Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Hourly Air Quality Prediction in Dhaka City using Time Series Forecasting Techniques: Deep Learning Perspectives

Md. Lisun-Ul-Islam[1], Md. Rahatul Hasan Rahat[2], Sanjida Esha[3], Abrar Faiyaz[4], Akinul Islam Jony[5*]

[1,2,3,4,5]*Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, Bangladesh*

Abstract: - Air pollution is a concern worldwide, especially in densely populated cities in developing nations like Dhaka, Bangladesh. Accurately predicting air quality is crucial for health and environmental management. This research explores the effectiveness of learning models based on time series forecasting to predict air quality levels in Dhaka, with a particular focus on feature selection and the Air Quality Index (AQI). It also compares the performance and adaptability of models such as long-short-term memory (LSTM), convolutional neural network (CNN), recurrent neural network (RNN), and gated recurrent unit (GRU) using various feature configurations. The study highlights that LSTM performs well in scenarios while CNN, RNN and GRU consistently exhibit strong predictive capabilities. Underscore their suitability for air quality forecasting particularly when considering the prediction of AQI. However, the study acknowledges the limitations of relying on data sources. Emphasizes the need for further research into hybrid models and how local factors impact air quality. The findings provide insights for practitioners and decision-makers aiming to tackle air pollution challenges in urban areas by emphasizing feature selection and AQIs role in enhancing predictive accuracy.

Keywords*: Time Series, Feature Selection, Air Quality Index, Hourly Forecasting, Deep Learning.*

1. Introduction

Air pollution is a concern that poses risks to both human well-being and the global ecosystem. Developing countries face vulnerability to the effects of air pollution due to rapid industrialization, urbanization, and population growth. Dhaka, which is the capital of Bangladesh and one of the world's most populated cities has been identified as among the most polluted in terms of air quality [16]. As a result, accurately predicting air quality levels in Dhaka is critical for lightening the impact of air impurity on health and the surroundings. Therefore, this research aims to explore how time series forecasting models can be adapted and implemented in settings with diverse sources of air pollution and weather conditions. The primary objective is to evaluate the effectiveness of time series forecasting techniques in predicting air quality levels specifically for Dhaka.

Deep learning perspective is now commonly used in research studies for model development (such as [24] and [25]). However, past studies on air quality forecasting had focused on deep learning models to predict AQI in cities across the globe. For instance, one study [1] proposed a LSTM multitask learning for PM2.5 prediction, where they utilized LSTM for real-time air quality prediction. Additionally, the research study in [5] utilized LSTM to predict O3 concentrations. But we assessed the performance of LSTM across pollutants and metrics. The importance of air quality forecasting is emphasized in reference [6], where historical data and deep learning techniques were explored. Moreover, reference [7] employed LSTM for air quality prediction in Dhaka, whereas in our study the research gap lies in comprehending their approaches to feature selection. Therefore, this research study aims to address the following research questions.

- How effective are different learning techniques for predicting air quality levels in Dhaka City in terms of accuracy and adaptability?
- What are the important attributes and feature selection methods that significantly impact the performance of learning models in air quality prediction?
- How can these knowledge be utilized to improve air quality forecasting in urban areas like in Dhaka?
- Which deep learning approaches show promise for predicting air quality levels?

To tackle these research question, this study starts with data collection, preprocessing techniques, and feature engineering for ensuring the quality of data and better identification of attributes. Deep learning models like LSTM, CNN, RNN and GRU are then employed to predict air quality levels, allowing for an evaluation of their effectiveness in capturing patterns [17]. The research has two outcomes: hourly predictions of air quality in Dhaka for decision-making in environmental and public health fields and applying the model's capabilities of detecting air quality to other cities or urban areas.

The rest of the paper is structured as follows. Section 2 provides a review of research on air quality prediction using deep learning. This section also identifies advancements and highlights gaps in the existing literature. Section 3 outlines the methodology used for data collection, preprocessing, feature selection and model development. Section 4 focuses on model development in detail, for example, explaining the accuracy and comparing the performance of all learning models. Section 5 analyses and discusses the findings and results obtained from the experiments. Finally, section 6 concludes the research work with remarks on limitations and future research directions.

