Volume 7 Number 1 (2025) E-ISSN: 2808-1390
January – Juni 2025
Page: 295-310 DOI: 10.37680/jcd.v7i1.6953

JCD
Journal Of Community Development and Disaster Management

LSTM-Based PM2.5 Prediction Enhanced by Polynomial Features: Case Study in South Tangerang

Wulan Kusuma Wardani[1]

[1]Institut Teknologi Sumatera, Indonesia
*wulan.wardani@tse.itera.ac.id

Article history Submitted: 2025/01/17; Revised: 2025/02/07; Accepted: 2025/02/21

Abstract

The significant impact of air pollution, particularly PM2.5, has driven mitigation efforts to reduce health and environmental risks through more accurate prediction systems. In this study, a Deep Learning approach using the LSTM method with the addition of Polynomial transformation features is proposed to predict PM2.5 concentrations. Historical PM2.5 data from South Tangerang City, Banten, was used to train and test the model. The results show that LSTM with polynomial features effectively captures temporal and non-linear patterns in the data, producing accurate and consistent predictions for both training and testing data compared to conventional machine learning methods such as XGBoost and SVR. Polynomial feature transformation significantly improved model performance, as evidenced by the reduction in prediction errors and increased accuracy compared to LSTM without polynomial features. The model also demonstrates adaptability to sudden fluctuations in air quality data. Although the prediction results closely align with actual values, slight discrepancies may arise due to external factors or model limitations. Therefore, the LSTM approach with polynomial feature transformation is an effective and promising method for PM2.5 prediction.

Keywords LSTM; PM2.5; Polynomial Transformation; Prediction;

1. INTRODUCTION

The global transition to renewable energy sources, particularly solar and wind power, plays a crucial role in combating climate change and reducing greenhouse gas emissions. However, the efficiency of these renewable energy systems can be significantly affected by environmental factors, notably air pollution. Particulate matter (PM), such as PM2.5, is one of the most significant pollutants due to its adverse effects on human health and the environment. Particulate matter consists of solid and liquid particles dispersed in the air, originating from a variety of sources, including fossil fuel combustion, motor vehicle emissions, road dust, and industrial and agricultural activities (Thangavel et al., 2022). The levels of PM, usually measured in micrograms per cubic meter ($\mu g/m^3$), are tracked by monitoring networks to evaluate ambient air quality (Iwaszenko et al., 2024). The smaller the size of particulate matter, the more toxic it can be, as it can penetrate deep into the lower lobes of the lung (Wang et al., 2021). Particulate matter (PM) can accumulate on solar panels, leading to reduced energy transmittance and efficiency. Similarly, airborne particles can diminish air quality, impacting the performance of wind turbines. Research (Shariah & Al-Ibrahim, 2023) has shown a correlation between the reduction in power output by photovoltaic modules and the increasing thickness of dust deposits. Dust can absorb and scatter photons, thereby reducing the amount of light reaching the solar cells. Additionally, dust accumulation can affect the performance of wind turbines and increase maintenance loads and operational costs, especially during high-temperature periods (Al-Khayat et al., 2021).

Due to the widespread and hazardous impact of particulate matter, the ability to predict accurately is essential. Reliable predictions can provide early warning and support future air pollution control policies. By utilizing previous data, particle concentrations are predicted by considering the possibility of similar patterns appearing in the future (Drewil & Al-Bahadili, 2022). Previous studies have explored various approaches to predict PM concentrations, with many utilizing machine learning techniques (Morapedi & Obagbuwa, 2023; Brokamp et al., 2018; Doreswamy et al., 2020). Several other studies have examined the prediction of particulate matter (PM) concentrations using deep learning, demonstrating its advantages in handling complex and non-linear data, particularly for time-series and spatial data (Sharma et al., 2025; Rad et al., 2025; Zhang et al., 2024). As a subset of machine learning, deep learning utilizes multi-layered neural networks to model complex data. A neural network should have three layers called input layer, hidden layer, and output layer, and neurons are the core entities of this neural network. One of the popular methods

for handling time series data over a long period is Long Short-Term Memory (LSTM) (Priyanka et al., 2021). Initially, LSTM was introduced as a development of Recurrent Neural Networks to overcome long-term dependency problems (Hochreiter & Schmidhuber, 1997). Issues like analyzing particulate matter concentrations, which often exhibit sequential dependencies, are particularly well-suited for the inherent memory capabilities of LSTM (Ayturan et al., 2018).

However, even though LSTM models are successful, their predictive performance is highly dependent on the quality and relevance of the input features. Feature engineering plays a critical role in improving the ability of these models to capture underlying patterns in the target data (Kurniawan et al., 2024). Feature engineering refers to the process of extracting, selecting and transforming features from raw data to create more representative and predictive input variables (He, 2024). Feature engineering consists of feature selection, feature extraction, feature construction, and feature scaling. Feature construction is carried out as an effort to create new features from existing raw data with the aim of enhancing the representation of information that can be utilized by the model, one of which is interaction features. Polynomial Features is one of the interaction feature techniques that generates new features by raising the power of input variables, for example, transforming x into x^2, x^3, and so on. These properties enable the employment of more straightforward modeling techniques as part of data preparation, as some of the complexity of interpreting the input variables and their relationships (Brownlee, 2020). The integration of LSTM and polynomial features is expected to enable the model to leverage the strengths of both approaches. LSTM can learn complex temporal patterns from time-series data, while polynomial feature engineering can enrich data representation by adding non-linear features. The combined use of LSTM and polynomial features is anticipated to optimize the capture of linear relationships among input variables, thereby producing accurate and reliable predictions.

