ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Edge-to-Cloud Long Short-Term Memory Model for Ambient Carbon Monoxide Level Prediction

Alauddin Maulana Hirzan[1], April Firman Daru[2], Susanto Susanto[3] and Ahmad Rifa'i[4]

ABSTRACT

Carbon monoxide (CO) is a harmful gas from incomplete fuel combustion, often found in motor vehicle emissions. Prolonged exposure can cause serious health issues or death. While existing Internet-of-Things (IoT) systems monitor CO levels, most lack predictive capability. One prior study used an Artificial Neural Network with limited accuracy (79%). To address this, a new IoT-based CO prediction model is proposed using a Long Short-Term Memory (LSTM) algorithm. The model predicts future CO concentrations based on seasonal patterns, empowering users to anticipate and proactively respond to potential exposure. By leveraging Edge-to-Cloud architecture, this approach enables low-power edge devices to send data to the cloud for accurate forecasting without local model deployment. Based on the evaluation, the model achieved 98.42% accuracy, outperforming previous approaches by 19.42%. It also showed superior performance against other algorithms, with the lowest MAE (0.026305), MSE (0.016004), RMSE (0.126506), and the highest R^2 (0.997647). Evaluation with AIC and BIC confirmed its reliability, scoring zero after MinMax scaling. The model demonstrates a substantial advancement in predictive CO monitoring, giving users actionable insights to protect health and safety.

Article information:

Keywords: Carbon Monoxide, Internet of Things, Long Short-Term Memory, Predictions

Article history:
Received: July 30, 2025
Revised: November 6, 2025
Accepted: December 16, 2025
Published: January 3, 2026
(Online)

DOI: 10.37936/ecti-cit.2026201.263289

1. INTRODUCTION

Carbon monoxide (CO) is a poisonous gas generated by the partial combustion of fuels in motorcycles or cars. External factors that affect CO concentration include the type of fuel used (such as gasoline or diesel), the number of engines operating, the speed and density of traffic, and ambient temperature. Although CO can be produced outdoors, it can also accumulate indoors through activities such as smoking, the use of gas heaters, and running a motorcycle in a garage. Unlike many gases, CO is colourless and odourless[1], [2], yet it is dangerous if a person is exposed to a high concentration for a long time. Since Carbon Monoxide is easier to mix with *haemoglobin* into *carboxyhaemoglobin*, the risk of Carbon Monoxide poisoning is higher than Carbon Dioxide. The WHO carbon monoxide report states that lower and higher exposures impact human life. Low Carbon Monoxide exposure caused significant focus impairments. *Angina Pectoris* (or known as chest pain) may occur as well[3], [4]. At higher exposure, Carbon Monoxide caused death for animals during lab tests.[5], [6].

For these reasons, many studies focused on monitoring ambient CO levels in the air. In 2020, a study designed a low-cost atmospheric pollution station. According to the evaluation, the developed model highly correlates with a fixed station of $r = 0.92$ and $r = 0.91$[7]. Within the same year, another study focused on creating a gas detection and warning system for older adults. Compared to a previous study, the developed model can transmit data through the MQTT protocol and gives alerts with a buzzer, vibration, and text notification during critical situations[8]. This model was similar to the model designed in article[9]. In 2021, the development of the CO monitoring model continues. An article used the Internet of Things (IoT) and Long-Range Wireless Area Network (LoRaWAN) to monitor CO concentration in a coal mine. The designed model is capable of sending an alert to the coal supervisor when the CO concentration reaches a critical level and saving

[1,2,3,4] The authors are with the Faculty of Information and Communication Technology, Universitas Semarang, Indonesia, Email: maulanahirzan@usm.ac.id, firman@usm.ac.id, susanto@usm.ac.id and rifai@usm.ac.id
[2] Corresponding author: firman@usm.ac.id

miners' lives[10]. A different model also implemented an Artificial Neural Network to predict CO concentration. Based on the evaluation, the designed model had an accuracy of up to 79%[11]. A carbon monoxide monitoring model is still actively developed in 2022. Article[12] designed an indoor monitoring model to monitor CO concentration. Based on the evaluation, the designed model consumed only 0.321mWh, and the notification can be received through a smartphone or personal computer. An article[13] in 2023 used a different approach to designing a monitoring model. Unlike the model in previous years, which only used one sensor, the model in the article implemented two gas sensors, MQ-135 and MQ-2. However, the designed model can only alert the user through a smartphone for easier access. The most recent CO monitoring models were designed in 2024 and 2025. The model developed in 2024 was implemented to monitor CO pollution from diesel exhaust using an ESP32 microcontroller and an IoT platform[14]. Meanwhile, a model developed in 2025 combined MQ-7, DHT-11, and ESP8266 to monitor Carbon Monoxide wirelessly through the ThingSpeak platform. This designed model is capable of controlling ventilation based on the CO concentration level[15].

