is discussed in the result and discussion.

IV. RESULT AND DISCUSSION

The results are evaluated with a real-time collective dataset observed from seasonal air quality measurements by Central Pollution Control Board (CPCB). The proposed algorithms were tested using a python framework with Deep learning libraries with an indexed quality assessment threshold margin rate from the Government of India. Table 1 shows the processed dataset values parameters used and environmental issues taken for the proposed study and its consideration.

TABLE I. ENVIRONMENT SETUP AND ITS CONSIDERATION

Parameters used	Values processed
Tool Processed	Jupiter Notebook
Dataset Log	AQI Dataset
Language environment	Python Framework
Features considered	Random attributes <=20
Quality By class	Year by Index class.

The progressive results are tested with confusion metrics. The performance results consider the resultant testing parameters such as precision accuracy, recall, rate, F-measure, false rate, time complexity and loss rate function are calculated by confusion matrix. The Mean Absolute Error (MAE) is represented by,

$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$ and Root Mean Squared Error (RMSE) is represented by,

$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ is used to evaluate the performance evaluation by considering the loss rate.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

By verifying the performance, the loss rate deficiency is considered an uncertainty principle, depending on prediction interval coverage probability (PICP), i.e. the mean error rate respectively at y_i *and* $\hat{y}_i$ scaling dependent feature values. Also, the mean prediction interval width (MPIW) has the superior prediction level on probability at scaling features. The evaluation be carried out,

$$PICP = \frac{1}{n} \sum_{i=1}^{n} c_i \quad with \begin{cases} c_i = 1, \; if \; y_i \in [L_i, U_i] \\ c_i = 0, \; else \end{cases}$$ as

respectively mean interval point in,

$$MPIW = \frac{1}{n} \sum_{i=1}^{n} (U_i - L_i)$$

The difference function is calculated between the Upper bound ground truth variation L_i *and* U_i respectively y_i and lower bound as same as fuzzy membership function.

The figure 4 shows the actual Air quality rate in recent definitions under NO_2, CO, and SO_2, which is marginalized under the ratio 0 to 0.25 mean level.

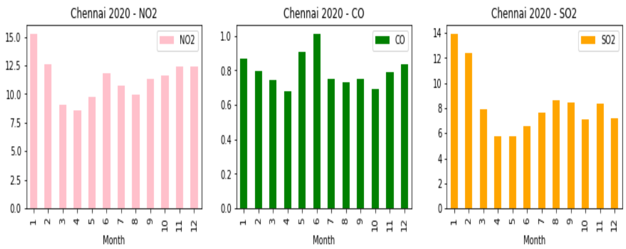

Figure 4. Actual successive rate of Air Quality

Figure 5 shows the successive feature variation state and seasonal variation trends; the upcoming prediction will be increased depending on parameters NO2, CO, and SO2 consecutively.

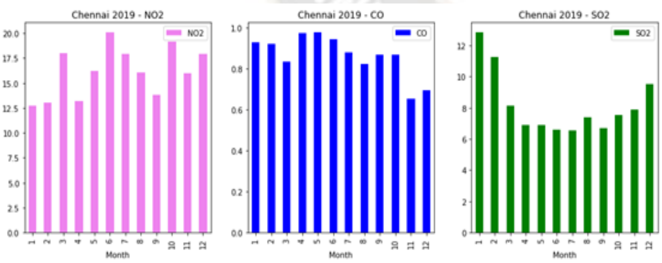

Figure 5. Successive feature variation state and seasonal variation trends

The average mean rate is varied according to the F-CNN classifier by the ARIMA rate. Figure 6 shows the successive feature variation state and seasonal variation trends.

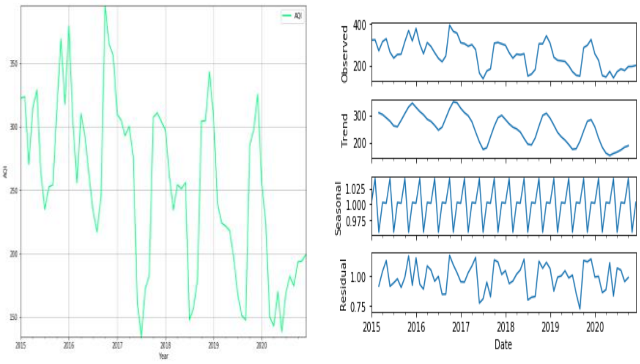

Figure 6. Successive feature variation state and seasonal variation trends

The subsequent rate differs from the feature index, which makes a linear index of variation in the Year, as shown in the figure. Also, the residual observation creates mutual feature scaling depending on seasonal trend the average rate be conditionally increased.

