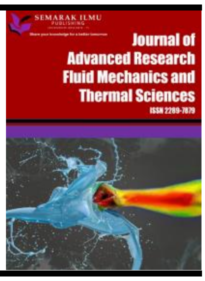
SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Fluid Mechanics and Thermal Sciences

Journal homepage:
https://semarakilmu.com.my/journals/index.php/fluid_mechanics_thermal_sciences/index
ISSN: 2289-7879



Analysis of Wind Speed Prediction using Artificial Neural Network and Multiple Linear Regression Model using Tinyml on Esp32

Chua Kiang Hong[1], Mohd Azlan Abu[1,*], Mohd Ibrahim Shapiai[2], Mohamad Fadzli Haniff[3], Radhir Sham Mohamad[1], Aminudin Abu[3]

[1] Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
[2] Pattern Recognition & Robotics Automation (PRA), Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
[3] Intelligent Dynamics & System (IDS) i-Kohza, Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 22 January 2023
Received in revised form 19 May 2023
Accepted 26 May 2023
Available online 15 June 2023

Keywords:
Embedded system; TinyML; Multiple Linear Regression; ANN; wind speed prediction; ESP32; air quality

ABSTRACT

This study provides the investigation of wind speed forecasting using an Artificial Neural Network (ANN) and Linear Regression Model with an ESP32 chip. A portable wind speed prediction system will aid in mitigating the risks associated with sudden gusts of wind by forecasting the maximum wind speed that may occur in the near future. This research also demonstrates the application of TinyML in the field of Artificial Intelligence (AI) by applying a Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) model to small, low-powered ESP32 microcontrollers to analyse data with low latency. The system predicts wind speed in Setapak, Kuala Lumpur, based on the measured temperature and humidity using a DHT22 sensor and displays forecast results and sensor readings on LCD screens. To measure the accuracy of the MLR and ANN models, the coefficient of determination (R^2), mean square error (MSE), and root mean square error (RMSE) between predicted and actual results are evaluated. Results indicate that the ANN model outperforms the MLR model for predicting wind speed.

1. Introduction

Measuring wind speed and direction is essential in the application of air quality monitoring system. It can help pinpoint the location of the pollution source and provide a clearer picture of what is occurring in the air [1]. Machine learning (ML) is an algorithm that enables computers to acquire knowledge from data or information. Without explicit programming, software applications can make accurate predictions using machine learning. The three basic types of machine learning are supervised, unsupervised, and reinforcement learning, which enables machines to learn from trial and error through feedback [2-3]. Compared to transferring information to the cloud, processing data

* *Corresponding author*
E-mail address: mohdazlan.abu@utm.my

https://doi.org/10.37934/araset.107.1.2944

directly on a device will require less time and resources. TinyML is an application of machine learning that emerged due to people's desire for smaller and more capable devices. TinyML emerged as a result of these demands. TinyML is an application of embedded systems and machine learning that optimises machine learning models. This application may run on small, low-powered devices [4] such as ESP32, Arduino development boards, and microcontrollers. TinyML's low power consumption enables machine learning models to run on edge devices on battery for long periods.

TinyML's future is promising due to its numerous features. Since machine learning is executed on the edge device, data is not stored on servers, ensuring the system's confidentiality. In addition, by running the model on the edge device, TinyML enables the system to operate with low power, low latency and bandwidth [4]. A linear regression model can be as straightforward as a single input and a single output, with the connection between input and output depicted in the equation below [5]

$$y = \beta_0 + \beta_1 x \qquad (1)$$

where y is output of the model (dependent variable), x is input of the model (independent variable), β_0and β_1 are unknown called regression coefficients. β_0 can be called the intercept (y's value when x is 0) and β_1 is the slope that indicates the change of y with the increase or decrease with one unit of x. For a linear regression model to provide accurate predictions, there must be a proportional relationship between the model's inputs and outputs, as the model assumes that their relationship is linear. The model will conduct a regression task to identify the linear best fit between data. Therefore, the distributed data will be needed as outlying data patterns that will introduce bias while finding the best fit. Examples of linear regression models include the forecasting of food sales based on warehouse conditions [6], the forecasting of blood pressure based on human weight [7], and the forecasting of Air Quality Monitoring System [8].

This paper will implement the Multiple Linear Regression (MLR) model on the ESP32 microcontroller development board to forecast wind speed depending on the surrounding temperature and humidity. The used model is a supervised learning model in which a training dataset containing data on temperature, humidity, and wind speed at various times is used to train the model. Multiple linear regression is a linear regression technique that consists of more than one independent variable and conducts regression tasks to find the greatest linear fit between data that results in the smallest overall model error [5]. The linear relationship of independent and dependent variables of the MLR model used in this study can be stated as below

$$\text{Wind speed} = \beta_0 \text{ (Temperature)} + \beta_1 \text{ (Humidity)} + \beta_2 \qquad (2)$$

where wind speed is the dependent variable (the model's output), measured in kilometres per hour, temperature and humidity are the independent variables (the model's inputs), measured in degrees Celsius and percentages, accordingly. β2 is undetermined as the wind speed when the temperature and humidity are both zero. β0 and β1 are the coefficients that indicate the slope of the relationship between temperature and humidity and the change in wind speed. The research motivation of this work is to implement AI and MLR model on the ESP32 hardware. So that the applications of AI and MLR can be carried out independently without the uses of computer. Thus, the complex application can be applied anytime, anywhere by using the small hardware.

1.1 TinyML on ESP32

TensorFlow Lite Micro (TFLM) [9], Tensor [10], Edge ML [11], and Eloquent TinyML[12] are among the libraries and frameworks that facilitate the building of machine learning models on microcontrollers. TensorFlow Lite Micro and Eloquent TinyML are the only libraries claimed to be accessible as an Arduino library and in the Arduino IDE for programming Arduino and ESP32 microcontrollers [13]. For this paper, ESP32 microcontrollers are programmed with Arduino IDE; thus, research on TinyML implementations for ESP32 will rely on the TensorFlow Lite Micro and Eloquent TinyML libraries. The possibility of implementing machine learning models on the ESP32 microcontroller by implementing the TFLM library is demonstrated by implementing an Artificial Neural Network (ANN) model to predict the specific volume of moulded product parts from pressure and temperature data from sensors [14]. Eloquent TinyML is an Arduino library that simplifies the deployment of TensorFlow Lite models on compatible microcontrollers [13, 14]. As of current, no research has been found on the use of AI models with Eloquent TinyML on ESP32 microcontrollers in the real world. Nonetheless, a method of deploying a Convolutional Neural Network (CNN) model on a Tensilica Xtensa LX6 microprocessor [15] has shown the viability of utilising different kinds of AI models on the ESP32 microcontrollers board. Tensilica Xtensa LX6 is a microcontroller implemented for the ESP32 microcontroller board family [16].