2. Related Works

This research study focuses on air quality testing using various deep learning models and feature selection techniques to evaluate accuracy. The section will compare existing works related to our study. Such as Xu and Yoneda (2021) [1] propose an LSTM autoencoder multitask learning model for predicting PM 2.5 time series in urban areas. While our research study also considers LSTM as one of the deep learning models, the research gap lies in the comparison and evaluation of our LSTM model with the specific LSTM autoencoder multitask learning approach proposed in this paper. We are exploring how our LSTM model performs in comparison to the multitask learning approach for predicting air quality, and potentially extending our analysis to include other pollutants in addition to PM 2.5.

Sonawani et al. (2021) [2] showed NO2 concentration in Pune, India, proposing an optimized bidirectional GRU model for real-time monitoring in smart cities. Our study investigates deep learning models for predicting air quality in Dhaka, emphasizing LSTM, CNN, RNN, and GRU, focusing on feature selection and AQI. The key distinction lies in the broader scope of this study, covering multiple pollutants and models for urban settings, while the other paper is specific to NO2 and emphasizes optimization and smart city applications.

Hamami and Dahlan (2020) [3] used IoT technology and a neural network model with LSTM to forecast air pollution concentrations in real-time by air quality monitoring. The research gap in our study lies in the implementation of deep learning models like LSTM in real-time AQI forecasting. We are exploring the effectiveness of LSTM and other deep learning models in a real-time monitoring setup and their predictive performance with the LSTM model proposed here [3]. Besides, Chang et al. (2020) [4] also introduced LSTM, aggregating local and external pollution sources for Taiwan's air quality prediction. It competes favourably with SVR, GBTR, and LSTM in predicting PM2.5. We address Dhaka's air quality using LSTM models, emphasizing AQI and robust feature selection. LSTM outperforms other models. Acknowledging data limitations, future research into hybrid models is suggested. Both papers utilize LSTM, but our focus on Dhaka, AQI, and comprehensive methodology sets it apart. Their paper's strength lies in holistic prediction through source aggregation. Together, they contribute diverse insights into LSTM's application in air quality forecasting.

Freeman et al. (2018) [5] focused on forecasting air quality time series using LSTM for predicting O3 concentrations. The research gap with our research work lies in comparing the predictive performance of LSTM for O3 concentrations with the other pollutants and air quality metrics. The strengths and weaknesses of LSTM in predicting various air quality indicators and comparing its performance with other deep learning models in this study. While our work considers LSTM for air quality testing, we have not specifically focused on predicting O3 concentrations. The research gap lies in evaluating the predictive performance of LSTM for O3 concentrations compared to other pollutants and air quality metrics we have explored. Additionally, we can investigate how LSTM compares to other deep learning models in predicting various air quality indicators in our dataset.

Yan et al. (2021) [6] presents a study on air quality forecasting in a busy urban traffic station in Helsinki. They utilized a parallel genetic algorithm and a multi-layer perceptron model to predict hourly nitrogen dioxide concentrations and highlighted the importance of accurate air quality forecasting in urban areas with significant traffic-related pollution. By using deep learning algorithms, the study contributed to developing effective strategies for mitigating air pollution's adverse impact on public health and the environment in densely populated urban settings. The research gap lies in the need to fully consider spatiotemporal distribution characteristics to enhance air quality forecasting accuracy in densely populated cities and exploring how historical data and deep learning algorithms can be effectively used for predicting hourly air quality levels in Dhaka city.

Tsai et al. (2018) [7] utilized deep learning models such as LSTM, for predicting air quality levels. However, the specific feature selection techniques are not mentioned or explored. On the other hand, we use multivariate time series analysis to forecast the AQI in Bangladesh. The researchers include various data dimensions (month, weekday, hour) alongside air quality-related data, but again, the feature selection methods are not explicitly stated. Both papers focus on air quality forecasting, but they differ in their feature selection approaches. The research gap lies in the lack of detailed descriptions of the feature selection techniques employed, making it essential to address this gap for better model interpretability and effectiveness in air quality forecasting.

3. Methods and Materials

The methos and materials adopted in this research study are discussed briefly in the following subsections. It starts with describing the overall methodology followed by dataset overview, and a brief description of deep learning models, feature selection and evaluation metrics.