Although the designed models in the past successfully monitored the CO concentration level, they had several weaknesses. The first weakness is the implementation of the sensor. Only one study properly used MQ-7 as the primary sensor to detect ambient CO concentration. However, the MQ-7 model was unsuitable for battery or low-power models due to cyclic heating. Furthermore, the supporting sensor for CO is only DHT11 to detect temperature and humidity. The DHT11 sensor has a limited range to detect temperature compared to other sensors[16], [17]. The second weakness is that the article implements an artificial neural network (ANN) to predict CO concentration. This algorithm relies on the correlation between variables to increase its accuracy. Thus, the prediction would not be accurate if the independent variables have low correlation with the dependent variable[18], [19]. The article only implements ANN; the other studies do not implement any algorithm. Thus, most previous models cannot predict the ambient CO concentration.

To address these weaknesses in previous models, this study aims to design an ambient CO monitoring model capable of predicting concentration levels by implementing Long Short-Term Memory (LSTM) with an Edge-Cloud architecture. Compared to previous models, the proposed model utilises MQ-9 rather than MQ-7. MQ-9 is easier to implement and allows faster deployment compared to MQ-7[20], [21]. Besides that, this model also utilises BME-280 to measure temperature, humidity, and air pressure for supporting data. Although the supporting data will not be used for the LSTM model, these data will help create predictions for comparison in the evaluation phase. Compared to DHT-11, BME-280 offers pressure measurement and a wider measurement range[20], [22]. To address the problem with the machine learning algorithm implemented in the article[11]. This study will implement the Long Short-Term Memory (LSTM) algorithm to predict future CO concentrations using time series data. Since the LSTM algorithm requires significant computing resources and is not suitable for direct deployment on resource-limited embedded systems such as the ESP32, this study will employ an Edge-to-Cloud architecture. In this setup, the ESP32 device reads CO concentration levels from the environment and transmits the readings to a cloud server via REST API. The cloud server receives the data, runs the LSTM prediction model on the incoming data, and then stores both the original and predicted values for permanent retention.

The gaps between past and proposed models are methodological and evidence gaps. First, most previous models did not use any machine learning algorithm except one, so they could not predict outcomes. One model used an ANN, but its accuracy was low. This study, in contrast, will use LSTM, which predicts time series data accurately, so independent variables are not needed. Second, past models, including the one with ANN, did not train using time series data. This study will use the time series CO dataset for training, and the data type will impact the machine learning model's behavior.

The novelties of this study are implementing an Edge-Cloud architecture to produce LSTM predictions by combining an Edge device to collect environmental data and a Cloud to create predictions based on the received data[23], [24]. This study also implements an MQ-9 sensor as the primary sensor to collect ambient CO concentration in the air and a BME-280 sensor to collect supporting data for prediction comparison in the evaluation phase. Besides that, this study also utilises time series datasets for training and prediction instead of cross-sectional data.

This article is divided into different sections. The first section is the introduction, which explains the main problem, the state of the art (the most recent and advanced developments in the field), the problem with the current model, and the purpose of the study. Next, the method used by the study to ensure reproducibility is described, making it possible for future studies to repeat the process and achieve the same results. This is followed by the presentation and discussion of the proposed model's results. Finally, the overall study concludes with a summary.

2. METHOD

This section explains the research design, tools or software, data gathering process, LSTM model creation, and analysis. Each subsection is important to ensuring model and result reproducibility in future studies. The first subsection is the research design, which is the foundation of this study. The required tools or software subsection explains how to write a model or gather the data. Then, the data gathering process is used to collect the required training data from the environment. This study explains the training process in the LSTM model creation subsection, finding the correct parameters. Then, the last subsection is the analysis, in which this study evaluates the model's performance.

2.1 Research Design

This subsection explains the research design used as the foundation of this study. This study used an experimental design. The first step was to find the problem in an environmental issue. Environmental issues are common in urban areas, including Semarang city, Indonesia. Many vehicles were on the road and produced many different gases. Since this occurrence happened almost daily, it raised concerns about environmental issues. To narrow the problem, this study only focused on the environmental issues in Universitas Semarang since the number of students riding motorcycles is high during class. This study observed and found environmental problems in Universitas Semarang. The next step was to start the data gathering process and experimentation with the LSTM model and Internet of Things prototype. This study evaluated them later in the evaluation step.

2.2 Required Software and Tools

This subsection explains the tools and software required to gather environmental data for the proposed model. This study breaks down the required items into two categories: the IoT prototype, which collects environmental data, and the LSTM model, which processes this data for analysis. To gather the required data, this study must first build the IoT prototype. This step can be done easily by using an ESP32 development board as the central processing board to collect environmental data through sensors. This development board is connected with two sensors: the MQ-9 sensor to collect ambient CO concentration in the air and the BME-280 to collect supporting data like Temperature, Humidity, and Pressure. This prototype connects to a wireless access point to send the collected data using the MQTT protocol. To support longer operation, the prototype is powered by a 20,000mAh battery. This study used a private MQTT broker to prevent high traffic or data mixed with public data. Figure 1 illustrates the block diagram of the data gathering prototype.

