

Evaluation of Transformer Based Neural Network Architecture in Forecasting Pollution Trends

Pritthijit Nath, 001810501028, BCSE UG-IV, 2022

Abstract—With rising pollution concerns in recent times, producing refined and accurate predictions as a part of United Nations Sustainable Development Goals 11 (Sustainable Cities and Communities) and 13 (Climate Action) have gained utmost importance. As new model architectures become available, it becomes difficult for an average policymaker to evaluate various models comprehensively. One such architecture is the Transformer neural network which has shown an exceptional rise in various areas such as natural language processing and computer vision. This paper explores the performance of such a Transformer based neural network (referred in the paper as PolTrans) in the domain of pollution forecasting. Experiments based on four univariate city pollution datasets (Delhi, Seoul, Skopje and Ulaanbaatar) and two multivariate datasets (Beijing PM_{2.5} and Beijing PM₁₀) are performed against baselines consisting of widely used statistical, machine learning and deep learning methods. Findings show that although PolTrans performs comparatively better compared to existing deep learning methods such as BiDirectional long short-term memory networks (LSTM), LSTM AutoEncoder, etc. for modelling pollution in cities such as Beijing, Delhi and Ulaanbaatar, in the majority of cases, the PolTrans architecture lags behind statistical and machine learning methods such as AutoRegressive Integrated Moving Average (ARIMA), Random Forest Regression, Standard Vector Regression (SVR), etc. by a range of 1.5 - 15 units in terms of Root Mean Square Error.

Index Terms—Deep Learning, Air Quality, Transformer Neural Network, Time-Series

I. INTRODUCTION

ENVIRONMENTAL risk assessment [1] aims to identify the likelihood of future environmental hazards and the long-term effects such events can pose to various communities and urban systems all over the planet. Out of multiple such possible hazards possessing a looming threat to modern human civilization, climate change and air quality degradation have been identified as the ones requiring prominent attention from key stakeholders. With leading nations now declaring urgent action to combat the ongoing climate change a key subject in their future government policies [2], the need for leading technological advancements to develop novel solutions to mitigate such climatic risks have become of utmost importance for long-term sustainability related causes. To develop leading edge computational approaches as efforts to address prominent environmental issues, new and exciting research under the umbrella of Computational Sustainability [3] has gained traction in the last decade. With ideas from Artificial Intelligence

Pritthijit Nath is affiliated with the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. (e-mail: prithijit.nath@ieee.org). This project was performed under the supervision of Prof. Sarbani Roy (email: sarbani.roy@jadavpuruniversity.in).

(AI) coupled with numerous applications in interdisciplinary domains, research has shown that AI-based computing holds extensive promise in providing key actionable insights that can help society plan new policies to lessen the harm posed by future hazards.

With regards to modelling air quality in urban environments, methods from AI have presently become pervasive in recent times. Various versions of long short term memory (LSTM) networks along with regression models such as standard vector machines (SVM), linear regression coupled with models taken from econometrics such as autoregressive integrated moving average (ARIMA) are widely used in producing estimates with a fair amount of accuracy. Although LSTMs suffer from a lack of data in certain situations [4], statistical models such as ARIMA and Holt-Winters have been found to perform substantially better in univariate time-series predictions. Fields of Natural Language Processing (NLP) and time-series modelling are inherently intertwined and advances made in one domain often find key applications in the other domain. For example, deep learning based methods such as recurrent neural networks (RNN) and LSTMs often are used in extensive applications such as sentiment analysis, spam detection and document summarization to name a few.

A recent key innovation in seq2seq modelling originally for NLP applications has been the Transformer [5] neural network. Unlike LSTMs and RNNs, Transformers eliminate recurrence which helps decrease model complexity and also decrease training time as well. Further, due to the presence of independent components, Transformers can be easily parallelized thus improving computation efficiency when dealing with large sequence datasets. Taking motivation of the wide usage in NLP and recent Computer Vision related tasks [6], this paper investigates the utility of Transformer neural networks in air quality modelling. Since the original auto-encoder architecture was specially designed for NLP related sequence modelling tasks, this paper builds upon the original model and utilizes an altered version (referred to as PolTrans in this paper) which is a modified architecture suited for tasks such as one-step ahead prediction usually used in pollution count estimation. Additionally, this paper also guides a detailed performance comparison of the time-series models used in air quality modelling and explores any relative improvement PolTrans provides over widely used baselines.

The organisation of the remaining parts of this paper is as follows : in Section II, relevant research studies in this field are briefly discussed. Section III contains a detailed description of PolTrans. The results of the study along with a discussion of the data are laid out in Section IV. The conclusion of this

study is ultimately presented in Section V.

II. RELATED WORKS

This section presents a brief survey of related research studies and architectures proposed for temporal modelling of pollution time-series using various learning based methodologies.

1) *Statistical Learning*: Peng et al. [7] thoroughly investigated through a simulation study on air pollution as well as mortality data to compare the efficacies of various statistical approaches with respect to mean squared error. Koo et al. [8] performed a comparative study on air pollution index in Kuala Lumpur using fuzzy time-series models and found them to outperform other approaches in error and computation time. Dominici et al. [9] used generative-additive models to model time-series concerning air pollution and health effects. They found the estimates to be similar in Poisson regression and strict convergence parameter settings but upward biased in various default convergence counterparts. Reikard [10] investigated the prediction performance of various models on SO₂ emissions by the Kilauea volcano in Hawaii and found methods such as ARIMA to produce accurate estimations on daily and hourly granularities. For PM, regressions and simple persistence methods were found to perform better. Fang et al. [11] used Bayesian Model Averaging (BMA) method and reported the GAMM+BMA combination to produce accurate estimates in learning the association between PM₁₀ and mortality.

2) *Machine Learning*: Dincer et al. [12] reported showing successful forecasting results using a proposed fuzzy based time-series model built on top of Fuzzy K-Medoid clustering algorithm. Shahriar et al. [13] performed a comparative study of ML methods such as L-SVM, M-SVM Gaussian Process (GP) and Random Forest regression models and reported GP to perform significantly better in predictions of PM_{2.5} and PM₁₀. Chen et al. [14] found SVMs to produce better results in prediction of CO₂ and TVOCs and were able to outperform GP, M5P and ANNs. Wang et al. [15] et al. developed an improved version of SVM capable of online learning and used the model to estimate pollution levels based on the air pollutant database in Hong Kong. Staffogia et al. [16] used an ensemble methodology to establish relationships between PM and satellite land use and meteorological parameters. Their study produces estimates in 1sq. km spatial resolution and reported finding improvements in the addition of small-scale predictors at monitor locations.

3) *Deep Learning*: Espinosa et al. [17] based on exactness and robustness criteria compared multiple forecasting models such as 1DCNN, GRU, LSTM along with different linear regression models to create a multi-criteria methodology for air quality prediction. Niska et al. [18] presented the use of parallel genetic algorithm (GA) to design the architecture of MLPs for forecasting NO₂ emissions reported in Helsinki. The findings showed promising results and found GA to be a capable tool in neural network design. Dunea [19] proposed a wavelet-feed forward neural network architecture to predict pollutants such as O₃, NO₂ and PM. Their study reported

improvements in ANN performance with wavelet integration and were found to present positive results for O₃ and NO₂. Soh et.al [20] created an adaptive deep learning model which combined ANNs, CNNs and LSTMs which was used to extract spatio-temporal air quality relations for improved performance. Lei et al. [21] performed a case study using a hybrid approach involving CNNs and Bi-LSTMs for spatio-temporal modelling of pollution counts in Fushung, Liaoning Province in China.

