atmosphere

MDPI

Article

A Comparison of Machine Learning Methods to Forecast Tropospheric Ozone Levels in Delhi

Eliana Kai Juarez [1,*] and Mark R. Petersen [2]

[1] V. Sue Cleveland High School, Rio Rancho, NM 87144, USA
[2] Computational Physics and Methods Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA; mpetersen@lanl.gov
* Correspondence: elianajuarez0523@gmail.com

Abstract: Ground-level ozone is a pollutant that is harmful to urban populations, particularly in developing countries where it is present in significant quantities. It greatly increases the risk of heart and lung diseases and harms agricultural crops. This study hypothesized that, as a secondary pollutant, ground-level ozone is amenable to 24 h forecasting based on measurements of weather conditions and primary pollutants such as nitrogen oxides and volatile organic compounds. We developed software to analyze hourly records of 12 air pollutants and 5 weather variables over the course of one year in Delhi, India. To determine the best predictive model, eight machine learning algorithms were tuned, trained, tested, and compared using cross-validation with hourly data for a full year. The algorithms, ranked by R^2 values, were XGBoost (0.61), Random Forest (0.61), K-Nearest Neighbor Regression (0.55), Support Vector Regression (0.48), Decision Trees (0.43), AdaBoost (0.39), and linear regression (0.39). When trained by separate seasons across five years, the predictive capabilities of all models increased, with a maximum R^2 of 0.75 during winter. Bidirectional Long Short-Term Memory was the least accurate model for annual training, but had some of the best predictions for seasonal training. Out of five air quality index categories, the XGBoost model was able to predict the correct category 24 h in advance 90% of the time when trained with full-year data. Separated by season, winter is considerably more predictable (97.3%), followed by post-monsoon (92.8%), monsoon (90.3%), and summer (88.9%). These results show the importance of training machine learning methods with season-specific data sets and comparing a large number of methods for specific applications.

Keywords: ozone prediction; pollutant forecasting; machine learning; atmospheric monitoring; air quality

check for updates

Citation: Juarez, E.K.; Petersen, M.R. A Comparison of Machine Learning Methods to Forecast Tropospheric Ozone Levels in Delhi. *Atmosphere* **2022**, *13*, 46. https://doi.org/10.3390/atmos13010046

Academic Editors: Valentine Anantharaj, Forrest M. Hoffman, Udaysankar S. Nair and Samantha Vanessa Adams

Received: 11 November 2021
Accepted: 24 December 2021
Published: 28 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

According to the World Health Organization, air pollution is a leading cause of premature deaths, responsible for approximately 4.2 million deaths annually worldwide due to lung cancer, heart disease, respiratory diseases, and more [1]. One of these harmful pollutants is tropospheric ozone (O_3), or "ground-level ozone", which is produced when nitrous oxides (NOx) and volatile organic compounds (VOCs) undergo chemical reactions with sunlight and heat. These pollutants are emitted from a combination of anthropogenic (cars, power plants, etc.) and biogenic (soil and vegetative) sources. In addition to causing heart and lung diseases, tropospheric ozone is a general throat and lung irritant, causing coughing, wheezing, and difficult breathing [2]. Ozone also damages plant leaves and can harm agricultural crops [3].

As a secondary pollutant, tropospheric ozone is a candidate for accurate forecasting on the time scale of hours to days. Unlike the primary pollutants that are directly emitted by human activities, such as carbon monoxide (CO), NOx, and VOCs, secondary pollutants cannot be directly reduced or appropriately regulated. Ozone production also depends on environmental conditions, particularly sunlight, but also air temperature inversions that

inhibit convection. The combination of precursor pollutants and the influence of varying environmental conditions make ozone prediction difficult with traditional approaches; hence, this is an excellent potential application for machine learning (ML) methods, which can provide both forecasting capabilities and deeper insights into the causes of high ozone levels. This information can help regulatory agencies to limit emissions of NOx and VOCs during high-risk periods. Machine learning methods can improve the accuracy of predictive warning systems of pollutants so that residents may avoid outdoor activity, particularly the elderly and those with respiratory problems.

ML methods differ fundamentally from more traditional modeling of physical systems using conservation laws in the form of partial differential equations (PDEs). ML algorithms are trained purely on historical data and incorporate no information on the underlying physical laws. PDE-based models are useful where these laws are known, e.g., in atmosphere and ocean modeling [4,5], compressible turbulence [6], and protein folding [7], to name a few. However, most data-driven applications lack fundamental prognostic equations, or the models are too idealized for practical use—consider stock market prices, consumer purchase preferences, or facial recognition. In the past ten years, ML methods have proven so useful that they are being incorporated into model parametrizations, i.e., the parts of physical models that are unresolved or not easily expressed by PDEs [8,9]. Indeed, combining PDE-based weather forecasting systems with ML-based methods for the prediction of air pollutants may combine the best of both approaches [10].

Machine learning methods may be classified into supervised (where input variables map to output variables) and unsupervised (where only input data exists and the algorithm models the underlying data structure) methods [11]. Further, supervised methods are split into classification problems, where the output is a category, or regression problems, where the output is a real value. All methods discussed here are supervised regression methods, as the output data are the future ozone concentrations, a real-valued field used for training and prediction.

This study compares eight machine learning algorithms, which are representative of the categories of ML methods and are those in popular use: linear regression, KNN, SVM, Decision Trees, Random Forest, AdaBoost, XGBoost, and LSTM. The simplest method is *linear regression*, which computes the coefficients of a hyperplane to best fit the data. This method typically has the highest errors because it does not account for local variations and nonlinearities in the data, but it remains a baseline for comparison in many studies [12–14]. Instance-based algorithms create a database of specific instances and rely on local relations rather than global rules or generalizations [15], and they include *K-Nearest Neighbor* (kNN) and *Support Vector Machines* (SVM). In kNN [16], output values are simply averaged from the nearest neighbors, as measured in some distance norm in the input space. SVM [17] constructs hyperplanes between classes of input data, which maximize the separation space between the two classes. For regression problems with real-valued output, the hyperplane is approximated as a nonlinear function [18]. *Decision Tree Regression* (DTR) [14] algorithms use a forking tree structure to classify data and can be sensitive to small changes in the training data. DTRs are very popular for classification, such as product recommendations, but may also be applied to regression predictions, including environmental forecasting [19,20]. *Artificial neural networks* (ANN) model complex connections between input and output data sets with a middle "hidden layer" analogous to biological neurons. There are a large variety of neural network methods [21], and many have been successfully applied to air pollution prediction [22–25]. Here, we test the *Long Short-Term Memory* (LSTM) neural network method, which is well-suited to time-dependant data sets such as air pollution and weather measurements [26–28].

Ensemble methods are composed of a number of underlying models that are individually trained, and their results are combined to produce a final prediction [11]. In this study, ensemble methods include Random Forest, Adaptive Boosting (AdaBoost), and extreme gradient boosting (XGBoost). Random Forest constructs a large number of decision trees and returns the average prediction from individual trees. AdaBoost and XGBoost add other

learning methods, and the tree-growing process is adaptive in that each stage classifies the hardness of training samples, and later stages focus on harder-to-classify samples [18]. XGBoost, in particular, is highly optimized for speed, is designed to handle missing data, and supports regularization to reduce the potential of over-fitting [29]. Ensemble methods have gained popularity in recent years, including in air pollution forecasting, because they often outperform single-algorithm methods of machine learning [14,30]. However, results vary by application and even data set, so it is best to test and compare a suite of ML methods for each new project [13,18]. When counting all variations, there are easily 70–100 ML algorithms [11].

A number of studies have compared machine learning methods for urban air pollution prediction, including ozone, in recent years. Elkamel et al. [12] compared Artificial Neural Networks to both non-linear and linear regression models to predict current ozone levels based on meteorological conditions and precursor concentrations for a period of 60 days in Kuwait. Capilla [31] predicted ozone 1, 8, and 24 h in advance in an urban area on the eastern coast of the Iberian Peninsula and compared multiple linear regression with a Multi-Layer Perceptron network. Aljanabi et al. [13] predicted ozone in Amman, Jordan, one day in advance, comparing a Multi-Layer Perceptron neural network (MLP), SVM, DTR, and the XGBoost algorithm, and found that MLP performed the best. They applied feature selection to reduce the run time by 91% by only using the previous day's ozone, humidity, and temperature. Jumin et al. [14] predicted 12 and 24 h ozone concentrations in Malaysia and found that Boosted Decision Tree outperformed linear regression and neural network algorithms for all stations. They obtained R^2 values up to 0.91 for the 12 h dataset. Ozone prediction is not limited to ground measurements and may incorporate satellite data in remote areas [32].

We chose the city of Delhi, India, for this study on ozone prediction. India has some of the world's highest levels of air pollutants, resulting in increased risks of respiratory diseases for its large, densely populated urban populations. The country includes more than half of the 50 most polluted cities in the world (based on the most harmful pollutant, PM2.5) [33]. Delhi reports some of the highest ground-level ozone concentrations in the world, with values regularly exceeding 100 micrograms/m^3, the 8 h Indian National Air Quality Standard set by the Central Pollution Control Board [34]. This makes air quality research an urgent priority.

High-quality data are a prerequisite for this type of study. Critically, the Central Pollution Control Board of India reports hourly measurements of 12 pollutants in Delhi, including tropospheric ozone, with limited interruptions (approximately 96.4% complete), and these are freely available online [35]. Regular, consistent measurements of past weather data are also readily available for Delhi [36].

There has been other research on machine learning applications for air pollution prediction specifically in Delhi. Generally, most studies tested either one or two methods over the course of 1–2 years. Several have had a focus on predicting Particulate Matter 2.5 due to its extreme toxicity. SVM is a commonly used machine learning method for studies based in Delhi [37,38], perhaps due to its robustness to outliers and relatively flexible implementation. Studies by Sinha et al. [39,40] compared several machine learning algorithms for the daily prediction of several pollutants in Delhi. Shukla et al. [41] tested linear regression and Random Forest regression for the prediction of the pollutants NO, NO_2, and O_3, in which site-specific predictions using Random Forest had the best results. Krishan et al. [26] predicted one-hour forecasts of O_3, PM2.5, NOx, and CO in Delhi using LSTM, and included vehicular emissions and traffic data, as well as the more typical pollutant levels and meteorological conditions, in their training sets. They found that LSTM is quite efficient at capturing most aspects of air quality prediction for a one-hour forecast, with R^2 values ranging from 0.92 for ozone to 0.98 for PM2.5.

This research differs from other studies in several ways. This is one of the first studies to split the training data sets across India's four meteorological seasons, resulting in large improvements for winter forecasts and smaller improvements for the other three seasons.

Eight machine learning methods were analyzed and tuned for optimal performance in this study, as compared to 2–3 methods in most other studies, offering more robust analyses for comparing models for real-world implementation [12,26,31,41]. For ozone prediction in Delhi in particular, this work provides a useful comparison with previous work [26,40,41]. XGBoost, a relatively new algorithm that has not been as widely used in pollutant forecasting, is one of the methods tested here. A time-series analysis is also compared to current popular regression models with LSTM. Details of the selection of useful variables and their initial linear correlations with the target pollutant are also provided in this study.

This paper is organized as follows: Section 2 introduces the methods of data collection, preparation, and analysis. Section 3 describes the results, with daily averages and correlations of fundamental variables and the application and comparison of seven machine learning methods. Section 4 compares our results against past research. Finally, Section 5 concludes with remarks on the advantages of relevant methods and the prospects for machine learning for ozone forecasting.

2. Methods and Experimental Design

The goal of this experiment is two-fold. Firstly, we compare the predictive skill of a number of machine learning methods in the application of urban ozone forecasting. Secondly, we measure the improvement of seasonal training and prediction over annual training and prediction. Specifically, the experimental design is to train eight machine learning methods on a standard set of air quality and weather data and then predict the ozone concentration 24 h into the future. Feature selection was used to reduce the total available training data set to 12 pollutant and 4 weather variables. Parameter tuning was then used to optimize the performance of each method individually. Experiments were run for one year (2015) for the annual comparisons and five years (2015–2019) for the four seasonal-focused experiments, making a 24 h ozone prediction for every hour in that time span (Figure 1). Each of these experiments was repeated ten times for each machine learning method, where the training data were randomly assigned into 10 partitions and each experiment trained with 90% of the data and tested with the remaining 10%. The skill of the machine learning methods was assessed by computing various error metrics between the predicted ozone concentration and the actual concentration observed 24 h later. The details of the experimental design and the reasoning for these choices are explained in the following two sections. LSTM is a special case because it is trained with sequences of time-dependant data, and the experimental design is given in Section 3.4.