Figure 1 illustrates the connection between the ESP32 development board, sensors, battery, internet access point, and database. According to Figure 1, the ESP32's analog input (A0 pin) is connected to the analog output of the MQ-9 sensor, from which it

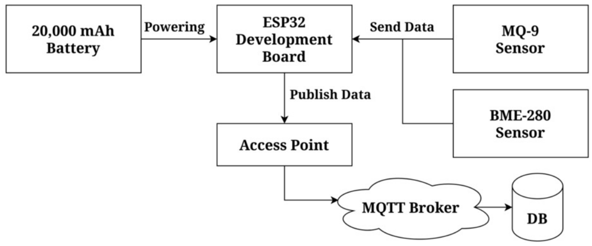

Fig.1: *Block Diagram for Data Gathering Prototype.*

receives CO data. The ESP32 is also connected to the BME-280 sensor via the I2C interface, receiving temperature, humidity, and pressure data. To power the entire prototype, this study used a power bank with a 20,000 mAh capacity, allowing for longer operation time. The study also used a battery-powered portable wireless modem to provide connectivity to a private MQTT broker server. Once all components are properly connected and booted, the prototype IoT can gather the required data and store it in a database. In this case, the database uses Comma-Separated Value format for easier preprocessing later.

After designing the IoT prototype, the next step was to reprogram the ESP32 board to gather data from each sensor and publish it to the MQTT broker. The data gathering process is illustrated in the flowchart in Figure 2.

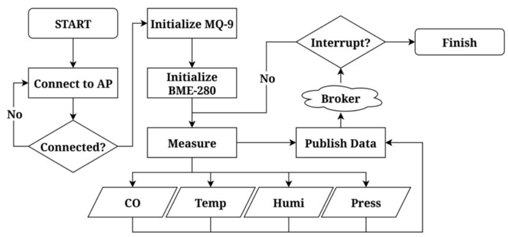

Fig.2: *IoT Prototype Flowchart.*

Figure 2 illustrates the flow process inside the IoT prototype programmed with the MicroPython language. This study decided to use MicroPython instead of other languages (C++ or Lisp) because it was faster to deploy and did not require compilation. The flow started by initialising wireless connectivity to the access point. This step was critical to ensure that the IoT prototype could send data to the MQTT broker. The prototype would try to connect when the Internet was unreachable before proceeding to the next step. After the Internet became reachable, the prototype initialised the sensor modules before measuring environmental data. Then, the environmental data was measured and pushed to the MQTT broker. Once the data is correctly pushed to the MQTT broker, the prototype measures again and loops the process until an interruption occurs. Once the prototype is completed, the next step is to gather the data.

2.3 Data Gathering

This subsection explains the data gathering process, including how much data was gathered. An incorrect data gathering process may lead to an incorrectly trained LSTM model and inaccurate predictions. This study configured the IoT prototype to gather environmental data such as CO, Temperature, Humidity, and Pressure for 24 hours with 10-second intervals. Table 1 below is the sample data received from the prototype.

Table 1: *Sample Data from IoT Prototype.*

Timestamp	Temp (°C)	Humi (%RH)	Press (hPa)	CO (ppm)
11/04/2025 08:00	26.19	57.73	1007.67	0.02
11/04/2025 08:00	26.18	57.47	1007.69	0.01
11/04/2025 08:00	26.18	57.36	1007.72	0.02
...				
12/04/2025 07:59	31.21	47.82	997.56	0.13
12/04/2025 07:59	31.18	47.91	997.61	0.11
12/04/2025 08:00	31.18	47.95	997.61	0.08

Table 1 contains a partial part of the whole dataset gathered by the IoT prototype. Out of a total of 9,326 rows collected, only the Carbon Monoxide data column was selected for LSTM model training, as LSTM only required the target feature. This dataset was also used to create a comparison model for evaluation. Once the necessary Carbon Monoxide data were collected, the study continued with preprocessing. Data was split using a 70% training and 30% testing ratio. Data denoising or normalisation was conducted using the Exponential Moving Average (EMA), considering a look back of 30 data points for each case. EMA was applied to both training and testing data before initiating the training phase to reduce noise in the dataset.

2.4 Long Short-Term Model

This subsection explains the LSTM model creation process, inference mechanism, and deployment using Edge-to-Cloud architecture. In order to train an LSTM model, the required data must be gathered beforehand. After obtaining the primary training variable, this study created the LSTM model. During the LSTM model creation process, this study used a brute-force approach by testing possible internal and external parameters, epochs, and denoising methods to obtain the best parameters to predict CO concentration. Table 2 shows the best configuration that this study found to create a prediction model.

Table 2 shows the configuration for the LSTM model with input and output dimensions of 1, making it a regression model. The model includes eight hidden dimensions and two layers. The denoising method uses an Exponential Moving Average (EMA) of 30 (5 minutes, based on a 10-second interval). Training and testing require three-dimensional data, so the sequential length was set to six, allowing the model to use six consecutive data points as input. The model was trained for 1684 epochs, with the best evaluation result achieved. Figure 3 shows the loss during training.