While extensive studies have been organized in the past, the majority of them have focussed on the utility of classical methodologies such as econometric models, shallow ML models and diverse versions of LSTM in emission modelling and forecasting. Due to the inherent nature of pollutant behaviour, majority studies were regional and lacked a global perspective. This paper aims to investigate the utility of a transformer based model PolTrans in air quality modelling and how the prediction performance varies with respect to widely used baselines in major pollution risk areas around the world.

III. MODEL ARCHITECTURE

This section presents a description of the PolTrans model architecture outline in Fig. 1.

A. Input Data Pre-processing

1) *In2Out Sequence Modelling*: To convert the time-series data P_{data} into a series of sequences for supervised training, a new dataset is engineered from the original data by breaking into chunks using a rolling window λ_n of size n . As the window λ_n is slid over the 1-dimensional time-series P_{data} , the series $\{P_{data,k}\}_{k=i}^{k=n-i-1}$ is taken as the feature vector while $P_{data,n-i}$ is made to serve as the label for the i th row of the new dataset. Here, $1 \leq i \leq T - n + 1$ where T stands for the number of time-steps present in P_{data} . The new dataset Q_{data} can be then split and sampled into training, validation and test sets for evaluation purposes as and when needed.

2) *Time2Vec Encoding*: To create a model agnostic vector representation of time, the supervised feature vectors are passed through the Time2Vec [22] Encoding layer. For a scalar time notion τ , the resulting Time2Vec encoding $t2v(\tau)$ is a vector of size n that can be defined as follows:

$$t2v(\tau)[j] = \mathfrak{F}(\omega_j\tau + \varphi_j), 1 \leq j \leq n \quad (1)$$

where \mathfrak{F} represents the periodic function, $t2v(\tau)[j]$ represents the j th element of $t2v(\tau)$. As the layer is trainable, φ_j and ω_j are taken to be the learnable parameters. In PolTrans, \mathfrak{F} is taken as the sine function, as in case of $\sin(\omega_j\tau + \varphi_j)$, being inherently periodic, recurrent behaviors can be captured without an explicit need for feature engineering. The benefit of the Time2Vec encoding lies in three main properties : periodicity, invariance to time rescaling and finally simplicity. Being easily consumable the $t2v(\tau)$ representation can be used inserted in any model with minimal overhead.

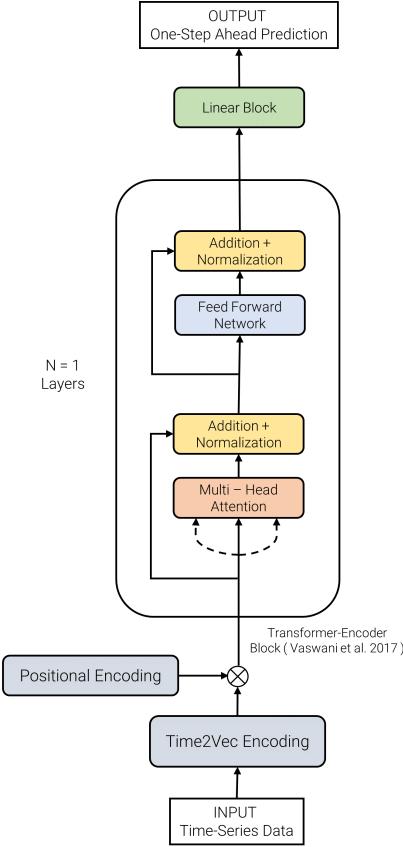


Fig. 1. PolTrans model architecture describing the flow of input, application of two kinds of encoding, modelling through the Transformer-Encoder block and output result as a one-step ahead forecast from the last linear block

3) Positional Encoding: To inject relative or absolute positioning information about the elements in the vector representation derived from the Time2Vec layer, positional encodings are added to the input embeddings. As presented in the original paper [5], PolTrans also uses a similar encoding \mathfrak{P} given as functions of different frequencies.

$$\begin{aligned}\mathfrak{P}_{(\text{pos}, 2i)} &= \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \\ \mathfrak{P}_{(\text{pos}, 2i+1)} &= \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right)\end{aligned}\quad (2)$$

Here, pos refers to the position in the vector representation while i stands for the dimension. By converting each position to a sinusoid, the relative difference can be expressed as a linear function of $\mathfrak{P}_{\text{pos}}$ enabling the model to easily learn the relative positions.

B. Encoder and Decoder Layer Stack

1) Encoder: Unlike the NLP Transformer, PolTrans uses only a stack of $N = 1$ layer. The transformer encoder block has two sub-layers. The first sub-layer employs a multi-head attention wrapped around by a residual connection followed by a normalization layer. Using a multi-head attention layer helps the encoder to attend to all positions as well as parallelise

through numerous attention layers, decreasing the overall computational cost. The second sub-layer uses the result from the first-layer and passes it through a fully connected feed-forward network wrapped around by a skip connection and followed by a normalization layer just like the first layer. The feed-forward network consists of two layers with the first layer containing a ReLU activation layer. The network output $y(x)$ with weights W_1, W_2 along with biases b_1 and b_2 can be mathematically expressed as follows :

$$y(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (3)$$

In PolTrans the model dimension d_{model} is fixed to 250. As there are multiple residual connections involved along with various forms of encoding, a single d_{model} eases the tracking of dimensions.

2) Decoder: In contrast to the NLP Transformer, as the main output is concerned only with a single continuous output, the decoder block consists of only a single linear layer with an output dimension of $d_{\text{model}} \times 1$.

3) Multi-Head Attention: In the original Transformer paper [5], multi-head attention promised extensive reductions in the total computational cost involved from values having dimension $d_v = d_{\text{model}}$ to $d_v = \frac{d_{\text{model}}}{h}$ where h is the number of parallel attention layers. In this technique, attention is computed in parallel thereby producing value of dimension d_v . Since, multiple computations can be done simultaneously on parallel CPU cores, the computation cost effectively reduces to that of single-head attention with full dimensionality. The technique of multi-head attention can be mathematically expressed in the following equations as follows :

$$\begin{aligned}\text{MultiHead}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) &= \text{Concatenate}(\mathcal{H}_1, \dots, \mathcal{H}_h) M^O \\ \mathcal{H}_i &= \text{Attention}(\mathcal{Q}M_i^Q, \mathcal{K}M_i^K, \mathcal{V}M_i^V) \\ \text{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) &= \text{softmax}\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}}\right)\mathcal{V}\end{aligned}\quad (4)$$

Here, the projections are parameter matrices $M_i^Q, M_i^K, M_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $M^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. \mathcal{Q}, \mathcal{K} and \mathcal{V} stand for query, key and value matrices respectively while \mathcal{H}_i stands for the i th head out of a total of h heads.

IV. RESULTS

This section presents detailed insights into the experiments performed with the PolTrans model and its relative performance against various baselines.