Figure 7 shows the Successive loss rate Function. The drop ratio scaling is 0.25 margin related to feature threshold values based on the error rate evaluation.

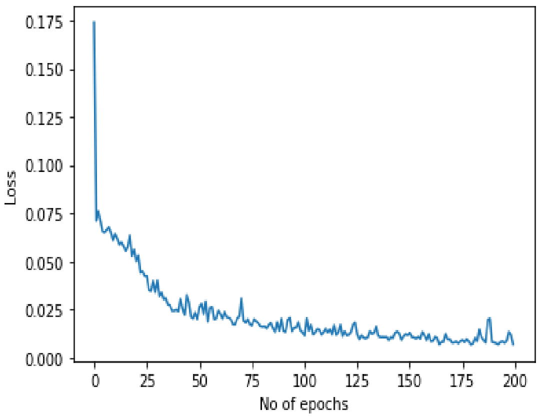

Figure 7. Successive loss rate Function

So the fuzzification reduces the Loss depending on the failure rate in the proposed system to attain high performance with a low loss rate. This proves the low-level loss rate in high performance in Prediction accuracy. Figure 8 defines the overall performance of classification accuracy.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

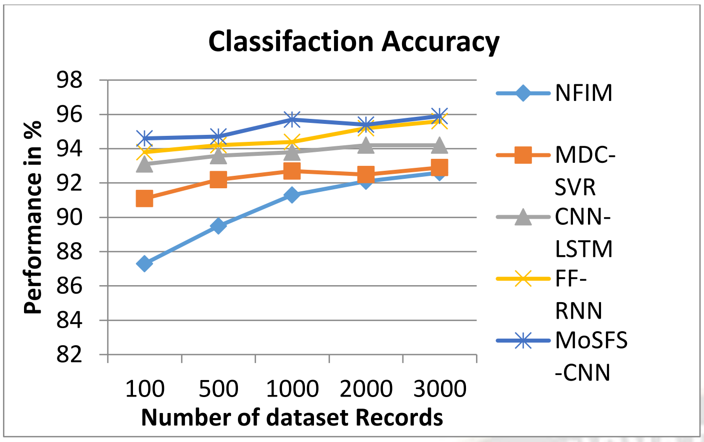

Figure 8. Classification accuracy of records

The classification results process the results of the proposed system. The proposed MoSFS-CNN attains the province average mean rate of sensitivity and specificity rate carried by provenance security. Table 2 shows that the accuracy of the classification, which is compared in different ways, offers the best performance with the recommended accuracy of the crop.

TABLE II. CLASSIFICATION ACCURACY IN PERCENTAGE FOR ALL RECORDS

Methods /dataset records	Classification Accuracy in %				
	NFIM	MDC-SVR	CNN-LSTM	FF-RNN	MoSFS-CNN
100	87.3	91.1	93.1	93.8	94.6
500	89.5	92.2	93.6	94.2	94.7
1000	91.3	92.7	93.8	94.4	95.7
2000	92.1	92.5	94.2	95.2	95.4
3000	92.6	92.9	94.2	95.6	95.9

An additional mutual measurement process for categorical evaluation is compassion. An actual positive, further correlation to the true value of the negative range. Sensitivity ratings were performed as well as other methods. Figure 9 describes the sensitivity advancements for dissimilar dataset record findings in functional approximations to classify results with higher presentation rates of the intentional method than prevailing additional methods.

Table 3 describes the sensitivity analysis of the proposed algorithm and the existing algorithm comparison presented.

Figure 10 defines the specificity performance the proposed and existing algorithm result present in the graph.

In the graph x-axis presents number of data and y-axis presents performance in %. The proposed method produces high performance than existing methods. Table 4 represents the variability of the singularity governed by different datasets creates different representations differently.