In order to determine the most suitable location for this study, we first evaluated daily air quality data from 25 Indian cities [35]. Delhi, Mumbai, Patna, and Amaravati reported the highest number of days over the acceptable pollutant limits, as defined by the NAAQS [42], making these cities the best candidates to evaluate the accuracy of forecasting methods. Out of these, Delhi was chosen due to the quality of its data, having close to 99% availability for most variables.

Hourly pollutant levels in Delhi were obtained for January 2015 to June 2020 from the Central Pollution Control Board of India [35], which is freely available for download, for station DL007 at latitude and longitude coordinates (28°33′06.2″ N 77°16′22.2″ E). In order to prepare the data fields for the regression models, invalid values with not-a-number (nan) entries were replaced with a linear interpolation between the surrounding valid values. These interpolated values were tracked, and never exceeded 3% of the full data. For future studies having data sets with a larger percentage of missing data, it is advisable to apply a data decomposition method to fill missing values [43].

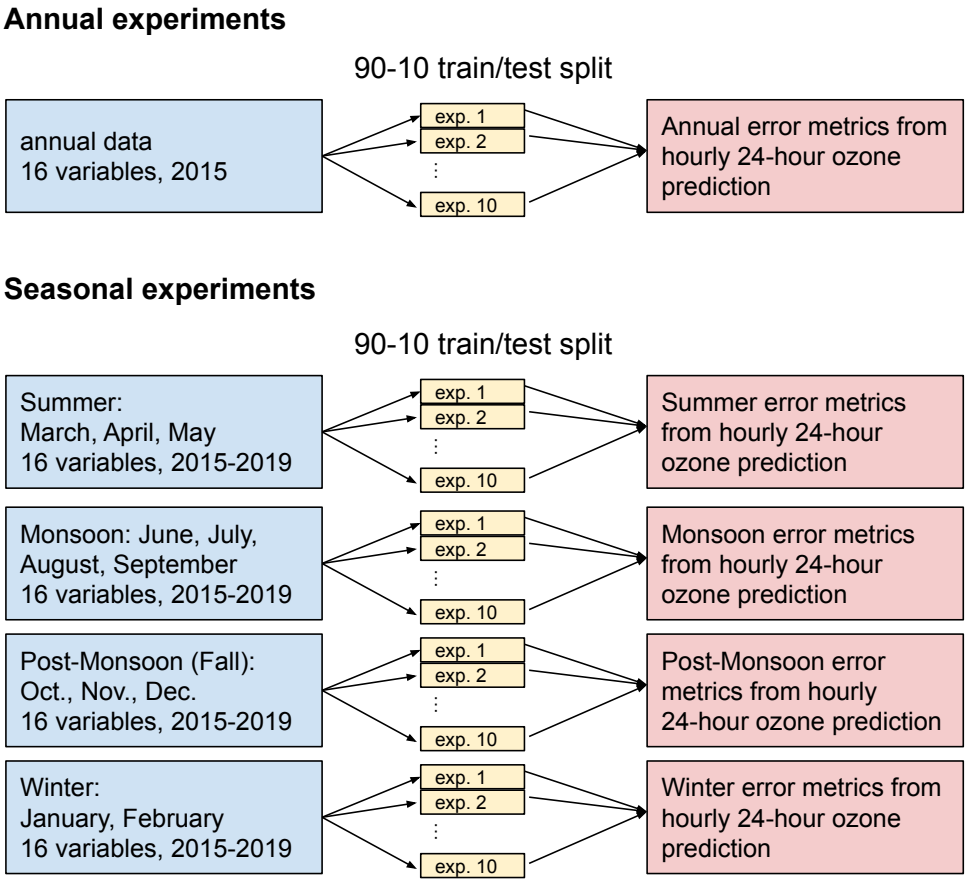

Figure 1. Diagram of the experimental design. This process was repeated for all machine learning methods tested in this study.

The meteorological data were obtained from the Visual Crossing Weather application program interface [36], using weather station GK2 with coordinates (28°31′12.0″ N 77°15′00.0″ E). The weather station was chosen to be within a 10 km radius of the station recording pollutant levels. The data were downloaded as a comma-separated values (CSV) file of hourly weather data in Delhi for the years 2015–2020, and the same interpolation process was applied for invalid values. The final data set included 12 pollutant variables and 5 weather variables, shown in Table 1, which were compiled into a single CSV file and manipulated using the Python library Pandas (Figure 2).

Eight machine learning algorithms were applied to the pollutant and weather data in order to predict ozone concentrations from 1 to 24 h in the future. Table 2 shows the parameters chosen from the Python scikit learn library. For each algorithm, each possible combination of hyperparameters was tested on a smaller sample of the first 4 months of data (to reduce training time), and the R^2 values were compared. The best combinations were recorded and later used when training and testing the models on the entire data set. For the algorithms that required it (KNN and SVM), three scaling methods (StandardScaler, MinMaxScaler, RobustScaler) were tested, with RobustScaler being chosen for reasons discussed later. Further methodological details are also provided, with the corresponding results, in the next section.

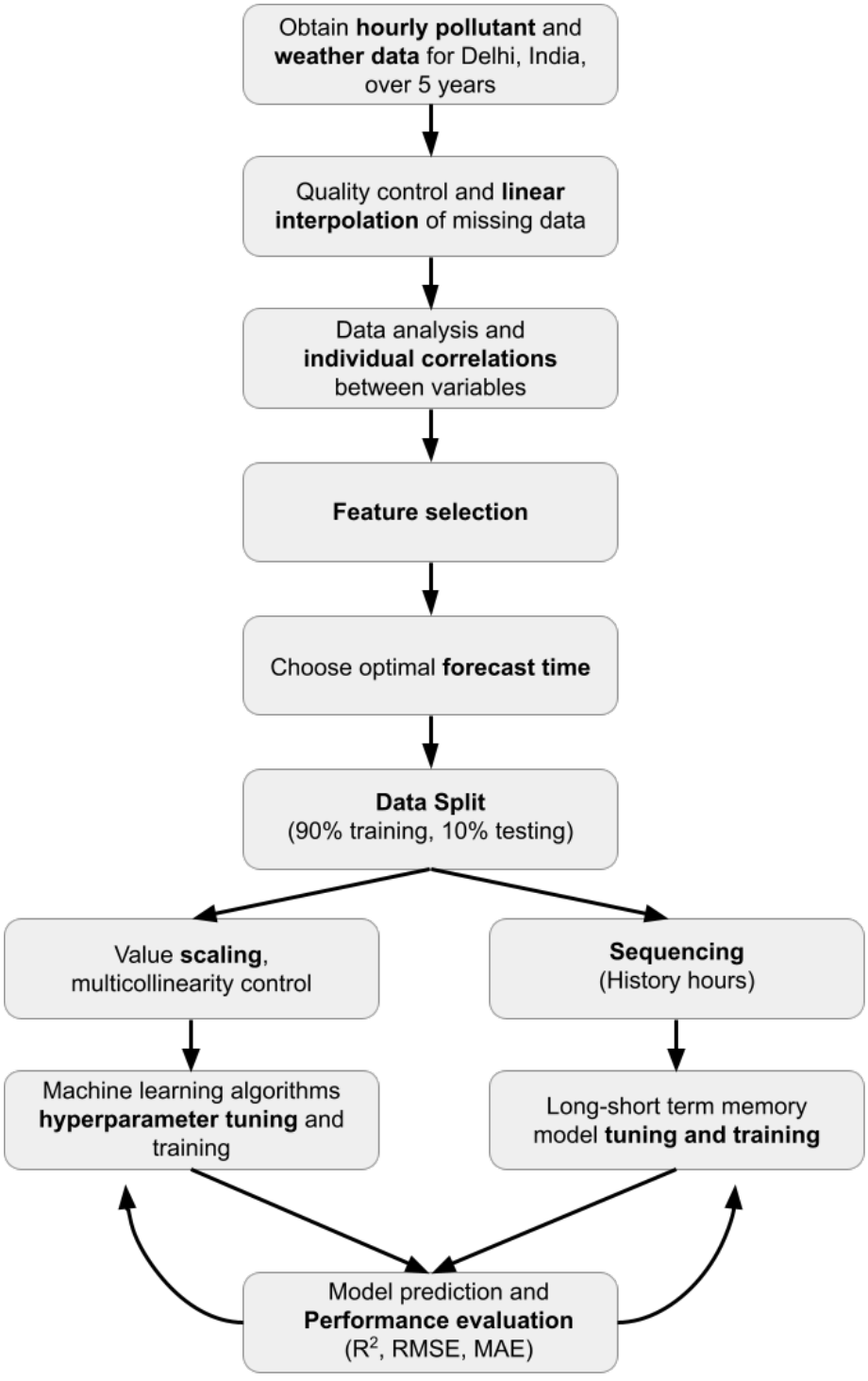

Figure 2. Flow chart of input data processing and analysis of the results.

Table 1. List of ozone correlations for the year 2015 of pollutant and weather variables used in this study. Asterisk indicates that the variable was excluded from ML models affected by multi-collinearity (linear regression, SVM, KNN).

Variable	Abbv.	Units	Corr. w/ Current O_3	Corr. w/ O_3 in 24 h	Used in ML?
Ozone (current)	O_3	$\mu g/m^3$	1.000	0.579	Yes
Ozone (in 24 h)	O_3P24	$\mu g/m^3$	0.579	1.000	N/A
Particulate matter (<10 microns)	PM10	$\mu g/m^3$	0.390	0.271	Yes
Particulate matter (<2.5 microns)	PM2.5	$\mu g/m^3$	0.170	0.125	Yes *
Nitrogen oxide	NO	$\mu g/m^3$	0.298	0.145	Yes *
Nitrogen dioxide	NO_2	$\mu g/m^3$	0.473	0.357	Yes
Any nitric x-oxide	NOx	ppb	0.331	0.202	Yes
Ammonia	NH_3	$\mu g/m^3$	−0.044	−0.088	Yes
Carbon monoxide	CO	$\mu g/m^3$	−0.320	−0.281	Yes
Sulfur dioxide	SO_2	$\mu g/m^3$	0.448	0.352	Yes
Benzene	Benzene	$\mu g/m^3$	0.093	0.046	No
Toluene	Toluene	$\mu g/m^3$	0.228	0.099	Yes
Xylene	Xylene	$\mu g/m^3$	−0.108	−0.175	No
Temperature	Temp	deg C	0.242	0.224	Yes
Cloud cover	Cloud	% cover	−0.123	−0.065	No
Humidity	Humid	% humidity	−0.253	−0.221	Yes
Sea level pressure	Press	Millibars	−0.102	−0.112	No

Table 2. Machine learning algorithm parameters used in Python scikit-learn and keras.

Machine Learning Method	Parameters
Linear Regression	n/a
KNN	n_neighbors = 4, metric = 'minkowski', p = 1
SVM	C = 10, gamma = 0.1, kernel = 'rbf'
Random Forest	max_depth = 50, random_state = 0, n_estimators = 250
Decision Tree	random_state = 0, max_depth = 6
AdaBoost	random_state = 0, learning_rate = 0.1, n_estimators = 100
XGBoost	learning_rate = 0.1, max_depth = 10, n_estimators = 300, random_state = 0, silent = True
BD-LSTM	batch_size = 72, epochs = 25, n_neurons = 256 (first layer), 128 (second layer), dropout = 0.2, n_hours = 8, n_steps = 3 optimizer = 'adam', loss = 'mse'

3. Results

3.1. Data Exploration and Correlations

The first step in creating a predictive model is to explore the data for quality, coherence, and agreement with expected correlations. In order to elucidate the relationships between a variety of meteorological and pollutant variables with ozone, we created scatter plots of these correlations (Figure 3). Correlations were computed with hourly data for the full year of 2015 and are shown in Table 1.

