

## Distant temperature and humidity monitoring: prediction and measurement

Farrukh Hafeez<sup>1</sup>, Usman Ullah Sheikh<sup>2</sup>, Attaullah Khidrani<sup>3</sup>, Muhammad Akram Bhayo<sup>4</sup>,  
Saleh Masoud Abdallah Altbawi<sup>5</sup>, Touqeer Ahmed Jumani<sup>6</sup>

<sup>1,2,5</sup>School of Electrical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia

<sup>3</sup>Faculty of Electrical Engineering, Balochistan University of Engineering and Technology, Khuzdar, Pakistan

<sup>4</sup>Department of Electrical Engineering, Quaid-e-Awam University of Engineering, Science and Technology,  
Nawabshah, Pakistan

<sup>6</sup>Department of Electrical Engineering, Mehran University of Engineering and Technology, Khairpur, Pakistan

---

### Article Info

#### Article history:

Received Jun 5, 2021

Revised Oct 14, 2021

Accepted Oct 27, 2021

---

#### Keywords:

Android app

Recurrent neural network

Remote sensing

Robotic car

Sensor

---

### ABSTRACT

Sensing environmental measuring parameters has a pivotal role in our everyday lives. Most of our daily life activities depend upon environmental conditions. Accurate information about these parameters also helps in several industrial applications like ventilation rate calculation, energy prediction, stock maintenance in warehouses, and saving from harmful conditions. The emergence of machine learning can make it easy to predict such time series problems. This paper describes the design of a remotely controlled robotic car for measuring and predicting humidity and temperature. A customized app for accessing the robotic car is designed to indicate predicted and real-time measured values of humidity and temperature. A sensor installed built-in helps in the measurement. The recurrent neural network (RNN) model is used to predict humidity and temperature. For this purpose, experiments are carried out in both outdoor and indoor settings. Accuracy of 85% and 90% is achieved in an outdoor environment and indoor settings.

*This is an open access article under the [CC BY-SA](#) license.*



---

### Corresponding Author:

Farrukh Hafeez

School of Electrical Engineering

Universiti Teknologi Malaysia

81310 Skudai, Johor Bahru, Johor-Malaysia

Email: hafeez@graduate.utm.my

---

## 1. INTRODUCTION

In the last decade, artificial intelligence-based techniques development had a great impact on all fields. Machine learning algorithms are based on investigating patterns, data, and trends that produced promising results in the fields of language processing, predictions, self-driving car, health care, classifications, and surveillance systems [1]-[4]. Popular search engines like Youtube, Google, and social media sites, such as Instagram, Twitter, and Facebook own their specific learning tools [5]-[7]. Weather forecasting, earthquake predictions, rain expectations have also been explored in the metrology field [8]-[13]. Internet of things (IoT) is another widely used technology around the globe. According to some analysts, the IoT revolution is much significant than the industrial revolution. Some of its application includes Smart homes, banking, retails, health care, agriculture, and industries. IoT development was more prominent in recent years. However, high-speed and affordable smartphones are one of the crucial factors. IoT applications can be easily assessed by mobile phone apps than web apps. However, both of these technologies are being used by scientists and researchers [14]-[17]. When artificial intelligence (AI) is used in conjunction with IoT,

as it is done on clouds, such as predictive engines, the end device or temporary node stores real-time data provided by sensors. The data is sent to the cloud by nodes or end devices, where it is processed by a predictive engine leading to a decision that is sent to the control devices. All of this happens without human interference, results in a decline in maintenance cost. This paper provides a model for predicting and measuring the humidity and temperature in any location by using Wi-Fi to connect to a robotic vehicle. This will be done in both an outdoor and indoor environment. The recurrent neural network is used to predict humidity and temperature. The robotic car's sensor is used for the measurements. The DHT22 sensor for humidity and temperature assessment, an internet protocol (IP) camera for driving the robotic car to the required place, and a node microcontroller unit (NodeMCU) ESP32 for monitoring and connecting, are the main components of the robotic car. The customized app is designed to indicate the car's current position with measured and predicted values of humidity and temperature. Variation in temperature ranges from 4-48 °C and variation in humidity ranges from 36-80%. In this work, instead of relying on measurement or prediction that were used mostly in previous researches, both approaches; measurement as well as prediction are used simultaneously, to do comparison and analysis for getting maximum accuracy and precision in the desired sensing parameters.

The remaining paper is structured in the following manner: section 2 discusses the overview of the research conducted on weather forecasting and IoT machine learning development. The first subsection describes the technical definition of the robotic car; the second subsection discusses App development for remote location access, and the third subsection introduces the recursive neural network. In section 4, given the experimental results and evaluation are outlined. Eventually, the conclusion of the research and future perspective are presented.