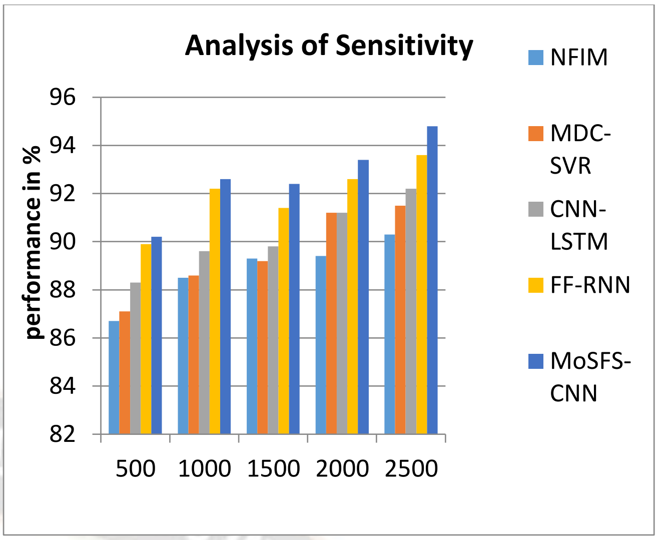

Figure 9. Sensitivity analysis

TABLE III. SENSITIVITY ANALYSIS

Methods/No of records	Sensitivity Analysis in %				
	NFIM	MDC-SVR	CNN-LSTM	FF-RNN	MoSFS-CNN
100	86.7	87.1	88.3	89.9	90.2
500	88.5	88.6	89.6	92.2	92.6
1000	89.3	89.2	89.8	91.4	92.4
2000	89.4	91.2	91.2	92.6	93.4
3000	90.3	91.5	92.2	93.6	94.8

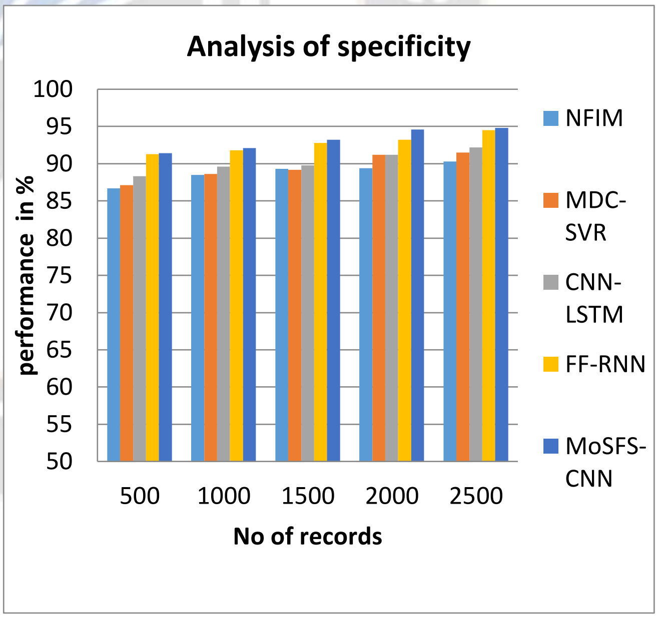

Figure 10. Specificity Evaluation

TABLE IV. SPECIFICITY EVALUATION

Methods/No of records	Analysis of Specificity in %				
	NFIM	MDC-SVR	CNN-LSTM	FF-RNN	MoSFS-CNN
100	82.3	87.3	89.3	91.3	91.4

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

500	83.8	87.6	91.2	91.8	92.1
1000	84.2	88.5	92.6	92.8	93.2
2000	85.3	88.9	92.8	93.2	94.6
3000	86.3	90.2	93.5	94.5	94.8

Measurements represent coherent representations of true positives and false negatives and depend on accuracy and recall.

Figure 11 shows that the difference between the false rate generated by different approaches and the predicted method produces a lower F-number than the remaining additional approaches.