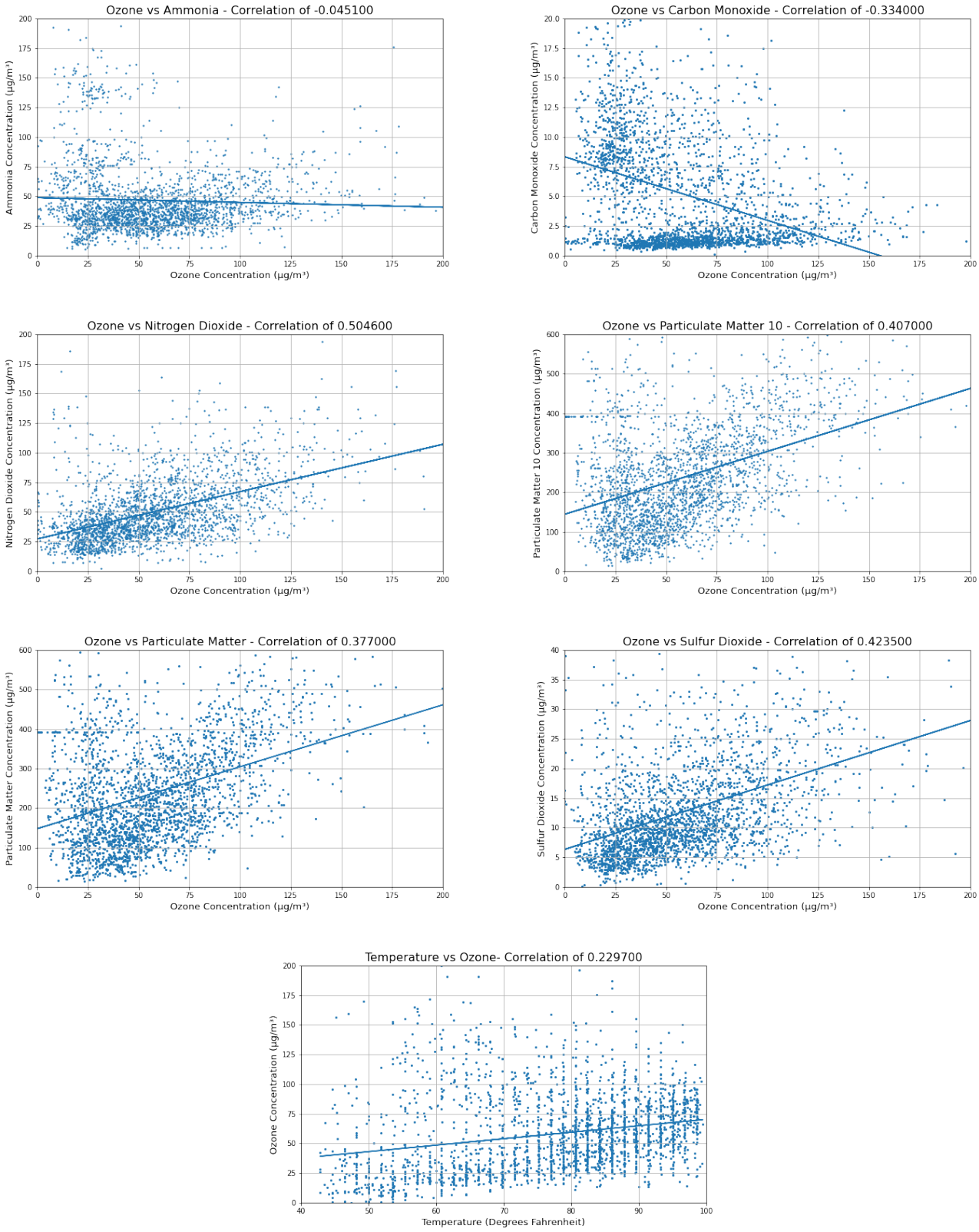

Figure 3. Individual correlations between variables and ozone, with best fit line and correlation in label. Each point is a single hour, and data for the year 2015 are shown.

The pollutants involved in the chemical formation of ozone are carbon monoxide and nitrogen dioxide (Figure 4). Nitrogen dioxide is positively correlated with ozone

at 0.5, while carbon monoxide is negatively correlated at −0.3. Some other noteworthy correlations occurred with particulate matter 10 (R = 0.41), sulfur dioxide (R = 0.42), and temperature (R = 0.23). Although their individual correlations with ozone are rather weak, they can be combined to train predictive models to achieve the best possible results. Particulate matter 10 and sulfur dioxide are not directly involved in the chemical equations of ozone production, so this appears to be an example of correlation rather than causation, where pollutants exist together because they are emitted together, while atmospheric conditions can remove the pollutants as a group. The weak positive correlation with temperature is likely due to the role of direct sunlight in ozone production and the correlation between sunlight and air temperature. Variables with low correlations and variables with a majority of data missing were discarded in this step. This includes benzene, precipitation, wind speed, and sea level pressure.

$$\bullet OH + CO \rightarrow \bullet HOCO$$
$$\bullet HOCO + O_2 \rightarrow HO_2\bullet + CO_2$$

$$HO_2^{\bullet} + NO \rightarrow {}^{\bullet}OH + NO_2$$
$$NO_2 + h\nu \rightarrow NO + O(^3P),\ \lambda<400nm$$
$$O(^3P) + O_2 \rightarrow O_3$$

$$CO + 2O_2 \rightarrow CO_2 + O_3$$

Figure 4. Ground ozone chemical equation [44]; highlighted chemicals were involved in this study.

In order to further explore the temporal relationship between ozone and the other pollutants involved in its formation, we created plots of the hourly concentrations against one another. The strongest correlating variable, nitrogen dioxide, showed a time lag of approximately 5 h between the primary and secondary pollutants (Figure 5). These preliminary results show that the precursor constituents will likely have predictive power, but it is not clear from mere visual inspection that a 24 h forecast will be possible.

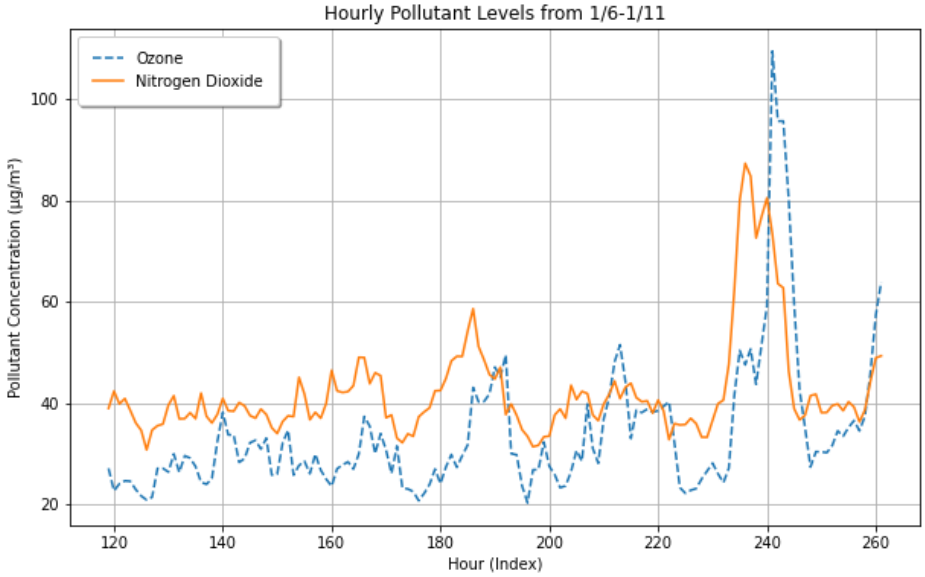

Figure 5. Hourly pollutant levels of ozone (blue) and nitrogen dioxide (orange) in a week in January 2015. Note that peaks in ozone lag nitrogen dioxide by approximately 5 h.

In order to predict future ozone levels, this project used regression analysis (as well as a time-series analysis, detailed in Section 3.4). Regression analysis is a type of predictive modeling in which a relationship is determined between one dependent variable (ozone)

and one or more independent variables (weather and pollutant data). There are different algorithms by which this process can be done, and the simplest one is multiple linear regression. In linear regression, as well as some other algorithms (SVM, KNN), multi-collinearity can be an issue. This arises when there are strong correlations between the independent variables, diminishing the statistical significance of each variable. Since linear regression analyzes the relationship between each independent variable and the dependent variable, multi-collinearity blurs which variable a change in the dependent variable can be attributed to. In order to avoid this, it is important to analyze the relationships between each independent variable before attempting any linear regressions. To do so, we used the Seaborn Python library to generate a heat map of the correlation between each variable in a data frame (Figure 6). This chart was also used to see which variables could potentially cause difficulties later on.

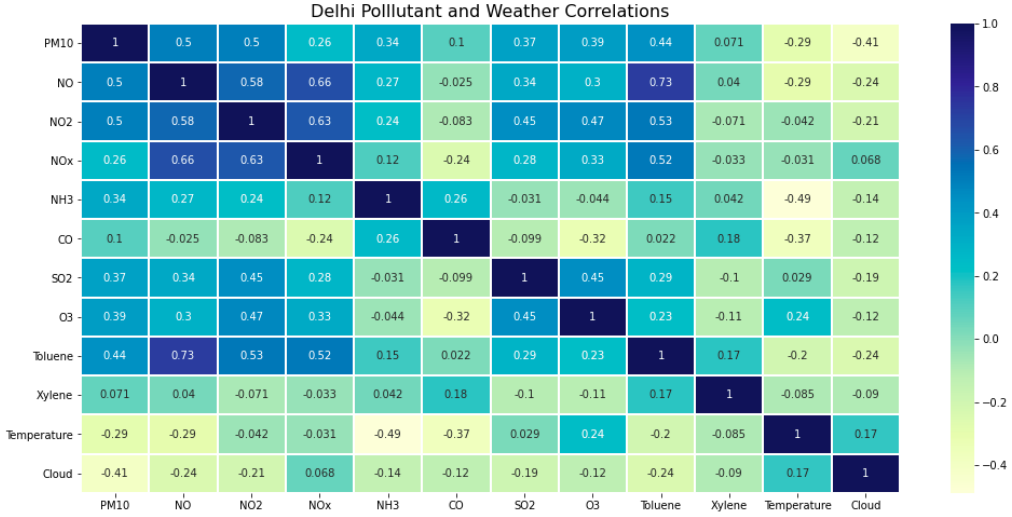

Figure 6. Correlation of each variable with all others for the year 2015. Some high values denote causal relationships, such as O_3 and NO_2, while many others are simply emitted in tandem.

Feature selection was then used to choose the most relevant variables for the machine learning algorithms in order to maximize the computational efficiency. The original data frame included 13 pollutant and 14 weather variables, which had been reduced to 12 pollutant and 4 weather variables after removing variables that were missing over 10% of data, including precipitation, dew point, and wind variables. Next, the variables were evaluated for their usefulness in explaining ozone concentrations using a significance level of 0.05 with ANOVA. Variables with p-values greater than 0.05 were removed. To reduce the impact of multi-collinearity, the variance inflation factor (or VIF) of each variable was determined. If the VIF was equal to or greater than 10, the variable was either removed or other less relevant variables that it correlated with were removed.

Another step in the initial data exploration was the creation of the average ozone concentration at each hour during the seasons to see potential patterns and cycles that ozone goes through over a day. As seen in Figure 7, the post-monsoon (fall) season generally had the highest ozone concentrations throughout the day, and winter had the lowest. During all seasons, there was also a notable increase during sunlight hours, followed by a decrease in the night, most likely due to the amount of solar radiation causing the formation reaction of the secondary pollutant ozone to take place. The size and timing of the peak ozone concentration varies substantially by season. This could be influenced by the timing and location of the emissions, the wind direction, and the timing and intensity of clouds and rainfall, all of which vary by season.

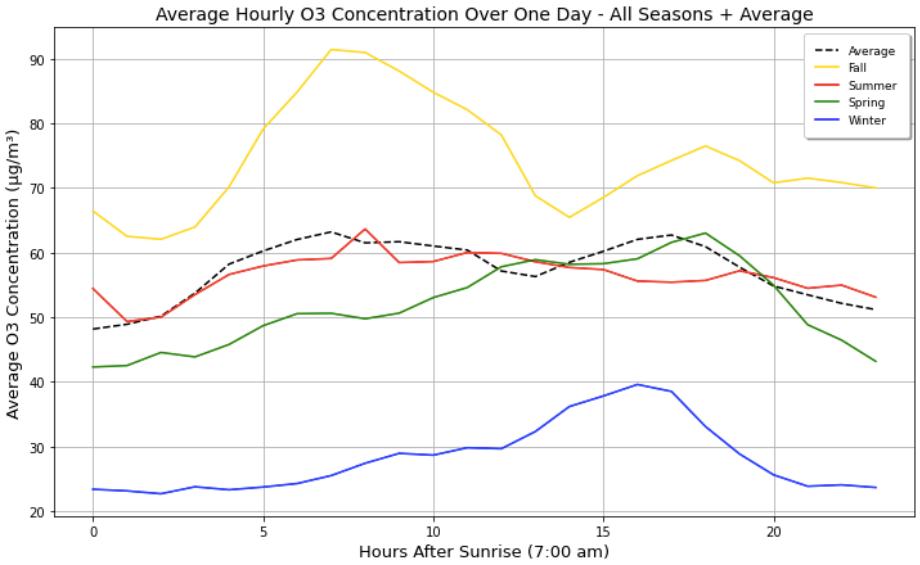

Figure 7. Average hourly O_3 concentration over one day for all seasons, showing elevated ozone in the fall and low values in winter. Ozone production requires solar radiation, which causes peaks during the day and a drop at night.