1.2 Wind Speed Prediction using Linear Regression Model and ANN Model

Prior to this study, other researchers have utilised the Multiple Linear Regression (MLR) and Artificial Neural Network models to estimate wind speed based on other meteorological data. Verma *et al.,* [17] demonstrated the use of the Multiple Linear Regression (MLR) model and the Artificial Neural Network (ANN) to estimate wind speed in Chhattisgarh, India, based on humidity, air pressure, wind direction, precipitation, and temperature. The research achieved a coefficient of determination (R^2) of 0.779 for the MLR model and 0.901 for the ANN model between predicted and actual results.

Two cities in Morocco, Northern Africa, Tangier and Tarfaya, produce better study results [18]. Based on other meteorological data, such as temperature and humidity, MLR and ANN models were utilised to forecast future wind speeds. R^2, MSE, and root mean squared error (RMSE) between anticipated and actual results are used to evaluate the models. R^2 was 98.16% in Tangier and 97.2% in Tarfaya for the model of multiple linear regression used to forecast wind speed. By comparing MLR and ANN models to the Autoregressive Integrated Moving Average (ARIMA) model, Arzu *et al.,* [19] demonstrated that MLR and ANN models could be accurate algorithms for predicting wind speed. Eshan [20] found that the application of the MLR model to forecast wind speed based on temperature, humidity, and air pressure may reach a high degree of accuracy. The investigation was conducted in 17 American cities, including Paxton, Biggs, Trimont, Lakefield, etc. As a result, the MLR model attained an average accuracy of 92.3% based on the coefficient of determination (R^2) between predicted and actual results, which is greater than other methods such as ridge regression (87.2%), convolutional neural network (82.0%) and hubber regression (9.19%) [20]. Al-Zadjali *et al.,* [21] demonstrate an ANN model capable of predicting wind speed based on temperature, humidity, and air pressure, with a correlation of 98.73% between predicted and measured wind speed. In [22], it is demonstrated that RNN has a superior performance for wind speed nowcast prediction than a standard ANN model, as measured by a smaller RMSE between predicted and actual wind speeds.

1.3 Correlation Between Wind Speed, Temperature and Humidity

In order for a Multiple Linear Regression (MLR) model to forecast wind speed based on surrounding temperature and humidity, there must be a proportional relationship between the meteorological variables, either directly proportional or inversely proportional. Figure 1 depicts connections between wind speed, temperature, and humidity according to research conducted in Poland by Dec *et al.,* [23]. During the day, as the temperature rises, the value of wind speed is at its peak. When wind speed is strong, relative air humidity demonstrates a lower value.

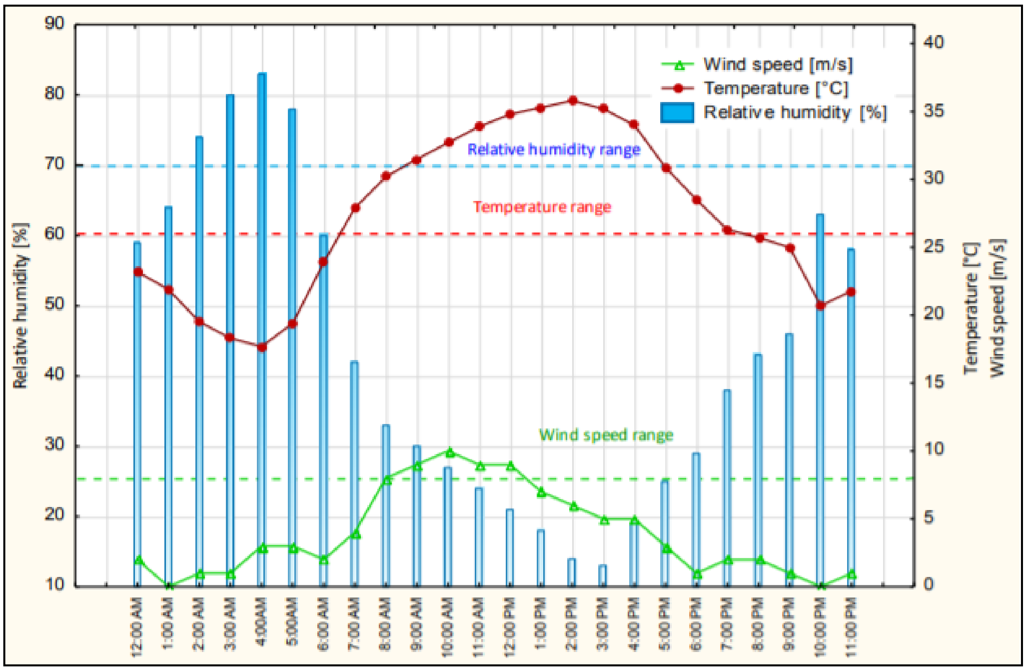


Fig. 1. Hourly air parameters at Rzeszów-Jasionka airport, Poland [23]

According to research conducted by Zakariah *et al.,* [24], the similar association exists between wind speed, temperature, and humidity. The link between wind speed, humidity, and temperature is shown in Table 1, based on data of monthly wind speed in Klang Valley, Selangor, Malaysia, in 2017. The research revealed a negative association between wind speed and relative humidity (-0.256) and a positive link between wind speed and temperature (0.278). Figure 2 depicts the 24-hour data analysis of wind speed, humidity, and temperature. Based on the data, it was hypothesised that as wind speed increased, humidity readings would decrease and temperature would rise.

Table 1
Correlation relationship between air parameters [24]

	Correlation (maximum 1.000)
Wind speed	1.000
Humidity	-0.256
Temperature	0.278

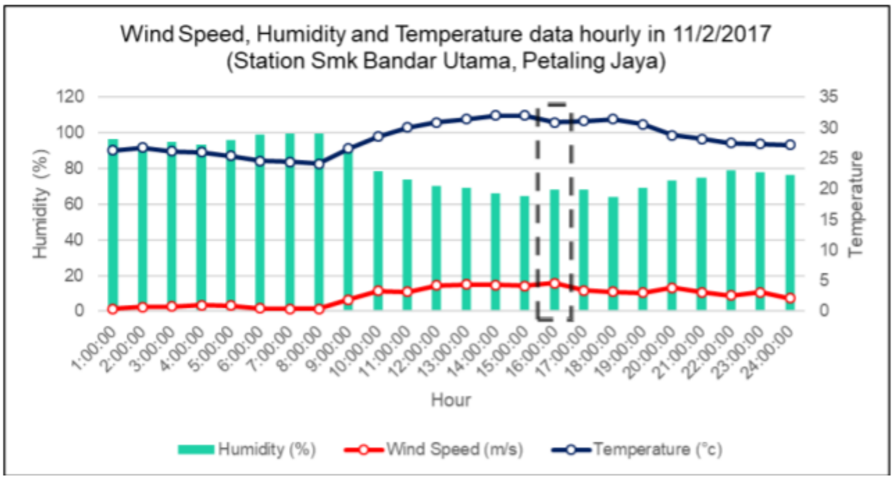


Fig. Error! No text of specified style in document.**.** Hourly air parameters analysis in Petaling Jaya, Malaysia [24]

There are currently no studies demonstrating the practical implementation of Artificial Intelligence (AI) models on the ESP32 microcontroller using the Eloquent TinyML library. In addition, there are no studies that apply Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) models to the prediction of real-time wind speed in Malaysia based on temperature and humidity. This paper will answer the research question to investigate whether ANN and MLR can be implemented on ESP32 microcontroller.