Table 2: *LSTM Model Parameters.*

Parameter	**Configuration**
Epochs	1684
Hidden Dimension	8
Layer Dimension	2
Input Dimension	1
Output Dimension	1
Sequence Length	6
Exponential Moving Average Denoise	30

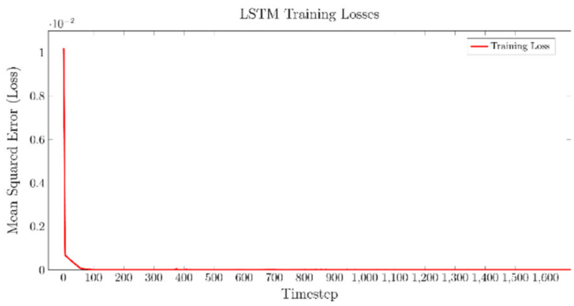

Fig.3: *LSTM Model Training Losses.*

Figure 3 illustrates the LSTM model training losses with the required dataset. Based on this figure, the LSTM minimised its mean squared error loss from 1.0192×10-2 to 1.4784×10-5. This result indicated that the LSTM model successfully learned the time series pattern inside the CO dataset. After successful training, the LSTM deployment to a cloud server involved three steps: (1) exporting the model's state weights, (2) saving them in a portable format, and (3) uploading the portable file to the cloud environment. These steps ensured that the model could be easily deployed while preserving the exact trained weights. Figure 4 illustrates the Edge-to-Cloud topology for online inferencing.

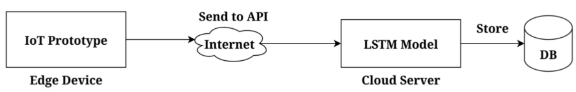

Fig.4: *Edge-to-Cloud Architecture for LSTM Model Inferencing.*

Figure 4 illustrates the architecture that allows the LSTM model to infer without being implemented directly into the edge device. Since no neural network algorithm could be deployed directly on the ESP32 board, the most possible deployment method was via an external server with a REST API endpoint. Based on Figure 4, the IoT prototype, as an edge device, sends sensor data to the server via REST API. The server, deployed with an LSTM model and a REST API endpoint, received and inferred the data. All prediction results were stored in a database for easier analysis in the evaluation phase.

2.5 Evaluation

Evaluation is the most important step, indicating whether the proposed model is better than the previous model or vice versa. This study used several methods to evaluate the proposed model. The first method compares the LSTM prediction with real data. The second method compares the LSTM prediction with other time series-based algorithms, like Autoregressive Integrated Moving Average (ARIMA) and Fuzzy Time Series Chen (FTS Chen). The last evaluation is to compare with another algorithm like Artificial Neural Network (ANN), Linear Regression and Random Forest. Besides comparing side-by-side with another algorithm, this study also evaluates with a statistical approach using Mean Average Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error, and R^2[25], [26]. The following equations are the calculation formulas for each statistical analysis.

$$MAE = \frac{1}{num} \sum_{iter=1}^{num} |actual_{iter} - pred_{iter}| \tag{1}$$

$$MSE = \frac{1}{num} \sum_{iter=1}^{num} (actual_{iter} - pred_{iter})^2 \tag{2}$$

$$RMSE = ^2\sqrt{\frac{1}{num} \sum_{iter=1}^{num} (actual_{iter} - pred_{iter})^2} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{iter=1}^{num} (actual_{iter} - pred_{iter})^2}{\sum_{iter=1}^{num} (actual_{iter} - actual_{iter})^2} \tag{4}$$

Where num refers to the total number of the dataset, iter refers to the iteration number or index. Each formula has a different purpose. MAE measures the difference between actual and predicted errors. MSE measures the average squared error between two values. RMSE measures MSE but is aware of the scaling. R2 was also known as the coefficient of determination, measures the proportion of variance in prediction. This study also used Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to validate the best model among all algorithms.[27], [28], [29]. The following formula was used to calculate AIC and BIC.

$$AIC = 2 \cdot param - 2 \cdot \ln(\hat{L}) \tag{5}$$

$$BIC = \ln(num) \cdot param - 2 \cdot \ln(\hat{L}) \tag{6}$$

Where param is the number of parameters used in the model, followed by the maximum likelihood value's natural logarithm (ln). BIC shared a similar equation structure with AIC, with an additional number as the number of observations (or the tested data)

3. RESULTS AND DISCUSSION

This section consists of two subsections. The first subsection explains the proposed model's evaluation result compared to actual data and with other time-series algorithms. This study also provides statistical evaluation results to compare which algorithm best predicts ambient CO concentration. Meanwhile, the second subsection discusses the results, implications, strengths, weaknesses, and future studies of this type of research.

3.1 Results

This subsection explains the evaluation result from the proposed model. The first evaluation result was comparing the LSTM model with the actual data. Figure 5 illustrates the difference between actual data and LSTM prediction.