A. Experimental Setting

1) Datasets: To test PolTrans on a wide range of regional variations, throughout the world, five highly polluted cities were chosen: Delhi, Seoul, Skopje, Ulaanbaatar and Beijing with the first four datasets being univariate while the last being multivariate in nature. Being the national capital of India, Delhi ($28.7041^\circ\text{N}, 77.1025^\circ\text{E}$) has been the hub for socio-economic activities. With lax pollution measures and a lack of social consensus, Delhi has been a regular feature in the

world's most polluted cities list published annually by the World Health Organisation (WHO). Out of all developed cities, Seoul (37.5665°N , 126.9780°E) has been regarded to have the worst air quality among different developed cities around the world. The Korean government has been responsible for monitoring the pollution counts and has plans to decommission most of their coal power plants by 2025. Air pollution in Skopje, North Macedonia (41.9981°N , 21.4254°E) has been the highest in the European Union with pollutant levels sometime exceeding EU limits for more than 200 days in a year. Geographical location near a mountain and overreliance on wood and coal are the main contributors to the pollution in the city. Ulaanbaatar (47.8864°N , 106.9057°E), the capital city of Mongolia also is a regular feature of the most polluted cities list. With daily averages reaching to almost 27 times WHO recommended levels, the pollution in the city is a major public health emergency for the local government to solve in the coming years. Beijing (39.9042°N , 116.4074°E) before 2021 suffered from $\text{PM}_{2.5}$ levels reaching 90 times WHO's recommended daily levels, with the sky mostly filled with haze and heavy air pollution. Such adverse effects lead to higher morbidity and mortality and became an important area of concern for the Chinese authorities. Although the situation has drastically improved now, the city still suffers from albeit short heavy pollution spells.

Most of the data sourced for these cities came from publicly available datasets. City wise data for the period of 2018 - 2020 from different datasets in Kaggle [23] [24] [25] [26] as well as from national air quality monitoring agencies such as Central Pollution Control Board [27] and AirNow [28] were used. The multivariate dataset for station in Tiantan, Beijing belonging to the period from 2013-2017 was sourced from UCI Machine Learning Repository [29] [30] containing various pollutants and other meteorological factors of which $\text{PM}_{2.5}$, PM_{10} , Temperature, Dew Point and Pressure were chosen with the pollutant ($\text{PM}_{2.5}$ or PM_{10}) being the target variable.

2) Evaluation Metrics: As modelling time-series is inherently a regression task, the following four metrics: Root Mean Squared Error (RMSE), Mean Average Error (MAE) and Median Absolute Error (MedAE) were used to evaluate various method performances. For visualization purposes, Relative station-wise performance graphs in MedAE (only for univariate datasets since for Beijing, only a single station was used) along with Taylor diagrams were plotted to compare methods against each other.

3) Hardware and Software: The test setup used to carry out multiple experiments comprised of a 3.6 GHz 6C/12T processor with 16 GB 3000 MHz RAM and 8 GB RTX 2070 Super GPU for parallel matrix computation purposes. The project code was entirely based on Python with extensive support from libraries such as PyTorch, TensorFlow, Sci-kit Learn, Statsmodels and NumPy.

4) Implementation Details: For PolTrans, because of ReLU, weights were first initialized using Kaiming (He) initialization [31] method. A $n_{\text{heads}} = 10$ for multi-head attention in the encoder layer on extensive experimentation was found to provide optimum results. A learning rate $\alpha = 5 \times 10^{-3}$

was used inside an Adam optimizer [32] with weight decay enabled. A learning rate scheduler with stepsize = 1 and $\gamma = 0.95$ were taken as parameters to help in faster convergence. For experimental purposes PolTrans was trained with batchsize = 64 on 200 epochs. The In2Out sequences were constructed with window size $n = 2$. For testing purposes, the last 30% of the original time-samples was used as the test set while the first 70% was used for training and validation purposes.

5) Baselines: To measure the relative performance increase, three sets of evaluations were carried out. The first of evaluation placed PolTrans against popular deep learning methods such as Multi-Layer Perceptron (MLP), Bi-directional LSTM and LSTM Auto-Encoder. The second set and third set of evaluations involved using ML models such as SVM, Polynomial, Linear, Decision Tree and Random Forest regression along with statistical approaches such as ARIMA, AR and Holt-Winters respectively for each type of dataset. For multivariate datasets, in case of statistical approaches, ARIMAX and ARX were used due to their support for exogenous variables. For PolTrans the pollutant variable was used to perform modelling and subsequent forecasting in both kinds of datasets.

B. Performance in Univariate Time-Series Datasets

1) Comparison with Deep Learning Models: Upon performing detailed comparisons with various deep learning based methods, PolTrans (in terms of RMSE as can be seen in Table I) was found to outperform competing deep learning models for Delhi and Ulaanbaatar. The biggest difference in MedAE performance was witnessed in Ulaanbaatar, where PolTrans fared significantly better than other deep learning models by a significant margin. For Ulaanbaatar, PolTrans managed an RMSE of around 61.167 compared to 64.7 - 67.6 as in the case for other deep learning methods. In the case of Seoul and Skopje, PolTrans performed around 1.09 - 3.87 units worse in comparison to the best performing deep learning model.

TABLE I
PERFORMANCE METRICS AVERAGED OVER ALL STATIONS CITY-WISE FOR DEEP LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|-------------|------------------------|---------------|---------------|---------------|
| Delhi | BiDirectional LSTM | 31.958 | 49.658 | 18.439 |
| | LSTM AutoEncoder | 31.900 | 49.477 | 18.253 |
| | Multi Layer Perceptron | 34.017 | 51.746 | 20.956 |
| | PolTrans | 33.114 | 49.020 | 21.426 |
| Seoul | BiDirectional LSTM | 10.891 | 21.716 | 7.312 |
| | LSTM AutoEncoder | 10.869 | 21.538 | 7.305 |
| | Multi Layer Perceptron | 11.181 | 20.936 | 8.017 |
| | PolTrans | 11.855 | 22.597 | 8.398 |
| Skopje | BiDirectional LSTM | 11.491 | 19.416 | 6.086 |
| | LSTM AutoEncoder | 11.539 | 19.450 | 5.973 |
| | Multi Layer Perceptron | 11.726 | 18.942 | 6.533 |
| | PolTrans | 14.435 | 22.126 | 9.820 |
| Ulaanbaatar | BiDirectional LSTM | 43.269 | 64.750 | 31.713 |
| | LSTM AutoEncoder | 43.344 | 64.715 | 31.751 |
| | Multi Layer Perceptron | 46.575 | 67.627 | 34.814 |
| | PolTrans | 35.200 | 61.167 | 13.637 |

Inspecting the MedAE station-wise performance in Fig. 2, shows that PolTrans had mixed results in Delhi and Seoul.