3.2. Ozone Forecasting Results

The first step in ozone forecasting was to determine how many hours in advance an accurate forecast could be made. Generally, a longer lead time results in a drop in accuracy. Figure 8 shows this relationship, where the R^2 value was determined for forecasts with 1 to 24 h of lead time, by season. As expected, there is a decrease in predictive ability the more hours in advance that the forecast predicts, leveling off slightly at around the ten-hour mark. However, it is notable that, although it dips slightly, the significance of the model increases by the 24 h mark to be almost equal to the 10 h mark. This increase in predictive ability may have been due to more consistent 24 h cycles in air pollution and weather factors. Taking this into consideration, it was decided that the models would aim to predict the tropospheric ozone concentration in 24 h, since its predictive ability is similar to that in 10 h, yet is over double the lead time.

Figure 9 displays the strength of the linear correlations between the input variables with the ozone concentration 24 h later. The strongest relationships with future ozone concentration were with current ozone, nitrogen dioxide, sulfur dioxide, particulate matter 10, and carbon monoxide pollutant concentrations. Some of these correlations may result from different pollutants tending to rise and fall together or indicate some causal relationship such as an increase in precursor chemicals driving an increase in ozone concentration.

An important part of some machine learning models involves scaling the data. This is done to place all independent variables into a fixed range so that different variables with varying units can be handled appropriately. There are different methods of scaling, as mentioned in the Methods Section, and RobustScaler was found to have the best results, likely due to its use of statistics that are robust to outliers when scaling the data. Outliers often occur in air pollution data due to sudden changes, such as precipitation temporarily removing air pollutants through wet deposition, or the appearance of nearby sources of smoke.

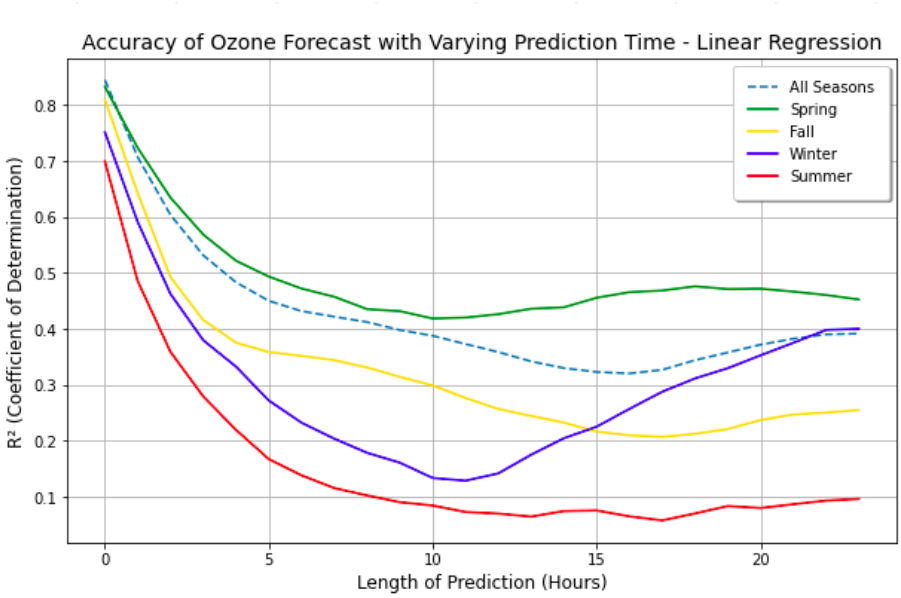

Figure 8. Accuracy of ozone forecast with varying lead time. Prediction skill drops from 0 to 10 h out, but then recovers out to 24 h. This makes it possible to forecast ozone 24 h in advance.

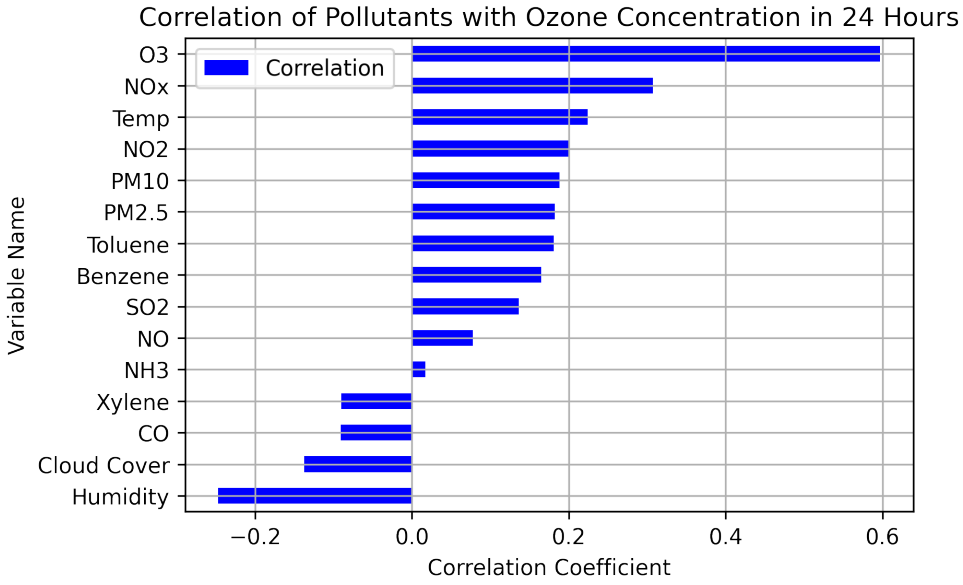

Figure 9. Correlations between variables and ozone 24 h later, ordered by correlation. The two chemical precursors of ozone, NO_2 and CO, have some of the highest correlations (positive and negative), along with the current ozone concentration.

3.3. Tuning, Training, and Testing

The final step was to tune, train, and test the machine learning models. These models included linear regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forests, Decision Tree, AdaBoost, and XGBoost. Bidirectional Long Short-Term Memory (LSTM) has a different training process, as explained in Section 3.4. Excluding linear regression, these models have a variety of hyperparameters that can be adjusted to improve model performance. In order to determine the best combination of parameters, the Python Sklearn library's function called GridSearchCV was used for each machine learning model. This function evaluates each combination of parameters using the cross-validation method (detailed below). Some of these parameters were also adjusted and tested manually to verify that they returned the highest accuracy scores. Each of the hypertuned

test models was trained on one year of data (2015). An algorithm was created to try different combinations of parameters and return the best ones, also being adjusted and tested manually (see Table 2). Following standard metrics for air pollution evaluation [13,14,18,25,45], the regression models were evaluated with R-squared (R^2), adjusted R-squared, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). R-squared, also known as the coefficient of determination, is a measure of how closely data fit the fitted regression line; it is the proportion of variance in a dependent variable explained by the independent variables in a regression model. The adjusted R-squared is similar, but it takes into account the variables that do not help the model by lowering the score. If the adjusted R-squared value is similar to the R-squared value, it means that the input variables are useful and contribute to the model. The Mean Absolute Error (MAE) represents the absolute value of the average difference between the expected and predicted values. The Root Mean Square Error is a measure of the standard deviation of residuals and is more affected by outliers than the MAE. Generally, lower RMSE and MAE scores and higher R-squared scores imply better models.

Regression analysis with machine learning was done by randomly splitting the data set into 90% for training and 10% for testing. The testing portion is the data that the model has not yet seen, and the model attempts to predict the dependent variable for this portion. The predicted and actual results are then compared to assess the accuracy of this model, conveyed through the aforementioned metrics. In order to reliably evaluate the accuracy of each model, we trained and tested each model 10 times, repeating the 90–10 split 10 times to test all of the data at a certain point. This was done in order to eliminate the possibility that the random 10% of data simply happened to match the predicted values more than in another round, giving this model an arbitrarily higher accuracy score for a particular trial. This process of repeated training and testing is called cross-validation and was done to fairly evaluate the performance of each model. The average performance of each model is listed in Table 3, from highest to lowest R-squared values, and shown graphically in Figures 10 and 11, and time taken in Figure 12. Table 3 also provides the statistics from cross-validation, and the correlation coefficient was calculated from the linear relationship between actual versus predicted ozone values. Taylor diagrams [46] were used to compare model skill, where each model's standard deviation was plotted against its correlation with observations (Figure 13). All models presented here under-predict the standard deviation. This is a common problem for forecasting models, because extremes are harder to reproduce than events near the mean. This can be seen directly in Figure 14, where KNN, XGBoost, and Random Forests, with higher standard deviations, span nearly the same range of ozone predictions as the observations, whereas AdaBoost has the lowest standard deviation and the predicted values fall within a narrow range from 30 to 110.

Table 3. Average performance of the 24 h ozone prediction from each machine learning model, tested for every hour for the full year of 2015.

Model Name	Correlation Coefficient	R^2	R^2 Adjusted	RMSE µg/m^3	MAE µg/m^3	Time s
XGBoost	0.784	0.6161	0.6156	20.78	13.67	315.6
Random Forest	0.782	0.6041	0.6035	21.11	13.99	740.1
KNN	0.739	0.5126	0.5120	23.41	15.55	1.9
SVM	0.695	0.4633	0.4626	33.96	25.76	172.7
Decision Tree	0.656	0.4032	0.4023	25.91	17.54	1.8
Linear Regression	0.626	0.3937	0.3929	26.12	18.06	0.3
AdaBoost	0.623	0.3523	0.3514	26.99	20.70	91.5
LSTM	0.393	0.1550	NA	44.5	33.70	429.8

Computing times varied widely across the seven methods (Figure 12). XGBoost and Random Forest obtained nearly identical fit and error scores, but XGBoost was 2.3 times

faster. Surprisingly, KNN obtained a slightly lower fit score than the top two methods, but was 170 times faster than XG boost.

In order to determine the performance of each model on an actual prediction sample, the models were run on a 90–10% split once, and the results were plotted and explored more deeply (Figure 14). Each plot displays the relationship between actual vs. predicted values—the closer to the line of the perfect fit, the better. As mentioned earlier, some models simply happened to perform better on different sections, and, in this case, KNN gave the best results. When tested, the KNN model could predict O_3 concentrations 24 h in advance, where 68% of the predictions had a percentage error of less than 25%.

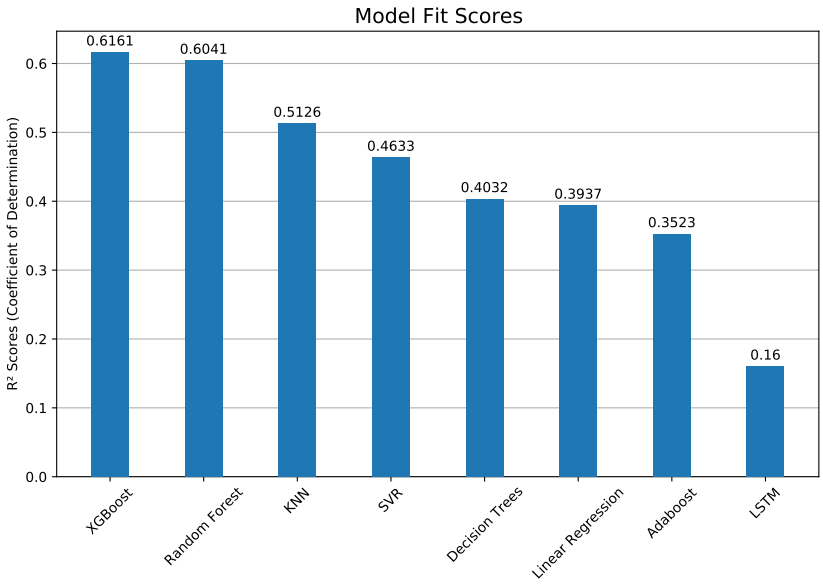

Figure 10. Model fit (R^2) scores, showing that XGBoost and Random Forest produce the best forecasts. The models were tested for hourly predictions over the full year of 2015.

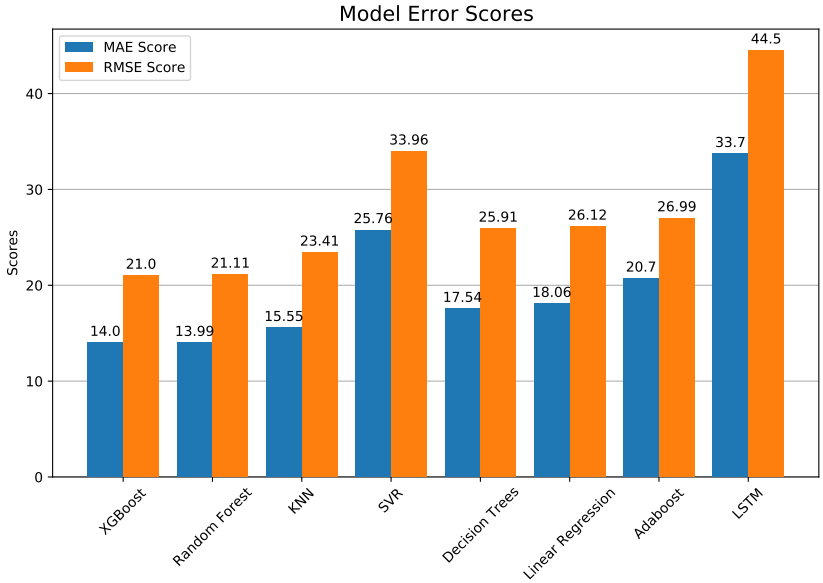

Figure 11. Same as Figure 10 but for annual model error scores of MAE and RMSE. Both are in units of $\mu g/m^3$.