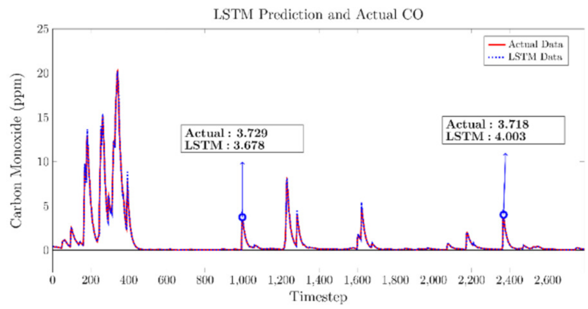

Fig.5: *Comparison of Actual Data and LSTM Prediction.*

Figure 5 compares actual data from the IoT prototype to the LSTM prediction. At the 996th timestep, the LSTM predicted 3.678, closely matching the actual value of 3.729. At the 2368th timestep, the LSTM predicted 4.003 compared to the actual value of 3.718. The deviation between actual and predicted values was minimal. To further evaluate LSTM's performance, Figure 6 compares it with other time-series algorithms.

Figure 6 shows the comparison among time-series algorithms. The evaluation was fair since this study used LSTM, specifically tuned to detect temporal

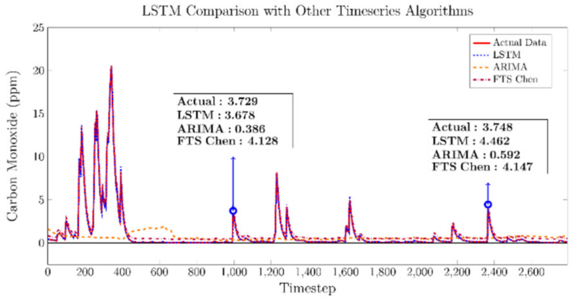

Fig.6: *Comparison of LSTM with other Time Series Algorithms.*

patterns. At the 996th step, LSTM predicted a CO concentration of 3.678 ppm, followed by FTS Chen at 4.128 ppm, and ARIMA at 0.386 ppm. This indicates that LSTM and FTS Chen had predictions closest to the actual data for that step. At the 2368th step, FTS Chen was closest to the actual value with 4.147 ppm, followed by LSTM at 4.462 ppm. However, these results do not establish LSTM or FTS Chen as the best algorithm overall. The findings support that time-series algorithms can capture patterns in Carbon Monoxide data over time. On the other hand, ARIMA performed poorly, likely due to the lack of exogenous variables and the need for more independent data to enhance prediction. Figure 7 then compares a different algorithm not designed for time-series datasets.

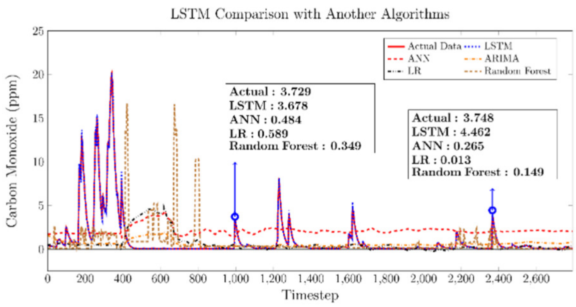

Fig.7: *Comparison of LSTM with Another Algorithm.*

Figure 7 illustrates the comparison result between LSTM prediction and other algorithms. According to the Figure, only the LSTM prediction was close to the actual data. In the 996th timestep, an Artificial Neural Network predicted 0.484 ppm, linear regression predicted 0.589, and Random Forest predicted 0.349 ppm. These results were far from the actual data, with a result of 3.729 ppm. Algorithms like Artificial Neural Networks, linear regression, and Random Forest require supporting data to act as independent variables that affect carbon monoxide. However, the results were far from the actual data. This result is caused by the supporting data, which correlates poorly with the carbon monoxide data. Thus, a significant change only affects a small portion of the Carbon Monoxide concentration. Figure 8 illustrates the statistical comparison between all algorithms.

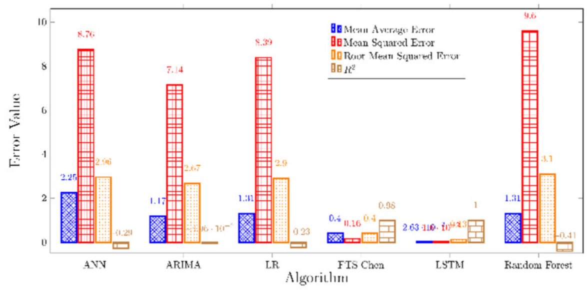

Fig.8: *Statistical Comparison between Algorithms.*

Figure 8 illustrates the statistical evaluations from all algorithms. Several parameters, such as MAE, MSE, RMSE, and R2, were included. The interpretation of each parameter was different for R2. If the model with the lowest MAE, MSE, and RMSE was considered the best, then a model with the highest R2 (close to 1) was considered the best. Based on these interpretations, the proposed model with LSTM prediction was the best model with an MAE of 0.026305, an MSE of 0.016004, an RMSE of 0.126506, and an R2 of 0.997647. FTS Chen was in second place with an MAE of 0.399192, an MSE of 0.159354, an RMSE of 0.399192, and an R2 of 0.976572. The rest of the algorithms were considered weak models with high errors. The accuracy of the LSTM prediction model was calculated using the Mean Average Percentage Error and subtracting it from 100. With that calculation, the accuracy of the LSTM model was 98.42%. This study calculated the AIC and BIC to validate the best model among all algorithms to strengthen this proof. Figure 9 illustrates the AIC and BIC comparison results between algorithms.