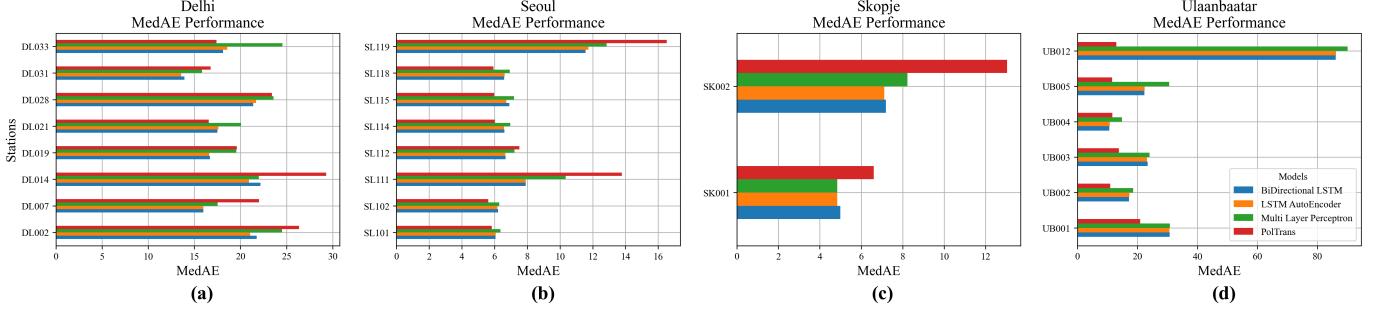


Fig. 2. MedAE performance of PolTrans vs other deep learning models (in $\mu\text{g}/\text{m}^3$) for different stations in (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

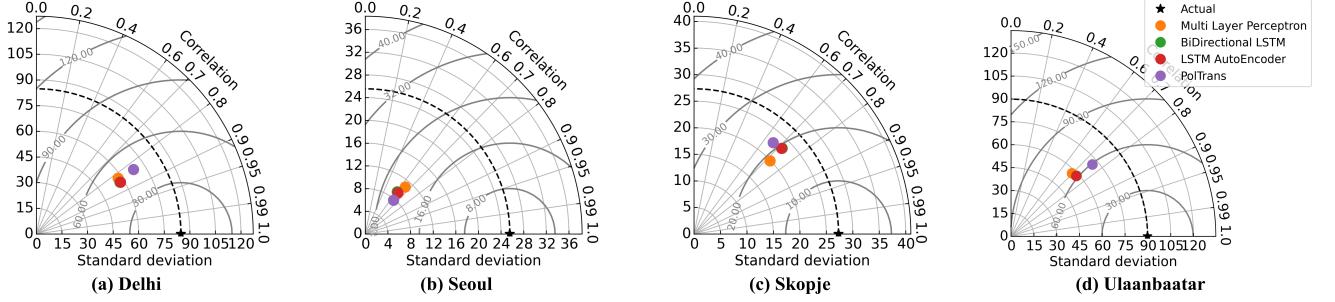


Fig. 3. Taylor diagrams comparing various deep learning methods vs PolTrans for (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

In Skopje, PolTrans consistently performed worse while in case of Ulaanbaatar, PolTrans consistently outperformed all other deep learning models in every station examined. In case of Seoul, for stations, SL119 and SL111, PolTrans showed MedAE deviation in the average of around 4 units from the competition. Conversely, in case of Ulaanbaatar, PolTrans showed improvements of as much as 60 units for UB012 in comparison. The two versions of LSTM used namely, LSTM Auto-Encoder and Bi-Directional LSTM were found to perform close to each other in almost all stations for all cities with minute differences separating the two.

On examining the Taylor diagrams in Fig. 3, except in case of Seoul, PolTrans showed higher standard deviation in comparison to other deep learning models. Although RMSE and standard deviation performance might have been off w.r.t. competition, the correlation of predicted values in comparison to actual values could be found to be similar without much deviation.

2) Comparison with Machine Learning Methods: In this comparison, as evident from Table II, PolTrans fared worse than its competition in every city examined. Although there does not seem to be present a clear winner, mostly SVR and Random Forest Regression performed the best out of all. Glancing overall comparisons PolTrans deviated with RMSE in the 2.19 - 4.09 units range. Out of all models experimented upon, Decision Tree Regression was found to perform the worst with RMSE values off by huge margins when compared to the best performing models.

Looking at individual station performances in Fig. 4, although PolTrans may not have performed the best, it certainly was also not the worst one being experimented upon. For

TABLE II
PERFORMANCE METRICS AVERAGED OVER ALL STATIONS CITY-WISE FOR MACHINE LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|-------------|---------------|---------------|---------------|---------------|
| Delhi | Decision Tree | 45.666 | 69.912 | 27.575 |
| | Linear | 31.725 | 48.267 | 19.574 |
| | PolTrans | 33.114 | 49.020 | 21.426 |
| | Polynomial | 32.703 | 51.767 | 19.556 |
| | Random Forest | 32.667 | 49.121 | 21.138 |
| | SVR | 31.582 | 47.862 | 19.828 |
| Seoul | Decision Tree | 13.851 | 25.654 | 8.377 |
| | Linear | 10.269 | 20.407 | 6.109 |
| | PolTrans | 11.855 | 22.597 | 8.398 |
| | Polynomial | 16.712 | 52.338 | 5.973 |
| | Random Forest | 10.253 | 21.175 | 6.186 |
| | SVR | 10.502 | 22.834 | 5.991 |
| Skopje | Decision Tree | 17.365 | 28.644 | 10.532 |
| | Linear | 11.675 | 18.533 | 6.864 |
| | PolTrans | 14.435 | 22.126 | 9.820 |
| | Polynomial | 12.084 | 20.162 | 6.751 |
| | Random Forest | 12.185 | 20.502 | 6.852 |
| | SVR | 11.666 | 19.998 | 6.361 |
| Ulaanbaatar | Decision Tree | 41.375 | 76.775 | 13.412 |
| | Linear | 31.908 | 58.695 | 12.576 |
| | PolTrans | 35.200 | 61.167 | 13.637 |
| | Polynomial | 33.995 | 64.730 | 12.136 |
| | Random Forest | 31.017 | 57.071 | 11.744 |
| | SVR | 31.973 | 60.221 | 11.715 |

station UB012, PolTrans was the best performing of all machine learning models tested. In majority of the stations, Decision Tree Regression showed the highest MedAE error among all with SVR and Random Forest being the best in

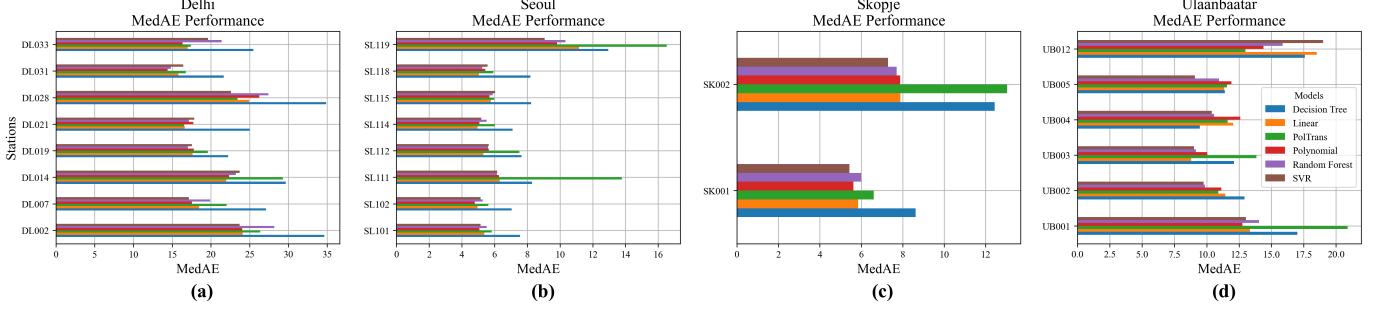


Fig. 4. MedAE performance of PolTrans (indicated in green) vs other machine learning models ($\mu\text{g}/\text{m}^3$) for different stations in (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