3.1. Research Approach

The research process begins by collecting data, which refers to the data that has been previously collected by others (records or publicly available information). This secondary data is prepared to address missing values, outliers, and normalization. These steps ensure that the data can be effectively utilized by learning models. After this stage, a correlation matrix is computed to determine the relationships between features. This knowledge aids in selecting the features to be used in deep learning models.

To identify the set of features for prediction, various feature selection techniques are employed. These techniques encompass utilizing the correlation matrix, which is a feature selection method, backward feature elimination, wrapper method, information gain, forward feature selection, recursive feature elimination (RFE), and L1 regularization (LASSO). Once the appropriate features have been chosen multiple deep-learning models are trained with the aim of predicting the target variable, which is AQI. The models include LSTM, CNN, RNN and GRUs. Ultimately, the model that exhibits performance on a held-out test set is selected as the model. In general, the methodology process tree offers an outline of the stages in creating a deep learning model for AQI. This approach is methodical and thorough, with the intention of generating a dependable model as presented step-by-step in Figure 1.

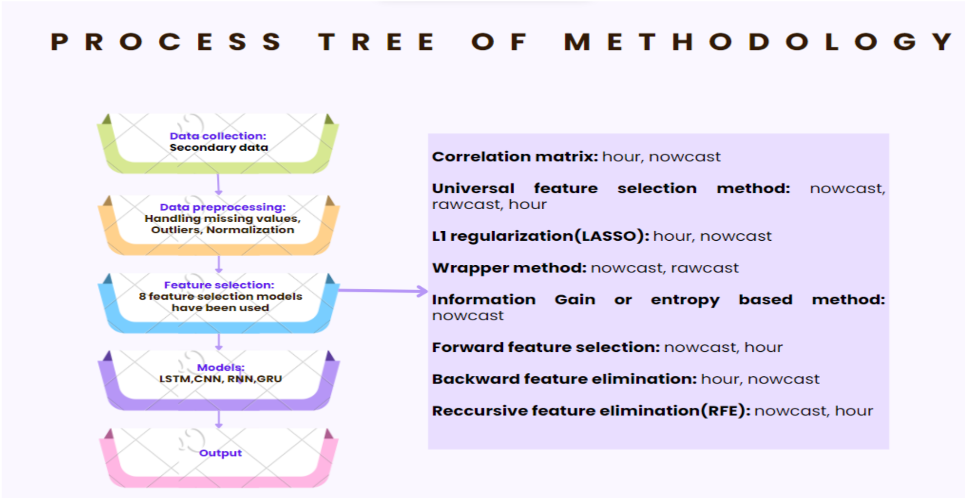

Figure 1: Overview of Research Process

3.2. Dataset Overview

In this research work, we plan to utilize a method for collecting data. The dataset that will be utilized in our research is sourced from Kaggle (to access the dataset, see the "Data Availability Statement"). This dataset consists of 8 columns, Date Nowcast Concentration, Raw Concentration, Concentration Unit, AQI, AQI Category and QC Name. We have gathered a total of 51,923 instances. The use of nowcasting concentrations can greatly contribute to air quality management and public health. It provides real-time information about the state of air quality conditions. This information proves valuable in issuing warnings, informing policy decisions, and taking actions to acknowledge the impact of air pollution.

Air quality monitoring stations collect data on concentrations using equipment like sensors, analysers and other devices. These devices aid in the identification and quantification of substances like nitrogen oxides, sulphur dioxide, tiny particles and ozone [20]. The AQI is a measurement used to explain the condition of the air we breathe. It offers insights into how air pollution can impact our well-being, over a duration. The main goal of the AQI is to inform people about the impact of air quality on their well-being [20]. We also have an AQI category, which depends on the range of AQI. It has some string values, and the ranges are, AQI level (0-50) means good; (51-100) means moderate; (101-150) means unhealthy for some sensitive groups; (151-200) means unhealthy; (201-300) means way to unhealthy; and (301+) means hazardous [15]. We also have Date and QC name variables. QC name in for the data authentication part is the data got from the authenticate source or not. In the case of Date, it is important for visualizing the data in our time-series forecast.

3.3. Data Preprocessing

To initiate our analysis, we first examined the dataset to identify any gaps in information or irregularities that might impact our conclusions. We resolved any data by implementing data processing techniques such as cleansing, filtering, managing outliers and smoothing the data. Afterward, we carefully selected the features for our analysis, eliminating any unnecessary ones using feature selection models. In the end, we assessed the data's quality once we finished going through all the steps.