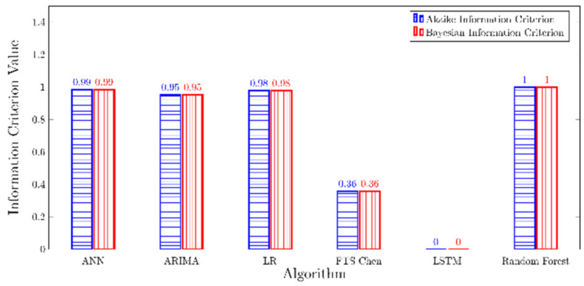

Fig.9: *Validation with AIC and BIC.*

Figure 9 illustrates the AIC and BIC results between algorithms. The interpretation of AIC and BIC was based on the low score. However, there was no baseline result for AIC and BIC. Thus, this study used MinMax scaling to interpret the results more straightforwardly. Based on the validation result, LSTM has the lowest AIC and BIC of 0, clearly demonstrating its superiority in CO prediction. FTS Chen follows with AIC and BIC of 0.359271, while the remaining algorithms have higher AIC and BIC scores above 0.97. These results, together with previous evaluation and validation findings, firmly establish the proposed model as the best performer among the compared algorithms. Further explanations will be provided in the Discussion subsection.

3.2 Discussion

This subsection discusses the results, implications, strengths, weaknesses, and future studies. The first section highlights that the proposed model had the nearest regression prediction to the actual data, with the FTS Chen model as the next closest. Both the proposed model and FTS Chen could predict CO concentration accurately by learning the time-series patterns in the dataset. Autoregressive Integrated Moving Average underperformed among time-series algorithms due to its reliance on unavailable exogenous data during training. Some algorithms with supporting data still performed poorly due to weak correlation between those variables (Temperature, Humidity, and Pressure) and Carbon Monoxide concentration, leading to low accuracy.

The second discussion compares the designed model in previous studies. Most of the previous models, except one article, were not equipped with a prediction algorithm. Thus, the proposed model in this study already exceeds the previous models in terms of implementing a prediction algorithm. However, to compare the performance of the prediction algorithm, this study used article(11) as the baseline for comparison. The model in that article successfully predicted Carbon Monoxide concentration up to 79%. In comparison, the proposed model successfully predicted Carbon Monoxide concentration with an accuracy of 98.42%. With a 19.42% accuracy difference, the proposed model successfully surpassed the previous model's accuracy.

The third discussion concerns the implications of this result for the practical or policy side. The data gathering alone indicated that the CO concentration in Universitas Semarang's parking lot was not high, but that concentration may cause a milder risk if exposed longer. The data gathering result may affect the new policy for Universitas Semarang to mitigate the CO concentration by adding additional exhaust fans or a more open area in the parking lot. The practical implication of this study lies in implementing the proposed model. The prototype of the LSTM prediction model can be implemented to create a mitigation alert whenever the CO concentration reaches the warning zone. Thus, the students or staff may avoid the affected area to mitigate CO poisoning.

The fourth discussion is about the strengths and weaknesses of the proposed model. The strength of the proposed model lies in the capability of the LSTM model to learn time-series patterns from the data set. Thus, additional data were not required. Furthermore, increasing the row number of datasets will increase the accuracy of the prediction model. Another strength of the proposed model is the low-power operation. Since the ESP32 is a low-power processing board, the battery usage for this board is relatively low. Thus, the board can be used longer than a System-on-Chip-based board (for example, Raspberry Pi). The subsequent strength of the proposed model is that a stronger processing board is not required at all. Since the prediction mechanism was done in the cloud server, the edge node only needs to send the data. There are several weaknesses in the proposed model. The first weakness is the algorithm itself. The LSTM algorithm is a type of algorithm that leverages seasonality or trends within a dataset. Without this pattern, LSTM would not be able to learn the pattern and would produce low-accuracy predictions. Thus, it is critical to capture the seasonality of the data. The second weakness is the application of LSTM. The model is only applicable in the designed location and time. If the proposed model predicts in different locations (for example, the canteen area), the result may differ and produce low accuracy. Besides that, an LSTM that only learns hourly patterns will not be able to predict daily patterns. Thus, daily patterns are required to predict daily results. The third weakness is the accuracy of LSTM prediction. Although the accuracy of the LSTM model reached 98.42%, it was deemed sufficient. As shown in Figure 6, there were several incorrect predictions in time steps 996th and 2368th. The fourth weakness of this model is the internet requirement to send the data to the cloud server, since it is impossible to process the prediction onboard.

The last discussion concerns future studies on Carbon Monoxide prediction. Several improvements could enhance the model. First, using a different onboard machine learning model would enable independent predictions. Second, including more LSTM data may allow additional future predictions. Third, using supplemental data may improve accuracy in various algorithms. Fourth, adding sensors such as NOx or PM2.5 could increase model variation, but each sensor should be trained separately. In summary, these changes could further refine predictions. The study concludes that the LSTM-based model accurately predicts CO concentration in Universitas Semarang's parking lot at 98.42%, a 19.42% improvement from the previous study.

