Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 12-23
http://ilkogretim-online.org.tr
doi: 10.17051/ilkonline.2021.05.01

Adaptive kernel fuzzy weighted particle swarm optimized deep learning model to predict air pollution $PM_{2.5}$.

S. Jeya[1]
Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, Coimbatore, India

Dr. L. Sankari[2]
Associate Professor, Department of Computer Science, Sri Ramakrishna College of Arts andScience for Women, Coimbatore, India

Abstract. Precise prediction by deep learning algorithms will help regulate the pollutant $PM_{2.5}$ (particulate matter, inhalable particles with diameter 2.5μm micrometres or smaller) a global threat to the entire environment by adulterating the air causing respiratory, cardiovascular, numerous other diseases and long term exposure to the pollutantin turn leads to human mortality in the globe. A combination of both 1D CNN and BIGRU model used to forecast the menacing pollutant $PM_{2.5}$. For extemporizing proposed model accuracy, precise choice of hyper parameter selection is inevitable. Hyper parameters of ID CNN and BIGRU are automatically selected by a novel Adaptive kernel fuzzy weighted particle swarm optimization (AKFWPSO).Both $PM_{2.5}$ and meteorological hourly data set of Beijing from UCI Machine learning repository is exploited for this analysis. For measuring model performance three measurement approaches such as RMSE, MAE and SMAPE are used. The model accuracy is considered superior comparing theexisting model with estimation of error metrics.This model can be applied not only to oversee and regulate the $PM_{2.5}$but also alert the public when the amount of the pollutant level goes beyond the level prescribed.

Keywords: Adaptive kernel fuzzy weighted particle swarm optimization, 1D Convolutional Neural Network, Bidirectional Gated Recurrent Unit, Particulate Matter ($PM_{2.5}$)

I. INTRODUCTION:

Speedy industrialisation and urbanisation in developing countries pollutes the air and causes severe lung, heart diseases to people leads to premature death[1].Especially $PM_{2.5}$due toits tiny size easily penetrate alveoli causes great damage to lungs and pass through other organs of the body and affects that too in the long run [2]. Predicting$PM_{2.5}$accurately is highly challenging without the spatiotemporal correlations and long term dependencies [3].Recent studies implied that exposure to $PM_{2.5}$ causes various neuro degenerative diseases. Individual base leaners are weak in prediction when compared to the ensemble model which comprises composite machine learning algorithms [4]. Air pollution and weather factors are closely related especially wind direction, wind speed, temperature, air pressure, humidity, rainfall are some of the meteorological factors greatly influencing the pollutants to greater extent. Forecasting model has to be very accurate and till date numerous methodologies are adopted for predicting the air quality starting from statistical models to machine learning, deep learning models [5]. Combining CNN and RNN model to predict the $PM_{2.5}$ exploits convolutional neural networkfeature extraction ability besides expertise of time series prediction by Recurrent Neural Networks [6].Deep learning models require bestnetwork configuration, appropriate hyper parameter selection[10] and the proposed Adaptive kernel fuzzy weighted particle swarm optimization (AKFWPSO) is used for automatically optimize 1D CNN BIGRUhyper parameters and the combined model is used for anticipating $PM_{2.5}$.The Paper structure is organised as follows, Section II. Related Study III. Data Cleaning &Preparation IV. Correlation analysis V. Proposed Methodology. VI. Experimental Results. VII. Conclusion.

II.RELATED STUDY:

Shengdong Du et al. considering the non-linear, dynamic nature of pollutant data and to extract the spatiotemporal correlations and dependencies, suggested a hybrid model which is aamalgamation of 1D-CNN and Bi-LSTM to forecast $PM_{2.5}$ a menace to the environment and found the model's improved performance [7]. Athira V proposed a model to forecast PM_{10} pollutant by applying RNN, LSTM and GRU. Also,it is inferred through prediction results that three models have been performed reasonablyin a better way [8]. Particle swarm optimization helps automatic optimization of hyper parameters of 1D convolutional neural network effectively. Each iteration of PSO helps learn to achieve better hyper parameter values of 1D CNN[13]. Hyper parameter and layer selection of 1D CNN by Particle swarm optimization to get better rainfall prediction compared to ACCESS-S1 model in terms of MAE, RMSE and r[11].Xianming Zhu et al suggested a new strategy forcontrolling the inertia weight of PSO from Butterworth filter inspiration BPSO in order to have a balance among exploration and exploitation of the algorithm. Better performance is achieved when tested with some benchmark functions [12]. The CNN-LSTM model is applied for inventory forecasting.Model accuracy enhancement is achieved by automating optimal CNN-LSTM network architectures, tuning parameters by PSO and two differential evolution variants.The evolved CNN-LSTM model outperformed the performance by SARIMA model [15].