2. Methodology

2.1 Flowchart of The Study

The study began with the development of a training dataset for Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) models. To train Artificial Intelligence (AI) models with greater precision, data filtering and hyperparameter tuning are performed on the MLR and ANN models. During the training phase, datasets are used to evaluate the testing accuracy of AI models. If the model cannot achieve an acceptable level of accuracy, the procedure will be repeated. If the testing accuracies of MLR and ANN models reach 70% (evaluated using the coefficient of determination between predicted wind speeds and recorded wind speeds in the dataset). Finally, the models will be deployed on ESP32 microcontrollers and the performance of wind speed prediction by AI models will be analysed. MLR and ANN models will be compared and if their accuracy reaches 70%, the process will be stopped. Otherwise, the AI model training procedure will be resumed.

2.2 Training Dataset

Due to the fact that the Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) models utilised in this study are Artificial Intelligence (AI) models built using a supervised learning approach, a training dataset is required. The dataset used for this study is comprised of historical weather data for Setapak city, Kuala Lumpur, Malaysia, obtained from www.timeanddate.com[25]. The data consists of 1696 rows of meteorological information collected every six hours between 1 January 2021 and 28 February 2022. The average temperature and humidity over a 6-hour timeframe are employed as independent variables (inputs) in AI models, whereas wind speed is the dependent variable (output).

Figure 3 is a screenshot of historical weather data for Setapak city, Kuala Lumpur, Malaysia, obtained from www.timeanddate.com. This website provides the data required to construct the training data set used to train the AI models used in this project, including temperature, humidity, and wind speed at 6-hour intervals every day from January 1, 2021, to February 28, 2022. Wind velocity is measured in kilometres per hour (km/h), temperature is measured in degrees Celsius (°C), and humidity is measured in percentage (%). Figure 3 was a screenshot of the training dataset created using historical weather data from www.timeanddate.com [25].

Date	Time	Highest Temperature	Lowest temperature	Humidity	Average Temperature	Barometer	Wind Speed	Weather
1/1/2021	00:00:00	26	25	90	25.5	1010	1	Passing clouds
1/1/2021	06:00:00	27	25	87	26	1010	2	Passing clouds
1/1/2021	12:00:00	31	27	64	29	1009	5	Passing clouds
1/1/2021	18:00:00	30	26	76	28	1010	5	Passing clouds
2/1/2021	00:00:00	26	25	90	25.5	1010	3	Passing clouds
2/1/2021	06:00:00	28	25	84	26.5	1011	5	Passing clouds
2/1/2021	12:00:00	29	28	73	28.5	1009	6	Passing clouds
2/1/2021	18:00:00	28	25	91	26.5	1010	3	Light rain
3/1/2021	00:00:00	25	24	94	24.5	1010	5	Light rain
3/1/2021	06:00:00	24	24	93	24	1011	4	Light rain
3/1/2021	12:00:00	24	23	91	23.5	1009	10	Light rain
3/1/2021	18:00:00	24	23	93	23.5	1010	5	Passing clouds
4/1/2021	00:00:00	23	23	94	23	1010	2	Light rain
4/1/2021	06:00:00	27	23	88	25	1010	2	Passing clouds
4/1/2021	12:00:00	30	27	68	28.5	1009	7	Passing clouds
4/1/2021	18:00:00	28	26	83	27	1009	2	Passing clouds
5/1/2021	00:00:00	26	24	92	25	1009	2	Passing clouds
5/1/2021	06:00:00	31	24	80	27.5	1010	8	Passing clouds
5/1/2021	12:00:00	32	26	65	29	1008	14	Passing clouds
5/1/2021	18:00:00	26	26	90	26	1009	5	Passing clouds

Fig. 3. Screenshot of the training dataset

2.3 Training Platform

The Artificial Intelligence (AI) models for this study are trained using Google Colab. Google Colab is a platform that allows users to write and execute Python programmes via web browser [26]. This research employs two AI models, a Multiple Linear Regression (MLR) model and an Artificial Neural Network model (ANN).

Before the model is trained, the wind speed, temperature, and humidity data are plotted in plot diagrams to determine if there is a link between the data, as illustrated in Figure 4. On the basis of the produced diagrams, proportional connections between wind speed, temperature, and humidity were observed. As depicted by the plot diagram enclosed by a red box in the lower-left corner of Figure 4, wind speed increases as the average temperature rises. In addition, when high humidity is detected, wind speed tends to be low, as depicted in the plot diagram encircled by a black box in Figure 4. The connections between wind velocity, temperature, and relative humidity are identical to those obtained in [29-31].