3.4. Deep Learning Models

LSTM is a type of network designed to effectively handle long-term dependencies in sequential data [21]. It addresses the problem of gradient vanishing commonly encountered in RNNs by incorporating memory cells and gate mechanisms that regulate the flow of information. LSTMs find applications in domains such as natural language processing, speech recognition and time series prediction.

The equations for the gates in LSTM are:

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

$$f_t = \sigma \ (w_f[h_{t-1}, x_t] + b_f) \tag{2}$$

$$o_t = \sigma \ (w_o[h_{t-1}, x_t] + b_o) \tag{3}$$

The following lists the cell state, candidate cell state, and final output equation:

$$\underline{c}_t = tanh\ (w_c[h_{t-1}, x_t] + b_c) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \underline{c}_t \tag{5}$$

$$h_t = o_t * tanh\ (c^t) \tag{6}$$

In these equations,

i_t = input gate, f_t = forget gate, o_t = output gate, σ = sigmoid function,
w_x = gates' respective weight,
h_{t-1} = output for LSTM block where timestamp t – 1,
x_t = current timestamp on input,
b_x = gates' respective biases(x),
c_t = timestamp (t) on cell state, and
$\underline{c}_t$ = at timestamp(t), represents candidates for cell state.

CNN is a type of network that is frequently utilized for the analysis of forms of data including images or videos [22]. By utilizing the layers, its functionality involves extracting features from the input data by means of applying filters. Then, it utilizes pooling layers to reduce the output size. CNNs have shown efficacy such as image generation, image classification and object detection. If we use ω as a filter (m × m), convolutional layer output size will be (N−m+1) × (N−m+1) (N−m+1) ×(N−m+1). To compute some unit x_{ij}^{l} from the pre-nonlinearity input in our layer, we need the contributions from the previous layer cells using the following equation.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab}\, y_{(i+a)(j+b)}^{l-1} \tag{7}$$

In this equation,

m = upper limit of the summation, indicating the number of terms to be added,
Σ = sigma symbol denotes a summation, indicating that the terms following it should be added together,
a = index for the summation from 0 to m-1,
b = Index for the summation from 0 to m-1,
e = The natural logarithm's base roughly equivalent to 2.71828,
ab = product of variables a and b,
(i+a) = sum of variables i and a, and
(j+b) =The sum of variables j and b.

RNN is a network created especially to manage data. They achieve this by incorporating feedback connections, which create a loop structure allowing them to retain states. This capability enables RNNs to effectively capture the dependencies in the data they process. RNNs find applications in tasks such as language modeling, machine translation and speech recognition [23]. The following are the required equations, where W_{IH} ,W_{HH} and W_{HO} are the connection weight matrices, $f_H$and $f_O$are the hidden and output activation functions, x(t) and y(t) are network input and output vectors.

$$h(t) = f_H\left(W_{IH}\, x(t) + W_{HH}\, h(t-1)\right) \tag{8}$$

$$y(t) = f_o\left(W_{HO}\, h(t)\right) \tag{9}$$

GRU is a special type of network for handling data that identifies the issues of vanishing gradient that are commonly faced in RNNs through incorporating gating mechanisms. These gates enable the network to update and forget information thereby improving its capability to capture long-range dependencies. GRUs, regulate the information flow with an update gate and a reset gate [19]. Due to their training capabilities, GRUs find usage in machine learning applications. The following are the required equations, where z_t = update gate, r_t = rest gate, H_t = candidate activation, and h_t = hidden state. Besides, the output y of GRU is given as y = softmax (h_t).

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{10}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{11}$$

$$H_t = tanh\left(W_r x_t + U_r(r_t\, h_{t-1})\right) \tag{12}$$

$$h_t = (1 - z_t)h_{t-1} + z_t H_t \tag{13}$$

3.5. Feature Selection

Feature selection plays a role in improving machine learning models. It helps identify the features in our dataset leading to various important benefits. The following is the description of feature selection models.

Correlation Matrix: The Correlation Matrix [8] model examines the relationships between variables using correlation coefficients. It shows that "Hour" and "NowCast" have a correlation with either the target variable or other variables.