## 2. PREVIOUS WORK

Forecasting weather is not new; it has existed since the beginning of time, but it is not limited to humans. All animals, whether on-air, water, or land predict future conditions. Humans used to use nature clues and sayings to forecast the weather before the development of scientific methods. The design of sophisticated instruments resulted in the scientific analysis of meteorology. Machine learning-based and statistical approaches to temperature prediction are dual means of modern approaches. Regression approaches such as non-linear and multiple linear regressions are the basis of most statistical models [18], [19]. Gardner *et al.* [20] used Gaussian process and Kriging (KRIG) regression methods for temperature prediction. Further, generalized additive modeling (GAM) is used by some researchers. Wang *et al.* [21] proposed the GAM in the prediction model and consequently got successful in finding an effective association between environmental variables and predictive variables. Other researchers used hybrid methods to get optimized results besides these statistical techniques. Despite this, forecasts were found to be too far off the ground. In the year 2010, a new age of the modern machine and deep learning began to develop. The popularity of machine learning in other fields urges metrological researchers to apply in the area. According to Dagli [22], it is not possible to assimilate all the data using conventional methods.

Weather forecasting involves parameters such as humidity and temperature, snow or rain forecasting, barometric pressure, and wind speed forecasting. Researchers have proposed various temperature models for diverse environmental conditions based on a wide range of studies. For instance, the models proposed in [23], [24] forecasted the indoor temperature, and the model in [25], [26] evaluated the temperature prediction for the long-term. The conventional signal processing methods are replaced by the latest deep learning models like gated recurrent units (GRU), long short-term memory (LSTM), and convolutional neural network (CNN). Hossain *et al.* [27] used deep neural networks along with standard neural networks and stacked denoising auto-encoders (SDAE) using different parameters. SDAE achieved an average accuracy of 97.94%, while conventional ANN achieved an accuracy of 94.92%. Recurrent neural network (RNN), convolutional neural network (CNN) models, and conditional restricted boltzmann machine (CRBM) were contrasted in Salman *et al.* [28] research. The accuracy of rainfall prediction using recurrent neural is found to be maximum. Because of its superior success in multivariate time-series forecasting, LSTM, a variant of RNN, has gotten a lot of publicity. Goap *et al.* [29] proposed a smart irrigation model based on a field sensor and meteorology. Using weather prediction and soil moisture data from the internet, the model predicted field irrigation is required. Rahayu *et al.* [30] applied long short-term memory (LSTM) to predict daily temperatures over the next three days with five classes, namely "Cold", "Cool", "Normal", "Warm" and "Hot". Testing accuracy is found to be 90.92% and training is claimed to be 80.36% for testing data. Zhang *et al.* [31] applied convolutional LSTM to measure surface, but the subsurface temperature to predict 3-D ocean temperature. Results demonstrated the overall higher accuracy than earlier studies.

### 3. METHODOLOGY

#### 3.1. Working mechanism

Figure 1 depicts a block diagram of the method. The humidity-temperature measuring sensor DHT22 on the robotic car measures real-time humidity-temperature data at the current spot. A mounted IP camera allows the robotic car to be transported to any position inside or outside. The NodeMCU system, which is mounted on the robotic car, handles all of the computing and information processing. Custom applications built on the Android platform are used to view real-time data and monitor the robotic car. RNN is applied to forecast humidity and temperature using real-time recorded data. This predicted value would appear alongside the measured value in the event of sensor failure or to compare the predicted and measured values. The lab is chosen as an inside location for prediction, while for the outside location, the outside building location is considered. For both locations, data is collected for a year. The mounted sensors collect indoor position data, while the local meteorological data center obtains outdoor data. For training and research, a set daytime is used. In the following segment, the RNN algorithm is discussed in detail. There are three sections of the functional definition. The first section addressed device hardware; the second section elaborated on customizable application development, and the third section clarified the RNN algorithm's application.