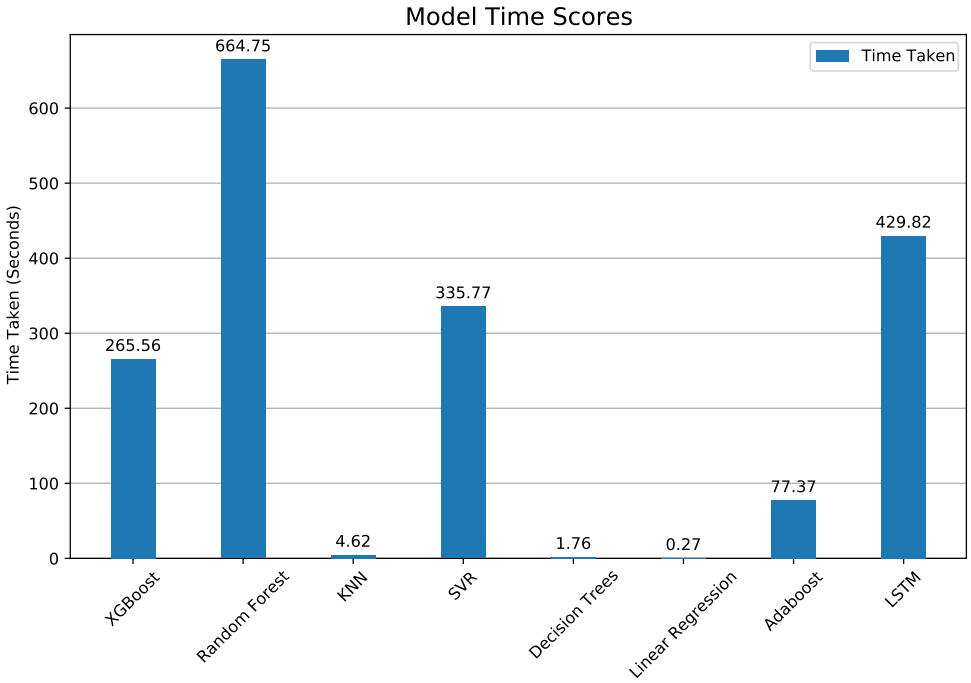

Figure 12. Time taken for each model for one year of hourly predictions, including training.

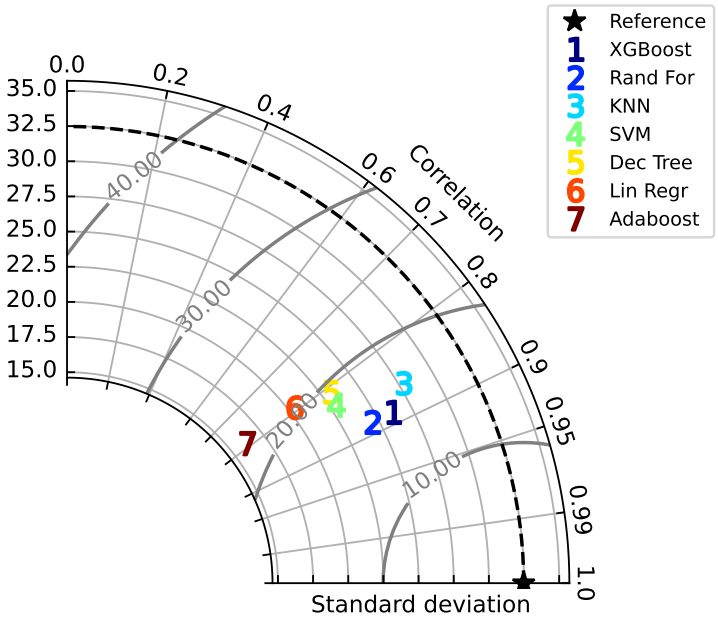

Figure 13. Taylor plot of annual data with seven models, comparing standard deviation (in micrograms/m^3 ozone) versus correlation.

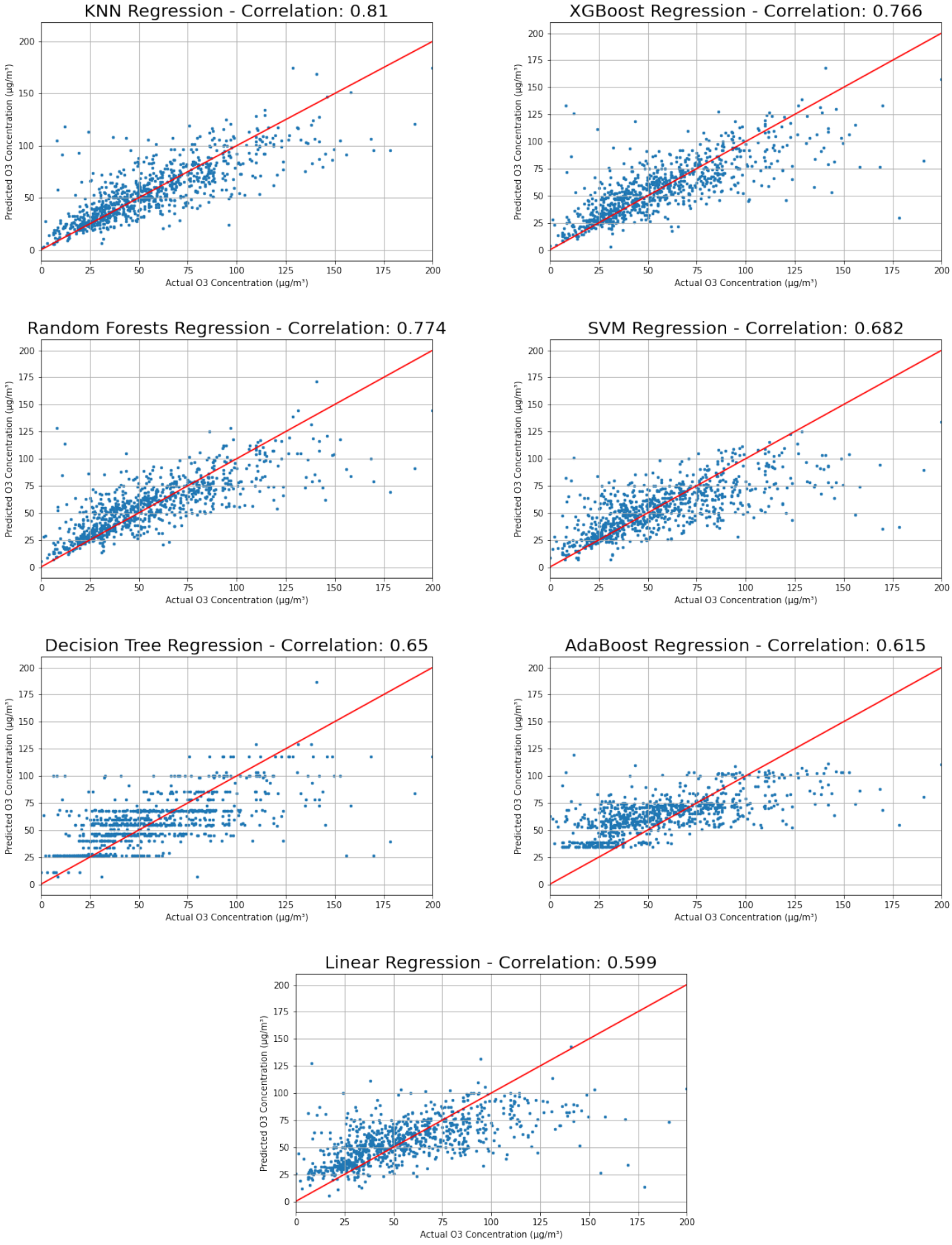

Figure 14. Actual versus predicted results of each model, with best-fit line and correlations.

3.4. Long Short-Term Memory (LSTM)

The set-up for LSTM differs substantially from the seven models presented above. Three types of LSTM models were tested: a simple LSTM network model, a bidirectional LSTM (BD-LSTM), and an encoder–decoder LSTM (ED-LSTM). LSTM models are trained to predict a target variable based on historical sequence data for a given number of previous hours (the *n_hours* variable) [11]. They are a type of recurrent neural network (RNN), but have the advantage of learning more long-term patterns by solving the issue of vanishing gradients, a difficulty that arises when the error signals that are used to train the network decrease exponentially. LSTM was created to overcome this challenge, making use of memory cells with forget gates, which determine whether previous step information is needed or should be forgotten. Because of this, LSTM models are able to store longer time steps in memory by regulating how much previous data is used for each time step.

While simple LSTMs are unidirectional and use only data from previous sequences to predict future time steps, bidirectional LSTMs use data from the past in the forward direction (as with LSTM) as well as from the future in the backward direction. Bidirectional LSTMs put together two independent RNNs of backward and forward information in order to update weights in both directions, giving the network more thorough knowledge of the relationships between previous/current and future values [47,48].

Another type of RNN is the encoder–decoder LSTM network. ED-LSTM was created specifically to address sequence-to-sequence predictions, in which a model converts an input sequence to an output sequence with differing numbers of items. ED-LSTM consists of two LSTM models: an encoder, which processes an input sequence to an encoded state, and a decoder, which produces an output sequence from the encoded state [49,50].

LSTM models have been used in past air pollution studies, especially for producing real-time predictions, which are also called one-step predictions. For instance, Xayasouk et al. [28] used LSTM with a deep autoencoder model to predict current particulate matter levels based on previous relationships with weather variables such as humidity, wind speed, wind direction, temperature, and other conditions. Tiwari et al. [51] studied multi-step-ahead LSTMs and found that multivariate bidirectional LSTM models had the best performance.

The experimental design for LSTM is as follows:

1. Test three types of LSTM models (LSTM, BD-LSTM, ED-LSTM) and compare their RMSE scores to determine the best option;

2. Improve the best model by preventing overfitting, hypertuning, using different training periods (seasons vs. annual), comparing time step lengths (e.g. 24 steps of 1 h versus 8 steps of 3 h);

3. Choose the best parameters for annual and seasonal predictions, run each model for ten iterations, and compute average metrics.

In order to compare unidirectional LSTM, BD-LSTM, and ED-LSTM, we evaluated the RMSE values of each model for predictions from 1 to 24 h in advance, all trained on the past 24 h of history. Due to the stochastic nature of the LSTM code, each of the models was run for 10 iterations, and the average RMSEs, MAEs, and R2 values were calculated. It was found that for longer future prediction times, the BD-LSTM outperformed both LSTM and ED-LSTM.

Tuning experiments were conducted to adjust BD-LSTM parameters to optimize performance. The model was first trained to predict 24 h (24 steps) in advance based on the past 24 h using data from all of 2015, and this configuration was found to have low performance. This was likely due to seasonal variations in the input variables, so the model was trained on relationships that were not consistent throughout the year. To address this, the model was then trained on 4 years of seasonal data (2015–2018) and tested on each season of the last year (2019). This drastically improved upon annual results, but still gave ozone predictions that were relatively close to the mean of the data. To reduce overfitting, dropout layers were added and the number of variables was reduced to ensure that the

model trained on the most important variables only. This was done using the feature importance of variables in XGBoost, the highest-performing regression model.

To reduce the complexity of the input data, the model was tested on each season again with data from once every three hours (as opposed to every hour). This reduced the number of time steps being predicted in advance from 24 1 h steps to 8 3 h steps, which improved the results for all seasonal models. The average metrics from 10 iterations are shown in the BD-LSTM rows of Tables 3–7 and Figure 10. The BD-LSTM's final parameters can be found in Table 2. Seasonal predictions from BD-LSTM were more accurate than the other seven models, showing that the sequential time dependence of the input variables is an important factor in improving ozone forecasts.