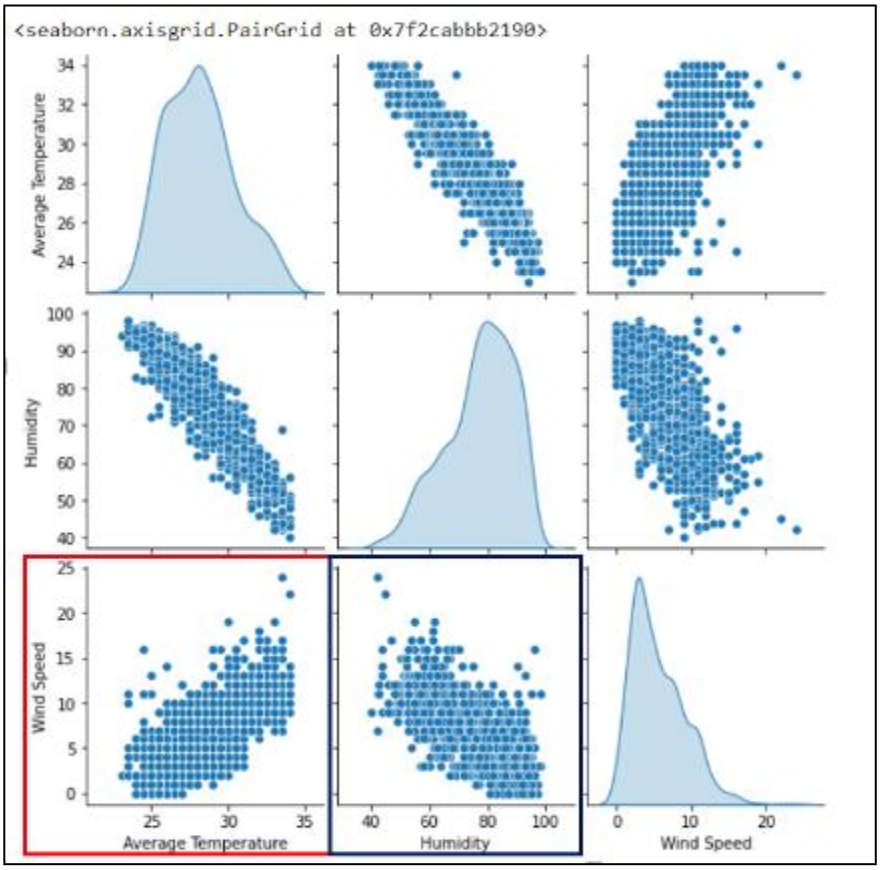


Fig. 4. Plotting of data before data filtering

Since an MLR model performs regression by identifying the best-fitting linear relationship between wind speed, temperature, and humidity, it is preferable for the temperature and humidity data to remain closer to the best-fitting linear relationship that represents their relationship with wind speed data. Extremely large or small data values that are too far from the line of best fit may confound the AI by throwing off its sense of proportion. In addition, there may be incorrect data in the dataset when negative wind speed values are recorded; data rows with negative wind speed values will be removed from the dataset. In this investigation, data with wind speeds below 0 km/h and above 15 km/h will be excluded from the dataset.

Figures 5 and 6 depict the weather data acquired at 6 a.m. and 6 p.m. and 12 a.m. and 12 p.m. daily after rainy weather data and data with wind speed less than 0 km/h and greater than 15 km/h are discarded. The plot diagrams of Figure 5 and Figure 6 reveal that weather data collected at 6am and 6pm have more data far from the best fit line and less balance than weather data obtained at 12am and 12pm. Thus, weather data captured at 12am and 12pm will be used to train the MLR model following data filtration. To ensure a fair comparison between the MLR model and the Artificial Neural Network (ANN) model, both are trained with the identical data.

Journal of Advanced Research in Fluid Mechanics and Thermal Sciences
Volume 107, Issue 1 (2023) 29-44

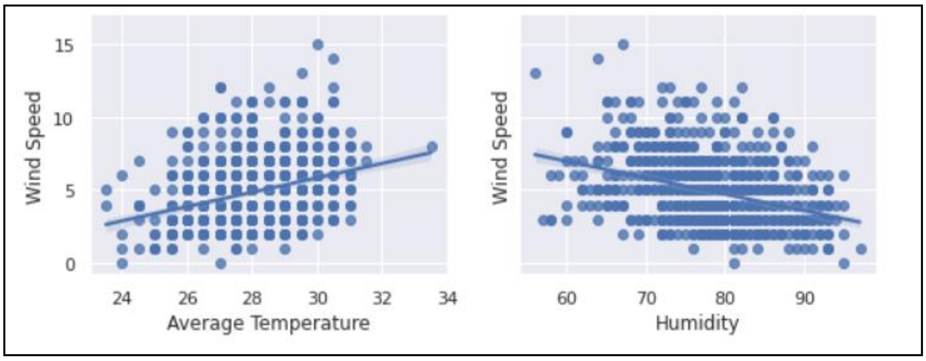


Fig. 5. Weather data recorded at 6am and 6pm everyday

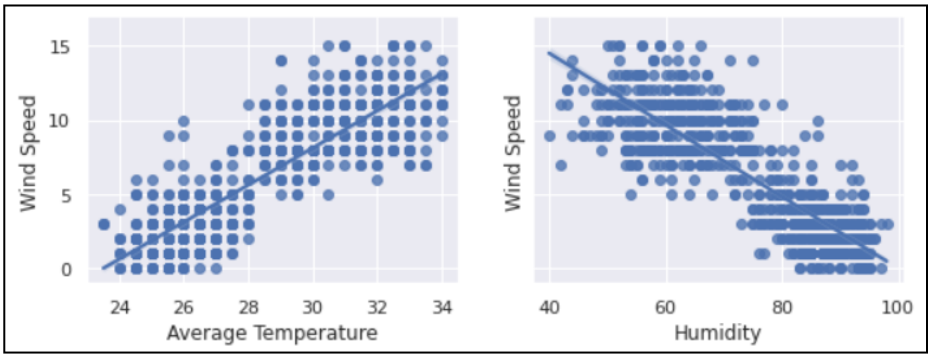


Fig. 6. Weather data recorded at 12am and 12pm everyday

Figure 7 depicts the outcome of data after data filtration. The proportional relationship between wind speed, temperature, and humidity may be viewed clearly after removing variables that can cause confusion during the AI model training phase. Figure 7's plot diagram surrounded by a red box depicts an increase in wind speed with an increase in temperature, whereas the plot diagram surrounded by a black box depicts a drop in wind speed with an increase in humidity. The dataset initially contains 1696 data. Following data filtering, 801 data remain in the dataset. Next, the dataset will be matched to AI models for training purposes.

A Multiple Linear Regression (MLR) model can be constructed using a TensorFlow Sequential model. TensorFlow is an open-source Python library for machine learning and artificial intelligence developed by the Google team [27]. Figure 8 depicts the MLR Model's modelling. The dense layer's input dimension is set to 2 to accommodate temperature and humidity data. When the dense layer output size is set to 1, the model will provide only one output: the expected wind speed. Number of epochs, which means the number of times the algorithm will be run for the model is set to 150 with batch size 3 to reach the best result.

Multiple Linear Regression Model is compared to an Artificial Neural Network (ANN) model to evaluate the MLR model's ability to forecast wind speed depending on temperature and humidity. Figure 9 illustrates the modelling of the ANN model. The ANN model is built using a sequential model from the TensorFlow framework. The ANN model began with a single input layer that had an input dimension of 2 and 128 input units. The ANN's hidden layer is comprised of four dense layers with output units of 64, 32, 16, and 8 correspondingly. The input layer and the hidden layer use the rectified linear activation function (RELU), which allows the data to converge more quickly than with other activation functions. The model's conclusion consists of a single output layer with output unit 1, which is the expected wind speed. The ANN model's hyperparameters are determined using the same tuning procedure as the Multiple Linear Regression (MLR) model. The hyperparameters that allow the ANN model to have the smallest loss value will be utilised. Number of epochs, or the number of times the algorithm will be executed for the model, is set to 150 with a batch size of 3 in order to achieve the optimal outcome. Adagrad is utilised as the optimizer for the ANN model's automatic tweaking of the learning rate.