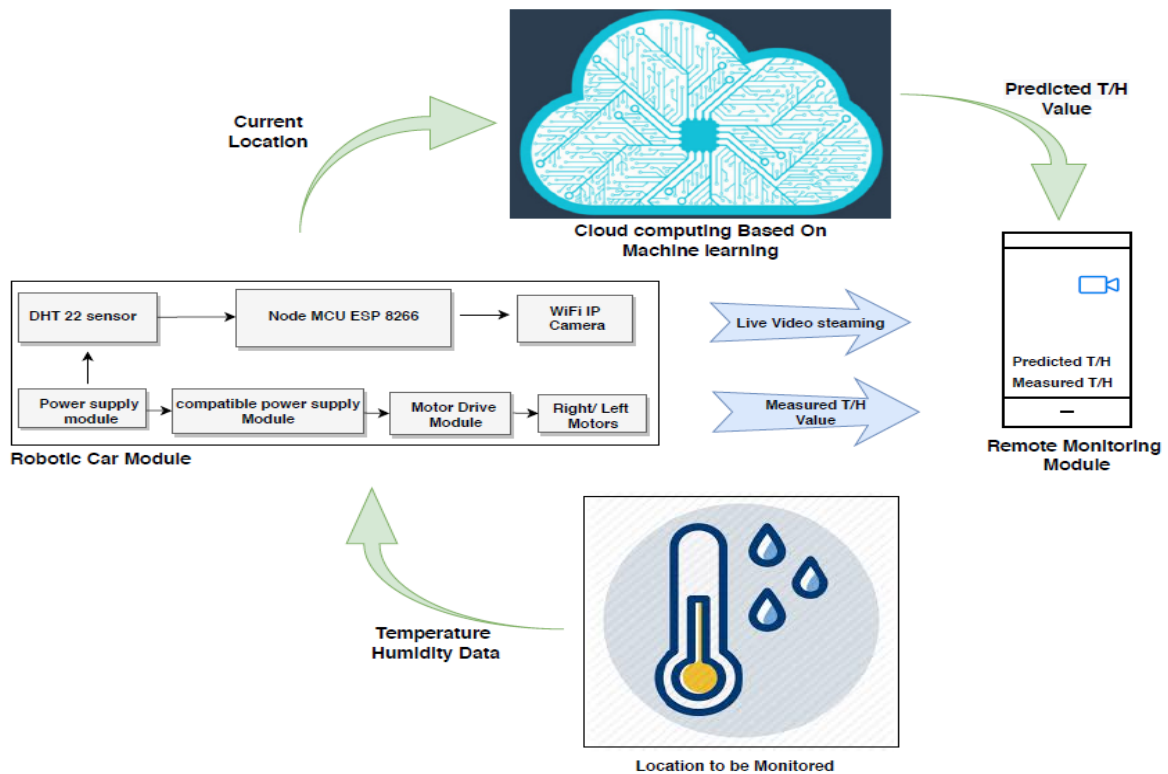


Figure 1. Block diagram with main components

#### 3.2. Robotic car module functional description

The NodeMCU ESP 8266 acts as the processing and controlling device for the robotic car, while the IP camera allows monitoring to work from a remote position, the DHT22 allows humidity and temperature measurement, the motor and driver module with DC motor for the robotic car's motion. The NodeMCU incorporates and monitors hardware while still communicating with the mobile app and cloud server. The NodeMCU is programmed in LUA, and its built-in Wi-Fi feature allows for easier data sharing between the mobile app and cloud server. Figure 2 depicts the primary functional phases. DHT22 sensor measures humidity and temperature with a precision of  $\pm 1\%$  and  $\pm 1^\circ\text{C}$ , respectively, from 0-100 percent and  $-40^\circ\text{C}$  to  $80^\circ\text{C}$ . DHT22 measures temperature with a negative temperature coefficient NTC temperature sensor and humidity with a combination of two electrodes and a moisture-holding substrate.

The sensor of 40 bits of serial data is processed; the current value of humidity and temperature values are obtained and sent to the mobile app. The motor driver module is linked between the motors and NodeMCU

to ensure that the voltage-current requirements of the DC motor. An IP camera is attached to a robotic car for location tracking and motion. The camera's Wi-Fi, IP addresses, and baud rate are set first, followed by the rest of the parameter values. For remote mobile app control, specified parameter values are used.