III. DATA CLEANING & PREPARATION:

The repository for the dataset is UCI machine learning for this analysis comprised of US Embassy in Beijing $PM_{2.5}$ hourly recordingsand Beijing Capital International Airportmeteorological data. Number of features are 13 which includes row number, year, month, day, hour, $PM_{2.5}$ concentration, Dew point, temperature, pressure, combined wind direction, cumulated wind speed, cumulated hours of snow, cumulated hours of rain for period from Jan 1st, 2010 to Dec 31st, 2014. Number of occurrences is 43824 rows [9]. Wind direction data encoded as float data, assigned −10 for NW, 0 for CV, 10 for SE and 20 NE for the sake of assessment. Data of previous timestamp is used to fill missing values of $PM_{2.5}$ due to sensor errors. If features are on different scales, identifying the trends in the data by deep learning algorithms may not produce the correct results. In order to overcome this problem all the data in the data set is normalised.

$$x_{std}^{i} = \frac{x^{i} - x_{mean}^{i}}{\sigma_{x}^{i}} \tag{1}$$

σ_x^i Is the variance and x_{std}^i is the standard deviation of the i^{th} variable [6].

IV. CORRELATION ANALYSIS: Irrelevant features in the dataset normally decrease the model performance. Selecting the best features i.e., features that are strongly related with the prediction variable increase the accuracy of the prediction analysis. Selecting

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 12-23
http://ilkogretim-online.org.tr
doi: 10.17051/ilkonline.2021.05.01

appropriate input variables may lead to effective model performance by fast training and reduce overfitting. Identifying the relationship between meteorological variables temperature, wind speed, wind direction, dew point, pressure, rain as well assnow with $PM_{2.5}$ being the target variable is done by correlation coefficient.

Table 1

$PM_{2.5}$, Meteorological correlation®	$PM_{2.5}$
$PM_{2.5}$	1
Dew Point	0.16
Temperature	-0.09
Pressure	-0.04
Wind Direction	0.11
Wind speed	-0.24
Snow	0.02
Rain	-0.05

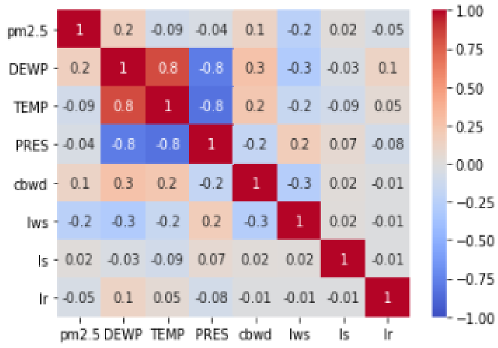

Fig.1-Heat map - Co-relation between $PM_{2.5}$ & Meteorological variables

The correlation coefficient of pressure, snow and rain with $PM_{2.5}$ is small not well correlated.
In order to avoid the model complexity and poor performance pressure, snow and rain fall are omitted. The pollutant $PM_{2.5}$and meteorological data such as well correlated Dew Point, Wind direction, negatively correlated Temperature,Wind speed are taken for analysis [6].

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 12-23
http://ilkogretim-online.org.tr
doi: 10.17051/ilkonline.2021.05.01

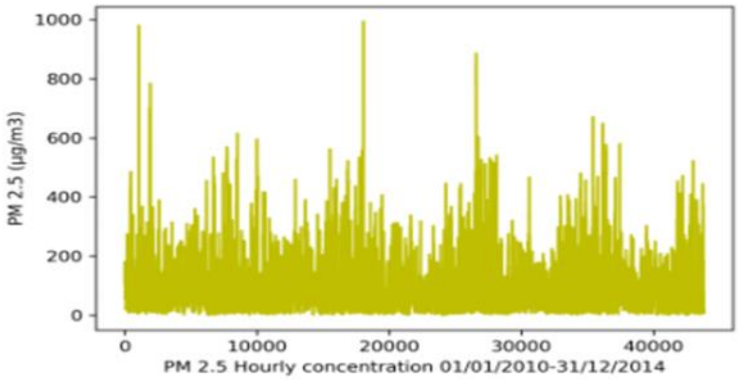

Fig.2 - $PM_{2.5}$ concentration.