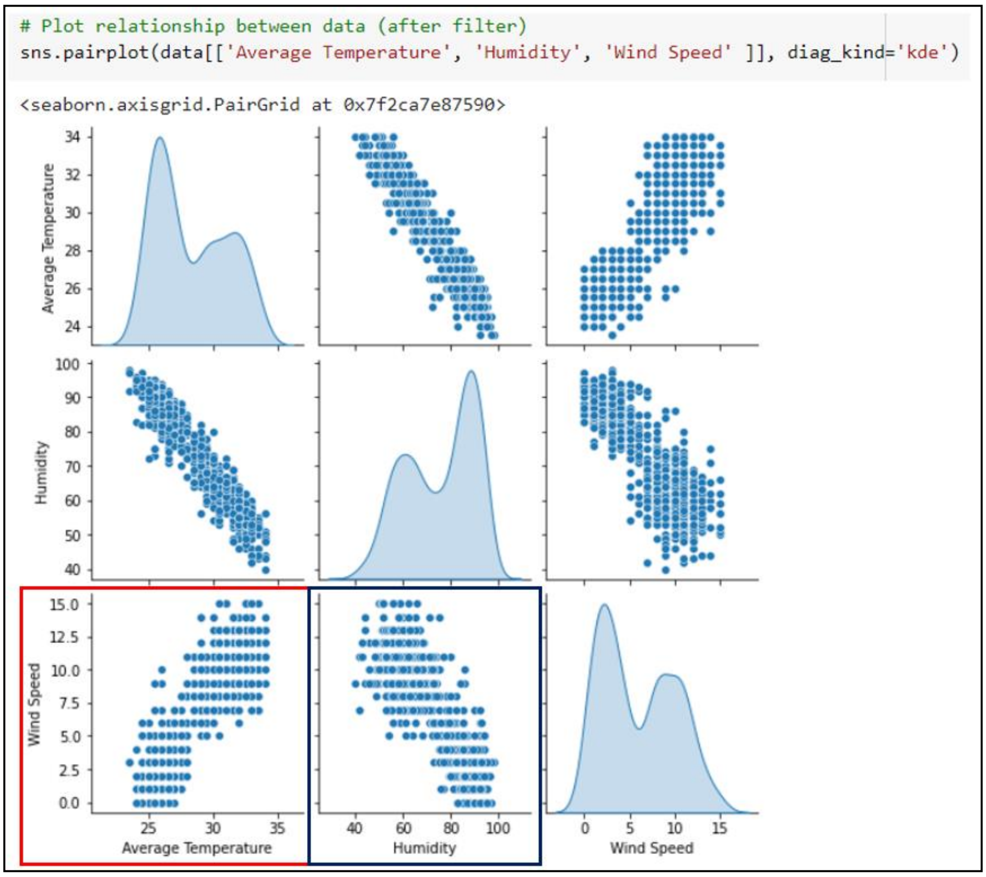


Fig. 7. Plotting of data after filtering process

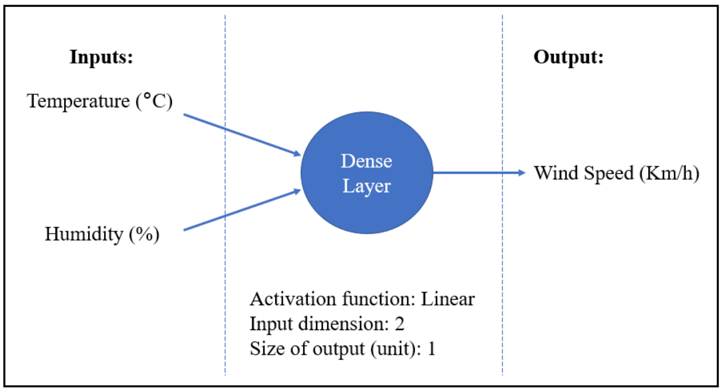


Fig. 8. MLR model approach

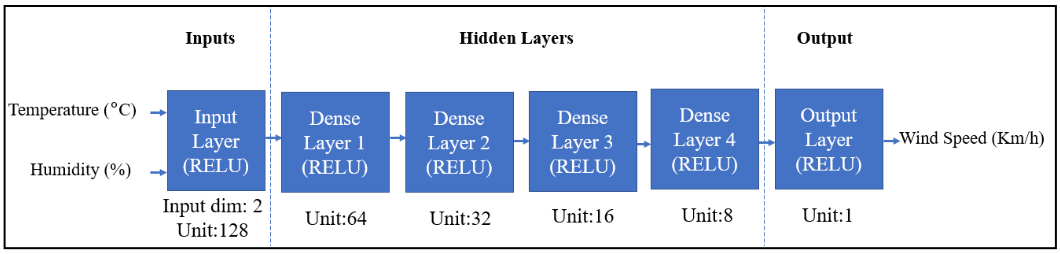


Fig. 9. ANN Model using TensorFlow

2.4 AI Models on ESP32

Multiple Linear Regression (MLR) and Artificial Neural Network (ANN) models will be downloaded in TensorFlow Lite FlatBuffer (.tlite) format from Google Colab. Due to the requirement of the C++ programming language for ESP32 microcontroller programming, the microcontroller cannot utilise the learned linear regression model in.tflite format. To use trained models on ESP32, it was necessary to convert the model to C array (.cpp or.h) format. This is possible through the use of a Python programming environment. For this project, the trained model will be translated into C array format by executing the following command at the system console [28]

"xxd -i model.tflite > model.h"

where xxd is a Linux command that generates a hex dump from a given file. "model.tflite" is the name of the trained model in Google Colab, while "model.h" is the name of the trained model converted to a C array.

Using the Arduino IDE, the EloquentTinyML library is utilised to deploy the linear regression model on the ESP32 board [20]. In order to utilise the trained model, the C array file must be included in the ESP32 microcontroller's programme. In addition, the model's number of inputs and outputs had to be provided. The MLR and ANN models for this project will have two inputs and one output. For the model to operate with the data, three 1024-bit areas will be preallocated, one for the output (wind speed) and two for inputs (temperature and humidity). Next, to ensure that the AI models deployed on the ESP32 microcontroller are identical to those trained in Google Colab, pre-set inputs of temperature and humidity will be used and it will be observed whether the models on ESP32 make the same predictions as the model in Google Colab based on the pre-set inputs. Because the models from Google Colab and ESP32 are identical, it is anticipated that they will produce same predictions based on identical inputs.