Therefore, this study aims to explore the effectiveness of using polynomial feature engineering in improving the accuracy of PM2.5 predictions using an LSTM model, with a case study in South Tangerang City. As a buffer zone for the capital city of Jakarta, Indonesia, South Tangerang is an area with high population density, industrial zones, and rapid commercial growth. This has triggered environmental issues in the city, including air pollution, particularly particulate matter. Accurate and reliable predictions are crucial for developing effective air quality management strategies and providing valuable insights for the implementation and maintenance of renewable energy systems, especially for this region. By combining LSTM and

polynomial feature engineering techniques, the model is expected to better capture complex patterns in air pollution data collected from South Tangerang City.

2. Research Methodology

2.1. Study area and datasets

This study uses data from South Tangerang City, located in Banten Province, Indonesia. This city is a rapidly growing urban area, with a growing population and various industrial, transportation, and residential activities that affect air quality. As one of the fastest growing cities in the Jabodetabek area, South Tangerang faces major challenges in controlling air pollution, even the city is recorded as the defending champion as the most polluted city in Indonesia (*Mengintip Juara Bertahan Polusi Udara Tangsel*, *Buruk Di Tengah Malam*, 2024). Based on IQAir observations on January 31, 2025 at 15:00 WIB, the main pollutant in this city is PM2.5 at 21 $\mu g/m^3$. This concentration is 4.2 times higher than the normal limit of PM2.5 by WHO. This highlights the urgent need for effective air quality management strategies in the region.

This study uses two main data, namely air pollution data, which includes PM2.5 target data and meteorological data. Both datasets are critical for understanding the relationship between atmospheric conditions and air pollution levels in South Tangerang. Daily air pollution data were obtained from the Kaggle dataset for the period 2020-2022, providing a comprehensive picture of air pollution fluctuations, including PM2.5 in South Tangerang City during that period. Meanwhile, meteorological data were obtained from the BMKG (Meteorology, Climatology, and Geophysics Agency) Online Database Center, precisely taken from the Banten Climatology Station. The meteorological data used include daily data on temperature, humidity, rainfall, wind speed, atmospheric pressure, rainfall, and sunshine duration. These variables are essential for understanding how atmospheric conditions effect the dispersion, accumulation, and chemical transformation of air pollutants (Rodríguez-Sánchez et al., 2024). For example, wind speed and direction can affect the transport of pollutants, while rainfall can contribute to the removal of particulate matter from the atmosphere through wet deposition. These two main data are used in PM2.5 prediction to improve model quality by providing more relevant information. Description of the datasets used related to naming during modeling and the units used are described in table 1.

Table 1. Desciription of datasets

Parameter	Dataset Name	Unit
Particulate Matter 2.5	PM2.5	$\mu g/m^3$
Particulate Matter 10	PM10	$\mu g/m^3$
Sulfur Dioxide	SO_2	$\mu g/m^3$
Carbon Monoxide	CO	$\mu g/m^3$
Sodium Dioxide	NO_2	$\mu g/m^3$
Average temperature	Temp_avg	°C
Relative humidity	Humidity	%
Precipitation	Prec	Mm
Sunshine duration	Sunshine	hour
Average wind speed	Wind_avg	m/s

2.2 Preprocessing

Preprocessing data is carried out to ensure that the data used is clean and consistent by applying various transformations, making it ready to be processed by the model (Kanellopoulos & Pintelas, 2006). The first step is to identify and handle any data points in a dataset that are clearly problematic in terms of measurement (outliers or missing values) using interpolation methods, which involve estimating and filling in problematic values based on surrounding data. This method is effective for handling time series data as it considers temporal patterns. Additionally, incomplete or noisy data is also cleaned to avoid bias in model training. Secondly, polynomial feature transformation is applied to capture non-linear correlations between features by expanding the feature space through the combination of original numerical characteristics in polynomial way (Parvathi et al., 2024). This step helps enhance the model's flexibility in learning complex patterns in the data. This technique is performed by generating new features by increasing the original features to various powers up to a specified degree (n) and creating combinations of these powers. As a result, new features are created, including x1, x2, x3, ..., xn. These features are then combined with all possible combinations (interaction features), such as the multiplication of x^2 and x^3, or other combinations (Parvathi et al., 2024). The number of polynomial feature combinations grows exponentially as the degree n increases. This means that the higher the polynomial degree chosen, the newer features are generated. In this study, n is set to 3, meaning polynomial features up to the third degree are created from the existing features.