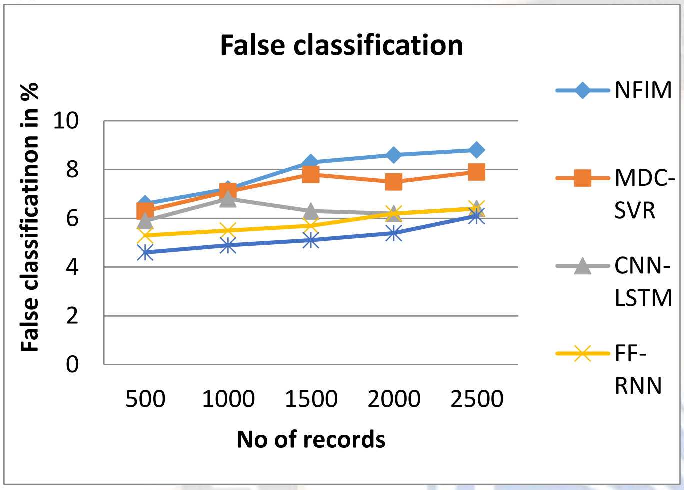

Figure 11. F-measure Evaluation

Table 5 defines the comparison ratio based on the number of actual positive values and records.

TABLE V. F-MEASURE EVALUATION

Methods/No of records	Analysis of False classification in %				
	NFIM	**MDC-SVR**	**CNN-LSTM**	**FF-RNN**	**MoSFS-CNN**
100	6.6	6.3	5.9	5.3	4.6
500	7.2	7.1	6.8	5.5	4.9
1000	8.3	7.8	6.3	5.7	5.1
2000	8.6	7.5	6.6	6.2	5.4
3000	8.8	7.9	6.9	6.4	6.1

Figure 12 shows the execution of the time complexity of number of records of various networks.

The execution complex nature during feature selection and classification takes time for evaluation. The performance time is referred to as feature screening and classification time during the overall time taken. The proposed system produces the best performance than other methods

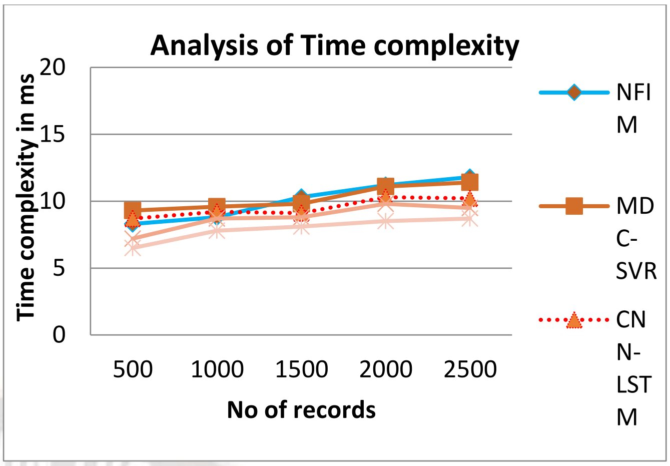

Figure 12. Time complexity Evaluation

TABLE VI. TIME COMPLEXITY EVALUATION

Methods/No of records	Analysis of Time Complexity (ms)				
	NFIM	**MDC-SVR**	**CNN-LSTM**	**FF-RNN**	**MoSFS-CNN**
100	8.3	9.3	8.7	7.2	6.5
500	8.8	9.6	9.2	8.7	7.8
1000	10.3	9.8	9.1	8.8	8.1
2000	11.2	11.1	10.3	9.8	8.5
3000	11.8	11.4	10.2	9.5	8.7

The time complexity representation of the proposed MoSFS-CNN system produces high performance than existing approaches. The hybrid model combines the feature evaluation and classification in redundant time taken verified with O(n) time complexity measure by completing the execution. Other methods produce high performance compared to the other system. Table 6 represents the time complexity evaluation.

V. CONCLUSION

To conclude, the proposed system produces higher compared to the previous design. The air quality progress attains a higher importance rate in classification accuracy by predicting usage based on Air quality prediction. This Multi-Objective Staked Feature Selection Approach (MoSFS) is applied to indicate the importance of the feature to reduce the dimension. The Deep Featured Neural Classifier (DFNC) model predicts air pollution. y the Successive feature defect scaling rate (SFDSR) was carried out Autoregressive Integrated Moving Average (ARIMA) rate for finding variation dependencies supported for high-performance evaluation to predict the air quality index. The proposed MoSFS-CNN system improves the higher performance in classification accuracy, precision rate, recall rate and redundancy in false rate and time complexity.