There are numerous extensions of LSTM and machine learning algorithms that could be added to this evaluation in the future. The ant lion optimizer model (LSTM-ALO) [52,53] optimizes the number of hidden layer neurons and the learning rate of the LSTM. The Extreme Learning Machine with Gray Wolf Optimization (ELM-GWO) [54,55] is a meta-heuristic algorithm that imitates the hunting behavior of wolves, and it uses fewer adjustment parameters and a powerful global search capability. The adaptive neuro-fuzzy inference system (ANFIS) [56] creates an input–output mapping based on human knowledge using fuzzy if-then rules and specified input–output data pairs. These methods have been used for a number of geophysical applications, particularly watershed streamflow prediction [57–59], landslide susceptibility [60,61], and agricultural metrics such as evapotranspiration [62,63].

3.5. Seasonal Model Evaluation

All of the results so far were obtained by training the models with data over one year (2015) in Delhi, India. However, as seen in Figure 8, the predictive ability of the model differed considerably across seasons, performing the lowest during the summer. This makes sense in the context of India's unique meteorological seasonal cycles, particularly including a four-month-long monsoon season. Because of this, we decided to train and test the models again, but with each of the four seasons separately across five years of data. The seasonal training and prediction used the same input variables and tuning parameters as the annual training and prediction. The year-long model contained 8760 hourly entries for the full year of 2015. For the seasonal split, after all 5 years of data were cleaned, the data were separated into each respective season. Seasons ranged from two to four months, but they each contained between 7000 to 12,000 time entries, which was comparable to the 8760 entries in the original annual data. The new data were split as follows:

- Summer: March, April and May;
- Monsoon: June, July, August, September;
- Post-Monsoon (Fall): October, November, December;
- Winter: January, February.

The results are shown in Tables 4–7, examples of XGBoost correlations in Figure 15, and a summary of all methods and seasons in Figures 16 and 17. Winter is most predictable for all methods, with R^2 scores typically 0.1 higher than the other seasons, which vary in predictability by model. Generally, XGBoost and Random Forest are the highest for the annual prediction and all seasons.

Table 4. Summer cross-validation results (March, April, and May).

Model Name	R	R^2	R^2 Adjusted	RMSE μg/m^3	MAE μg/m^3	Time s
XGBoost	0.7814	0.6106	0.6102	21.65	14.17	259.6
Random Forest	0.7803	0.6088	0.6084	21.70	14.33	795.1
KNN	0.7469	0.5579	0.5575	22.89	14.82	2.8
SVM	0.7186	0.5164	0.5159	35.20	26.50	262.8
Decision Tree	0.6923	0.4793	0.4787	25.08	16.74	2.0
AdaBoost	0.6626	0.4390	0.4384	25.23	19.27	87.2
Linear Regression	0.6856	0.4700	0.4694	25.79	17.30	0.4
BD-LSTM	0.8805	0.7750	NA	14.79	11.62	618.0

Table 5. Monsoon cross-validation results (June, July, August, September).

Model Name	R	R^2	R^2 Adjusted	RMSE μg/m^3	MAE μg/m^3	Time s
XGBoost	0.7967	0.6347	0.6344	16.02	9.72	306.1
Random Forest	0.7917	0.6268	0.6265	16.03	9.70	1113.1
KNN	0.7469	0.5722	0.5719	17.00	10.14	5.1
SVM	0.7369	0.543	0.5426	25.12	18.01	775.5
Decision Tree	0.7121	0.5071	0.5067	18.18	11.26	2.5
Linear Regression	0.6733	0.4534	0.4530	18.93	12.10	0.3
AdaBoost	0.6602	0.4359	0.4354	19.76	13.41	126.1
BD-LSTM	0.7528	0.5667	NA	11.78	8.27	613.2

Table 6. Post-monsoon (fall) cross-validation results (October, November, December).

Model Name	R	R^2	R^2 Adjusted	RMSE μg/m^3	MAE μg/m^3	Time s
XGBoost	0.798	0.6374	0.6368	25.23	14.65	200.8
Random Forest	0.797	0.6350	0.6344	25.08	14.91	604.3
KNN	0.761	0.5783	0.5777	26.63	15.70	1.9
SVM	0.677	0.4583	0.4575	43.07	30.84	158.2
Decision Tree	0.681	0.4642	0.4633	29.90	17.70	1.9
Linear Regression	0.626	0.3925	0.3916	32.80	20.60	0.6
AdaBoost	0.704	0.4951	0.4942	31.51	20.77	57.9
BD-LSTM	0.8187	0.6703	NA	13.62	10.26	126.9

Table 7. Winter cross-validation results (January, February).

Model Name	R	R^2	R^2 Adjusted	RMSE μg/m^3	MAE μg/m^3	Time s
XGBoost	0.8686	0.7545	0.7542	19.19	10.89	288.6
Random Forest	0.8645	0.7474	0.7471	19.37	11.08	852.7
KNN	0.8389	0.7038	0.7035	21.68	12.35	2.1
SVM	0.7980	0.6368	0.6364	39.17	25.25	247.7
Decision Tree	0.7892	0.6229	0.6224	23.34	13.58	1.9
Linear Regression	0.7622	0.5809	0.5804	24.75	14.82	0.4
AdaBoost	0.7407	0.5487	0.5482	24.13	15.96	288.9
BD-LSTM	0.7235	0.5235	NA	11.98	9.72	497.3

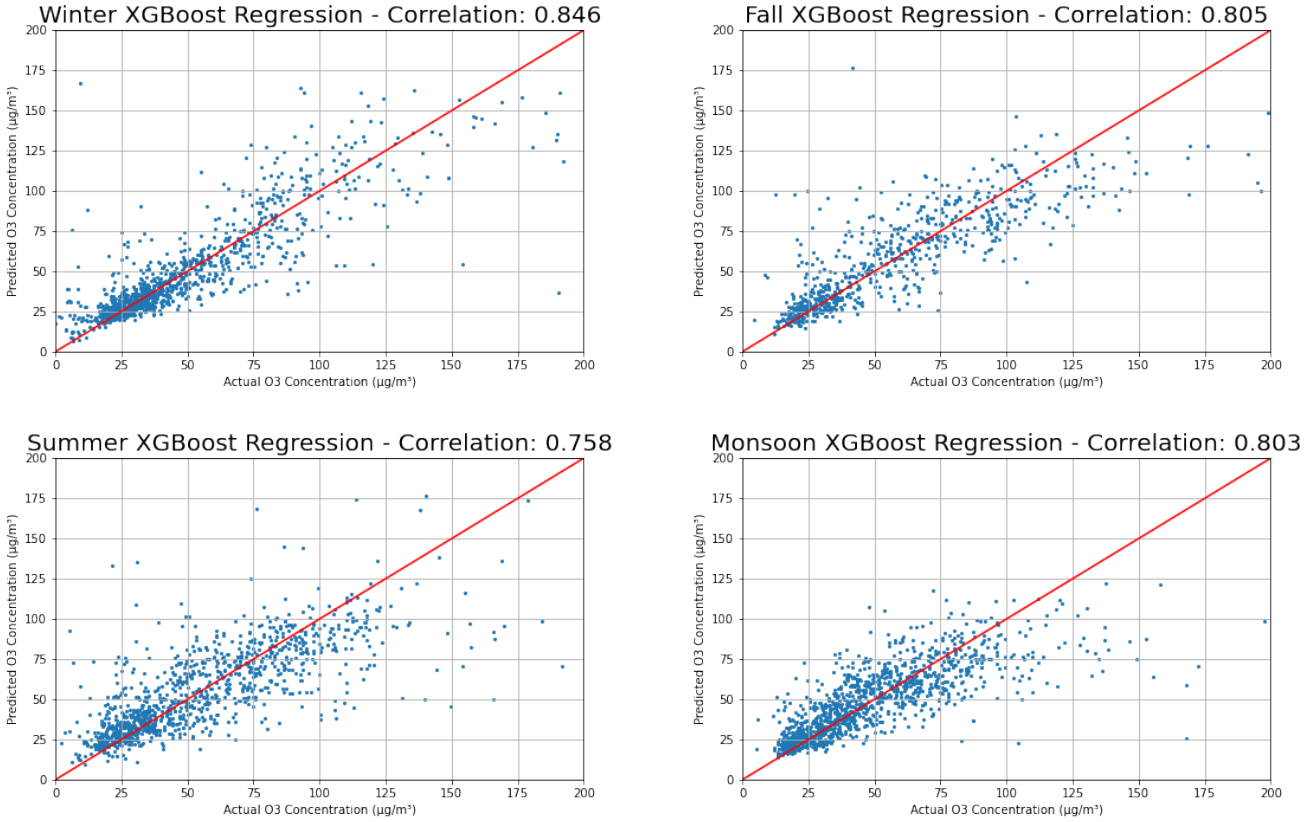

Figure 15. Actual versus predicted results of XGBoost for each season.

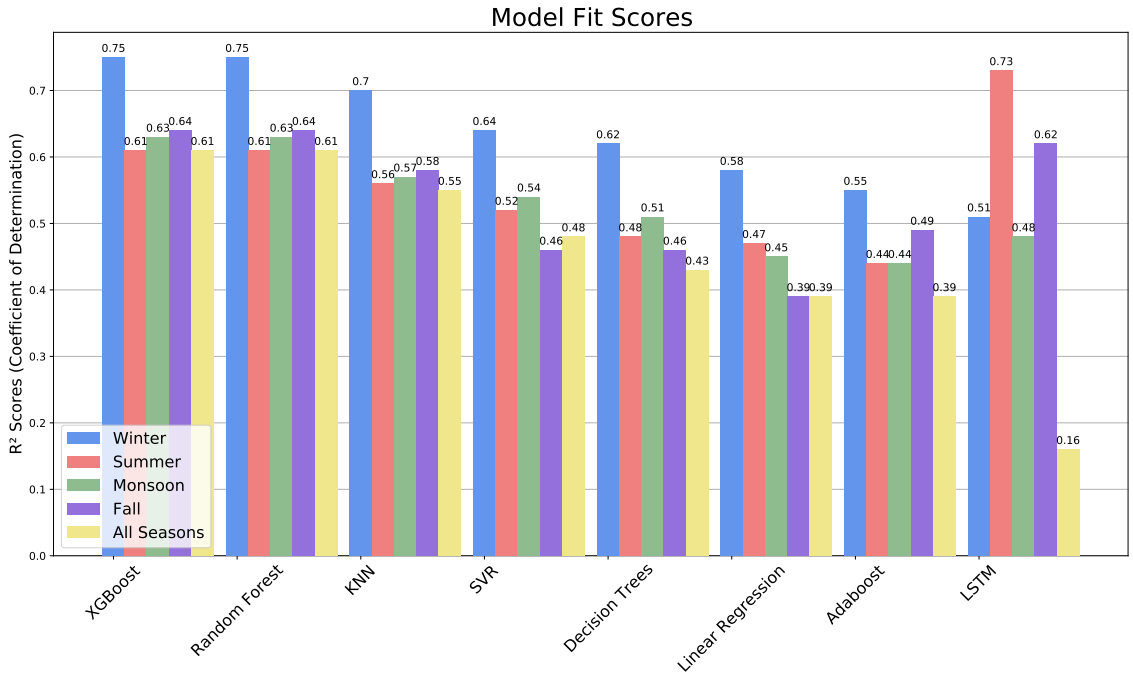

Figure 16. Model fit scores for each season and model.

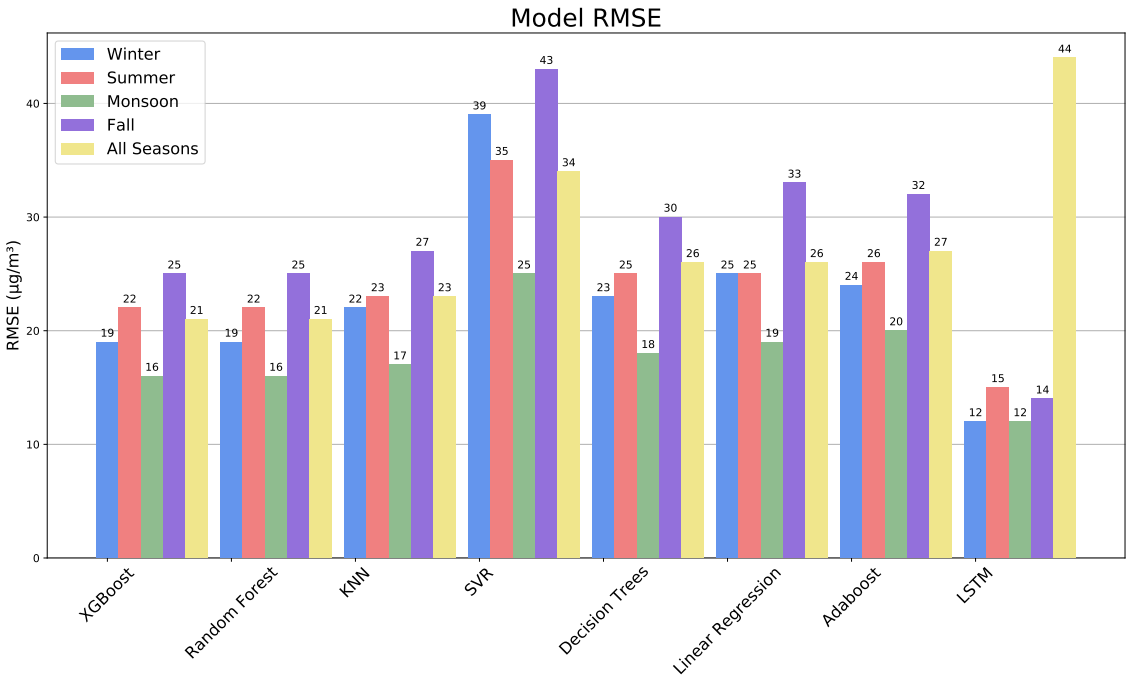

Figure 17. Model RMSE scores for each season and model.