The next crucial step in preprocessing is to standardize the data scale using normalization methods such as StandardScaler. Normalization is essential because the

features in a dataset often have different scales. Data normalization involves transforming numerical data into new data with smaller values and a predefined range (Alshdaifat, 2020). The StandardScaler is implemented from z-score normalization by transforming each feature to have a mean of 0 and a standard deviation of 1 using the following equation:

$$y_i = z = \frac{y_i - \bar{y}_i}{\sigma} \quad (1)$$

Where $\bar{y}_i$ and σ representing the mean and standard deviation of each data point (Mohamad & Usman, 2013). This normalization step is crucial to ensure that all features have a uniform scale, so no feature dominates the model's learning process due to significant scale differences. Thus, the data that has gone through the preprocessing stage is ready to be processed by the LSTM model to predict PM2.5 concentrations. The dataset summary of each parameter is presented in Table 2.

Table 2. Dataset summary

Parameter	Min	1st Quartile	Median	mean	3st Quartile	Max
PM2.5	19.0	35.0	45.0	44.2	53.0	66.0
PM10	3.0	13.0	17.0	18.7	23.0	60.0
SO_2	0.0	1.0	14.0	10.6	16.0	28.0
CO	0.0	10.0	12.0	18.8	21.0	164.0
NO_2	0.0	0.0	3.0	2.4	4.0	8.0
Temp_avg	24.6	27.1	28.0	27.9	28.7	32.5
Humidity	53.0	77.0	82.0	81.2	86.0	98.0
Prec	0.0	0.6	4.4	11.4	15.8	208.9
Sunshine	0.0	2.5	4.8	4.7	6.8	12.5
Wind_avg	0.0	1.0	1.0	1.5	2.0	4.0

2.3 LSTM Implementation

Long Short-Term Memory (LSTM) is an advancement of Recurrent Neural Networks (RNN) designed to address the vanishing gradient problem caused by its limitations in capturing complex temporal relationships, especially those involving long-term dependencies (Hochreiter & Schmidhuber, 1997). RNN itself is a type of artificial neural network that has the ability to retain information from previous steps through a feedback loop mechanism (Grossberg, 2013). The concept behind LSTM is its ability to maintain memory state over long periods of time due to the inclusion of memory cells and gating mechanisms. This memory state incorporates gates that control the flow of data within the cell. Each LSTM cell contains this memory state,

which adjusts the information from previous states based on the current input and the gates' operations (Drewil & Al-Bahadili, 2022). In order to capture long-term dependencies, LSTM uses a three-layer method (input layer, forget layer, and output layer) that regulates information flow and permits modifications to the cell state vector, which is transmitted iteratively (Lindemann et al., 2021).

As shown in Figure 1, at each time step t, the data flow starts with the input X_t (black circle) and the previous hidden state h_{t-1} (orange circle) into the LSTM cell, along with the previous cell state C_{t-1} (light blue circle). The data is first processed by the Forget Gate, which uses a sigmoid function (σ) to determine which part of the previous cell state should be forgotten or retained, with values between 0 (forgotten) and 1 (retained). Next, the data is processed by the Input Gate, which also consists of sigmoid and tanh functions; the sigmoid determines which elements will be updated, while tanh generates the candidate new value C_t that will be added to the cell state. After that, the cell state is updated with a combination of the retained information from the old cell state and the new information deemed relevant, according to the formula $C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t$. After the cell state update, the Output Gate determines the new hidden state h_t (red circle) using the sigmoid to select the portion of the cell state to be output, and then the result is multiplied by the normalized cell state value through the tanh function. The hidden state h_t becomes the output of the LSTM cell and will be used for the next time step or as input to the next layer in the network.

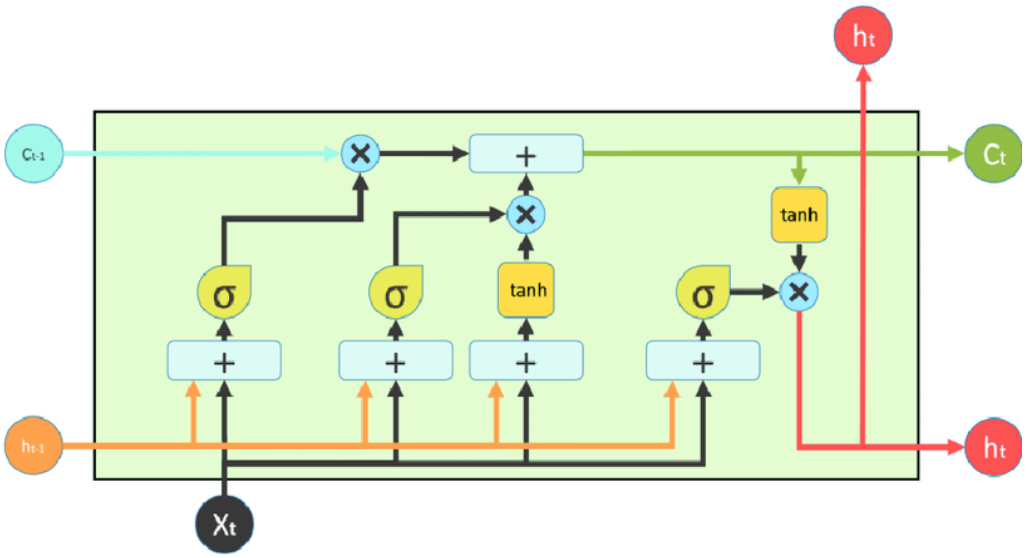


Figure 1. Architecture of LSTM (Guo et al., 2020)