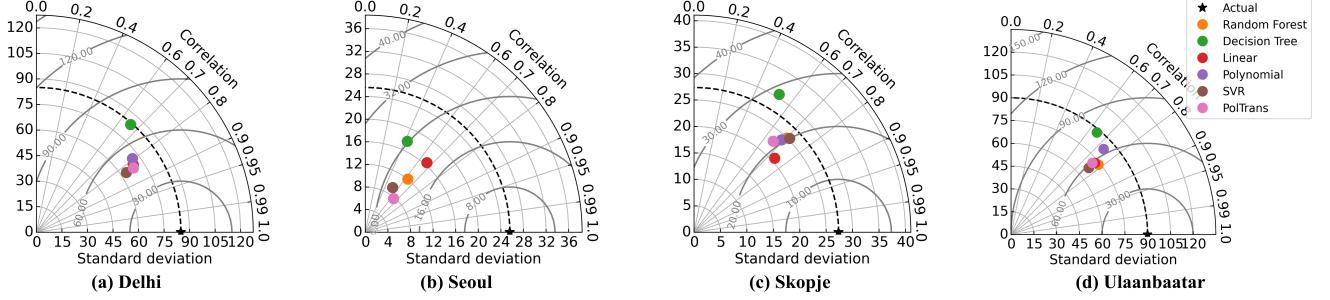


Fig. 5. Taylor diagrams comparing various machine learning methods vs PolTrans for (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

almost all stations.

Taylor diagrams for this comparison in Fig. 7 show that the performance of machine learning methods (except Decision Tree Regression) and PolTrans were close to each other barring a few deviations. One notable observation can be found for Decision Tree Regression, wherein all cities the model consistently showed higher standard deviation coupled with lower correlation scores compared to others.

3) *Comparison with Statistical Methods:* Analysis of the performance metric in Table III shows that econometric methods such as AR and ARIMA consistently performed well in all stations experimented on. With RMSE deviations in the range from 1.16 - 4.37 units, PolTrans in comparison to statistical methods has been found to lag behind the best performing statistical counterpart by a narrow margin.

Even though in most stations, PolTrans (as evident from Fig. 6) has been found to show the highest MedAE error out of all, in specific stations such as DL028 and UB012 in the cities of Delhi and Ulaanbaatar respectively, PolTrans has been able to outperform the best performing statistical method used. Although the position of best performing model station-wise was jointly owned by AR, ARIMA and Holt-Winters in proportional amounts, for most of the Ulaanbaatar stations specifically UB012, UB005 and UB004, AR was found to show disproportionately high deviations in MedAE error compared to the competition.

Except Seoul, predictions made by PolTrans had correlations close to each other, along 0.6 - 0.8 levels as can be evident from Fig. 7. For Seoul, although correlations between predicted and actual values were almost the same along the 0.6 to 0.7 mark, standard deviations varied a lot, with Holt-

TABLE III
PERFORMANCE METRICS AVERAGED OVER ALL STATIONS CITY-WISE FOR STATISTICAL LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|-------------|--------------|---------------|---------------|---------------|
| Delhi | AR | 31.501 | 47.821 | 19.665 |
| | ARIMA | 30.994 | 47.234 | 18.698 |
| | Holt-Winters | 32.021 | 49.848 | 19.041 |
| | PolTrans | 33.114 | 49.020 | 21.426 |
| Seoul | AR | 10.452 | 21.431 | 6.187 |
| | ARIMA | 10.549 | 21.562 | 6.575 |
| | Holt-Winters | 12.102 | 26.199 | 6.711 |
| | PolTrans | 11.855 | 22.597 | 8.398 |
| Skopje | AR | 11.573 | 18.437 | 6.490 |
| | ARIMA | 11.531 | 18.601 | 6.575 |
| | Holt-Winters | 12.560 | 20.496 | 6.614 |
| | PolTrans | 14.435 | 22.126 | 9.820 |
| Ulaanbaatar | AR | 32.491 | 58.698 | 14.128 |
| | ARIMA | 30.765 | 56.794 | 10.911 |
| | Holt-Winters | 31.843 | 59.312 | 11.291 |
| | PolTrans | 35.200 | 61.167 | 13.637 |

Winters and PolTrans being on the extreme ends with 24 to 8 units respectively.

C. Performance in Multivariate Time-Series Datasets

1) *Comparison with Deep Learning Models:* In the case of Beijing PM₁₀ and Beijing PM_{2.5} multivariate datasets, as evident from Table IV, PolTrans has been found to outperform other deep learning methods with great margin in all metrics of MAE, RMSE and MedAE. BiDirectional LSTM for both the multivariate Beijing datasets was found to perform consistently worse with RMSE values 83.391 and 75.782 respectively.

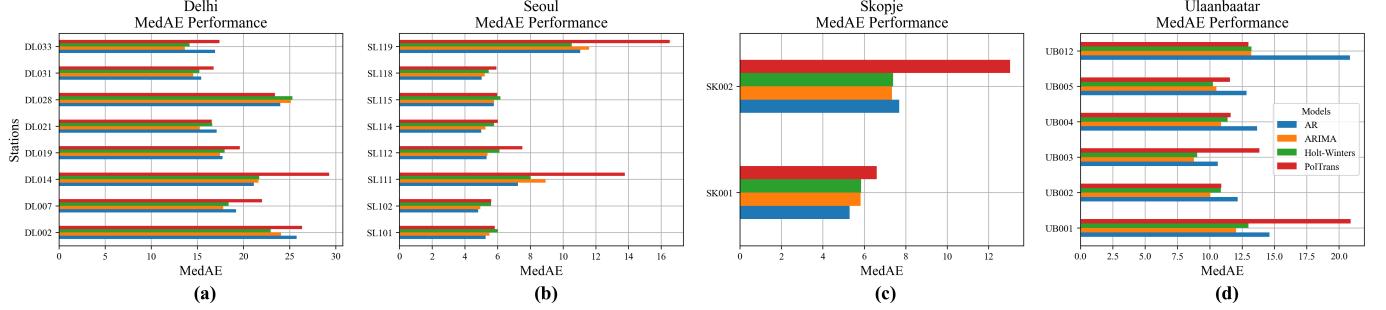


Fig. 6. MedAE performance of PolTrans (indicated in red) vs other statistical models ($\mu\text{g}/\text{m}^3$) for different stations in (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

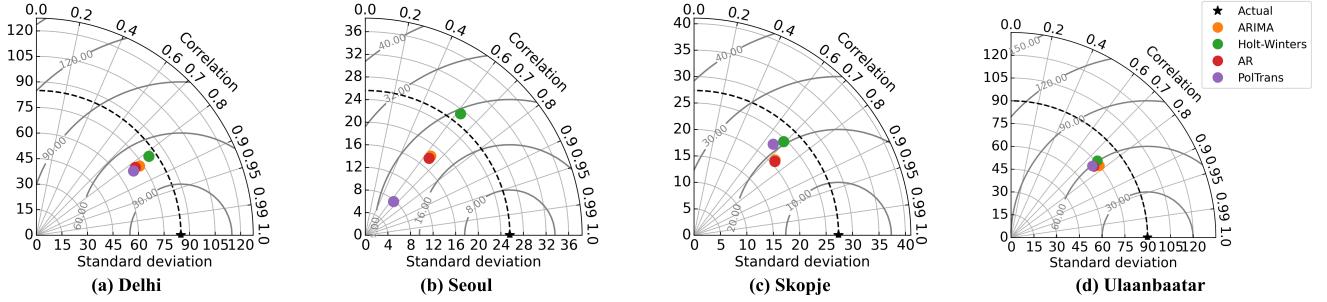


Fig. 7. Taylor diagrams comparing various statistical methods vs PolTrans for (a) Delhi, (b) Seoul, (c) Skopje and (d) Ulaanbaatar cities