V. PROPOSED METHODOLOGY:

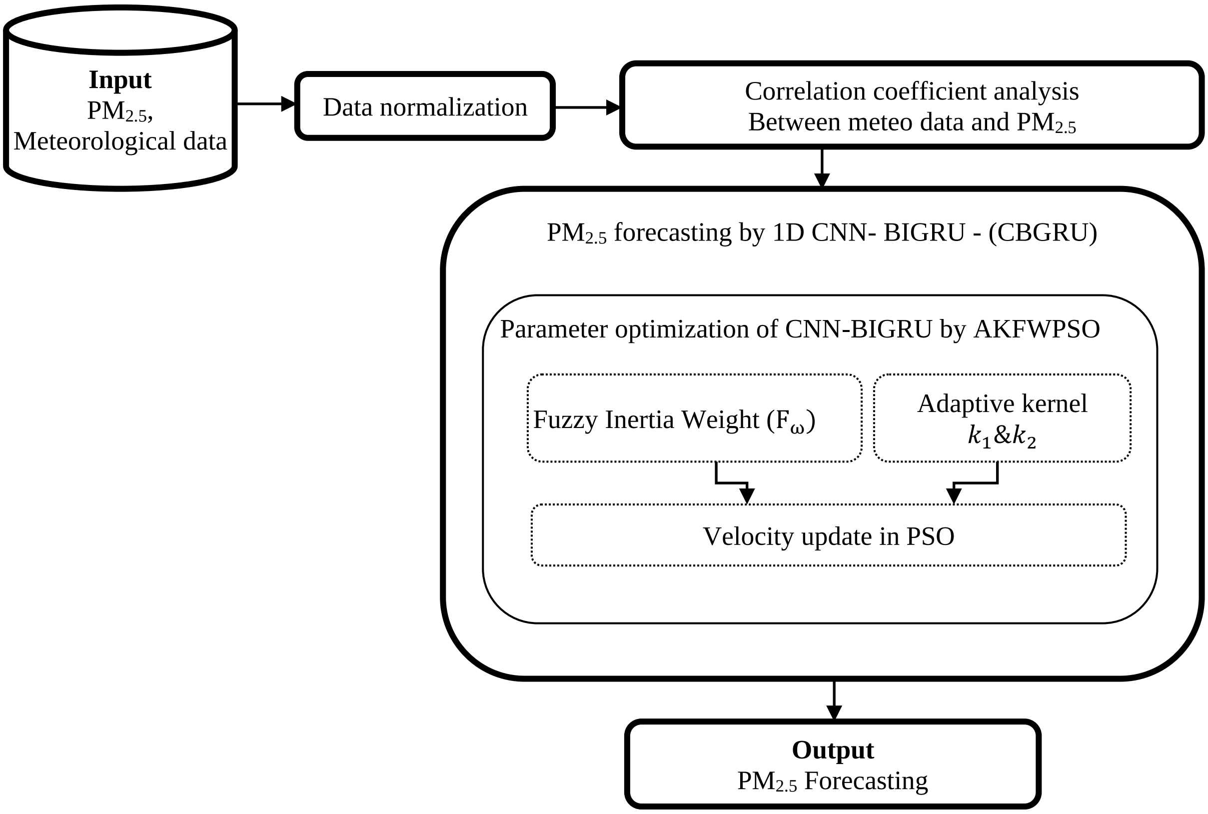

Fig.3- Proposed architecture - CBGRU model with AKFWPSO for $PM_{2.5}$ prediction

PSEUDOCODE - PARTICLE SWARM OPTIMIZATION (PSO)

```
Initialize all particles as CBGRU hyper parameter of
the swarm with randomly generated position and
velocity
For t=1: Maximum Generation
  For each particle (i-1=1,...N) in the swarm
```

If $f\left(x_{i,d}(t)\right) < f\left(p_{i,d}(t)\right)$ then $p_{i,d}(t) = x_{i,d}(t)$

$$f(p_{g(t)}) = \min_t \left(f(p_t(t))\right)$$

End

For $d = 1: dimension$

Update the velocity position of the particle

$$v_{i,d}(t+1) = v_{i,d}(t) + c_1 r_1(t)[p_{i,d}(t) - x_{i,d}(t)] + c_2 r_2 [p_g(t) - x_{i,d}(t)]$$