Universal Feature Selection Method: The Universal Feature Selection Method considers a range of techniques for feature selection. In this case based on this method, "Hour " "NowCast," and "RawCast" were identified as features. These features likely contain information. Exhibit strong predictive power.

L1 Regularization: L1 regularization, also known as LASSO [9] penalizes model coefficients to encourage sparsity. It reveals that "Hour" and "NowCast" were chosen as features suggesting they have an impact on the target variable while other features may be less relevant or redundant.

Wrapper Method: The Wrapper method [10] involves training models with subsets of features. Evaluating their performance to make selections. The choice is based on observing improvements in model performance. It suggests that the features "Hour" and "NowCast" were chosen as factors because they significantly contribute to the model's ability to make predictions.

Information Gain: Information gain is used in decision trees to select features. It measures the reduction in uncertainty or impurity when a feature is used to split the data. The selection process identified "Hour". NowCast" as having information gain indicating their relevance and their ability to distinguish the target variable.

Forward Feature Selection: The Forward method of feature selection [11] begins with a set of features. Gradually add one feature at a time based on how it improves the model's performance. In this case, the features "Hour" and "NowCast" were chosen as they resulted in the enhancement of model performance.

Backward Feature Elimination: The Backward method is an interesting way for feature elimination [13]. It starts with all the features. Then based on the impact of a feature on the model's performance, it removes one feature at a time. After using this process "Hour" and NowCast" are considered features for maintaining optimal model performance.

Recursive Feature Elimination: The suitable features are "Hour" and "NowCast." These two features have consistently shown capability, making them the best choice for accurate and efficient predictions. By selecting these features [12] our model can maintain its forecasting accuracy while simplifying its input variables.

In general, it is worth noting that the features "Hour" and "NowCast" consistently emerge as factors in most of the applied feature selection models. This suggests that these attributes hold information and exert an influence on the target variable within the dataset.

3.6. Evaluation Metrics

The term "evaluation metrics" refers to a set of metrics used to assess how well deep learning models perform with unseen data. For our study, we are going to use Accuracy, Precision, Recall and F1-Score evaluation metrics.

Accuracy: The concept of accuracy involves measuring how many instances are predicted correctly out of the number of instances.

$$\text{Accuracy}=\frac{True\ Positives+True\ Negatives}{Total\ Instances\ Accuracy} \tag{14}$$

Precision: Precision focuses on the percentage of positive predictions by comparing all the positive predictions that were made.

$$\text{Precision}= \frac{True\ Positives}{True\ Positives+False\ Positives} \tag{15}$$

Recall (Sensitivity): Recall is known as sensitivity, where the true positive rate refers to the percentage of true positive predictions from all the actual positive instances.

$$\text{Recall}=\frac{True\ Positives}{True\ Positives+False\ Negatives} \tag{16}$$

F1-Score: The F1 Score is an indicator that computes precision and recall by calculating their mean. It is particularly useful in situations where there is an imbalance between classes.

$$\text{F1-Score}=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{17}$$

These equations [18] provide a quantitative way to estimate the performance of a binary classification model. They provide information about evaluation to understand how well the model is at correctly classifying instances, identifying positive instances, and balancing precision and recall.

4. Model Development

We have used eight feature selection models (such as Correlation Matrix, Universal Feature Selection Method, L1 Regularization, Wrapper Method, Information Gain, Forward Feature Selection, Backward Feature Elimination, and Recursive Feature Elimination). Seven of the models selected the same attributes, which are "Hour" and "NowCast" attributes. On the other hand, the Universal Feature Selection Method selected three attributes, which are "Hour", "NowCast", and "RawCast".

4.1. LSTM

Table 1 shows the evaluation metrics for LSTM based on two sets of attributes such as evaluation metrics based on Hour, NowCast, RawCast features, and evaluation metrics based on Hour, NowCast features.

Evaluation Metrics based on Hour, NowCast, RawCast features: The model demonstrated remarkable performance with an accuracy of 97.93%, precision of 97%, recall of 99%, and an F1-score of 98. It showcased strong predictive capabilities on the specified features. Additionally, the model's Mean Squared Error (MSE) was calculated at 3382.32, confirming its effectiveness in forecasting.