In this study, the Long Short-Term Memory (LSTM) model was implemented using the TensorFlow/Keras deep learning library with Python. Polynomial Features transformation was applied up to the 4th order to capture non-linear interactions

between variables, followed by normalization using StandardScaler. The data was split into training data (70%) and testing data (30%) using the *train_test_split* method without shuffling (*shuffle=False*) to preserve the temporal order of the data. To handle time series data, the TimeseriesGenerator is used to structure the data with a lookback length of 7, meaning the model utilizes the past 7 days to predict the 8th day. The batch size is set to 35 to determine the number of samples processed before updating the model weights.

The developed LSTM model consists of:

1. A first LSTM layer with 150 units, using ReLU activation and L1 regularization (0.0005) to reduce model complexity and prevent overfitting.
2. A dropout layer (15%) after the LSTM to enhance model generalization.
3. A Dense layer (10 neurons) with linear activation, serving as an additional layer before the final output.
4. A dropout layer (10%) after the first Dense layer.
5. A Dense layer (1 neuron) as the output layer with linear activation to continuously predict PM2.5 values.

The model was compiled using the *RMSprop* optimizer with a learning rate of 0.0005 and the Mean Squared Error (MSE) loss function. During training, two callbacks were used:

- *EarlyStopping* to stop training if val_loss does not improve after 10 epochs.
- *ReduceLROnPlateau* to reduce the learning rate by 50% if val_loss does not improve after 8 epochs, with a minimum learning rate of 0.0001.

The model was trained for a maximum of 300 epochs with validation using test data. After training was completed, the model was evaluated using R^2 (Coefficient of Determination), RMSE (Root Mean Squared Error), and MAPE (Mean Absolute Percentage Error) metrics to measure prediction accuracy on both training and test data. The mathematical equations for each evaluation metric are explained below (Chicco et al., 2021):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_i^*)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - y_i^*)^2}{N}}, \tag{3}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - y_i^*}{y_i}\right|, \tag{4}$$

Where y_i as actual values; y_i^* as predicted values; $\bar{y}$ as mean of the actual values; N as number of data points. The R^2 is the Pearson correlation between the predicted and actual PM2.5 concentration, and it represents how close predicted values to the actual

PM2.5 concentration. MAPE indicates the magnitude of the prediction error relative to the actual value in percentage. Meanwhile, RMSE provides a clear picture of the model's error in the original units. The effectiveness and validity of the proposed LSTM model are also evaluated by comparing its performance with several machine learning methods, including XGBoost (Extreme Gradient Boosting) and SVR (Support Vector Machine).

3. RESULTS AND DISCUSSION

3.1. Prediction Accuracy

Table 3 shows the evaluation metric results from the LSTM, XGBoost, and SVR methods, both on the training and testing data. Overall, the LSTM model with polynomial feature transformation provides the best performance compared to XGBoost, LSTM without polynomial features, and SVR. The LSTM with polynomial feature transformation not only superior in prediction accuracy but also show good consistency between the training and testing data.

Table 3. Evaluation Metric Results of Models

Methods	Training Dataset			Testing Dataset		
	RMSE	R^2	MAPE	RMSE	R^2	MAPE
LSTM with Polynomial Features	2.43	0.95	4.32%	2.86	0.90	5.86%
XGBoost	2.88	0.93	5.23%	3.13	0.88	6.55
LSTM saja	4.01	0.87	7.58%	3.26	0.87	6.64%
SVR	4.62	0.83	7.37%	3.67	0.83	7.21%

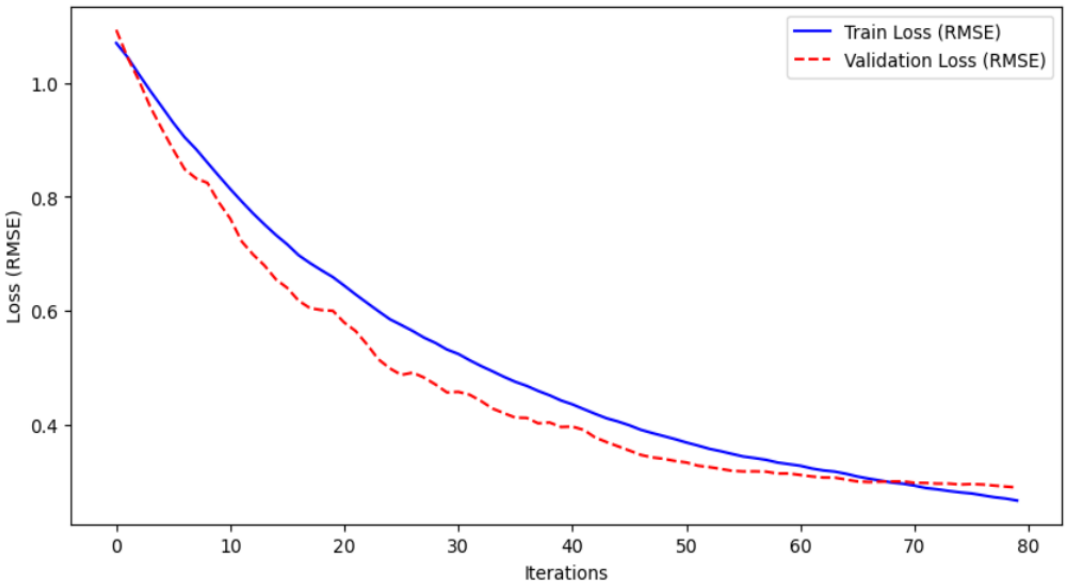
Lower prediction errors and higher correlation with the actual values indicate that the additional features from the polynomial transformation contribute positively to the LSTM learning process. This is clearly seen when comparing the evaluation results of the LSTM model with polynomial features and the LSTM without these features, LSTM model without feature transformation shows a significant performance decline. The decline in performance of the LSTM model without polynomial features indicates that polynomial feature transformation plays a crucial role in improving prediction accuracy. This transformation allows the model to better capture non-linear relationships in the PM2.5 data, which might be difficult for the LSTM to learn directly. By expanding the features before processing them into the model, the LSTM can more effectively learn the complex patterns present in the data.

