Arduino based Weather Monitoring and Forecasting System using SARIMA Time-Series Forecasting

[1]Danish Khan, [2]Dr. Rupesh C. Jaiswal

[1]Student, [2]Professor,
[1]Department of Electronics and Telecommunication,
[1]Pune Institute of Computer Technology, Pune, India.

Abstract: Arduino is an open-source electronics platform based on easy-to-use hardware and software. A DHT-11 temperature and humidity sensing module, monitors and stores the data onto a computer every ten minutes. A program is written in Python, which stores the minimum and maximum for the day. In addition to the data obtained from the DHT-11 sensor module, historical data for 4 years is used in order to account for seasonality trends in the weather, the two are combined into a single dataset. Using SARIMA Time Series Forecasting Model, the forecasts are generated.

Index Terms **– Arduino Uno, Weather Forecasting, SARIMA, Time-Series Forecasting.**

I. INTRODUCTION

Weather forecast plays an important role in our daily lives. On an everyday basis, weather forecasts are used to determine what clothes to wear, what mode of transport to use, mobile networks and satellite communication, travel and logistics are all dependent on the weather. Airplanes, ships, trains, all depend on weather forecasts, and cannot run during unfavorable weather conditions. Therefore, understanding the weather and having forecasts beforehand are essential in today's world.

Temperature is one the most commonly used quantity to describe the weather of a certain geographic location. One can draw inferences about the weather conditions of a place on basis of the temperature patterns of region. Temperatures in the 20-35°C indicates a warm and comfortable weather, a temperature range of upwards 35°C indicates that the weather conditions are hot in that region, similarly regions with temperatures below 15°C indicate cold weather. Although there are several parameters that define the weather of a region, such as humidity, rainfall, air pressure, wind speeds, latitude, longitude, altitude, etc., not all of these are useful for generating forecasts. The geographic co-ordinates and the altitude of a place are constant, they are therefore not considered in the forecasting model. Parameters such as humidity, wind speed, air pressure, etc. are often interdependent and exhibit a strong correlation with each other. As a principle in Machine Learning, the data must be free from multicollinear features in order to get the best results. Therefore, the proposed system aims to predict the maximum and minimum temperature in the city of Pune, India, using data collected from the DHT-11 sensor module, and 4-year historical data obtained from the Indian Meteorological Department.

As previously discussed, the weather is seasonal in nature. Therefore, the proposed system employs a Seasonal ARIMA (SARIMA) model which takes this seasonal nature of the weather into account, to generate the forecast. Another model called the One-Step-Ahead Model, which is based on the assumption that the temperature today depends on the temperature on the previous day, similarly, the temperature on the previous day depends on the day before yesterday, and so on., is created and its results are compared with those of the SARIMA model.

II. LITERATURE REVIEW

The studies in [1] and [2] proposes Machine Learning algorithms to predict the weather. Machine Learning is well known to understand complex, elusive patterns, and generate results, which is the reason for its widespread application. Weather is one such application where there are seasonal trends in the data. The study in [2] shows how Artificial Neural Networks (ANN) and the Day-Ahead System models are effective to generate predictions.

A study in [3] compares the results obtained with ARIMA and Seasonal-ARIMA (SARIMA) models, and found that the latter outperforms the former in forecast accuracy since it takes the seasonal nature of the weather into account. The four-year historical data, i.e. 2015 to 2018, of the city Pune, India, has been obtained from the India Meteorological Department [4]. The specifications of the DHT-11 temperature and humidity sensor module used, is one of the most widely used module for measuring temperature and humidity, are mentioned in [5]. The studies in [6], [7] and [8] demonstrate the different applications and techniques that can be employed while programming the Arduino Uno development board.