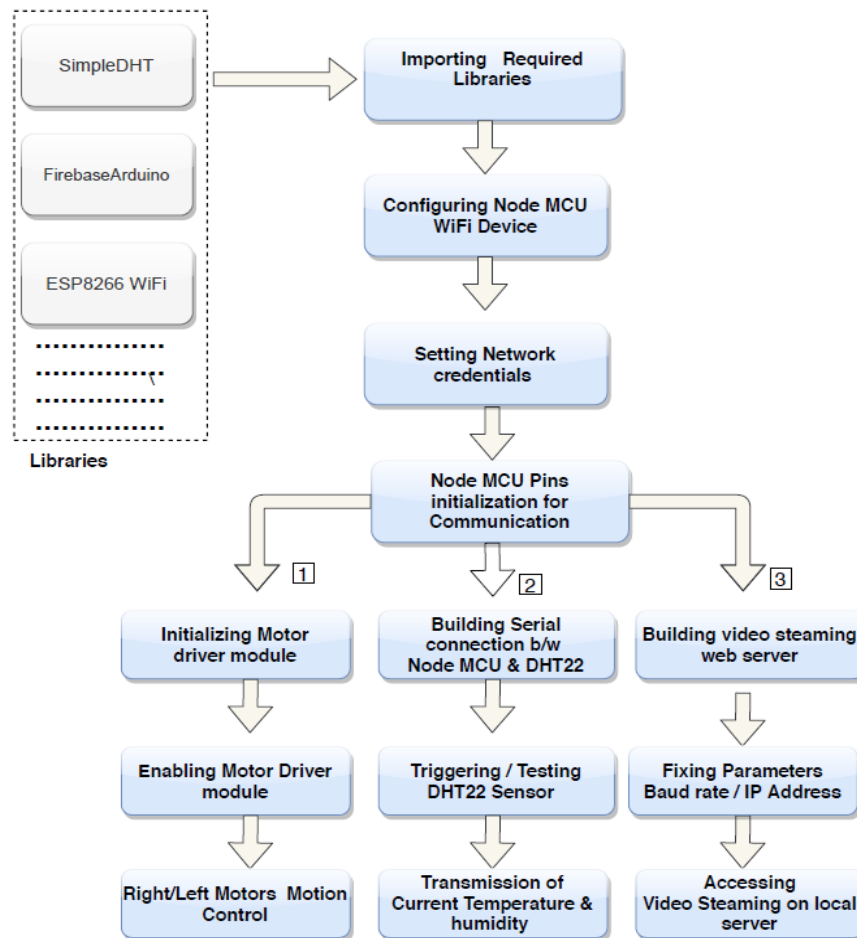


Figure 2. Robotic car functional block view

### 3.3. Mobile app architectural working

The customized app is based on the Android platform for accessing remote locations. Figure 3 shows a flowchart with four branches, each specifying a process's mobile app steps in sequence. The first branch explains how to link an IP camera to a mobile app. After searching for a live streaming module with Camera admin credentials, a live video session is created using content delivery networks (CDN) for offering premium video quality and an additional layer of protection. As video streaming begins, the video's pixel, resolution, and bit rate are scaled to fit the app's size. Branch 2 depicts a robotic car taking care of the pre-defined steps. The NodeMCU is used to initialize the app based on the motor pins specified in the hardware. Both the App and NodeMCU parameters are synchronized and the board's test LED is used for monitoring. Four-way directions are tested using four arrows on the mobile app after the link is established and LED checking. Branch no. 3 depicts the measures involved in collecting temperature and humidity values from the NodeMCU and displaying them on the app. Before connecting with the app, the sensor sends serial data transmission and NodeMCU must be checked. Once the app link is established, a moving robotic car is used to test the continuous updating of humidity and temperature values in various conditions. The steps in Branch No. 4 explain the initial steps in connecting the app to cloud data to display expected temperature and humidity values. A cloud server is made accessible, and the obtained data must be converted to the format of a mobile app.