TABLE IV
PERFORMANCE METRICS IN MUTI-VARIATE DATASETS FOR DEEP LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|---------------------------|------------------------|---------------|---------------|---------------|
| Beijing PM ₁₀ | BiDirectional LSTM | 56.432 | 83.391 | 39.139 |
| | LSTM AutoEncoder | 50.548 | 70.668 | 40.072 |
| | Multi Layer Perceptron | 50.877 | 70.085 | 42.114 |
| | PolTrans | 48.725 | 68.457 | 35.999 |
| Beijing PM _{2.5} | BiDirectional LSTM | 54.237 | 75.782 | 39.579 |
| | LSTM AutoEncoder | 51.290 | 71.671 | 38.711 |
| | Multi Layer Perceptron | 48.665 | 64.222 | 41.845 |
| | PolTrans | 45.161 | 62.305 | 34.179 |

was found to show the lowest standard deviation for equivalent correlation values in comparison to the rest.

2) *Comparison with Machine Learning Models:* Similar to the scenario for univariate datasets mentioned in Section IV-B2, PolTrans as evident from Table V was unsuccessful in faring better than its machine learning counterparts. With SVR and Polynomial Regression dominating in terms of MAE and RMSE respectively, PolTrans was found to be off by around 2.7 units in RMSE and 0.75 units in MAE compared to the best performing counterparts. Out of all, Decision Tree Regression was found to perform worse in terms of RMSE and MAE with huge margins in comparison to others.

TABLE V
PERFORMANCE METRICS IN MUTI-VARIATE DATASETS FOR MACHINE LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|---------------------------|---------------|---------------|---------------|---------------|
| Beijing PM ₁₀ | Decision Tree | 61.370 | 87.291 | 43.240 |
| | Linear | 48.372 | 67.374 | 36.312 |
| | PolTrans | 48.725 | 68.457 | 35.999 |
| | Polynomial | 47.474 | 66.626 | 36.055 |
| | Random Forest | 49.589 | 70.555 | 37.424 |
| | SVR | 47.130 | 69.578 | 32.953 |
| Beijing PM _{2.5} | Decision Tree | 56.388 | 81.938 | 37.395 |
| | Linear | 44.928 | 61.244 | 34.881 |
| | PolTrans | 45.161 | 62.305 | 34.179 |
| | Polynomial | 43.586 | 59.532 | 34.757 |
| | Random Forest | 44.485 | 61.574 | 32.474 |
| | SVR | 42.830 | 61.028 | 30.583 |

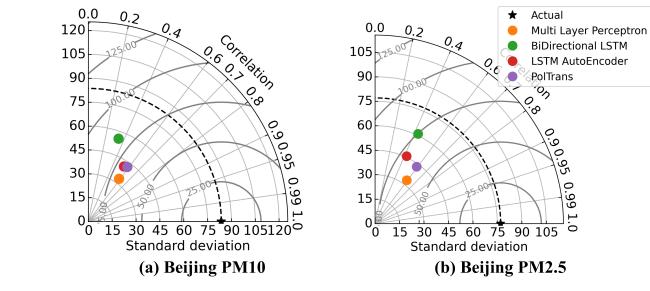


Fig. 8. Taylor diagrams comparing various deep learning methods vs PolTrans for (a) Beijing PM₁₀ and (b) Beijing PM_{2.5}

For Taylor diagrams in Fig. 8, even though most correlation values were in the range 0.6 - 0.7, the correlation for BiDirectional LSTM was found to be a little lower in the middle of 0.3 - 0.4 and alongside exhibited higher standard deviation compared other methods. The Multi Layer Perceptron method

A similar picture prevailed for Taylor diagrams as well in Fig. 9. With Decision Tree Regression being left with a

lower correlation in the 0.4 - 0.6 level and higher standard deviation around the 75 mark, other models clustered at standard deviation levels of 45 units with correlations in the 0.6 - 0.7 levels.

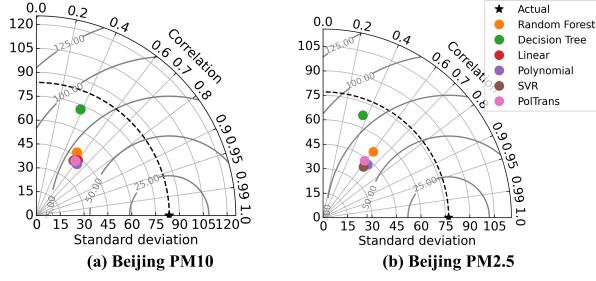


Fig. 9. Taylor diagrams comparing various machine learning methods vs PolTrans for (a) Beijing PM₁₀ and (b) Beijing PM_{2.5}.

3) *Comparison with Statistical Learning Models:* The exogenous equipped ARIMAX outperformed all other models used in this specific comparison, as can be seen in Table VI. PolTrans was the worst performer of all three with a margin of 12 - 13 units in MAE and 14 - 15 units in RMSE errors. Although ARX was not the worst performing model in the experiments, the difference in the performance metrics for ARX was lower compared to PolTrans and was close in the 5 - 6 units interval for MAE and RMSE.

TABLE VI

PERFORMANCE METRICS IN MUTI-VARIATE DATASETS FOR STATISTICAL LEARNING METHODS AND POLTRANS

| City | Model | MAE | RMSE | MedAE |
|---------------------------|----------|---------------|---------------|---------------|
| Beijing PM ₁₀ | ARIMAX | 36.718 | 54.600 | 25.198 |
| | ARX | 41.590 | 60.391 | 29.748 |
| | PolTrans | 48.725 | 68.457 | 35.999 |
| Beijing PM _{2.5} | ARIMAX | 32.750 | 47.804 | 22.407 |
| | ARX | 37.428 | 52.983 | 27.033 |
| | PolTrans | 45.161 | 62.305 | 34.179 |

The Taylor diagrams in Fig. 10 show a noticeable difference in the model performance for ARIMA, ARX and PolTrans. While PolTrans had correlation values around the 0.6 range, ARX and ARIMAX had values in the 0.7 and 0.8 mark. Regarding standard deviation, PolTrans showed lower values within 30 - 45 while ARX and ARIMAX had values ranging in 45 - 60 and 60 - 75 respectively.

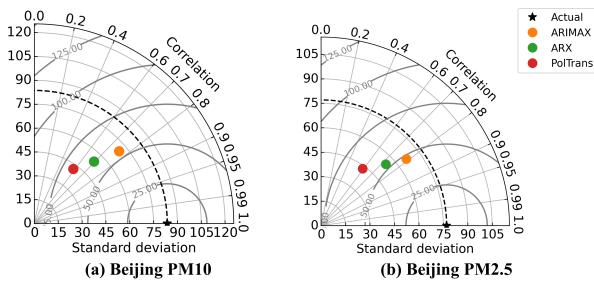


Fig. 10. Taylor diagrams comparing various statistical methods vs PolTrans for (a) Beijing PM₁₀ and (b) Beijing PM_{2.5}.