Meanwhile, XGBoost shows fairly good performance, although slightly below LSTM with polynomial features. This model can explain most of the data variation, but there is increasing prediction error when tested on the testing data. Aalthough strong in handling tabular data, XGBoost is less optimal in capturing the temporal pattern complexities present in the PM2.5 data. On the other hand, SVR ranks last in terms of performance. Although it can still provide good predictions, higher prediction errors and lower accuracy values indicate that SVR is less suitable for time series prediction of PM2.5 compared to previous methods.

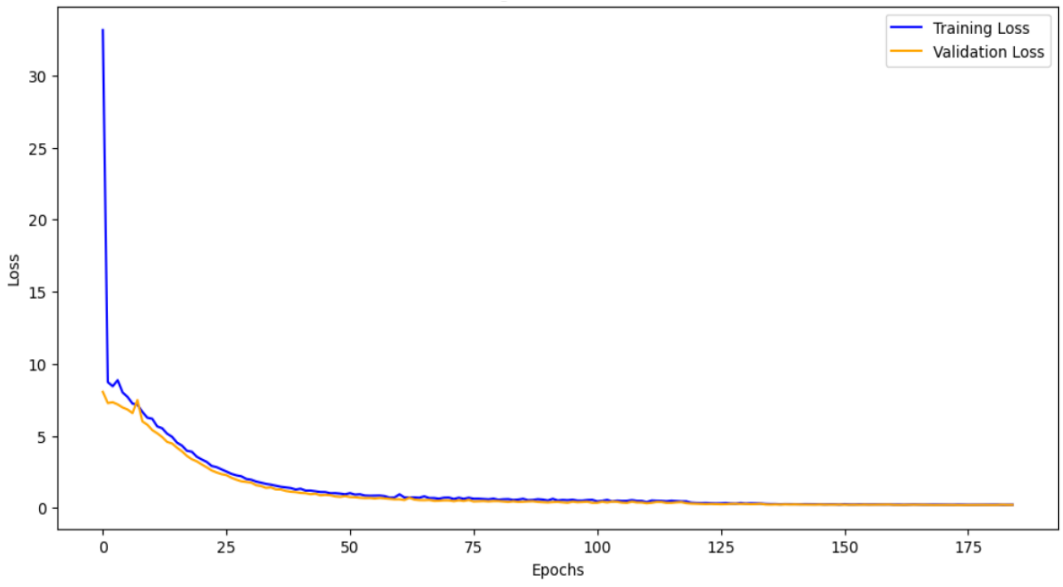
The success of LSTM with polynomial features in handling time series data makes it a more optimal choice for PM2.5 prediction compared to conventional machine learning methods like XGBoost and SVR. However, further exploration is still needed, such as hyperparameter optimization or the use of other techniques like the attention mechanism, to further improve the model's performance.

3.1. Model Performance During Training

Figure 2. (a) Training loss and validation loss curves of XGBoost models (b) training



(a)



(b)