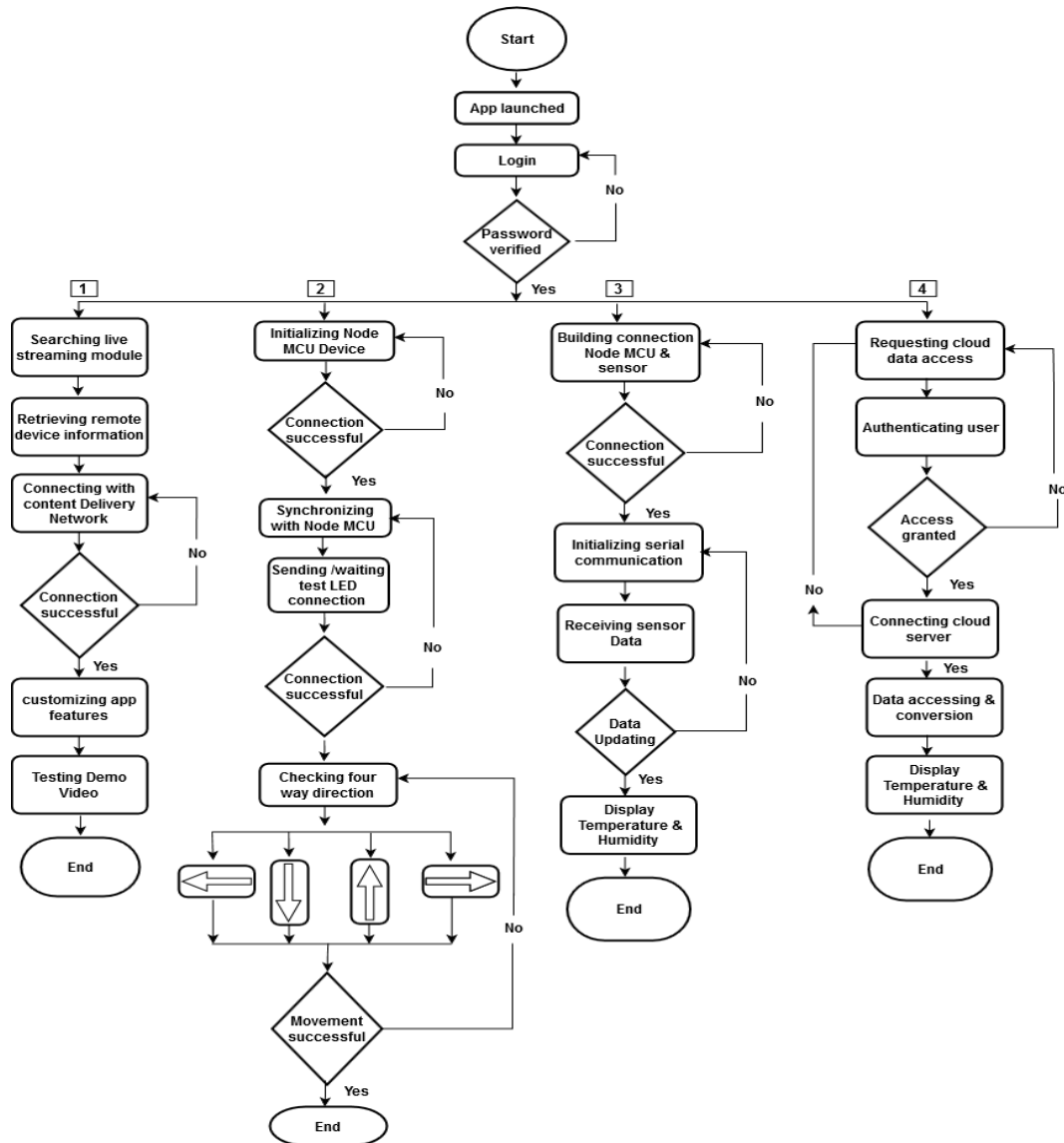


Figure 3. Android application process flow chart

**3.4. Temperature and humidity prediction system using RNN**

The RNN model is used to estimate humidity and temperature. The indoor data collection is made up of lab data from the mounted sensor, while the outdoor data is sourced from a global weather database. To keep the network's training quality, appropriate data splitting is needed for training, authentication, and analysis. For testing and training, the data set is divided into 20% and 80%, respectively. Tensor flow version 2.3.0 is used to design the model in Python 2.7 with Keras Library. Table 1 shows certain hyperparameters that contribute to minute differences. Hyper-parameters such as epoch, mean absolute error, and learning rate are considered to be key elements for forecasting efficiency.

Table 1. Summary of RNN model's parameter values

Hyper parameters	Values
Loss Function	MSE
Learning Rate	0.00069
Recurrent Step's Activation Function	sigmoid
Activation Function	tanh
Neurons in Layer 1	32
Neurons in Layer 2	54
Epochs	250
Optimizer	Adam

One of the initial steps after data preprocessing is to find the relation between variables and underlying factors. This is accomplished by the use of factor analysis (FA). The Kaiser-Meyer-Olkin (KMO) test is used to determine the dataset factorability. For selecting the number of variables, the Kaiser criteria and the scree plot are used. Table 2 summarizes the results of different learning rates for performance following error variation. During training, various learning rates were selected in this model. During training, learning rates of 0.00025, 0.00069, 0.00075, 0.001, 0.01, and 0.033 generated minimum losses with reasonable training speed. The model's output was evaluated using mean absolute error loss to identify acceptable hyper-parameters. The difference in various learning rates from models 1-6 is obvious; revealing that model 6 performs the best with a learning rate is of 0.00069.

Table 2. Analysis of various learning rates in terms of performance

Model	MSLE	RMSE	learning rates	MAE	Epoch
1	0.0019	3.8831	0.01	1.4042	250
2	0.0118	21.6369	0.033	2.9329	250
3	0.0047	7.8743	0.001	1.859	250
4	0.0117	11.0040	0.00075	2.1676	250
5	4.8762	23.5233	0.00025	3.0356	250
6	0.0070	5.8501	0.00069	2.1815	250

Testing for forecasting is conducted after training and tuning of the RNN network. Mean squared logarithmic error (MSLE), root mean squared error (RMSE), and mean absolute error (MAE) between  $Y_{\text{predicted}}$  (predicted value) and  $Y_{\text{real}}$  (real value) are used to evaluate the performance at time  $t$  and  $N$ , total evaluation numbers. The formulae for all of them is expressed as:

$$\text{RMSLE} = \frac{1}{N} \sum_{t=1}^N \left( \log(Y_{\text{real}} + 1) - \log(Y_{\text{predicted}} + 1) \right)^2 \quad (1)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^N (Y_{\text{predicted}} - Y_{\text{real}})^2}{N}} \quad (2)$$

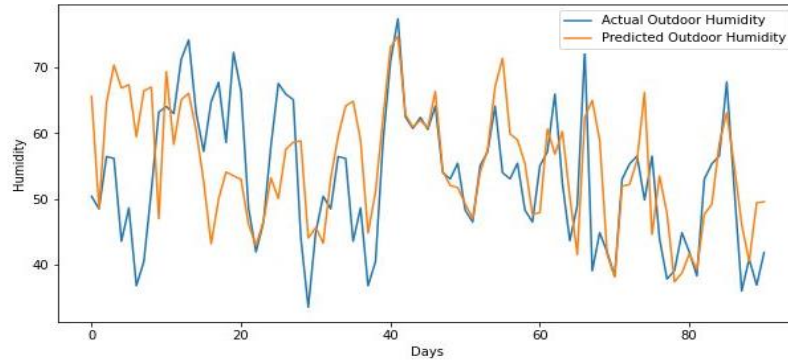
$$\text{MAE} = \frac{1}{N} \sum_{t=1}^N |Y_{\text{predicted}} - Y_{\text{real}}| \quad (3)$$