4. CONCLUSIONS

Partial or incomplete fuel combustion in motor vehicles emits carbon monoxide, a poisonous gas. This gas alone may cause health problems like chest pain and focus impairment. Higher exposure can cause death. For that reason, several studies have focused on CO concentration monitoring with Internet of Things technology. Most models implement a monitoring capability, but do not create predictions for future alerts. Meanwhile, only one article implemented an Artificial Neural Network to create predictions based on a dataset. However, the accuracy of the past model was limited to 79%. Because of this, this study proposes an Internet-of-Things-based prediction model with a Long Short-Term Memory algorithm to create future predictions. With Edge-to-Cloud architecture, a low-processing board can create a prediction by sending the data to the cloud without deploying the prediction model onboard. Based on the evaluation step, the proposed model produced a regression prediction close to the actual data, achieving an accuracy of 98.42%. This is a 19.42% improvement compared to the previous model. Besides that, this study also compared other time series and non-time series algorithms to evaluate performance. The LSTM model outperformed all algorithms with the lowest MAE of 0.026305, MSE of 0.016004, RMSE of 0.126506, and the highest R2 of 0.997647. To validate the result, this study used AIC and BIC to determine the best algorithm. Both AIC and BIC scored zero after MinMax scaling, deeming the proposed model with the LSTM algorithm as the best among the rest. This explanation shows that the proposed model with the LSTM algorithm is the best and successfully outperforms both time series and non-time series algorithms.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Institute for Research and Community Service (LPPM), Universitas Semarang, for the financial support provided through the Internal Competitive Research Grant (*Penelitian Hibah Kompetitif Internal*) under contract number 010/USM.H7.LPPM/L/2025. The authors also appreciate the institutional encouragement that fosters innovation and applied research to address real-world environmental and public health challenges.

AUTHOR CONTRIBUTIONS

Conceptualisation, A.F.D. and S.S.; methodology, A.F.D.; software, A.R.; validation, AM.H. and S.S.; formal analysis, A.F.D.; investigation, A.R.; data curation, A.R.; writing—original draft preparation, A.M.H.; writing—review and editing, A.R. and A.M.H.; visualisation, A.F.D. and A.M.H.; supervision, A.F.D.; funding acquisition, A.M.H.; All authors have read and agreed to the published version of the manuscript.

References

[1] H. Kinoshita *et al.*, “Carbon monoxide poisoning,” *Toxicology Reports*, vol. 7, pp. 169–173, 2020.

[2] L. K. Weaver, “Carbon monoxide poisoning,” *Undersea & amp; hyperbaric medicine : journal of the Undersea and Hyperbaric Medical Society, Inc*, vol. 47, no. 1, p. 151—169, 2020.

[3] L. M. Chu, S. Shaefi, J. D. Byrne, R. W. Alves de Souza, and L. E. Otterbein, “Carbon monoxide and a change of heart,” *Redox Biology*, vol. 48, p. 102183, Dec. 2021.

[4] M. Yang *et al.*, “A systematic review and meta-analysis of air pollution and angina pectoris attacks: identification of hazardous pollutant, short-term effect, and vulnerable population,” *Environmental Science and Pollution Research*, vol. 30, no. 12, pp. 32246–32254, Mar. 2023.

[5] “Carbon monoxide,” World Health Organisation; obtainable from WHO Publications Centre USA, Geneva : Albany, N.Y, 13, 1979.

[6] G. Reumuth *et al.*, “Carbon monoxide intoxication: What we know,” *Burns*, vol. 45, no. 3, pp. 526–530, 2019.

[7] G. D. Astudillo, L. E. Garza-Castañon and L. I. Minchala Avila, “Design and Evaluation of a Reliable Low-Cost Atmospheric Pollution Station in Urban Environment,” in *IEEE Access*, vol. 8, pp. 51129-51144, 2020

[8] T. Joseph *et al.*, “Portable gas detection and warning system for olfactory disabled people,” in *2020 International Conference for Emerging Technology, INCET 2020*, 2020.

[9] Y. A. Koedoes, S. Jie, M. N. A. Nur, Bunyamin and A. Astari, “Design of Prototype System for Monitoring Air Quality for Smart City Implementation,” *IOP Conference Series: Materials Science and Engineering*, vol. 797, no. 1, p. 012023, Mar. 2020.

[10] T. Porselvi, S. G. CS, J. B, P. K and S. B. S, “IoT-Based Coal Mine Safety and Health Monitoring System using LoRaWAN,” in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, pp. 49–53, May 2021.

[11] H. Etemadfard, V. Sadeghi, F. Hassan Ali and R. Shad, “CO Emissions Modelling and Prediction using ANN and GIS,” *Pollution*, vol. 7, no. 3, Jul. 2021.

[12] A. N. Hikmah and C. B. Dwi Kuncoro, “Carbon Monoxide Monitoring System based on IoT with Low Power Sensor Node for Indoor Applications,” in *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 316–320, Oct. 2022.

[13] Karuna, G., Kumar, R.P. Ram, Gopaldas, Steven, Parvathaneni, Vasista and Lokesh, Teddu, “Air Quality and Hazardous Gas Detection using IoT for Household and Industrial Areas,” *E3S Web Conf.*, vol. 391, p. 01146, 2023.