2.5 Evaluation of Model Accuracy

The accuracy of Artificial Intelligence (AI) models during training and real-world application is determined by calculating the coefficient of determination or R-squared (R^2), mean square error (MSE), and root mean square error (RMSE) between anticipated and measured wind speed. R^2 measures the similarity between the expected and real data. R^2 can range from 0% to 100%; the greater the R^2 value, the closer the predicted values are to the actual values. Consequently, the R^2 percentage will be utilised as the model's accuracy percentage during the training process. R^2 can be calculated using the following Eq. (3)

$$R^2 = \left(1 - \frac{\sum(\text{actual value} - \text{predicted value})^2}{\sum(\text{actual value} - \text{mean of actual values})^2}\right) \times 100\% \tag{3}$$

Mean square error (MSE) can be used to evaluate AI models because it represents the square of the distances (errors) between projected and actual wind speeds. MSE is calculated by averaging the set of errors. The lower the MSE score, the more closely predicted data match actual data. MSE can be proven using Eq. (4), where n is the number of data points.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (actual\ value - predicted\ value)^2 \tag{4}$$

2.6 Anemometer measurement

During the training of AI models, the 801 wind speed data contained in the training dataset are compared to the AI models' anticipated wind speeds. During real-world applications, anemometer-measured wind speeds will be used to evaluate the performance of AI models for predicting wind speed. As shown in Figure 10, an anemometer is used to determine wind speed. The anemometer is placed in the opposite direction of the wind; if there is wind, the pinwheel on the anemometer will whirl, and the anemometer will display the actual wind speed. The maximum wind speed measured within 10 minutes of the commencement of data collection will be recorded and compared to the projected wind speed.

Fig. 10. Measure wind speed using anemometer

2.7 Hardware and Circuit Design

For circuit design, two ESP32 NODEMCU development boards, one DHT22 temperature and humidity sensor, and two 2004 I2C LCE modules are utilised in this work. Two ESP32 microcontrollers are utilised, one for the Multiple Linear Regression (MLR) model and the other for the Artificial Neural Network (ANN) model. The signal from the DHT22 temperature and humidity sensor is linked to the General Purpose Input Output (GPIO) pin 4 of both ESP32 microcontrollers. The positive power source of the DHT22 sensor can be connected to either of the ESP32 microcontrollers, but the ground pin of DHT22 must be connected to the ground pin of both ESP32 microcontrollers as a common ground in order for both ESP32 microcontrollers to receive the same signal from the DHT22 sensor.

Both ESP32 microcontrollers are linked to LCD displays for displaying DHT22 sensor readings and wind speed predictions. The I2C module enables ESP32 to be connected to an LCD screen using only four pins: GPIO pin 22 is connected to the SCL, pin 21 is connected to the SCA of the I2C module, and two pins are utilised for the power source: positive voltage and ground. SCL is the clock line used to synchronise I2C bus data, whereas SCA is the data line. I2C bus enables the ESP32 development board to have additional GPIO pins for additional functionality. Figure 11 shows the circuit schematic diagram for this paper.

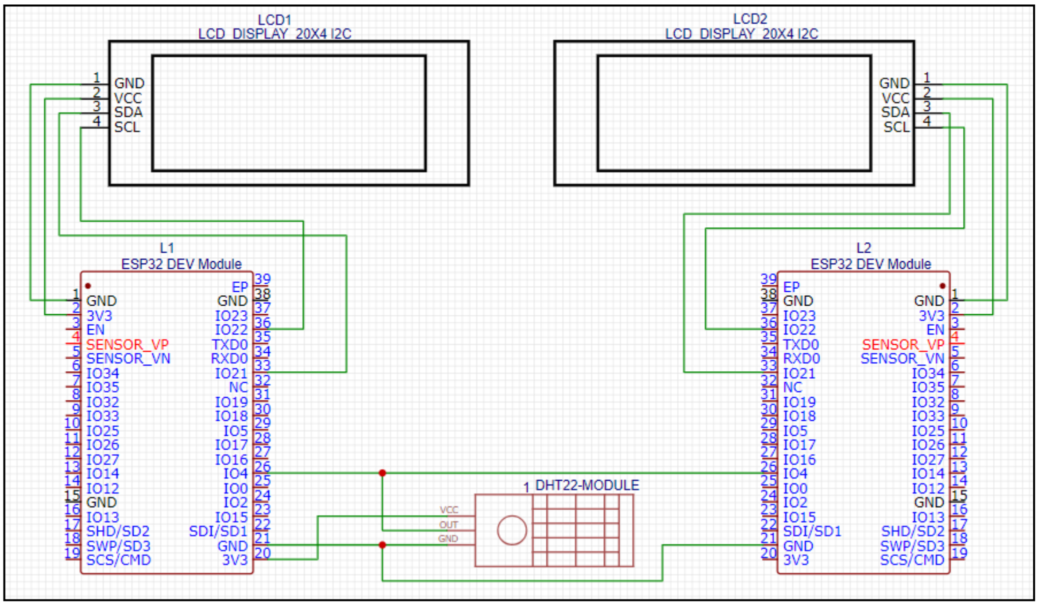


Fig. 11. Circuit Schematic Diagram

In order to estimate wind speed, the MLR and ANN models in the ESP32 microcontrollers will continuously receive temperature and humidity measurements from the DHT22 temperature and humidity sensor. The anticipated wind speed will be shown on LCD screens, allowing it to be recorded by observations as illustrated in Figure 12 and compared with anemometer-measured wind speed.

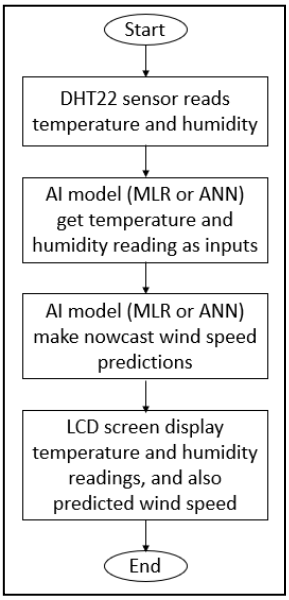


Fig. 12. Flowchart of wind speed prediction on ESP32

3. Results

There are 50 data taken between 8 May 2022 and 21 May 2021 at random times and weather conditions. The information is depicted in Table 2. The rows in Table 2 that are coloured pink represent data collected during periods of precipitation.