#### 4. RESULTS AND DISCUSSION

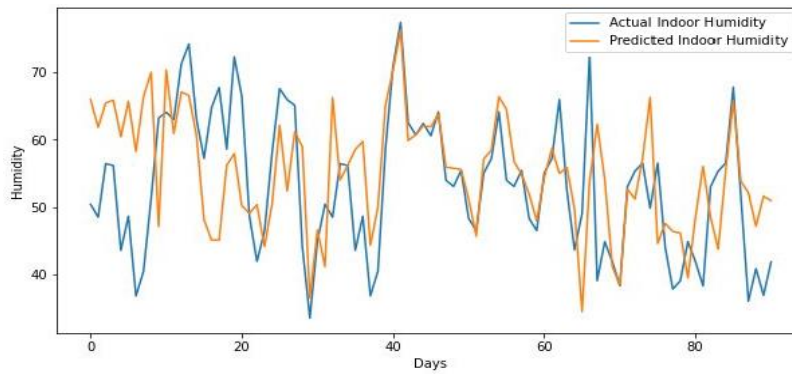
Figure 4 shows the difference between the (expected and real) temperatures and humidity over 90 hours. The result indicates that the RNN system is a reliable method to predict outdoor and indoor humidity and temperature. In the indoor area, the difference in (predicted and real) temperature value ranges from 0.85-5.36 °C. On the contrary, the difference in the humidity (predicted and real values) range from 1.56-7.5%. The result depicts that the outdoor settings have a little higher difference in humidity and temperature. For instance, the temperature difference ranges from 1.35-8.36 °C, and the humidity difference ranges from 2.56-8.3% in outdoor conditions. Table 3 shows the comparison results of the defined parameters.

Table 3. Performance of RNN model

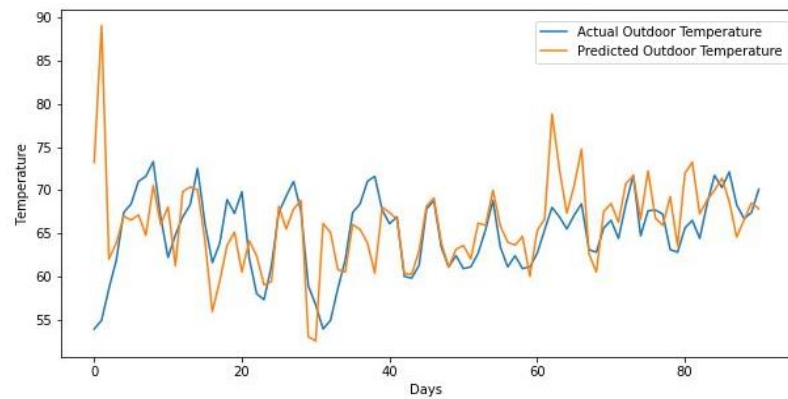
Location	Parameter	Criteria	Values
Lab	Temperature	Mean absolute error	1.38
		Root mean squared error (RMSE)	1.71
		Accuracy when prediction error is smaller than 3°C	90
	Humidity	Mean absolute error	2.31
		Root mean squared error (RMSE)	2.43
		Accuracy when prediction error is smaller than 3°C	92
Outside	Temperature	Mean absolute error	1.18
		Root mean squared error (RMSE)	1.63
		Accuracy when prediction error is smaller than 3°C	90
	Humidity	Mean absolute error	2.21
		Root mean squared error (RMSE)	2.56
		Accuracy when prediction error is smaller than 3°C	90



(a)



(b)



(c)

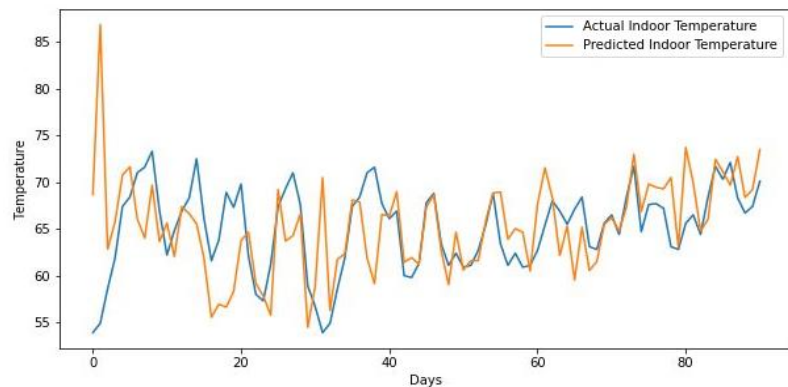


Figure 4. RNN Actual and predicted temperature and humidity graphs: (a) outdoor humidity, (b) outdoor temperature, (c) indoor humidity, and (d) indoor temperature