[14] D. M. B. Sidan, H. Hartono and S. Suyatmo, “Design And Development Of Diesel Power Plant Exhaust Gas Emission Monitoring Based

on IoT," *Int. Conf. Adv. Transp. Eng. Appl. Soc. Sci.*, vol. 3, no. 1, pp. 35–40, Dec. 2024.

[15] A. Rerkratn, V. Riewruja, W. Petchmaneelumka and S. Tammaruckwattana, "Wireless Carbon Monoxide Level Control and Monitoring System," in *2025 11th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 537–541, Apr. 2025.

[16] V. N. Saputri, F. A. Alifteria, A. I. Agusty, M. Anggaryani and M. N. R. Jauhariyah, "The use of DHT11 for making green ecosystem model," *AIP Conference Proceedings*, vol. 2858, no. 1, p. 040007, Aug. 2023.

[17] A. Sudaryanto, Yohanes Aditya Wisnu W, and Agung Kridoyono, "Accuracy of DHT11 Temperature and Humidity Sensor In Egg Incubator," *INFOTRON*, vol. 4, no. 1, pp. 1–6, May 2024.

[18] K. Han and Y. Wang, "A review of artificial neural network techniques for environmental issues prediction," *Journal of Thermal Analysis and Calorimetry*, vol. 145, no. 4, pp. 2191–2207, Aug. 2021.

[19] R. Maya, B. Hassan and A. Hassan, "Develop an artificial neural network (ANN) model to predict construction projects' performance in Syria," *Journal of King Saud University - Engineering Sciences*, vol. 35, no. 6, pp. 366–371, Sep. 2023.

[20] S. C. Agustinur, K. I. Khalifa, M. Yantidewi and U. A. Deta, "Literature Review: Air Oxygen Level Monitoring System," *Int. J. Res. Community Empower.*, vol. 1, no. 2, pp. 62–70, Jul. 2023.

[21] L. M. Easterline, A. A.-Z. R. Putri, P. S. Atmaja, A. L. Dewi and A. Prasetyo, "Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32," *Procedia Computer Science*, vol. 245, pp. 815–824, Jan. 2024.

[22] E. Evcin and Y. M. Erten, "Comparison of Different Weather Data Acquisition Methods," in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, Oct. 2024, pp. 1–6.

[23] D. Milojicic, "The Edge-to-Cloud Continuum," *Computer*, vol. 53, no. 11, pp. 16–25, Nov. 2020.

[24] M. N. Jamil, O. Schelén, A. Afif Monrat and K. Andersson, "Enabling Industrial Internet of Things by Leveraging Distributed Edge-to-Cloud Computing: Challenges and Opportunities," *IEEE Access*, vol. 12, pp. 127294–127308, 2024.

[25] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, Jul. 2021.

[26] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, 2022.

[27] S. I. Vrieze, "Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC)," *Psychological Methods*, vol. 17, no. 2, pp. 228–243, 2012.

[28] S. Portet, "A primer on model selection using the Akaike Information Criterion," *Infectious Disease Modelling*, vol. 5, pp. 111–128, Jan. 2020.

[29] H. Akaike, "Akaike's Information Criterion," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 41–42, 2025.

Alauddin Maulana Hirzan is a lecturer at the Faculty of Information and Communication Technology, Universitas Semarang, Indonesia. His primary research interests include the Internet of Things (IoT), reinforcement learning, Python programming, networking, and security. He has contributed to publications on IoT applications, including sensor-based temperature and humidity monitoring using DHT22 sensors and the Cayenne API, automatic irrigation systems for rice fields, gas leak detection, and network security optimizations using MikroTik firewalls. His work often emphasizes practical implementations for industrial, environmental, and educational enhancements.

April Firman Daru is a senior lecturer and researcher at the Faculty of Information and Communication Technology, Universitas Semarang, Indonesia. His research spans artificial intelligence, web technologies, Internet of Things (IoT)-based systems, e-learning platforms, mobile applications, environmental monitoring (including carbon monoxide detection and hydroponic systems), and sentiment analysis using natural language processing. He has authored numerous works focusing on technological solutions for education, safety, agriculture, and aquaculture, often integrating sensors, reinforcement learning for attack detection in IoT networks, and interactive online learning tools.

Susanto Susanto is a lecturer at the Faculty of Information and Communication Technology, Universitas Semarang, Indonesia. His research focuses on management information systems, IoT configurations and communication (including Telegram integration), learning platform selection using fuzzy multiple attribute decision making, genetic algorithms for scheduling practicum activities, the impact of grammar on academic writing, and network security (such as WiFi password analysis). His publications address practical improvements in educational tools, optimization in higher education settings, and community outreach on data privacy and digital security.

Ahmad Rifa'i is a lecturer at the Faculty of Information and Communication Technology, Universitas Semarang, Indonesia. He holds qualifications in computer science with a focus on applied technologies. His expertise lies in data mining, intelligent systems, web technologies, and databases. His research contributions include the development of information systems for organizational and educational applications, intelligent decision-making tools, and web-based solutions to support data processing and management needs.