Table 1: LSTM Performance Metrics

	Hours, NowCast					Hours, NowCast, RawCast				
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**
LSTM	99.17	97	100	98	3371.48	97.93	97	99	98	3382.32

Evaluation Metrics based on Hour, NowCast features: The model achieved an impressive accuracy of 99.17%. It's predicted well by a high F1 score of 98 also precision and recall rates of 97% and 100%. In terms of predictive accuracy, the model yielded a mean squared error (MSE) of 3371.48, indicating its effectiveness in forecasting.

4.2. CNN

Table 2 shows the evaluation metrics for CNN based on two sets of attributes such as evaluation metrics based on Hour, NowCast, RawCast features, and evaluation metrics based on Hour, NowCast features.

Evaluation Metrics based on Hour, NowCast, RawCast features: The CNN model continued to perform well, achieving an accuracy rate of 97.95%. It maintained its strong predictive capabilities with an F1 score of 87, reflecting its ability to make accurate and reliable predictions. The precision rate remained consistent at 81%, indicating its precision in making positive predictions. The recall rate, while slightly reduced compared to the previous feature set, remained high at 90%, signifying its proficiency in correctly identifying positive instances. The CNN model's mean squared error (MSE) on this feature set was calculated at 4227.67, further establishing its reliability and effectiveness in forecasting.

Table 2: CNN Performance Metrics

	Hours, NowCast					Hours, NowCast, RawCast				
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**
CNN	98.90	81	91	86	3736.29	97.95	81	90	87	4227.67

Evaluation Metrics based on Hour, NowCast features: The CNN model demonstrated an impressive level of performance with an accuracy rate of 98.90%. Its predictive capabilities were underscored by a high F1 score of 86, indicating its effectiveness in making accurate predictions. The precision rate of 81% showcased its ability to make precise positive predictions, while the recall rate of 91% highlighted its capacity to identify positive instances correctly. In terms of predictive accuracy, the CNN model yielded a mean squared error (MSE) of 3736.29, confirming its reliability and effectiveness in forecasting.

4.3. RNN

Table 3 shows the evaluation metrics for RNN based on two sets of attributes such as evaluation metrics based on Hour, NowCast, RawCast features, and evaluation metrics based on Hour, NowCast features.

Evaluation Metrics based on Hour, NowCast, RawCast features: With an accuracy of 97.25%, the RNN model maintained a strong precision (81%) and recall (92%). It exhibited an F1-score of 86, demonstrating reliable predictive capabilities when considering the extended feature set, "Hours," "NowCast," and "RawCast". The MSE for this model was 4140.81, underlining its forecasting effectiveness.

Table 3: RNN Performance Metrics

Hours, NowCast						Hours, NowCast, RawCast				
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**
RNN	99	86	93	88	3664.06	97.95	81	92	86	4140.81

Evaluation Metrics based on Hour, NowCast features: The RNN model achieved an impressive accuracy of 99%, with a strong balance of precision (86%) and recall (93%). It displayed an F1-score of 88, highlighting its robust predictive abilities. The model's MSE was 3664.06, confirming its effectiveness in forecasting.

4.4. GRU

Table 4 shows the evaluation metrics for GRU based on two sets of attributes such as evaluation metrics based on Hour, NowCast, RawCast features, and evaluation metrics based on Hour, NowCast features.

Evaluation Metrics based on Hour, NowCast, RawCast features: Maintaining a solid accuracy of 98.07%, the GRU model exhibited consistent precision (93%) and recall (97%). It obtained an F1-score of 92, indicating reliable predictive abilities while considering the extended feature set, "Hours," "NowCast," and "RawCast." The MSE for this model was 4645.97, reinforcing its forecasting capability.

Table 4: GRU Performance Metrics

Hours, NowCast						Hours, NowCast, RawCast				
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**
GRU	99.23	94	97	93	3404.54	98.07	93	97	92	4645.97

Evaluation Metrics based on Hour, NowCast features: The GRU model achieved an impressive accuracy of 99.23%, demonstrating strong predictive capabilities with precision (94%) and recall (97%). It yielded an F1-score of 93, emphasizing its well-rounded performance. The model's MSE was 3404.54, affirming its efficiency in forecasting.