ACKNOWLEDGMENT

None

REFERENCES

1. X. L. L. P. Y. H. J. S. T. Chi, "Deep learning architecture for air quality predictions", Environ Sci Pollut Res, vol. 23, pp. 22408-22417, 2016.
2. CJ Huang and PH Kuo, "A Deep CNN-LSTM Model for Particulate Matter (PM 25) Forecasting in Smart Cities ", Sensors (Basel), vol. 18, no. 7, pp. 2220, 2018.
3. Z. Yang and J. Wang, "A new air quality monitoring and early warning system: Air quality assessment and air pollutant concentration prediction", Environmental Research, vol. 158, pp. 105-117, 2017.
4. Thanongsak Xayasouk, HwaMin Lee and Giyeol Lee, "Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models", Sustainability, vol. 12, pp. 2570, 2020.
5. 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.
6. S. Munawar, M. Hamid, M. S. Khan, A. Ahmed and N. Hameed, "Health Monitoring Considering Air Quality Index Prediction Using Neuro Fuzzy Inference Model A Case Study of Lahore Pakistan", Journal of Basic & Applied Sciences, 2017.
7. E. Pardo and N. Malpica, "Air Quality Forecasting in Madrid Using Long Short-Term Memory Networks", Biomedical Applications Based on Natural and Artificial Computing, vol. 10338, 2017.
8. B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines", European Journal of Operational Research, vol. 265, no. 3, pp. 993-1004, Mar 2018.
9. B. C. Liu et al., "Urban air quality forecasting based on multi-dimensional collaborative Support Vector Regression (SVR): A case study of Beijing-Tianjin-Shijiazhuang", PLOS, 2017.
10. C R Aditya, Chandana R Deshmukh, D K Nayana and Praveen Gandhi Vidyavastu, "Detection and Prediction of Air Pollution using Machine Learning Models", International Journal of Engineering Trends and Technology (IJETT), 2018.
11. Zhili Zhao, Jian Qin, Zhaoshuang He, Huan Li, Yi Yang and Ruisheng Zhang, "Combining forward with recurrent neural networks for hourly air quality prediction in Northwest of China", Environmental Science and Pollution Research, 2020.
12. Y. Zhang et al., "A Predictive Data Feature Exploration-Based Air Quality Prediction Approach," in IEEE Access, vol. 7, pp. 30732-30743, 2019, doi: 10.1109/ACCESS.2019.2897754.
13. B. Liu et al., "A Sequence-to-Sequence Air Quality Predictor Based on the n-Step Recurrent Prediction," in IEEE Access, vol. 7, pp. 43331-43345, 2019, doi: 10.1109/ACCESS.2019.2908081.
14. J. Chen, K. Chen, C. Ding, G. Wang, Q. Liu and X. Liu, "An Adaptive Kalman Filtering Approach to Sensing and Predicting Air Quality Index Values," in IEEE Access, vol. 8, pp. 4265-4272, 2020, doi: 10.1109/ACCESS.2019.2963416.
15. 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," in IEEE Access, vol. 6, pp. 38186-38199, 2018, doi: 10.1109/ACCESS.2018.2849820.
16. El Fazziki, D. Benslimane, A. Sadiq, J. Ouarzazi and M. Sadgal, "An Agent Based Traffic Regulation System for the Roadside Air Quality Control," in IEEE Access, vol. 5, pp. 13192-13201, 2017, doi: 10.1109/ACCESS.2017.2725984.
17. K. Gu, J. Qiao and W. Lin, "Recurrent Air Quality Predictor Based on Meteorology- and Pollution-Related Factors," in IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 3946-3955, Sept. 2018, doi: 10.1109/TII.2018.2793950.
18. Mokhtari, W. Bechkit, H. Rivano and M. R. Yaici, "Uncertainty-Aware Deep Learning Architectures for Highly Dynamic Air Quality Prediction," in IEEE Access, vol. 9, pp. 14765-14778, 2021, doi: 10.1109/ACCESS.2021.3052429.
19. Y. Yang, G. Mei and S. Izzo, "Revealing Influence of Meteorological Conditions on Air Quality Prediction Using Explainable Deep Learning," in IEEE Access, vol. 10, pp. 50755-50773, 2022, doi: 10.1109/ACCESS.2022.3173734.