Update the position of the particle

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$$

If $v_{i,d}(t+1) > v_{max}$ then $v_{i,d}(t+1) = v_{max}$

Else if $v_{i,d}(t+1) < v_{min}$ then $v_{i,d}(t+1) = v_{min}$

End

If $x_{i,d}(t+1) > x_{max}$ then $x_{i,d}(t+1) = x_{max}$

Else if $x_{i,d}(t+1) < x_{min}$ then $x_{i,d}(t+1) = x_{min}$

End

End for

End for

Until (Stopping criteria met)

PSEUDOCODE - ADAPTIVE KERNEL FUZZY WEIGHTED PARTICLE SWARM OPTIMIZATION (AKFWPSO)

Initialize all particles as CBGRU hyper parameter of the swarm with randomly generated position and velocity

For t=1: Maximum Generation

For each particle (i-1=1,...N) in the swarm

If $f\left(x_{i,d}(t)\right) < f\left(p_{i,d}(t)\right)$ then $p_{i,d}(t) = x_{i,d}(t)$

$$f(p_{g(t)}) = \min_t \left(f(p_t(t))\right)$$

End

For $d = 1: dimension$

Update the velocity position of the particle

$$v_{i,d}(t+1) = F_\omega v_{i,d}(t) + c_1 r_1(t) * k_1[p_{i,d}(t) - x_{i,d}(t)] + c_2 r_2 k_2 [p_g(t) - x_{i,d}(t)]$$

Update the position of the particle

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$$

If $v_{i,d}(t+1) > v_{max}$ then $v_{i,d}(t+1) = v_{max}$

Else if $v_{i,d}(t+1) < v_{min}$ then $v_{i,d}(t+1) = v_{min}$

End

If $x_{i,d}(t+1) > x_{max}$ then $x_{i,d}(t+1) = x_{max}$

Else if $x_{i,d}(t+1) < x_{min}$ then $x_{i,d}(t+1) = x_{min}$

End

End for

End for

Until (Stopping criteria met)

A. Particle swarm optimization (PSO)

PSO algorithm is developed by Eberhart and Kennedy in the year 1995 is a meta-heuristic and population based algorithm used as an optimisation technique. Random particles which forms a swarm travels in search space foridentifyingfinest solution using some fitness function by updating generations. Each and every particle keeps an eye on its own best (personal best) and best of any other particle (global best), trying to balance exploration and exploitation. During the particles flight it updates the position and velocity using the formulae mentioned in the above pseudocode [14]. The performance of PSO is achieved prominentlythrough control parameters such as acceleration coefficients besides inertia weight.

For balancing the exploration and exploitation inertia weight linearly decreases from 0.9 to 0.4 with generations. Suitable PSO inertia weight strategy helps to improve the convergence speed. The search in PSO is specified throughequations for position(Equation (2)) and velocity update(Equation (3)) of every particle, correspondingly.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Where $x_i$represents particle i, t represents current iteration, $v_i$here denotes the velocity of particle i.

$$v_i(t+1) = \omega v_{ij}(t) + c_1 r_1(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_2[\hat{y}_j(t) - x_{ij}(t)] \tag{3}$$

Wherein the velocity$v_i$in the above equation represents of particle i in dimension j, c_1 represents cognitive factor (best previous position significance of particle), $c_2$notates social factor (best global position significanceof swarm), r_1, r_2 are random values generated by uniform distribution ranging amid [0, 1], y_{ij}represents particle best position i in dimension j, x_{ij}characterizes present position of the particle i and j represents dimension, $\hat{y}_j$denotes swarmbest global position in dimension j. Both components Cognitive and social of the particles velocity is maintained by following acceleration coefficients namely c1, c2[21].

Limitations of PSOalgorithm:In the PSO Algorithm, choosing of lesser inertia weight value trapped into local optima. It takes more time to repeat the algorithm, so it needs much time to get optimal results.

B. Adaptive kernel fuzzy weighted particle swarm optimization (AKFWPSO)

An Adaptive Kernel Fuzzy Weighted Particle Swarm Optimization (AKFWPSO) is proposed for optimized hyper parameter selection of (1D CNN, BIGRU) with two major modifications: Fuzzy Weight and Kernel function. In the kernel function, similarity between the features is computed via the use of the Radial Basis Function (RBF) which extraordinarilyoutclassestypical PSO. For searching qualityenhancement and to get effective results tuning of inertia weight in particle swarm optimization by fuzzy logic strategy is proposed.Fuzzy Inertia Weight (F_ω) by PSO is to tune the inertia weight dynamically to maintain good control of exploration and exploitation is needed during the optimization process. Careful attention to control premature convergence or trapped into local optima and convergence of swarm too quickly which may not be global optima is essential. So adjustment between global as well as local search is achieved by tuning F_ω adaptivelyin response to average relative kernel(k) [20].