## 5. CONCLUSION AND FUTURE WORK

The hybrid model approach to examine the humidity and temperature is proposed in this paper. The hybrid model aims to compare measured and expected values in a single study. RNN, a data-driven machine learning algorithm, is developed to predict humidity and temperature. Testing and assessment are done in both an outdoor and an indoor environment. In contrast to the outdoor world, the statistical result reveals that indoor environments achieved high accuracy. To verify that the proposed model's accuracy must be improved. This can be accomplished by preparing more data for a longer time. A more sophisticated meteorology-related algorithm may also be used to obtain reliable prediction values. However, in this article, we have used a hybrid approach rather than relying solely on expected results. Using a sophisticated and dedicated humidity and temperature sensor will improve measurement accuracy.

## REFERENCES

- [1] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning-based natural language processing [Review Article]," *IEEE Computational Intelligence Magazine, Institute of Electrical and Electronics Engineers Inc.*, vol. 13, no. 3, pp. 55-75, Aug-2018, doi: 10.1109/MCI.2018.2840738.
- [2] B. Norgeot, B. S. Glicksberg, and A. J. Butte, "A call for deep-learning healthcare," *Nature Medicine, Nature Publishing Group*, vol. 25, no. 1, pp. 14-15, 01-Jan-2019, doi: 10.1038/s41591-018-0320-3.
- [3] R. Vij and B. Kaushik, "A survey on various face detecting and tracking techniques in video sequences," *International Conference on Intelligent Computing and Control Systems, ICCS 2019, 2019*, pp. 69-73, doi: 10.1109/ICCS45141.2019.9065483.
- [4] N. Mehdiyev, J. Evermann, and P. Fettke, "A Novel Business Process Prediction Model Using a Deep Learning Method," *Bus. Inf. Syst. Eng.*, vol. 62, no. 2, pp. 143-157, Apr. 2020, doi: 10.1007/s12599-018-0551-3.
- [5] M. H. Abd El-Jawad, R. Hodhod, and Y. M. K. Omar, "Sentiment analysis of social media networks using machine learning," in *ICENCO 2018, 14th International Computer Engineering Conference: Secure Smart Societies*, 2019, pp. 174-176, doi: 10.1109/ICENCO.2018.8636124.
- [6] O. Kolchyna, T. T. P. Souza, P. Treleaven, and T. Aste, "Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination," *International Journal of Emerging Technologies in Learning*, vol. 12, no. 3, pp. 45-56, Jul. 2015.
- [7] M. Santillana, A. T. Nguyen, M. Dredze, M. J. Paul, E. O. Nsoesie, and J. S. Brownstein, "Combining Search, Social Media, and Traditional Data Sources to Improve Influenza Surveillance," *PLOS Comput. Biol.*, vol. 11, no. 10, p. e1004513, Oct. 2015, doi: 10.1371/journal.pcbi.1004513.
- [8] A. Sharaff and S. R. Roy, "Comparative Analysis of Temperature Prediction Using Regression Methods and Back Propagation Neural Network," *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics*, 2018, pp. 45-56, doi: 10.1109/ICOEI.2018.8553803.
- [9] E. Sreehari, "Prediction of Climate Variable using Multiple Linear Regression," *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018, 2018*, pp. 7-10, doi: 10.1109/CCAA.2018.8777452.
- [10] I. Park, H. S. Kim, J. Lee, J. H. Kim, C. H. Song, and H. K. Kim, "Temperature prediction using the missing data refinement model based on a long short-term memory neural network," *Atmosphere (Basel)*, vol. 10, no. 11, pp. 1-16, 2019, doi: 10.3390/atmos10110718.
- [11] S. P. Menon, "Prediction of Temperature using Linear Regression," *International Conference on Electrical, Electronics, Communication Computer Technologies and Optimization Techniques, ICEECCOT 2017*, pp. 17-27, 2017, doi: 10.1109/ICEECCOT.2017.8284588.
- [12] S. P. Menon, R. Bharadwaj, P. Shetty, P. Sanu, and S. Nagendra, "Prediction of temperature using linear regression," *Int. Conf. Electr. Electron. Commun. Comput. Technol. Optim. Tech. ICEECCOT 2017*, vol. 2018-Janua, pp. 670-671, 2018, doi: 10.1109/ICEECCOT.2017.8284588.
- [13] T. Khan, M. Rabbani, S. M. Tanvir Siddiquee, and A. Majumder, "An Innovative Data Mining Approach for Determine Earthquake Probability Based on Linear Regression Algorithm," *Proc. 2019 3rd IEEE Int. Conf. Electr. Commun. Technol. ICECCT 2019, 2019*, pp. 7-10, doi: 10.1109/ICECCT.2019.8869286.
- [14] S. S. Patil and S. A. Thorat, "Early detection of grapes diseases using machine learning and IoT," in *Proceedings - 2016 2nd International Conference on Cognitive Computing and Information Processing, CCIP 2016, 2016*, doi: 10.1109/CCIP.2016.7802887.
- [15] P. S. Pandey, "Machine Learning and IoT for prediction and detection of stress," in *Proceedings of the 2017 17th International Conference on Computational Science and Its Applications, ICCSA 2017, 2017*, doi: 10.1109/ICCSA.2017.8000018.
- [16] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 2923-2960, 2018, doi: 10.1109/COMST.2018.2844341.
- [17] N. Baracaldo, B. Chen, H. Ludwig, A. Safavi, and R. Zhang, "Detecting poisoning attacks on machine learning in IoT environments," in *Proceedings - 2018 IEEE International Congress on Internet of Things, ICIOT 2018 - Part of the 2018 IEEE World Congress on Services, 2018*, pp. 57-64, doi: 10.1109/ICIOT.2018.00015.
- [18] O. Giustolisi, A. Doglioni, D. A. Savic, and B. W. Webb, "A multi-model approach to analysis of environmental phenomena," *Environ. Model. Softw.*, vol. 22, no. 5, pp. 674-682, May 2007, doi: 10.1016/j.envsoft.2005.12.026.