loss and validation loss curves of LSTM with polynomial features

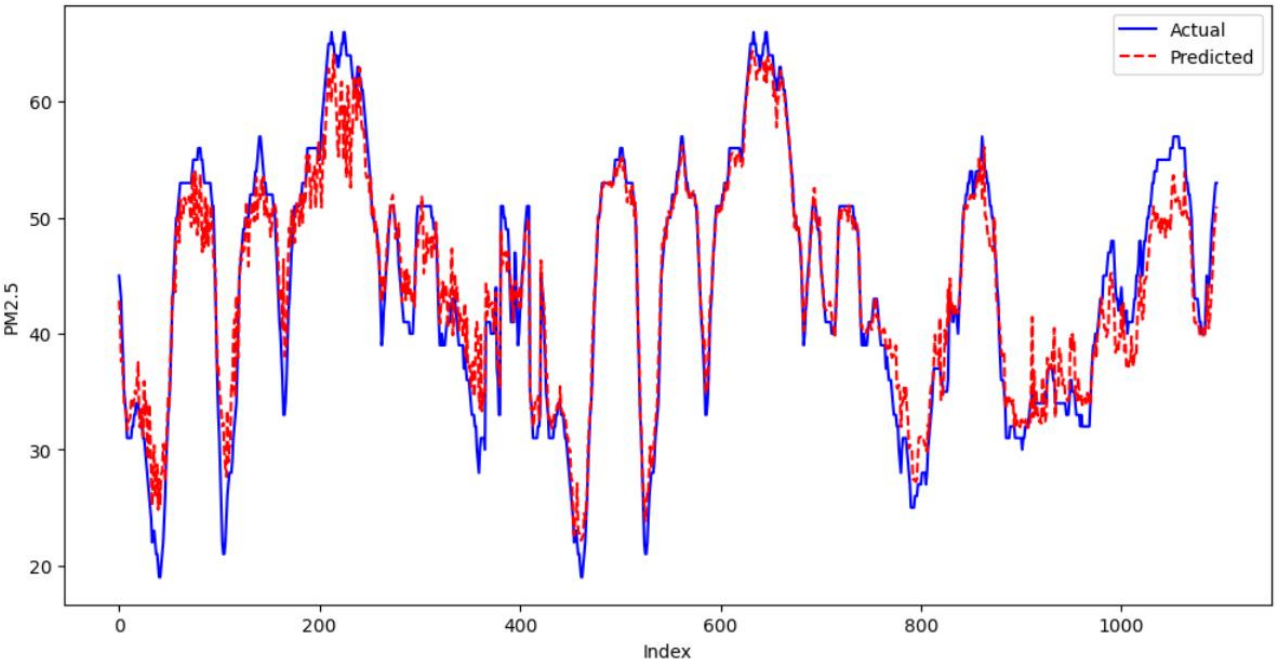
Figure 2 shows the training loss and validation loss curves during the model training process. Graph 2(a) illustrates the change in Root Mean Squared Error (RMSE) to the number of iterations in the XGBoost model. In the XGBoost model, both the training loss and validation loss decrease steadily as the number of iterations increases. However, the difference between the two losses remains consistent, indicating that the model does not suffer significant overfitting. That's XGBoost has effectively learned the patterns from the data while maintaining its performance on unseen data.

Meanwhile, the convergence pattern in the LSTM model with polynomial features

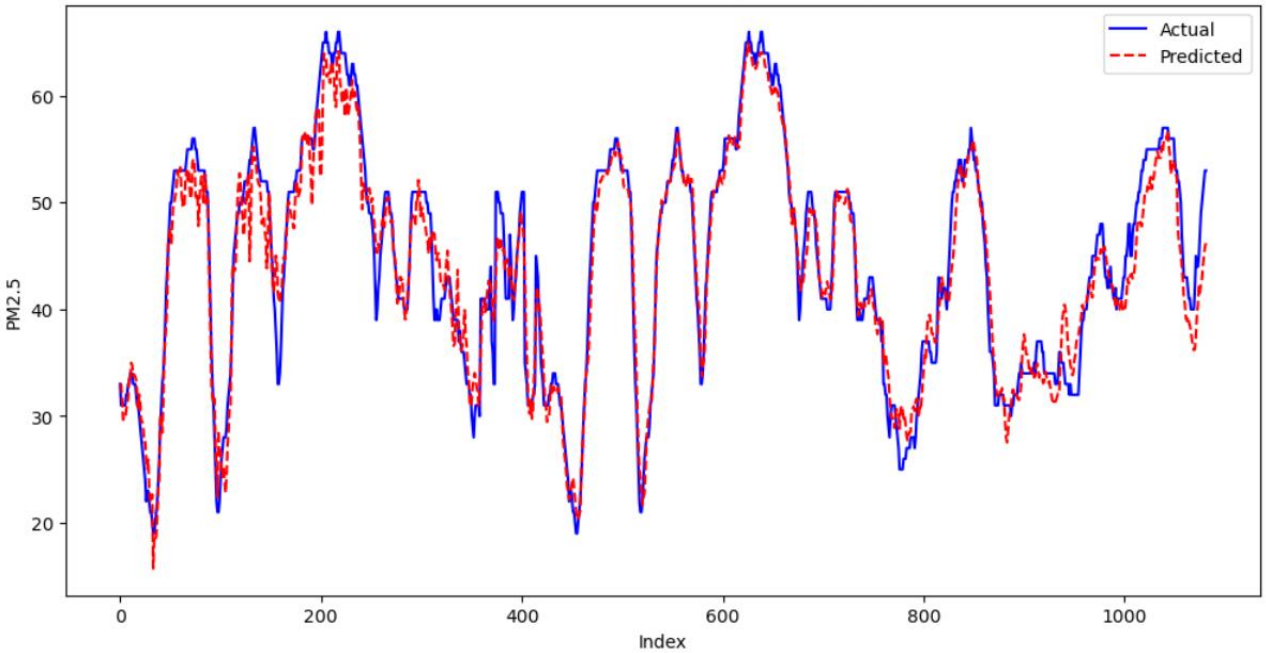
shows a sharp decline at the beginning of training before stabilizing close to zero as the epochs increase. The graph illustrates that the training loss and validation loss move in parallel without significant gaps, indicating that the model does not experience substantial overfitting or underfitting. In other words, the LSTM model successfully adjusts its weights optimally and able to generalize well to the validation data. From the two graphs shown in Figure 2, the LSTM model with polynomial features demonstrates a smoother and more stable long-term convergence pattern, which could indicate that this model is better at capturing complex patterns in PM2.5 data compared to XGBoost.