The particles velocity and position in PSO is updated based on kernel diversity in order to achieve higher performance.

The algorithm dynamics are achievedthroughfine-tuning each particle position on basis of experienceattained for both particle and its neighbours.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4}$$

Where $x_i$indicates particle i, t specifiescurrent iteration, $v_i$designates particle i velocity.

$$v_{i,d}(t+1) = F_\omega v_{i,d}(t) + c_1 r_1(t) * k_1\left[p_{i,d}(t) - x_{i,d}(t)\right] + c_2 r_2 k_2\left[p_g(t) - x_{i,d}(t)\right] \tag{5}$$

A fuzzy weight vector F_ω is derived whose membership function estimation is done through interval weight vectors from an interval/crisp comparison matrix. F_ω,a membership function for a fuzzy set A on discourse X universe is defined as μ_A: $F_\omega \rightarrow$ [0,1], where every element of ω is mapped to a value amid 0 and 1. Triangular function is defined by a lower limit a, an upper limit b, and a value m, where a < m < b.

$$F_\omega = \mu_A(\omega) = \begin{cases} 0, \omega \le a \\ \frac{\omega - a}{m - a}, a < \omega < m \\ \frac{b - \omega}{b - m}, m < \omega < b \\ 0, \omega \ge b \end{cases} \tag{6}$$

If a <ω< m, then the result for triangular function is (ω - a) / (m - a).

If m <ω<b, then (b - ω)/(b - m) [19].

If ω<= a (or) ω>= b, then 0. Where the values are set as a = -2, m = 0 and b = 2. *For* ω, values are random between 0 and 1 generated for max iteration and stored.When the above formula applied with the specific values mentioned above for the max iteration, values which is in between 0.9 to 0.5 generated which improves the convergence speed and avoids falling into local optima.

The kernel is well-defined by Equation (7), in which swarm distribution quantity is computed. The parameter γ value is set as 0.6. Spread of the kernel is set by this parameter. The bell shaped curve width is decided by γ. If γ value is high then the bell will be narrower and vice versa. When distance between points is higher, they are dissimilar and obviously the kernel value is < 1 and close to 0. Similar points have less distance and have a larger kernel value [18]. The Radial basis function kernel or a Gaussian functions is used. If two vectors are closer then, the Euclidean distance $[\![x - x']\!]$ will be less.The RBF kernel is defined as

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 12-23
http://ilkogretim-online.org.tr
doi: 10.17051/ilkonline.2021.05.01

$$k\left(x_{i,d}(t), \bar{x}_d(t)\right) = \frac{1}{N} * \exp\left[-\gamma \left\|x_{i,d}(t) - \bar{x}_d(t)\right\|^2\right] \quad (7)[17]$$

Based on the value of k we compute bothk_1&k_2. If the feature is most similar to global we can increase k_2 and then updatek_1, where γ notates a parameter that sets kernel "spread.

Advantages of the proposed system: It takes lesser time to perform feature selection task in order to get optimal results.Proposed model is enhanced by modifying inertia weight and PSO learning factors in the course of the iteration processfor attaining optimized results.