Journal of Advanced Research in Fluid Mechanics and Thermal Sciences
Volume 107, Issue 1 (2023) 29-44

Table 2
Data of temperature, humidity, predicted and measured wind speed

Date	Time	Temperature	Humidity	Predicted Wind Speed (MLR)	Predicted Wind Speed (ANN)	Measured Wind Speed
8/5/2022	8:00	30.2	72.0	7.42	9.13	8.2
8/5/2022	20:00	28.2	79.4	5.25	5.47	5.8
8/5/2022	22:00	28.0	78.2	5.30	5.63	6.2
9/5/2022	13:00	32.0	61.2	9.94	10.43	11.4
9/5/2022	21:00	28.4	75.2	5.94	6.97	6.5
10/5/2022	12:30	31.4	63.2	9.32	10.14	8.1
10/5/2022	15:30	37.7	42.0	15.88	11.54	14.4
11/5/2022	15:00	30.4	65.2	8.46	9.62	9.1
11/5/2022	21:20	28.1	83.0	4.70	4.27	5.2
12/5/2022	0:10	28.2	83.4	4.71	4.25	4.5
12/5/2022	15:00	31.9	67.0	9.10	10.16	9.1
12/5/2022	16:30	31.8	67.7	8.94	10.08	9.4
12/5/2022	17:20	29.7	66.3	7.91	9.27	8.2
12/5/2022	21:15	28.5	68.0	6.97	8.62	8.2
13/5/2022	13:00	33.0	59.3	10.79	10.77	11.2
13/5/2022	16:00	34.0	55.0	11.95	11.20	10.3
13/5/2022	20:00	28.5	74.1	6.15	7.42	6.5
14/5/2022	8:00	29.1	76.0	6.24	7.47	6.8
14/5/2022	14:30	32.6	56.1	10.99	10.67	10.9
14/5/2022	17:15	28.8	84.4	4.92	4.56	9.5
14/5/2022	20:15	28.4	72.6	6.29	7.78	6.8
14/5/2022	22:00	28.2	76.3	5.67	6.42	5.9
15/5/2022	12:00	32.2	61.3	10.05	10.56	10.4
15/5/2022	13:00	32.5	62.2	10.05	10.54	9.5
15/5/2022	17:00	30.7	69.3	8.11	9.54	8.2
15/5/2022	20:00	29.7	74.4	6.75	8.54	7.3
16/5/2022	1:00	28.6	78.2	5.59	6.32	5.7
16/5/2022	3:00	28.6	77.6	5.73	6.51	6.1
16/5/2022	13:00	26.8	87.6	3.32	2.83	9.1
17/5/2022	9:00	26.8	68.5	5.91	7.32	6.7
17/5/2022	12:00	32.2	49.4	11.66	10.58	8.9
17/5/2022	14:30	32.3	48.1	11.90	10.72	10.2
17/5/2022	22:00	32.2	62.0	8.74	9.69	8.9
18/5/2022	9:45	31.2	63.5	9.10	10.05	10.1
18/5/2022	12:15	31.8	57.5	10.33	10.34	9.5
18/5/2022	14:15	32.0	54.6	10.84	10.56	11.0
18/5/2022	15:15	32.1	58.2	10.34	10.48	9.7
19/5/2022	11:00	32.6	52.4	11.49	10.78	8.4
19/5/2022	16:45	31.6	61.2	9.71	10.33	8.8
19/5/2022	18:00	30.1	66.5	8.11	9.43	8.7
19/5/2022	22:00	28.2	77.5	5.51	6.05	6.5
20/5/2022	1:20	29.3	68.3	7.47	9.00	8.5
20/5/2022	9:45	32.2	75.8	8.08	9.77	8.0
20/5/2022	18:00	32.0	66.6	9.21	10.22	10.5
21/5/2022	8:00	26.5	72.9	5.14	5.68	6.0
21/5/2022	10:00	30.0	70.2	7.55	9.20	8.9
21/5/2022	12:00	33.2	47.5	12.51	10.87	10.4
21/5/2022	14:00	35.1	47.2	13.66	11.39	11.4
21/5/2022	16:00	33.1	56.3	11.25	10.85	11.0
21/5/2022	21:00	29.7	70.8	7.29	8.98	8.2

The performance of AI models is evaluated based on data collected under two conditions: data recorded while rainy conditions are included and data recorded during rainy conditions are excluded. Table 3 depicts the R^2, MSE, and RMSE values between predicted and measured data for the MLR and ANN models.

Table 3
Comparison of the accuracy for MLR and ANN Model

Parameter	Data during raining included (50 data)		Data during raining excluded (48 data)	
	MLR model	ANN model	MLR model	ANN model
R^2	43.048%	43.025%	71.029%	75.866%
MSE	2.200	2.200	1.159	0.965
RMSE	1.483	1.484	1.076	0.982

When data collected during a rainstorm is included in the AI models' accuracy analysis, both the MLR and ANN models perform poorly, with R^2 values below 50%. R^2 achieved by the MLR model increases from 43.084% to 71.029%, with MSE decreasing from 2.200 to 1.159 and RMSE decreasing from 1.483 to 1.019, as a result of excluding the data collected during periods of precipitation during the AI models' accuracy analysis. R^2 increases from 43.025% to 75.886% for the ANN model, while MSE and RMSE decrease from 2.200 to 0.965 and 1.484% to 0.982, respectively. The results indicate that both the MLR and ANN models function optimally only during non-rainy conditions and are not optimal for use during rainy conditions. This is due to the fact that data collected during periods of precipitation were omitted from AI model training. The AI algorithms produce forecasts based on the relationship in which wind speed increases as temperature rises and decreases when high humidity levels are identified. Consequently, the AI models will forecast low wind speed during rainy conditions with low temperature and high humidity. The average accuracy of AI models will be affected by the presence of strong wind speeds during rainy conditions. In addition, the ANN model outperforms the MLR model in terms of greater R^2, lower MSE, and lower RMSE when data acquired during rainfall is omitted. This may result from the fact that the relationship between wind speed, humidity, and temperature is not linear but rather complex. In this scenario, an ANN model might be more suited to deal with the nonlinearity between wind speed, temperature, and humidity [25].

4. Conclusions

In this paper, the MLR and ANN models were successfully deployed for predicting wind speed depending on the surrounding temperature and humidity at the conclusion of the investigation. The models' performance was evaluated by comparing the wind speed predicted by AI models to the wind speed recorded using an anemometer. During non-rainy conditions, the MLR model and ANN can reach an accuracy of 71.029% and 75.866%, respectively. With a difference in accuracy of 4.837%, ANN outperforms the MLR model. For future works, these methods can be used to forecast the air quality at atmosphere pollution control.