Figure 3 shows a comparison between the actual PM2.5 values (blue line) and the predicted values (red dashed line) generated by the XGBoost and LSTM with polynomial features models. It can be observed that both models perform quite well in following the data trends, but there are differences in the level of accuracy and responsiveness to data changes. The XGBoost model tends to provide more stable and smooth predictions. Although it can capture the main patterns of the PM2.5 data, there are some larger deviations during sharp changes in values. This model appears slower in responding to significant spikes and drops, which may indicate that XGBoost has limitations in understanding complex patterns in the data.

Meanwhile, the LSTM model with polynomial features demonstrated better performance in capturing more complex fluctuations. From Figure 3.b, the predictions of this model are closer to the actual values compared to XGBoost ini Figure 3.a, especially in areas with sharp spikes or drops. This capability indicates that LSTM is more adaptive to non-linear patterns in the data. However, its high sensitivity to data variations can also increase the risk of overfitting, especially if the model adapts too much to the patterns in the training data. So overall, the LSTM model with polynomial features outperforms XGBoost in predicting PM2.5, especially in capturing rapid and complex changes in the data. Nevertheless, the choice of model still depends on specific needs, where XGBoost may be a more efficient option if prioritizing speed and prediction stability, while LSTM is more suitable for scenarios requiring deeper modeling of non-linear patterns.



(a)



(b)

Figure 3. (a) Actual vs Predicted PM2.5 concentration by XGBoost model (b) Actual vs Predicted PM2.5 concentration by LSTM with polynomial features

4. CONCLUSION

In this study, LSTM model with polynomial features was applied to predict PM2.5 concentration based on historical data collected from South Tangerang City,

Banten. The results indicated that this approach is capable of capturing complex patterns in the data, including trends and fluctuations in PM2.5 values. The model demonstrated a strong ability to follow the movement of actual values, with predictions closely matching observational data. Additionally, it shows good consistency between training and testing data. This implies that the polynomial feature transformation is essential for improving data representation, especially when it comes to capturing intricate non-linear correlations, while the LSTM is successful in identifying temporal patterns in the data.

A significant decline in the performance of the LSTM model without polynomial feature transformation indicated that polynomial feature expansion greatly contributed to the improvement of prediction accuracy. This transformation allowed the model to better learn the characteristics of data that may be difficult for LSTM to capture, both in short-term and long-term variations in PM2.5 patterns. Additionally, this approach enabled the model to be more adaptive to sudden changes, which frequently occured in air quality data. Thus, the combination of LSTM and polynomial feature transformation has been proven effective in enhancing the predictive ability of the model.

Although the prediction results showed a high degree of alignment with the actual data, there are still some minor discrepancies that may be caused by external factors, that not included in the input features or limitations of the model in capturing certain patterns. Therefore, further development can be pursued through exploration of LSTM architecture optimization techniques, hyperparameter tuning, or the integration of additional features that are more representative of PM2.5 dynamics.

REFERENCES

Al-Khayat, M., Al-Rasheedi, M., Gueymard, C. A., Haupt, S. E., Kosović, B., Al-Qattan, A., & Lee, J. A. (2021). Performance analysis of a 10-MW wind farm in a hot and dusty desert environment. Part 2: Combined dust and high-temperature effects on the operation of wind turbines. *Sustainable Energy Technologies and Assessments*, *47*. https://doi.org/10.1016/j.seta.2021.101461

Alshdaifat, E. (2020). The Impact of Data Normalization on Predicting Student Performance: A Case Study from Hashemite University. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(4), 4580–4588. https://doi.org/10.30534/ijatcse/2020/57942020

Ayturan, Y. A., Ayturan, Z. C., & Altun, H. O. (2018). Air Pollution Modelling with Deep Learning: A Review. In *Int. J. of Environmental Pollution & Environmental Modelling* (Vol. 1, Issue 3).

Brokamp, C., Jandarov, R., Hossain, M., & Ryan, P. (2018). Predicting Daily Urban Fine Particulate Matter Concentrations Using a Random Forest Model. *Environmental Science and Technology*, *52*(7), 4173–4179. https://doi.org/10.1021/acs.est.7b05381

Brownlee, J. (2020, August 28). *How to Use Polynomial Feature Transforms for Machine Learning*. Machine Learning Mastery.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, 1–24. https://doi.org/10.7717/PEERJ-CS.623

Doreswamy, Harishkumar, K. S., Km, Y., & Gad, I. (2020). Forecasting Air Pollution Particulate Matter (PM2.5) Using Machine Learning Regression Models. *Procedia Computer Science*, *171*, 2057–2066. https://doi.org/10.1016/j.procs.2020.04.221

Drewil, G. I., & Al-Bahadili, R. J. (2022). Air pollution prediction using LSTM deep learning and metaheuristics algorithms. *Measurement: Sensors*, *24*. https://doi.org/10.1016/j.measen.2022.100546

Grossberg, S. (2013). Recurrent neural networks. In *Scholarpedia: Vol. 8(2)*.