D. Discussion

From the extensive experimentation performed in testing PolTrans against multiple baselines from areas of deep learning, machine learning and statistical methods, although PolTrans shows great promise as a deep learning method in forecasting time-series data in the pollution domain, other methods in the statistical and machine learning group of models perform better in the performance metrics used. However, the biggest advantage that PolTrans benefits from lies in its ability to rapidly train (w.r.t. other deep learning methods) as well as parallelize in distributed settings. When the volume of data increases (order $\geq 10^6$), statistical methods such as ARIMA, Holt-Winters, etc. take higher training times due to their inherent modelling characteristics.

Also, looking at the nature of predictions produced by PolTrans in Fig. 11, where ordinary deep learning models normally succumb to high variability and often lead to projections close to the mean, PolTrans was successfully able to model the variability fairly and predict with greater accuracy.

Although the pollutants in focus were PM_{2.5} and PM₁₀, similar results could be expected in case of other major pollutants for different cities. In the experiments, the chief focus was made to pick cities which had high variation in pollution counts. In urban environments where the pollution scenario is relatively stable and shows regular periodicity (eg. New York City, London, Washington DC, etc.), PolTrans as well as other methods experimented, would help in production of much more refined and accurate insights with relatively higher accuracy.

V. CONCLUSION

In this paper, a transformer based deep learning approach (referred in this paper as PolTrans) based on the popular NLP Transformer was experimentally evaluated with respect to multiple baseline methods from statistical, machine learning and deep learning domains which are widely used in pollution modelling. Experiments were run on both univariate and multivariate datasets belonging to different major cities all around the world, and the results showed that although PolTrans shows good progress in comparison to other commonly used deep learning approaches such as BiDirectional LSTM, LSTM Auto-Encoder, MLP, etc. it still lags in comparison to machine learning and statistical approaches such as SVR, ARIMA, Holt-Winters, etc. Even though other models show greater advantage in estimation of pollution counts, in certain situations PolTrans especially those involving high volume environments, PolTrans shows extensive promise due to high parallelizable nature of the underlying Transformer based architecture.

With pollution modelling and monitoring becoming important day by day, it is hoped that such a comparative evaluation made in this paper with different key urban settings in focus, will assist concerned policymakers (especially those working under SDG 11 and 13) in selecting appropriate pollution modelling strategies to gauge their own pollution scenarios effectively.

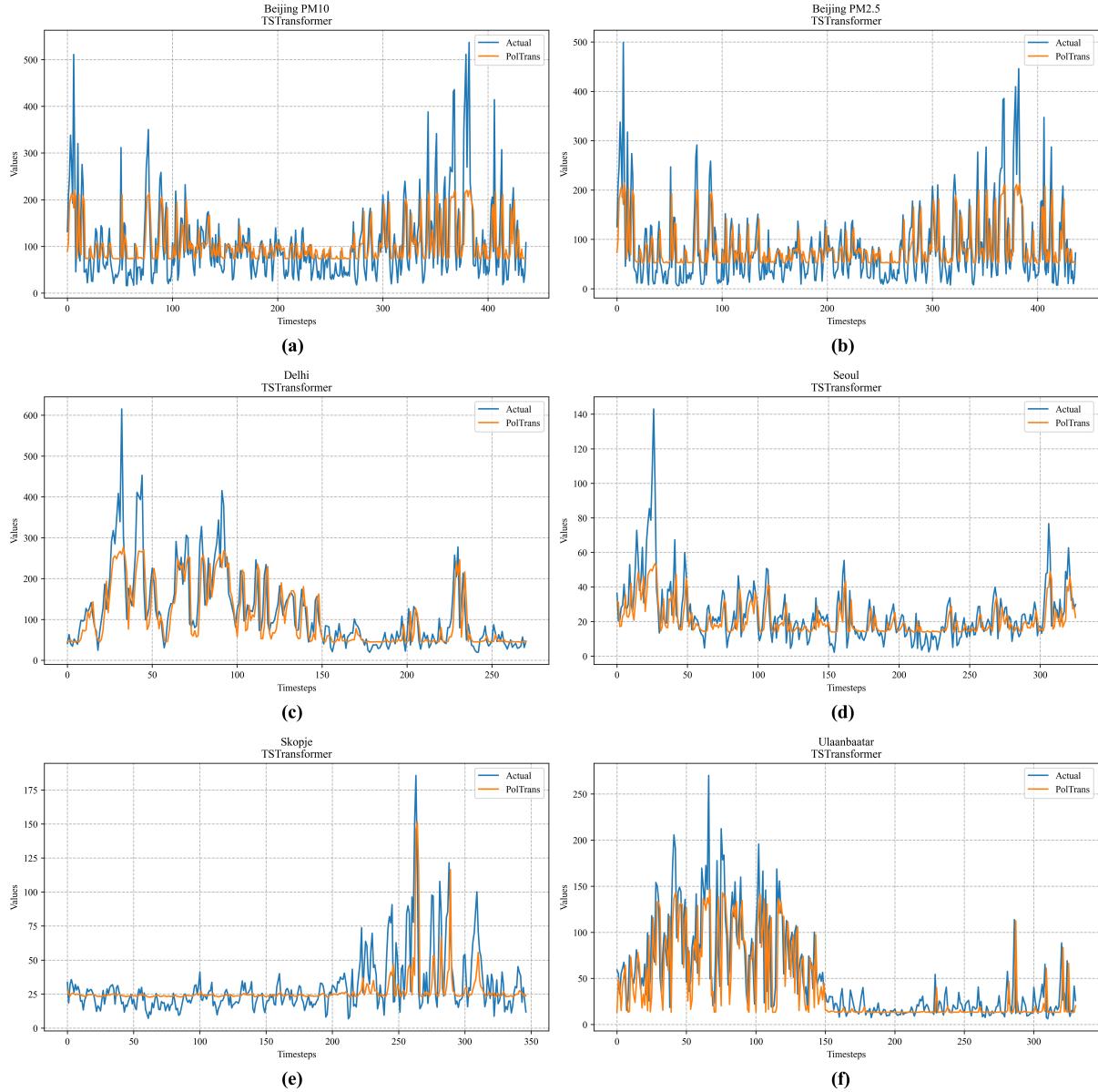


Fig. 11. Line plots demonstrating predictions made by PolTrans (indicated in orange) vs actual $\text{PM}_{2.5}$ / PM_{10} count as reported (in $\mu\text{g}/\text{m}^3$) for (a) Beijing PM_{10} , (b) Beijing $\text{PM}_{2.5}$ and a sample station in each of (c) Delhi, (d) Seoul, (e) Skopje and (f) Ulaanbaatar datasets

DATA AND CODE AVAILABILITY

The analysis code and data used for this paper can be found in the following link - <https://github.com/nathzi1505/PolTrans-Comparative-Study>. [Note: Since the repository is currently private due to confidential reasons, please email requesting for permission].

ACKNOWLEDGMENT

I am extremely thankful to Asif Iqbal Middya (Asif da) for taking out time to resolve doubts that came up as the project progressed. Needless to say the project would not be possible without his extensive mentorship and guidance.