Acknowledgement
This research was funded by a grant from Asia Technological University Network (ATU-Net) Young Researcher Grant (YRG) 2021 R.K130000.7343.4B776).

References

[1] Li, Wei, Shengyu Kang, Yueqiang Sun, Weihua Bai, Yuhe Wang, and Hongqing Song. "A Machine Learning Approach for Air-Quality Forecast by Integrating GNSS Radio Occultation Observation and Weather Modeling." *Atmosphere* 14, no. 1 (2022): 58. https://doi.org/10.3390/atmos14010058

[2] Adeh, Melisa Anak, Mohd Ibrahim Shapiai, Ayman Maliha, and Muhammad Hafiz Md Zaini. "Tracking-learning-detection using extreme learning machine with Haar-like features." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 5, no. 2 (2016): 1-11.

[3] Omairi, Amzar, and Z. H. Ismail. "Modeling battery state of charge in wireless sensor networks based on structured multi-layer perceptron." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 5, no. 2 (2016): 36-45.

[4] Arun. "An Introduction to TinyML,", November 10, 2020. https://towardsdatascience.com/an-introduction-to-tinyml-4617f314aa79/.

[5] Deepanshi, "All you need to know about your first Machine Learning model – Linear Regression," May 25, 2021. https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/

[6] Tsoumakas, Grigorios. "A survey of machine learning techniques for food sales prediction." *Artificial Intelligence Review* 52, no. 1 (2019): 441-447. https://doi.org/10.1007/s10462-018-9637-z

[7] Khalid, Syed Ghufran, Jufen Zhang, Fei Chen, and Dingchang Zheng. "Blood pressure estimation using photoplethysmography only: comparison between different machine learning approaches." *Journal of healthcare engineering* 2018 (2018). https://doi.org/10.1155/2018/1548647

[8] Lee, Mike, Larry Lin, Chih-Yuan Chen, Yu Tsao, Ting-Hsuan Yao, Min-Han Fei, and Shih-Hau Fang. "Forecasting air quality in Taiwan by using machine learning." *Scientific reports* 10, no. 1 (2020): 4153. https://doi.org/10.1038/s41598-020-61151-7

[9] David, Robert, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger et al. "Tensorflow lite micro: Embedded machine learning for tinyml systems." *Proceedings of Machine Learning and Systems* 3 (2021): 800-811.

[10] "uTensor/uTensor: TinyML AI inference library." https://github.com/uTensor/uTensor

[11] D. Dennis *et al.*, "EdgeML." https://github.com/microsoft/EdgeML

[12] "eloquentarduino/EloquentTinyML: Eloquent interface to Tensorflow Lite for Microcontrollers." https://github.com/eloquentarduino/EloquentTinyML/

[13] Estrebou, César Armando, Martín Fleming, Marcos David Saavedra, and Federico Adra. "A neural network framework for small microcontrollers." In *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 al 8 de octubre de 2021)*. 2021.

[14] Chen, Shia-Chung, Jibin Jose Mathew, Ching-Te Feng, and Tzu-Jeng Hsu. "An Innovative Method to Monitor and Control an Injection Molding Process Condition using Artificial Intelligence based Edge Computing System." In *2022 8th International Conference on Applied System Innovation (ICASI)*, pp. 41-45. IEEE, 2022. https://doi.org/10.1109/ICASI55125.2022.9774445

[15] Estrebou, César Armando, Martín Fleming, Marcos David Saavedra, and Federico Adra. "MbedML: A Machine Learning Project for Embedded Systems." In *IX Jornadas de Cloud Computing, Big Data & Emerging Topics (Modalidad virtual, 22 al 25 de junio de 2021)*. 2021.

[16] WIA, "Espressif ESP32." https://www.wia.io/things/espressif-esp32

[17] Verma, Manoj, Harish Kumar Ghritlahre, and Ghrithanchi Chandrakar. "Wind speed prediction of central region of Chhattisgarh (India) using artificial neural network and multiple linear regression technique: a comparative study." *Annals of Data Science* (2021): 1-23. https://doi.org/10.1007/s40745-021-00332-1

[18] Barhmi, Soukaina, Omkaltoume Elfatni, and Ismail Belhaj. "Forecasting of wind speed using multiple linear regression and artificial neural networks." *Energy Systems* 11 (2020): 935-946. https://doi.org/10.1007/s12667-019-00338-y

[19] Arzu, Arieni, Mohammed R. Ahmed, and Mohammad GM Khan. "Wind speed forecasting using regression, time series and neural network models: a case study of Suva." In *Australasian Fluid Mechanics Conference*, p. TBC. The University of Queensland, 2020. https://doi.org/10.14264/ccee311

[20] Ehsan, Md Amimul. "Predictive models for wind speed using artificial intelligence and copula." *arXiv preprint arXiv:2107.06182* (2021).

[21] Al-Zadjali, Saira, Ahmed Al Maashri, Amer Al-Hinai, Rashid Al Abri, Swaroop Gajare, Sultan Al Yahyai, and Mostafa Bakhtvar. "A fast and accurate wind speed and direction nowcasting model for renewable energy management systems." *Energies* 14, no. 23 (2021): 7878. https://doi.org/10.3390/en14237878

[22] Zucatelli, Pedro Junior, Erick Giovani Sperandio Nascimento, Alex Álisson Bandeira Santos, and Davidson Martins Moreira. "Nowcasting prediction of wind speed using computational intelligence and wavelet in Brazil." *International Journal for Computational Methods in Engineering Science and Mechanics* 21, no. 6 (2020): 343-369.

[23] Dec, Ewelina, Bożena Babiarz, and Robert Sekret. "Analysis of temperature, air humidity and wind conditions for the needs of outdoor thermal comfort." In *E3S Web of Conferences*, vol. 44, p. 00028. EDP Sciences, 2018. https://doi.org/10.1051/e3sconf/20184400028

[24] Zakaria, Nur Hidayah, S. A. Salleh, Arnis Asmat, Andy Chan, Nurul Amirah Isa, Nur Atiqah Hazali, and M. A. Islam. "Analysis of wind speed, humidity and temperature: variability and trend in 2017." In *IOP Conference Series: Earth and Environmental Science*, vol. 489, no. 1, p. 012013. IOP Publishing, 2020. https://doi.org/10.1088/1755-1315/489/1/012013

[25] "Past Weather in Setapak, Malaysia." https://www.timeanddate.com/weather/@1735162/historic

[26] Google. "Google Colab." https://research.google.com/colaboratory/faq.html

[27] "TensorFlow." https://www.tensorflow.org/

[28] B. Satrom. "Getting started with Machine Learning on MCUs with TensorFlow – Particle Blog." 2019. https://blog.particle.io/particle-machine-learning-101/