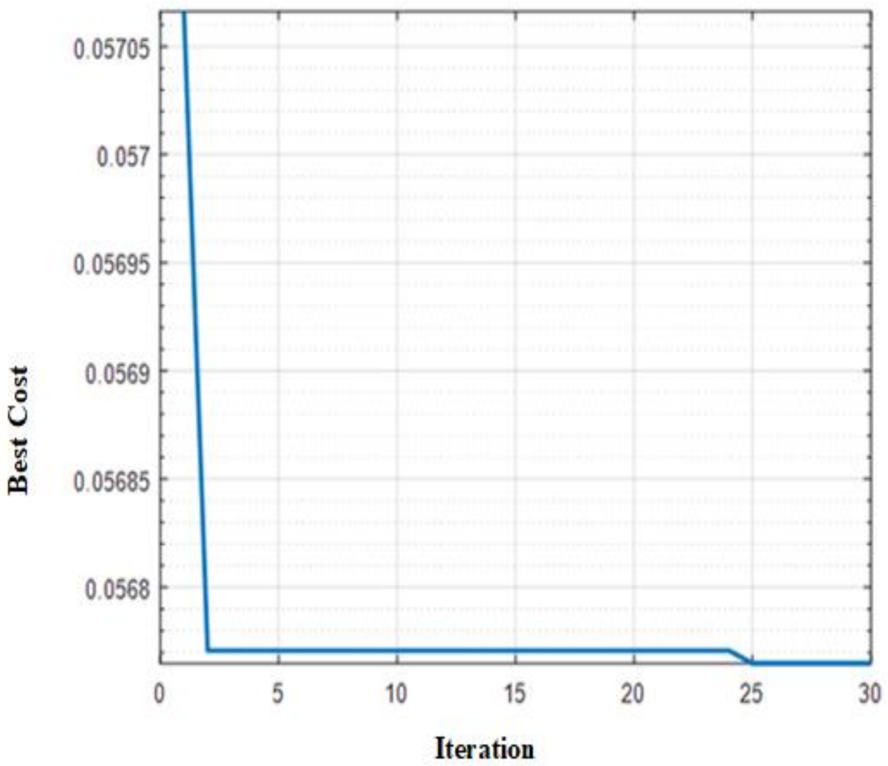

Fig.4-PSO Cost Convergence over iterations

C. 1D convolutional neural network & bidirectional gated recurrent unit

1D CNN are capable of performing local trend feature learning and sequence processing [6]. Certain applications with limited labelled data and extreme signal variations obtained from several sources, 1D CNN performs excellent. The main layers of 1D CNN are 1D convolution, Pooling (either maximum pooling or average pooling), drop out layer and fully connected layers [16]. The 1D CNN hyper parameters are number of layers, number of filters, the stride, and type of padding (same or valid), activation function, the dropout rate (dropout is generally to avoid overfitting and between in the range of 0 to 1 as real number), batch size and epochs [13].

Bidirectional Gated Recurrent Unit: GRU overcomes the traditional RNN vanishing gradient problem and the problem of learning long term dependency in some tasks. It consist of two gates update, and reset gate. No memory unit and this Bidirectional GRU are a perfect choice for time series forecasting. Its architecture is not complicated. Bidirectional GRU consists of two GRU and the input is processed from two directions chronologically and anti-chronologically then combines them which improve the prediction performance of pollution data efficiently [6].

Table 2: AKFWPSO Parameters

Swarm size	10
Iteration	30
C1	1.5
C2	2.0

Tuning
Table 3: 1D CNN possible hyper parametersbefore

Feature map	8......128
Filter size	2.....5
Stride	1,2
Pool type	Average, maximum
Pool filter size	2
Drop out	0.1,0.2,0.3,0.4
Optimizer	Adam

VI. EXPERIMENTAL RESULTS:

The mean absolute error commonly used error metric in most of the prediction tasks, Root mean squared error which is used to penalise in some cases where if the error is large . Symmetric mean absolute percentage error is scale independent.Proposed AKFWPSO is used for tuning the parameters of 1D CNN +BIGRUmentioned above. Both the strategy used for inertia weight and kernel strategy produced the effective predictive results which are summarised below.

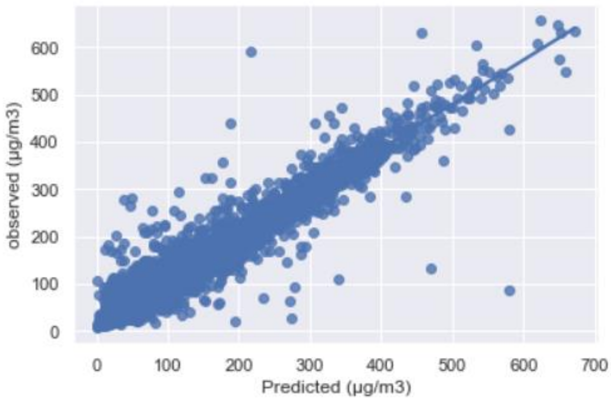

Fig.5- Scatter Plot for the observed vs predicted $PM_{2.5}$

For BIGRN neurons are 80.The Batch size is 24 and total number of epochs 20. Optimized MAE value obtained is 7.42, RMSE value found is 9.67 and SMAPE value is 0.1660 which is remarkably good when compared to existing MAE 7.53, RMSE 9.86 and SMAPE 0.1664.1D CNN parameters tuned for optimized result is zero padding, the

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 12-23
http://ilkogretim-online.org.tr
doi: 10.17051/ilkonline.2021.05.01

feature map is 8, filter size is 5 and stride is 1, Pooling layer filter size is 2, stride is 1, max pooling, Dropout rate is 0.1, activation function is RELU, optimiser is Adam.