REFERENCES

- R. N. Jones, "An environmental risk assessment/management framework for climate change impact assessments," *Natural hazards*, vol. 23, no. 2, pp. 197–230, 2001. [Online]. Available: <https://doi.org/10.1023/A:1011148019213>

- [2] "Climate change – united nations sustainable development." [Online]. Available: <https://www.un.org/sustainabledevelopment/climate-change/>
- [3] C. Gomes, T. Dietterich, C. Barrett, J. Conrad, B. Dilkina, S. Ermon, F. Fang, A. Farnsworth, A. Fern, X. Fern *et al.*, "Computational sustainability: Computing for a better world and a sustainable future," *Communications of the ACM*, vol. 62, no. 9, pp. 56–65, 2019. [Online]. Available: <https://doi.org/10.1145/3339399>
- [4] P. Nath, P. Saha, A. I. Middya, and S. Roy, "Long-term time-series pollution forecast using statistical and deep learning methods," *Neural Computing and Applications*, vol. 33, no. 19, pp. 12 551–12 570, 2021. [Online]. Available: <https://doi.org/10.1007/s00521-021-05901-2>
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [6] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings

- of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4055–4064. [Online]. Available: <https://proceedings.mlr.press/v80/parmar18a.html>
- [7] R. D. Peng, F. Dominici, and T. A. Louis, “Model choice in time series studies of air pollution and mortality,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 169, no. 2, pp. 179–203, 2006. [Online]. Available: <https://doi.org/10.1111/j.1467-985X.2006.00410.x>
- [8] J. W. Koo, S. W. Wong, G. Selvachandran, H. V. Long, and L. H. Son, “Prediction of air pollution index in kuala lumpur using fuzzy time series and statistical models,” *Air Quality, Atmosphere & Health*, vol. 13, no. 1, pp. 77–88, 2020. [Online]. Available: <https://doi.org/10.1007/s11869-019-00772-y>
- [9] F. Dominici, A. McDermott, S. L. Zeger, and J. M. Samet, “On the Use of Generalized Additive Models in Time-Series Studies of Air Pollution and Health,” *American Journal of Epidemiology*, vol. 156, no. 3, pp. 193–203, 08 2002. [Online]. Available: <https://doi.org/10.1093/aje/kwf062>
- [10] G. Reikard, “Volcanic emissions and air pollution: Forecasts from time series models,” *Atmospheric Environment: X*, vol. 1, p. 100001, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590162118300017>
- [11] X. Fang, R. Li, H. Kan, M. Bottai, F. Fang, and Y. Cao, “Bayesian model averaging method for evaluating associations between air pollution and respiratory mortality: a time-series study,” *BMJ Open*, vol. 6, no. 8, 2016. [Online]. Available: <https://bmjopen.bmjjournals.org/content/6/8/e011487>
- [12] N. Güler Dincer and Özge Akkuş, “A new fuzzy time series model based on robust clustering for forecasting of air pollution,” *Ecological Informatics*, vol. 43, pp. 157–164, 2018. [Online]. Available: <https://doi.org/10.1016/j.ecoinf.2017.12.001>
- [13] S. A. Shahriar, I. Kayes, K. Hasan, M. A. Salam, and S. Chowdhury, “Applicability of machine learning in modeling of atmospheric particle pollution in bangladesh,” *Air Quality, Atmosphere & Health*, vol. 13, no. 10, pp. 1247–1256, 2020. [Online]. Available: <https://doi.org/10.1007/s11869-020-00878-8>
- [14] S. Chen, K. Mihara, and J. Wen, “Time series prediction of co2, tvoc and hcho based on machine learning at different sampling points,” *Building and Environment*, vol. 146, pp. 238–246, 2018. [Online]. Available: <https://doi.org/10.1016/j.buildenv.2018.09.054>
- [15] W. Wang, C. Men, and W. Lu, “Online prediction model based on support vector machine,” *Neurocomputing*, vol. 71, no. 4, pp. 550–558, 2008, neural Networks: Algorithms and Applications 50 Years of Artificial Intelligence: a Neurononal Approach. [Online]. Available: <https://doi.org/10.1016/j.neucom.2007.07.020>
- [16] M. Stafoggia, T. Bellander, S. Bucci, M. Davoli, K. de Hoogh, F. de’ Donato, C. Gariazzo, A. Lyapustin, P. Michelozzi, M. Renzi, M. Scorticchini, A. Shtein, G. Viegi, I. Kloog, and J. Schwartz, “Estimation of daily pm10 and pm2.5 concentrations in italy, 2013–2015, using a spatiotemporal land-use random-forest model,” *Environment International*, vol. 124, pp. 170–179, 2019. [Online]. Available: <https://doi.org/10.1016/j.envint.2019.01.016>
- [17] R. Espinosa, J. Palma, F. Jiménez, J. Kamińska, G. Sciavicco, and E. Lucena-Sánchez, “A time series forecasting based multi-criteria methodology for air quality prediction,” *Applied Soft Computing*, vol. 113, p. 107850, 2021. [Online]. Available: <https://doi.org/10.1016/j.asoc.2021.107850>
- [18] H. Niska, T. Hiltunen, A. Karppinen, J. Ruuskanen, and M. Kolehmainen, “Evolving the neural network model for forecasting air pollution time series,” *Engineering Applications of Artificial Intelligence*, vol. 17, no. 2, pp. 159–167, 2004, intelligent Control and Signal Processing. [Online]. Available: <https://doi.org/10.1016/j.engappai.2004.02.002>
- [19] D. Dunea, A. Pohoata, and S. Iordache, “Using wavelet–feedforward neural networks to improve air pollution forecasting in urban environments,” *Environmental monitoring and assessment*, vol. 187, no. 7, pp. 1–16, 2015. [Online]. Available: <https://doi.org/10.1007/s10661-015-4697-x>
- [20] P. Soh, J. Chang, and J. Huang, “Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations,” *IEEE Access*, vol. 6, pp. 38 186–38 199, 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8392677>
- [21] F. Lei, X. Dong, and X. Ma, “Prediction of pm2. 5 concentration considering temporal and spatial features: A case study of fushun, liaoning province,” *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–11, 2020. [Online]. Available: <https://doi.org/10.3233/JIFS-201515>
- [22] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, “Time2vec: Learning a vector representation of time,” 2019. [Online]. Available: <https://arxiv.org/abs/1907.05321>
- [23] Vopani, “Air quality data in india (2015 - 2020),” Jul 2020. [Online]. Available: <https://www.kaggle.com/rohanrao/air-quality-data-in-india>
- [24] Bappe, “Air pollution in seoul,” Apr 2020. [Online]. Available: <https://www.kaggle.com/bappkekim/air-pollution-in-seoul>
- [25] S. Petrushevski, “Air pollution in skopje from 2008 to 2018,” Mar 2018. [Online]. Available: <https://www.kaggle.com/cokastefan/pm10-pollution-data-in-skopje-from-2008-to-2018>
- [26] R. Ritz, “Ulaanbaatar particulate matter pollution 2015-2018,” Nov 2018. [Online]. Available: <https://www.kaggle.com/robertritz/ulaanbaatar-particulate-matter>
- [27] Ministry of Environment, Forest and Climate Change, “Central control room for air quality management,” <https://cpcb.nic.in/>.
- [28] US Department of State, *Air Now International US Embassies and Consulates*. [Online]. Available: <https://www.airnow.gov/international/us-embassies-and-consulates/>
- [29] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [30] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, “Cautionary tales on air-quality improvement in beijing,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2205, p. 20170457, 2017.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html
- [32] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>