In order to judge the applicability of this method for an operational 24 h forecast, we evaluated how accurately the model could predict air quality index classifications (Figure 18). In practice, the public would modify their behavior based on these warnings, more so than numerical predictions of ozone concentrations. The best-performing model, XGBoost, performed quite well, as it was able to predict the exact air quality index 92% of the time and was able to predict within one index 98% of the time (Table 8). The seasonal breakdown shows the highest predictive capability in winter, predicting the correct index 97.3% of the time, and lowest for summer, at 88.9%.

O_3 Concentration (µg/m^3)	Air Quality
0-104	Good
105-134	Moderate
134-164	Unhealthy for Sensitive Populations
165-204	Unhealthy
205-380	Very Unhealthy

Figure 18. Air quality index categories for ozone [64].

Table 8. Percent time each index is correctly predicted in exactly the right air quality index, and within one index, using the XGBoost method.

Season	Correct Index	Within 1 Index
Annual	92.0%	98.0%
Winter	97.3%	99.6%
Monsoon	90.3%	98.5%
Post-Monsoon	92.8%	98.8%
Summer	88.9%	97.2%

4. Discussion

In order to assess the quality of our ozone forecast, we compared our results to a number of past studies. In the current study, XGBoost and Random Forest scored the highest, both having an R^2 value of 0.61 for annual, followed in order by KNN, SVM, Decision Trees, linear regression, AdaBoost, and BD-LSTM (Figure 10). XGBoost and Random Forest scored the highest for winter (R^2 = 0.75), while BD-LSTM was best in the summer and post-monsoon seasons (R^2 = 0.77 and 0.82). In previous research, the highest R^2 values for ozone have similarly been achieved with non-linear machine learning methods, including R^2 = 0.49 with Neural Power Networking [40]; R^2 = 0.72 with Random Forest [41]; R^2 = 0.84 with Boosted Decision Tree Regression [14]; and R^2 = 0.66 with a Multi-Layer Perceptron network [31]. Of course, these studies are not directly comparable because they were carried out at different locations with different data sets. However, the variety in their best-performing ML methods indicates that there is no clear "winner" and that many strategies need to be tested with each new application. The most similar past study to this one is by Srivastava et al. [65], which compared some of the same methods and found that SVM and Artificial Neural Networks were best suited for predicting the air quality in Delhi, over Random Forest (they did not test XGBoost or KNN).

This study is one of the first to design machine learning models of ozone with specific seasonal training data sets. Each set of seasonal models generally had higher predictive capability than the year-long models, and, for many cases, the seasonal predictions were much better. Based on these results, we recommend that this method of seasonal training be tested in other locations as well. Seasonal training may be particularly helpful in India, with its strong monsoons, but should be beneficial in temperate climates as well.

There is a noticeably higher predictive capability of the winter seasonal model, which may be due to the fact that January and February have some of the lowest percentages of rainfall during the year, only receiving approximately 1.5% of the annual rainfall each month [66]. The two months also experience some of the lowest temperatures of the year, perhaps limiting ozone formation and making it more dependent on precursor chemicals and conditions than in the summer, when temperatures are always high, leading to easier but less predictable ozone formation. Kumar and Goyal [67] created a forecast of air quality index (AQI) for Delhi using Principle Component Analysis. AQI includes ozone along with a number of other pollutant indicators. They found a wide range between seasons, with a normalized mean square error (NMSE) of 0.0058, 0.0082, 0.0241, and 0.0418 for winter, summer, post-monsoon, and monsoon, respectively. Their low error in winter agrees with this study, but the errors for the remaining seasons span a wider range than here (Figure 10).

This project faced several challenges. The low initial R^2 values obtained by the year-long models during the summer may have been due to India's rainy monsoon season, which temporarily clears the sky of air pollution but reduces predictability. However, when trained separately, the R^2 values for the same period (June, July, August) during monsoon season improved considerably, increasing from a maximum R^2 of 0.35 to 0.63. Additionally, there may be other influential variables that are not available in the data, so they are not included in the model. These include more seasonal weather patterns or factors affecting the emission of primary pollutants. This points to a direction to improve operational forecasts—creating ozone prediction models that include meteorological forecasting data. Indeed, Shukla et al. [41] found this improved pollution forecasts from poor to satisfactory for the prediction of NO, NO_2, and ozone in Delhi.

The direct measurement of solar radiation is an important predictive variable for ozone because it is required for its chemical formation (Figure 4). Solar radiation was not available from the weather stations used for this study, so we relied on proxies such as temperature. Time of day was also tested in the training data, but did not contribute sufficient predictive power to include. Other studies have found that ozone prediction has the highest dependence on relative humidity [41,68], followed by solar radiation, NO_2, NO, and benzene. In contrast, this study found current ozone concentration to be the best predictor 24 h in the future, followed by NO_2, SO_2, PM10, CO, and then humidity.

5. Conclusions

Tropospheric ozone has become an increasing cause of premature deaths for the past several decades in developing countries such as India. In this work, we analyzed eight machine learning models to predict ozone concentrations 24 h in advance. Beginning with hourly data for 12 pollutant and 5 meteorological variables, feature selection and hypertuning were conducted to optimize the performance of each model. As noted with the similar adjusted R^2 values and VIF values below 10, the models were not overfitted or too complex. The models were validated using cross-validation, with the XGBoost (R^2 = 0.614), Random Forest (R^2 = 0.611), and K-Nearest Neighbors Regression (R^2 = 0.546) models performing best with year-long training. The XGBoost model was able to predict the exact air quality index 92% of the time, and within one index 98% of the time, indicating great real-world applicability with AQI reporting.

Training the models by separate seasons rather than over an entire year improved their performance. The most reliable season was winter, where the highest-performing models were XGBoost (R^2 = 0.755), Random Forest (R^2 = 0.747), and K-Nearest Neighbors Regression (R^2 = 0.704). Thus, a recommendation from this work is that models should be trained with seasonal data rather than annual data. The Long Short-Term Memory model was the worst predictor with annual training but the best model for summer and post-monsoon with seasonal training.

One of the challenges of pollution prediction is that data from particular observation stations are often noisy or incomplete. This limits the accuracy of predictions based on a single station. For this study, Delhi was chosen because of its high-quality measurements, but this is a major limitation for many other cities. Incorporating a network of pollutant and weather stations, as well as satellite observations, would add resiliency to the prediction system. Additional preprocessing could also compensate for missing data with decomposition methods [43].

An important application of machine learning to ozone is to determine whether ozone production in a particular city is limited by NO_2 or VOCs. This would require a combination of tropospheric chemistry modeling with ground and satellite observations [69], potentially providing useful guidance for government policies to reduce ozone concentrations. The large swings in emissions during the COVID-19 pandemic provide an important test case for such studies [70].

The results of this study show that machine learning has great promise for ozone prediction. Recent advances in machine learning algorithms provide substantial improvements in forecasting skill. XGBoost and Random Forest demonstrated the best forecasting skill overall. It is worth noting that KNN was a close runner-up, but was hundreds of times faster, the impacts of which could be considered in future studies when compared to a slight increase in R^2 values. Future work would benefit from more detailed spatial analysis of emissions sources relative to the pollutant monitoring stations and the integration of weather forecasting into the training data.

Author Contributions: Conceptualization, E.K.J. and M.R.P.; methodology, E.K.J. and M.R.P.; software, E.K.J.; validation, E.K.J.; data curation, E.K.J.; writing—original draft preparation, E.K.J.; writing—review and editing, E.K.J. and M.R.P.; visualization, E.K.J.; supervision, M.R.P.; funding acquisition, M.R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the U.S. Department of Energy (DOE), Office of Science, Office of Biological and Environmental Research.

Data Availability Statement: Only publicly available data sets were analyzed in this study. Pollution monitoring data for Delhi were obtained from the Central Pollution Control Board of India [35], and Delhi weather data from Visual Crossing Weather API [36]. The processing code is available at https://github.com/kai-juarez/ozone-forecast (accessed on 23 December 2021). Specific resulting data sets presented in this study are available on request from the corresponding author.

Acknowledgments: Thanks to Ayush Agrawal for reviewing the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Martınez-Espana, R.; Bueno-Crespo, A.; Timon, I.; Soto, J.; Munoz, A.; Cecilia, J.M. Air-Pollution Prediction in Smart Cities through Machine Learning Methods: A Case of Study in Murcia, Spain. *J. Univ. Comput. Sci.* **2018**, *24*, 261–276.
2. Chen, T.M.; Kuschner, W.G.; Gokhale, J.; Shofer, S. Outdoor Air Pollution: Ozone Health Effects. *Am. J. Med. Sci.* **2007**, *333*, 244–248. [CrossRef]
3. Käffer, M.I.; Domingos, M.; Lieske, I.; Vargas, V.M. Predicting ozone levels from climatic parameters and leaf traits of Bel-W3 tobacco variety. *Environ. Pollut.* **2019**, *248*, 471–477. [CrossRef] [PubMed]
4. Golaz, J.C.; Caldwell, P.M.; Van Roekel, L.P.; Petersen, M.R.; Tang, Q.; Wolfe, J.D.; Abeshu, G.; Anantharaj, V.; Asay-Davis, X.S.; Bader, D.C.; et al. The DOE E3SM Coupled Model Version 1: Overview and Evaluation at Standard Resolution. *J. Adv. Model. Earth Syst.* **2019**, *11*, 2089–2129. [CrossRef]
5. Petersen, M.R.; Asay-Davis, X.S.; Berres, A.S.; Chen, Q.; Feige, N.; Hoffman, M.J.; Jacobsen, D.W.; Jones, P.W.; Maltrud, M.E.; Price, S.F.; et al. An Evaluation of the Ocean and Sea Ice Climate of E3SM Using MPAS and Interannual CORE-II Forcing. *J. Adv. Model. Earth Syst.* **2019**, *11*, 1438–1458. [CrossRef]
6. Petersen, M.; Livescu, D. Forcing for statistically stationary compressible isotropic turbulence. *Phys. Fluids* **2010**, *22*, 116101. [CrossRef]
7. Rohl, C.A.; Strauss, C.E.M.; Misura, K.M.S.; Baker, D. Protein Structure Prediction Using Rosetta. In *Numerical Computer Methods, Part D*; Methods in Enzymology; Academic Press: Cambridge, MA, USA, 2004; Volume 383, pp. 66–93. doi: 10.1016/S0076-6879(04)83004-0. [CrossRef]
8. Sonnewald, M.; Lguensat, R.; Jones, D.C.; Dueben, P.D.; Brajard, J.; Balaji, V. Bridging observations, theory and numerical simulation of the ocean using machine learning. *Env. Res. Let.* **2021**, *16*, 073008.doi: 10.1088/1748-9326/ac0eb0. [CrossRef]
9. Bolton, T.; Zanna, L. Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *J. Adv. Model. Earth Syst.* **2019**, *11*, 376–399.
10. Xi, X.; Wei, Z.; Xiaoguang, R.; Yijie, W.; Xinxin, B.; Wenjun, Y.; Jin, D. A comprehensive evaluation of air pollution prediction improvement by a machine learning method. In Proceedings of the 2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI), Yasmine Hammamet, Tunisia, 15–17 November 2015; pp. 176–181. [CrossRef]
11. Brownlee, J. Master Machine Learning Algorithms. 2016. Available online: https://machinelearningmastery.com/master-machine-learning-algorithms/ (accessed on 23 December 2021).
12. Elkamel, A.; Abdul-Wahab, S.; Bouhamra, W.; Alper, E. Measurement and prediction of ozone levels around a heavily industrialized area: A neural network approach. *Adv. Environ. Res.* **2001**, *5*, 47–59. [CrossRef]
13. Aljanabi, M.; Shkoukani, M.; Hijjawi, M. Ground-level Ozone Prediction Using Machine Learning Techniques: A Case Study in Amman, Jordan. *Int. J. Autom. Comput.* **2020**, *17*, 667–677. [CrossRef]
14. Jumin, E.; Zaini, N.; Ahmed, A.N.; Abdullah, S.; Ismail, M.; Sherif, M.; Sefelnasr, A.; El-Shafie, A. Machine learning versus linear regression modelling approach for accurate ozone concentrations prediction. *Eng. Appl. Comput. Fluid Mech.* **2020**, *14*, 713–725.
15. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [CrossRef]
16. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185.
17. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
18. Liang, Y.C.; Maimury, Y.; Chen, A.H.L.; Juarez, J.R.C. Machine Learning-Based Prediction of Air Quality. *Appl. Sci.* **2020**, *10*, 9151.
19. Liao, H.; Sun, W. Forecasting and Evaluating Water Quality of Chao Lake based on an Improved Decision Tree Method. *Procedia Environ. Sci.* **2010**, *2*, 970–979. doi: 10.1016/j.proenv.2010.10.109. [CrossRef]
20. Lindner, B.L.; Mohlin, P.J.; Caulder, A.C.; Neuhauser, A. Development and Testing of a Decision Tree for the Forecasting of Sea Fog Along the Georgia and South Carolina Coast. *J. Oper. Meteorol.* **2018**, *6*, 47–58. [CrossRef]
21. Cabaneros, S.M.; Calautit, J.K.; Hughes, B.R. A review of artificial neural network models for ambient air pollution prediction. *Environ. Model. Softw.* **2019**, *119*, 285–304. [CrossRef]
22. Guo, C.; Liu, G.; Chen, C.H. Air Pollution Concentration Forecast Method Based on the Deep Ensemble Neural Network. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, e8854649.
23. Kalajdjieski, J.; Zdravevski, E.; Corizzo, R.; Lameski, P.; Kalajdziski, S.; Pires, I.M.; Garcia, N.M.; Trajkovik, V. Air Pollution Prediction with Multi-Modal Data and Deep Neural Networks. *Remote Sens.* **2020**, *12*, 4142.
24. Rahman, P.A.; Panchenko, A.A.; Safarov, A.M. Using neural networks for prediction of air pollution index in industrial city. *IOP Conf. Ser. Earth Environ. Sci.* **2017**, *87*, 042016. [CrossRef]
25. Maleki, H.; Sorooshian, A.; Goudarzi, G.; Baboli, Z.; Tahmasebi Birgani, Y.; Rahmati, M. Air pollution prediction by using an artificial neural network model. *Clean Technol. Environ. Policy* **2019**, *21*, 1341–1352. [CrossRef] [PubMed]
26. Krishan, M.; Jha, S.; Das, J.; Singh, A.; Goyal, M.K.; Sekar, C. Air quality modelling using long short-term memory (LSTM) over NCT-Delhi, India. *Air Qual. Atmos. Health* **2019**, *12*, 899–908. [CrossRef]
27. Dua, R.D.; Madaan, D.M.; Mukherjee, P.M.; Lall, B.L. Real Time Attention Based Bidirectional Long Short-Term Memory Networks for Air Pollution Forecasting. In Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 4–9 April 2019; pp. 151–158. [CrossRef]