III. Flow Chart

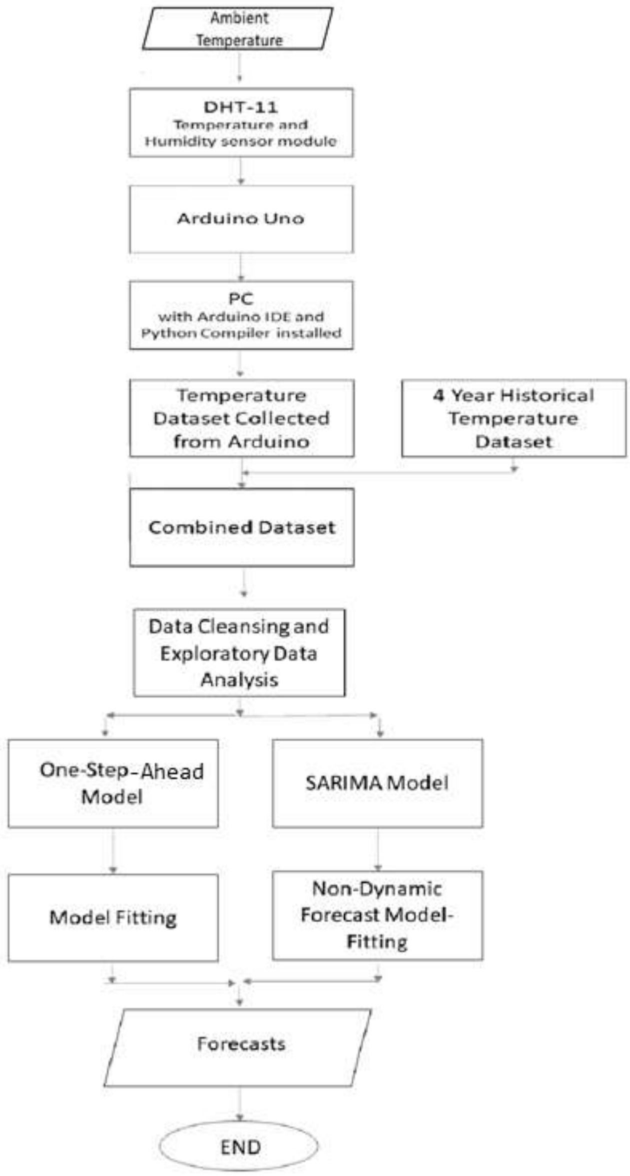


fig. 1. Flowchart of the proposed system

IV. Implementation

The DHT-11 sensor module is connected to the Arduino Uno development board, as show in fig. 2. The ambient temperature and humidity in the air is recorded by the module every 10 minutes, and is transmitted from the data pin of the module, to the digital-input port 2 of the Arduino Uno via a regular single strand wire. This temperature and humidity data is then transferred to a desktop PC from the Arduino Uno, using a USB Type-A cable, where it will then be monitored, and the forecasts will be generated. The PC has Arduino IDE, Python compiler and Spyder IDE installed on it. Arduino IDE is an Integrated Development Environment that is used to program the Arduino Uno. Similarly, the Python compiler is used by the computer to understand the Python code, and the Spyder IDE is used to develop the forecasting algorithm in the Python programming language.

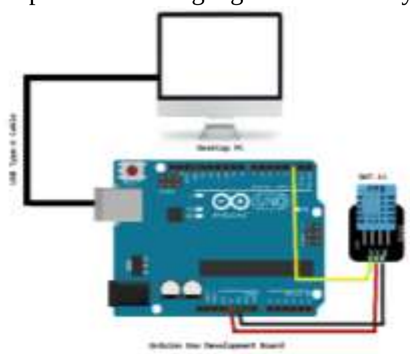
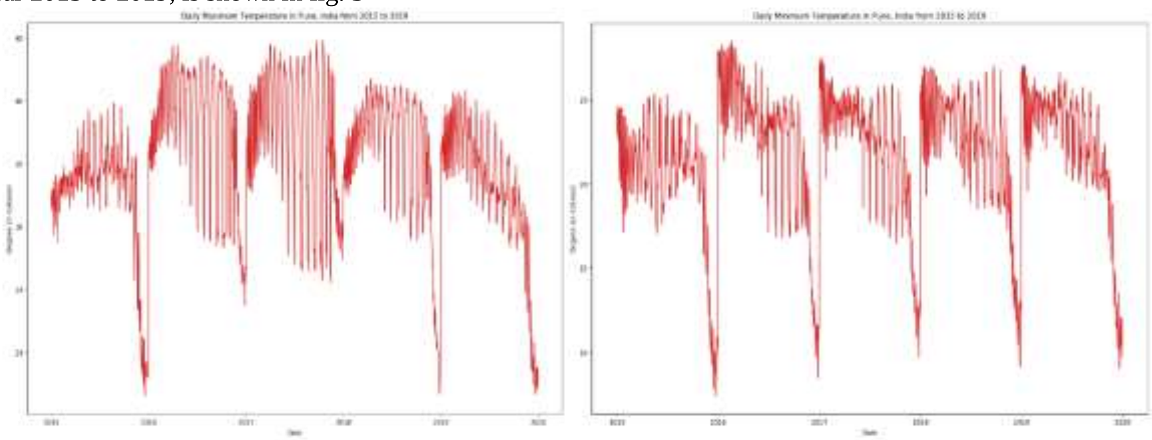