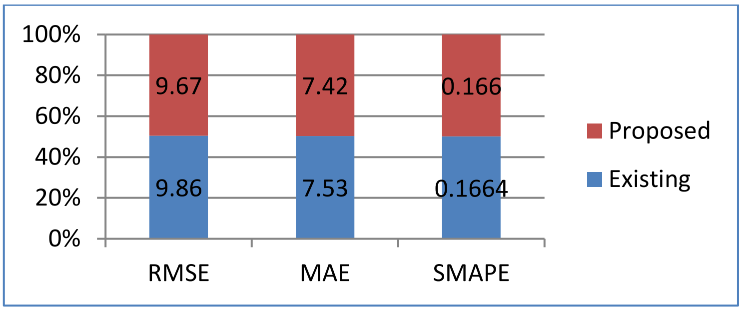

Fig. 6- Existing/Proposed Error evaluation Metrics

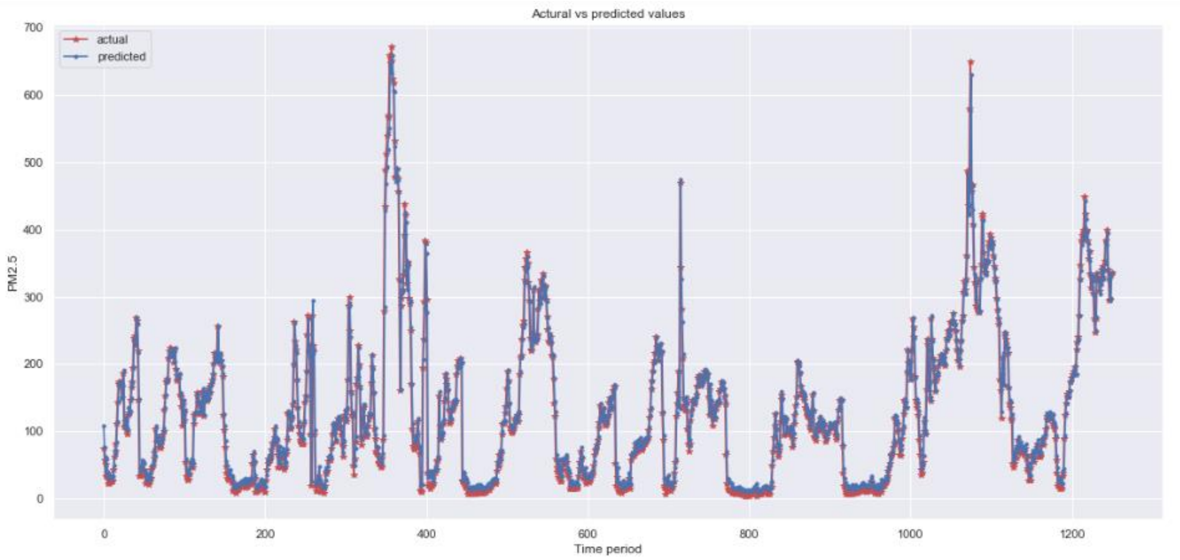

Fig.7-Forecasting results for $PM_{2.5}$ byAKFWPSO optimized CBGRU model

VII. CONCLUSION: The proposed AKFWPSO used for tuning the CBGRUhyper parameters and together 1D CNN BIGRU predicting the $PM_{2.5}$ time series data succeeds in accuracy by evaluating the RMSE, MAE and SMAPE. The improved prediction results of $PM_{2.5}$ by evolving the 1D CNN-BIGRUparameters are promising when compared to the existing methods. The proposed architecture optimized by AKFWPSO to predict $PM_{2.5}$ a highly volatile in nature data and is difficult to predict has achieved high performance with less computational speed. The outcome of this research helps to find out the accuracy in predicting the pollution level which in turn helps to monitor and control the pollution level by the government authorities.