Guo, Y., Cao, X., Liu, B., & Peng, K. (2020). El Nino index prediction using deep learning with ensemble empirical mode decomposition. *Symmetry*, *12*(6). https://doi.org/10.3390/SYM12060893

He, R. (2024). A Review of Feature Engineering Methods in Regression Problems. *Academic Journal of Natural Science Journal Home: Ajns.Suaspress.Org | CODEN: AJNSAE | NAAN*, *1*(1), 40704. https://doi.org/10.5281/zenodo.13905622

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Iwaszenko, S., Smolinski, A., Grzanka, M., & Skowronek, T. (2024). Airborne particulate matter measurement and prediction with machine learning techniques. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-70152-9

Kanellopoulos, D., & Pintelas, P. E. (2006). *Data Preprocessing for Supervised Learning*. https://www.researchgate.net/publication/228084519

Kurniawan, J. D., Parhusip, H. A., & Trihandaru, S. (2024). Predictive Performance Evaluation of ARIMA and Hybrid ARIMA-LSTM Models for Particulate Matter Concentration. *Jurnal Online Informatika*, *9*(2), 259–268. https://doi.org/10.15575/join.v9i2.1318

Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, *99*, 650–655. https://doi.org/10.1016/j.procir.2021.03.088

Mengintip Juara Bertahan Polusi Udara Tangsel, Buruk di Tengah Malam. (2024, July 31). CNN Indonesia.

Mohamad, I. Bin, & Usman, D. (2013). Standardization and its effects on K-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, *6*(17), 3299–3303. https://doi.org/10.19026/rjaset.6.3638

Morapedi, T. D., & Obagbuwa, I. C. (2023). Air pollution particulate matter (PM2.5)

prediction in South African cities using machine learning techniques. *Frontiers in Artificial Intelligence*, *6*. https://doi.org/10.3389/frai.2023.1230087

Parvathi, S. S. L., Devi, A. B., Kulkarni, G. L., Murugan, S., Vijayammal, B. K. P., & Neha. (2024). Exploring Feature Relationships in Brain Stroke Data Using Polynomial Feature Transformation and Linear Regression Modeling. *Journal of Machine and Computing*, *4*(4), 1158–1169. https://doi.org/10.53759/7669/jmc202404107

Priyanka, Kumari, A., & Sood, M. (2021). Implementation of SimpleRNN and LSTMs based prediction model for coronavirus disease (Covid-19). *IOP Conference Series: Materials Science and Engineering*, *1022*(1). https://doi.org/10.1088/1757-899X/1022/1/012015

Rad, A. K., Nematollahi, M. J., Pak, A., & Mahmoudi, M. (2025). Predictive modeling of air quality in the Tehran megacity via deep learning techniques. *Scientific Reports*, *15*(1), 1367. https://doi.org/10.1038/s41598-024-84550-6

Rodríguez-Sánchez, A., Santiago, J. L., Vivanco, M. G., Sanchez, B., Rivas, E., Martilli, A., & Martín, F. (2024). How do meteorological conditions impact the effectiveness of various traffic measures on NOx concentrations in a real hot-spot? *Science of the Total Environment*, *954*. https://doi.org/10.1016/j.scitotenv.2024.176667

Shariah, A., & Al-Ibrahim, E. A. (2023). Impact of Dust and Shade on Solar Panel Efficiency and Development of a Simple Method for Measuring the Impact of Dust. *Journal of Sustainable Development of Energy, Water and Environment Systems*, *11*(2). https://doi.org/10.13044/j.sdewes.d11.0448

Sharma, L., Singh, H., & Choudhary, M. P. (2025). Application of deep learning techniques for analysis and prediction of particulate matter at Kota city, India. *EQA*, *66*, 107–115. https://doi.org/10.6092/issn.2281-4485/20687

Thangavel, P., Park, D., & Lee, Y. C. (2022). Recent Insights into Particulate Matter (PM2.5)-Mediated Toxicity in Humans: An Overview. In *International Journal of Environmental Research and Public Health* (Vol. 19, Issue 12). MDPI. https://doi.org/10.3390/ijerph19127511

Wang, X., Xu, Z., Su, H., Ho, H. C., Song, Y., Zheng, H., Hossain, M. Z., Khan, M. A., Bogale, D., Zhang, H., Wei, J., & Cheng, J. (2021). Ambient particulate matter (PM1, PM2.5, PM10) and childhood pneumonia: The smaller particle, the greater short-term impact? *Science of the Total Environment*, *772*. https://doi.org/10.1016/j.scitotenv.2021.145509

Zhang, Y., Sun, Q., Liu, J., & Petrosian, O. (2024). Long-Term Forecasting of Air Pollution Particulate Matter (PM2.5) and Analysis of Influencing Factors. *Sustainability (Switzerland)*, *16*(1). https://doi.org/10.3390/su16010019