5. Results and Findings

This research thoroughly examines the effectiveness of LSTM, CNN, RNN and GRU models, in predicting air quality. It explores combinations of features. Consistently finds that "Hour" and "NowCast" have a significant impact. Additionally, a new method called "RawCast" is introduced as influential. The LSTM model stands out due to its adaptability achieving accuracy (97.93% and 99.17%) and demonstrating precision, recall and F1 score values across different sets of features. While the CNN model excels in accuracy (97.95% and 98.90%) it shows sensitivity to features as reflected in a mean squared error when utilizing a three-feature set. Both RNN and GRU models consistently perform well, underscoring the importance of considering "Hour" and "NowCast". After evaluation, the LSTM model emerges as a performer due to its versatility across various input configurations. The comparison between these models as shown in table 5 highlight the nuanced trade-offs between accuracy, model complexity and computational efficiency. Considering its adaptability, the LSTM model shows promise for realworld applications in air quality forecasting by providing insights into feature selection dynamics and model performance complexities.

Table 5: Comparison of Models based on Evaluation Metrics

	Hours, NowCast					Hours, NowCast, RawCast				
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**MSE**
LSTM	99.17	97	100	98	3371.48	97.93	97	99	98	3382.32
CNN	98.90	81	91	86	3736.29	97.95	81	90	87	4227.67
RNN	99	86	93	88	3664.06	97.95	81	92	86	4140.81
GRU	99.23	94	97	93	3404.54	98.07	93	97	92	4645.97

Our findings reinforce the efficacy of deep learning models, showcasing their ability to outperform traditional methods. The consistently high accuracy across different architectures underscores their suitability for air quality forecasting.

6. Conclusion

In summary, our research aimed to assess the efficacy of deep learning models for predicting hourly air quality levels in Dhaka, Bangladesh. We thoroughly examined architectures like LSTM, CNN, and GRU using an approach. Our research shows that features like 'Hour', 'Nowcast', and 'Rawcast' had an impact on our prediction. So, those features we have selected are the key features that can be the main asset for historical and real-time data.

However, in our research, we have used secondary sources for collecting our data, which increases the chances of biasness or inaccuracies in the data set. The main drawback of our research is that we didn't explore city-specific characteristics. Our focus was on quantitative data and qualitative aspects such as local emission sources. Future research might benefit from adding a wider range of comprehensive sets of data sources and considering the unique contextual factors influencing air quality.

To build upon our discoveries, future research could investigate deeply into refining the methods for selecting features and exploring models that combine the strengths of structures and extending the study to include other cities with diverse pollution characteristics. These approaches have the potential to enhance the effectiveness and adaptability of learning methods in forecasting air quality. In summary, our study outcomes did not contribute to conversations about predicting air quality. Also offers practical insights that can be valuable in devising strategies to reduce the adverse effects of air pollution in urban regions such as Dhaka. The impressive precision attained by our models establishes them as tools for policymakers and environmental experts.

Data Availability Statement

In this study, publicly available datasets were analysed, which can be found on
https://www.kaggle.com/datasets/shawkatsujon/dhaka-bangladesh-hourly-air-quality-20162022,
[accessed on September 02, 2023].

Funding

No external funding was received for this research.

Conflicts of Interest

There is no conflict of interest in this research.

Human and Animal Rights and Informed Consent

The research does not involve any human or animal as a subject of study.

References

1. X. Xu, M. Yoneda, "Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model," IEEE Transactions on Cybernetics, vol. 51, no. 5, pp. 2577-2586, 2021.

2. S. Sonawani, K. Patil, P. Chumchu, "NO2 Pollutant Concentration Forecasting for Air Quality Monitoring by Using an Optimized Deep Learning Bidirectional GRU Model," International Journal of Computational Science and Engineering, vol. 24, no. 1, pp. 64-73, 2021.

3. F. Hamami, I. A. Dahlan, "Univariate Time Series Data Forecasting of Air Pollution using LSTM Neural Network," in International Conference on Advancement in Data Science, E-learning and Information Systems (ICADEIS), Lombok, Indonesia, pp. 1-5, 2020.

4. Y. S. Chang, H. T. Chiao, S. Abimannan, Y. P. Huang, Y. T. Tsai, and K. M. Lin, "An LSTM-based Aggregated Model for Air Pollution Forecasting," Atmospheric Pollution Research, vol. 11, no. 8, pp. 1451-1463, ISSN 1309-1042, 2020.