- [19] D. Caissie, "The thermal regime of rivers: A review," *Freshwater Biology*, vol. 51, no. 8. John Wiley & Sons, Ltd, pp. 1389-1406, 2006, doi: 10.1111/j.1365-2427.2006.01597.x.
- [20] B. Gardner, P. J. Sullivan, and A. J. Lembo, Jr., "Predicting stream temperatures: geostatistical model comparison using alternative distance metrics," *Can. J. Fish. Aquat. Sci.*, vol. 60, no. 3, pp. 344-351, Mar. 2003, doi: 10.1139/f03-025.
- [21] S. Wang, G. Morishima, R. Sharma, and L. Gilbertson, "The Use of Generalized Additive Models for Forecasting the Abundance of Queets River Coho Salmon," *North Am. J. Fish. Manag.*, vol. 29, no. 2, pp. 423-433, Apr. 2009, doi: 10.1577/M07-055.1.
- [22] C. H. Dagli, "Machine Learning Methods for Data Assimilation," *Intell. Eng. Syst. through Artif. Neural Networks*, vol. 20, pp. 105-112, 2010, doi: 10.1115/1.859599.paper14.
- [23] P. Romeu, F. Zamora-Martínez, P. Botella-Rocamora, and J. Pardo, "Time-Series Forecasting of Indoor Temperature Using Pre-trained Deep Neural Networks," *Artificial Neural Networks and Machine Learning – ICANN 2013*, pp. 451–458, 2013, doi: 10.1007/978-3-642-40728-4\_57.
- [24] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy Build.*, vol. 83, pp. 162-172, Nov. 2014, doi: 10.1016/j.enbuild.2014.04.034.
- [25] M. Hossain, B. Rekabdar, S. J. Louis, and S. Dascalu, "Forecasting the weather of Nevada: A deep learning approach," in *Proceedings of the International Joint Conference on Neural Networks*, 2015, doi: 10.1109/IJCNN.2015.7280812.
- [26] H. Lin, Y. Hua, L. Ma, and L. Chen, "Application of ConvLSTM network in numerical temperature prediction interpretation," in *ACM International Conference Proceeding Series*, vol. Part F148150, 2019, pp. 109-113.
- [27] M. Hossain, B. Rekabdar, S. J. Louis, and S. Dascalu, "Forecasting the weather of Nevada: A deep learning approach," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2015-Septe, pp. 2-7, 2015, doi: 10.1109/IJCNN.2015.7280812.
- [28] A. G. Salman, B. Kanigoro, and Y. Heryadi, "Weather forecasting using deep learning techniques," *ICACSI 2015 - 2015 Int. Conf. Adv. Comput. Sci. Inf. Syst. Proc.*, pp. 281-285, 2016, doi: 10.1109/ICACSI.2015.7415154.
- [29] A. Goap, D. Sharma, A. K. Shukla, and C. Rama Krishna, "An IoT based smart irrigation management system using Machine learning and open source technologies," *Comput. Electron. Agric.*, vol. 155, pp. 41-49, 2018, doi: 10.1016/j.compag.2018.09.040.
- [30] I. Sri Rahayu, E. C. Djamal, R. Ilyas, and A. T. Bon, "Daily temperature prediction using recurrent neural networks and long-short term memory," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2020.
- [31] K. Zhang, X. Geng, and X. H. Yan, "Prediction of 3-D Ocean Temperature by Multilayer Convolutional LSTM," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1303-1307, 2020, doi: 10.1109/LGRS.2019.2947170.