28. Xayasouk, T.; Lee, H.; Lee, G. Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models. *Sustainability* **2020**, *12*, 2570.
29. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Association for Computing Machinery (KDD'16), New York, NY, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
30. Liu, R.; Ma, Z.; Liu, Y.; Shao, Y.; Zhao, W.; Bi, J. Spatiotemporal distributions of surface ozone levels in China from 2005 to 2017: A machine learning approach. *Environ. Int.* **2020**, *142*, 105823. [CrossRef]
31. Capilla, C. Prediction of hourly ozone concentrations with multiple regression and multilayer perceptron models. *Int. J. Sustain. Dev. Plan.* **2016**, *11*, 558–565. [CrossRef]
32. Li, R.; Cui, L.; Hongbo, F.; Li, J.; Zhao, Y.; Chen, J. Satellite-based estimation of full-coverage ozone (O_3) concentration and health effect assessment across Hainan Island. *J. Clean. Prod.* **2020**, *244*, 118773. [CrossRef]
33. World's Most Polluted Cities in 2020—PM2.5 Ranking | AirVisual. Available online: https://www.iqair.com/us/world-most-polluted-cities (accessed on 1 November 2020).
34. Central Pollution Control Board of India, Air Pollution Standards. Available online: https://cpcb.nic.in/air-pollution (accessed on 1 July 2021).
35. Central Pollution Control Board of India, Automatic Monitoring Data. Available online: https://cpcb.nic.in/automatic-monitoring-data/ (accessed on 1 November 2020).
36. Visual Crossing Weather API Documentation (Visual-Crossing-Corporation-Visual-Crossing-Corporation-Default). Available online: https://rapidapi.com/visual-crossing-corporation-visual-crossing-corporation-default/api/visual-crossing-weather (accessed on 1 November 2020).
37. Masood, A.; Ahmad, K. A model for particulate matter (PM2.5) prediction for Delhi based on machine learning approaches. *Procedia Comput. Sci.* **2020**, *167*, 2101–2110. [CrossRef]
38. Mahalingam, U.; Elangovan, K.; Dobhal, H.; Valliappa, C.; Shrestha, S.; Kedam, G. A Machine Learning Model for Air Quality Prediction for Smart Cities. In Proceedings of the 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 21–23 March 2019; pp. 452–457. [CrossRef]
39. Sinha, A.; Singh, S. Review on air pollution of Delhi zone using machine learning algorithm. *J. Air Pollut. Health* **2020**, *5*, 259–272. [CrossRef]
40. Sinha, A.; Singh, S. Dynamic forecasting of air pollution in Delhi zone using machine learning algorithm. *Quantum J. Eng. Sci. Technol.* **2021**, *2*, 40–53.
41. Shukla, K.; Dadheech, N.; Kumar, P.; Khare, M. Regression-based flexible models for photochemical air pollutants in the national capital territory of megacity Delhi. *Chemosphere* **2021**, *272*, 129611. [CrossRef]
42. National Ambient Air Quality Standards. Available online: https://www.epa.gov/criteria-air-pollutants/naaqs-table (accessed on 1 July 2021).
43. Caiafa, C.F.; Solé-Casals, J.; Marti-Puig, P.; Zhe, S.; Tanaka, T. Decomposition Methods for Machine Learning with Small, Incomplete or Noisy Datasets. *Appl. Sci.* **2020**, *10*, 8481.
44. Chemistry in the Sunlight. 2003. Available online: https://earthobservatory.nasa.gov/features/ChemistrySunlight/chemistry_sunlight3.php (accessed on 1 November 2020).
45. Iskandaryan, D.; Ramos, F.; Trilles, S. Air Quality Prediction in Smart Cities Using Machine Learning Technologies Based on Sensor Data: A Review. *Appl. Sci.* **2020**, *10*, 2401.
46. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [CrossRef]
47. Park, K.; Jung, Y.; Kim, K.; Park, S.K. Determination of Deep Learning Model and Optimum Length of Training Data in the River with Large Fluctuations in Flow Rates. *Water* **2020**, *12*, 3537.
48. Maddu, R.; Vanga, A.R.; Sajja, J.K.; Basha, G.; Shaik, R. Prediction of land surface temperature of major coastal cities of India using bidirectional LSTM neural networks. *J. Water Clim. Chang.* **2021**, *12*, 3801–3819. [CrossRef]
49. Liu, B.; Yan, S.; Li, J.; Qu, G.; Li, Y.; Lang, J.; Gu, R. A Sequence-to-Sequence Air Quality Predictor Based on the n-Step Recurrent Prediction. *IEEE Access* **2019**, *7*, 43331–43345.
50. Zhang, B.; Zou, G.; Qin, D.; Lu, Y.; Jin, Y.; Wang, H. A novel Encoder-Decoder model based on read-first LSTM for air pollutant prediction. *Sci. Total Environ.* **2021**, *765*, 144507. [CrossRef]
51. Tiwari, A.; Gupta, R.; Chandra, R. Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown. *arXiv* **2021**, arXiv:2102.10551.
52. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
53. Zhang, Z.; Yang, R.; Fang, Y. LSTM Network Based on on Antlion Optimization and its Application in Flight Trajectory Prediction. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 1658–1662. [CrossRef]
54. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
55. Zhou, J.; Huo, X.; Xu, X.; Li, Y. Forecasting the Carbon Price Using Extreme-Point Symmetric Mode Decomposition and Extreme Learning Machine Optimized by the Grey Wolf Optimizer Algorithm. *Energies* **2019**, *12*, 950. [CrossRef]
56. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man, Cybern.* **1993**, *23*, 665–685.

57. Yuan, X.; Chen, C.; Lei, X.; Yuan, Y.; Muhammad Adnan, R. Monthly runoff forecasting based on LSTM–ALO model. *Stoch. Environ. Res. Risk Assess.* **2018**, *32*, 2199–2212. [CrossRef]
58. Adnan, R.M.; Mostafa, R.R.; Kisi, O.; Yaseen, Z.M.; Shahid, S.; Zounemat-Kermani, M. Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization. *Knowl.-Based Syst.* **2021**, *230*, 107379. [CrossRef]
59. Belvederesi, C.; Dominic, J.A.; Hassan, Q.K.; Gupta, A.; Achari, G. Predicting River Flow Using an AI-Based Sequential Adaptive Neuro-Fuzzy Inference System. *Water* **2020**, *12*, 1622.
60. Zhang, L.; Chen, X.; Zhang, Y.; Wu, F.; Chen, F.; Wang, W.; Guo, F. Application of GWO-ELM Model to Prediction of Caojiatuo Landslide Displacement in the Three Gorge Reservoir Area. *Water* **2020**, *12*, 1860.
61. Jaafari, A.; Panahi, M.; Pham, B.T.; Shahabi, H.; Bui, D.T.; Rezaie, F.; Lee, S. Meta optimization of an adaptive neuro-fuzzy inference system with grey wolf optimizer and biogeography-based optimization algorithms for spatial prediction of landslide susceptibility. *CATENA* **2019**, *175*, 430–445. [CrossRef]
62. Adnan, R.M.; Mostafa, R.; Islam, A.R.M.T.; Kisi, O.; Kuriqi, A.; Heddam, S. Estimating reference evapotranspiration using hybrid adaptive fuzzy inferencing coupled with heuristic algorithms. *Comput. Electron. Agric.* **2021**, *191*, 106541. [CrossRef]
63. Goyal, M.K.; Bharti, B.; Quilty, J.; Adamowski, J.; Pandey, A. Modeling of daily pan evaporation in sub tropical climates using ANN, LS-SVR, Fuzzy Logic, and ANFIS. *Expert Syst. Appl.* **2014**, *41*, 5267–5276. [CrossRef]
64. Updates To The Air Quality Index (Aqi) For Ozone And Ozone Monitoring Requirements. Available online: https://www.epa.gov/sites/default/files/2015-10/documents/20151001_air_quality_index_updates.pdf (accessed on 1 July 2021).
65. Srivastava, C.; Singh, S.; Singh, A.P. Estimation of Air Pollution in Delhi Using Machine Learning Techniques. In Proceedings of the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 28–29 September 2018; pp. 304–309. [CrossRef]
66. Gajinkar, A. Exploratory Data Analysis of Indian Rainfall Data. 2019. Available online: https://medium.com/@anusha.gajinkar/exploratory-data-analysis-of-indian-rainfall-data-f9755f2cc81d (accessed on 1 November 2020).
67. Kumar, A.; Goyal, P. Forecasting of air quality in Delhi using principal component regression technique. *Atmos. Pollut. Res.* **2011**, *2*, 436–444. [CrossRef]
68. Abdullah, S.; Nasir, N.; Ismail, M.; Ahmed, A.; Jarkoni, M. Development of Ozone Prediction Model in Urban Area. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 2263–2267. [CrossRef]
69. Jin, X.; Fiore, A.M.; Murray, L.T.; Valin, L.C.; Lamsal, L.N.; Duncan, B.; Folkert Boersma, K.; De Smedt, I.; Abad, G.G.; Chance, K.; et al. Evaluating a Space-Based Indicator of Surface Ozone-NOx-VOC Sensitivity Over Midlatitude Source Regions and Application to Decadal Trends. *J. Geophys. Res. Atmos.* **2017**, *122*, 10439–10461.
70. Lovrić, M.; Pavlović, K.; Vuković, M.; Grange, S.K.; Haberl, M.; Kern, R. Understanding the true effects of the COVID-19 lockdown on air pollution by means of machine learning. *Environ. Pollut.* **2021**, *274*, 115900. [CrossRef] [PubMed]