5. B. S. Freeman, G. Taylor, B. Gharabaghi, J. Thé, "Forecasting Air Quality Time Series Using Deep Learning," Atmospheric Environment, vol. 175, pp. 104-112, 2018.

6. R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, F. Li, "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering," School of Geography and Planning, Sun Yat-Sen University, Guangzhou, Guangdong 510275, China, 2021.

7. Y.-T. Tsai, Y.-R. Zeng, Y.-S. Chang, "Air Pollution Forecasting Using RNN with LSTM," Journal of the Air & Waste Management Association, vol. 68, no. 6, pp. 558-573, 2018.

8. W. Zhang et al., "Exploring urban dynamics based on pervasive sensing: correlation analysis of traffic density and air quality," in 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2012.

9. J. K. Sethi, M. Mittal, "An efficient correlation based adaptive LASSO regression method for air quality index prediction," Earth Science Informatics, vol. 14, no. 4, pp. 1777-1786, 2021.

10. A. Z. Ul-Saufie et al., "Improving air pollution prediction modelling using wrapper feature selection," Sustainability, vol. 14, no. 18, 2022.

11. D. Balram, K.-Y. Lian, N. Sebastian, "Air quality warning system based on a localized PM2. 5 soft sensor using a novel approach of Bayesian regularized neural network via forward feature selection," Ecotoxicology and Environmental Safety, vol. 182, 2019.

12. S. Ketu, "Spatial Air Quality Index and Air Pollutant Concentration prediction using Linear Regression based Recursive Feature Elimination with Random Forest Regression (RFERF): a case study in India," Natural Hazards, vol. 114, no. 2, 2022.

13. K. Siwek, S. Osowski, "Data mining methods for prediction of air pollution," International Journal of Applied Mathematics and Computer Science, vol. 26, no. 2, pp. 347-359, 2016.

14. A. Gibiansky, "Convolutional Neural Networks," Andrew's Notes, Feb.2014. [Online]. Available: https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/.[Accessed: 02-Oct-2023].

15. R. Hassan, M. Rahman, and A. Hamdan, "Assessment of air quality index (AQI) in Riyadh, Saudi Arabia," IOP Conference Series: Earth and Environmental Science, vol. 1026, no. 1, IOP Publishing, 2022.

16. H. A. Haque et al., "Ambient air quality scenario in and around Dhaka city of Bangladesh." Barishal University Journal, Part-1, vol. 4, no. 1, pp. 203-18, 2017.

17. A. Aggarwal, D. Toshniwal, "A hybrid deep learning framework for urban air quality forecasting." Journal of Cleaner Production, vol. 329, 2021, p. 129660.

18. N. Elmrabit et al., "Evaluation of Machine Learning Algorithms for Anomaly Detection." in 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, 2020, pp. 1-8.

19. T. Can, K. Krishnamurthy, D. J. Schwab, "Proceedings of The First Mathematical and Scientific Machine Learning Conference, PMLR 107:476-511, 2020."

20. S. Kumari, M. K. Jain, "A Critical Review on Air Quality Index," in Environmental Pollution, V. Singh, S. Yadav, R. Yadava (Eds.), vol. 77, Springer, Singapore, 2018, pp. 1-29.

21. R. Navares, J. L. Aznarte, "Predicting air quality with deep learning LSTM: Towards comprehensive models," Ecological Informatics, vol. 55, 2020, p. 101019.

22. A. Chakma et al., "Image-based air quality analysis using deep convolutional neural network," in 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 3949-3952.

23. P. B. Weerakody, K. W. Wong, G. Wang, W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," Neurocomputing, vol. 441, 2021, pp. 161-178.

24. Zishan Ahmed, Shakib Sadat Shanto, Akinul Islam Jony. "Advancement in Bangla Sentiment Analysis: A Comparative Study of Transformer-Based and Transfer Learning Models for E-commerce Sentiment Classification", Journal of Information Systems Engineering and Business Intelligence, vol. 9, no. 2, 2023, pp. 181-194.

25. Kazi Tanvir, Akinul Islam Jony, Mohamed Kaisarul Haq, Farzana Nazera, Mahaaganapathy Dass, Valliappan Raju. "Clinical Insights through Xception: A Multiclass Classification of Ocular Pathologies", Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 4, 2023, pp. 5876-5885.