References:

Li, X., Peng, L., Hu, Y. et al. Deep learning architecture for air quality predictions. Environ SciPollut Res 23, 22408–22417 (2016). https://doi.org/10.1007/s11356-016-7812-9.

Huang CJ, Kuo PH. A Deep CNN-LSTM Model for Particulate Matter ($PM_{2.5}$) Forecasting in Smart Cities. *Sensors (Basel)*. 2018; 18(7):2220. Published 2018 Jul 10. Doi: 10.3390/s18072220.

Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, Yuan Hu, Chengzeng You, Tianhe Chi, "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation", Environmental Pollution 231 (2017) 997-1004.

Qian Di, Heresh Amini, Liuhua Shi, Itai Kloog, Rachel Silvern, James Kelly, M.BenjaminSabath,Christine Choirat,PetrosKoutrakis,Alexei Lyapustin, Yujie Wang, Loretta J. Mickley,Joel Schwartz, "An ensemble-based model of $PM_{2.5}$ concentration across the contiguous United States with high spatiotemporal resolution", Environment International 130 (2019) 104909.

T. Li, M. Hua and X. Wu, "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5)," in *IEEE Access*, vol. 8, pp. 26933-26940, 2020.

Q. Tao, F. Liu, Y. Li and D. Sidorov, "Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU," in *IEEE Access*, vol. 7, pp. 76690-76698, 2019.

S. Du, T. Li, Y. Yang and S. Horng, "Deep Air Quality Forecasting Using Hybrid Deep Learning Framework," in *IEEE Transactions on Knowledge and Data Engineering*.

Athira V, Geetha P, Vinayakumar R, Soman K P, "DeepAirNet: Applying Recurrent Networks for Air Quality Prediction", Procedia Computer Science 132 (2018) 1394–1403.

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. Proceedings of the Royal Society A, 471, 20150257.

Ye F (2017) Particle swarm optimization-based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data. PLOS ONE 12(12): e0188746. https://doi.org/10.1371/journal.pone.0188746

A. Haidar, Z. M. Jan and B. Verma, "Evolving One-Dimensional Deep Convolutional Neural Network: A Swarm based Approach," *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, 2019, pp. 1299-1305, doi: 10.1109/CEC.2019.8790036.

Xianming Zhu, Hongbo Wang, "A new inertia weight control strategy for Particle Swarm Optimization", AIP Conference Proceedings **1955**, 040095 (2018); https://doi.org/10.1063/1.5033759.

NebojsaBacanin ,TimeaBezdan , Eva Tuba , Ivana Strumberger and Milan Tuba, ”Optimizing Convolutional Neural Network Hyperparameters by Enhanced Swarm Intelligence Metaheuristics”, Algorithms 2020, 13 (3), 67.

Yudong Zhang, Shuihua Wang, Genlin Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications", Mathematical Problems in Engineering, vol. 2015, Article ID 931256, 38 pages, 2015. https://doi.org/10.1155/2015/931256.

N. Xue, I. Triguero, G. P. Figueredo and D. Landa-Silva, "Evolving Deep CNN-LSTMs for Inventory Time Series Prediction," *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, 2019, pp. 1517-1524, doi: 10.1109/CEC.2019.8789957.

SerkanKiranyaz, OnurAvci, Osama Abdeljaber, TurkerInce, MoncefGabbouj, Daniel J.Inman, “1D Convolutional Neural Networks and Applications: A Survey”, arXiv preprint arXiv:1905.03554.

Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1396-1400.

Vert, Jean-Philippe, Koji Tsuda, and Bernhard Schölkopf. "A primer on kernel methods." *Kernel methods in computational biology* 47 (2004): 35-70.

(2004) Elements of the Theory of Fuzzy Sets. In: Flexible Neuro-Fuzzy Systems. The International Series in Engineering and Computer Science, vol 771. Springer, Boston, MA. https://doi.org/10.1007/1-4020-8043-3_2.

P. Yadmellat, S. M. A. Salehizadeh and M. B. Menhaj, "A New Fuzzy Inertia Weight Particle Swarm Optimization," *2009 International Conference on Computational Intelligence and Natural Computing*, Wuhan, 2009, pp. 507-510, doi: 10.1109/CINC.2009.180.

F. Olivas, F. Valdez and O. Castillo, "An interval type-2 fuzzy logic system for dynamic parameter adaptation in particle swarm optimization," *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, Boston, MA, 2014, pp. 1-6, doi: 10.1109/NORBERT.2014.6893881.