fig. 2. Connecting the DHT-11 sensor to the Arduino and interfacing with the PC

The Arduino IDE uses C++ programming language to program the Arduino Uno. Using file handling, the temperature and humidity values, along with their timestamps, are stored in a text file, which is then taken as an input by the Python program, and the daily maximum and minimum temperatures are obtained. Since the weather is seasonal in nature, it is not possible to predict the temperature with short-term data. There are trends in the weather that need to be taken into account while generating forecasts. Trends are location specific, for instance; in India, the summer season lasts between the month of April and June, followed by monsoon which usually lasts till the month of October. Winter arrives right after the retreat of monsoon, which lasts until the month of March. Here, we see that there is an obvious trend that exists in the weather. However, in different countries, which are situated at different geographical locations, the weather patterns will be different. Therefore, seasonality needs to be accounted for. The proposed model makes use of a historical dataset which will help improve the model. The data obtained from the Arduino Uno, are appended to the historical dataset, and are then fed to the prediction model. Note: there are two separate datasets that are used, one for the daily maximum temperature, and another which stores the daily minimum temperature.

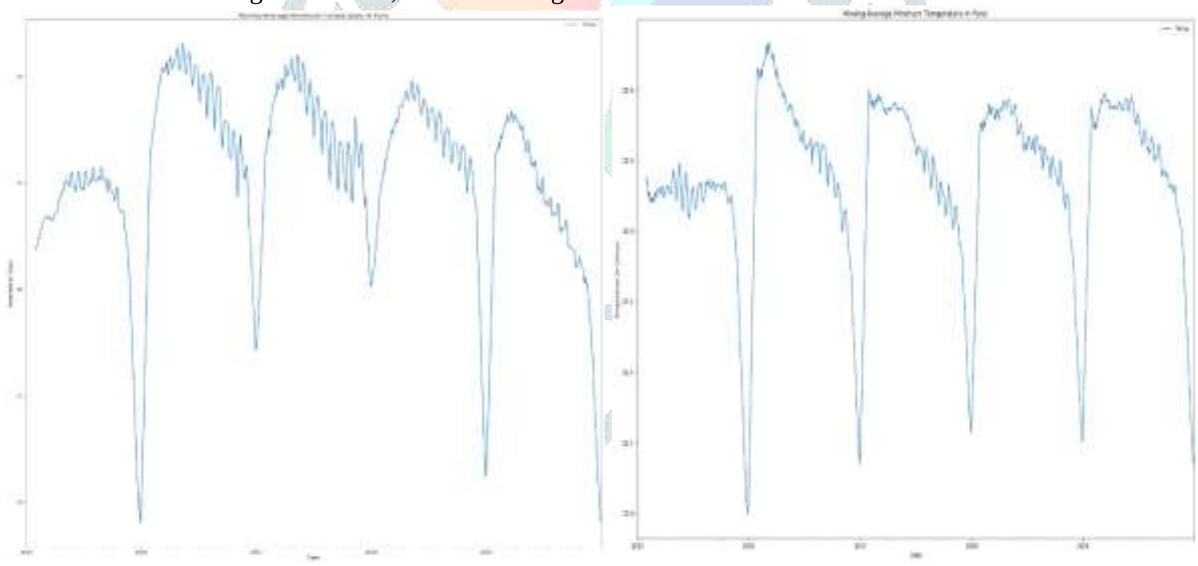
Exploratory Data Analysis (EDA) is performed on both the datasets, where they are checked for nulls and missing values. In case of nulls and missing values, a mean of the temperature on the preceding day of the null/missing-valued day, and the temperature of the day following the null/missing-valued day is taken. Taking the mean preserves the distribution of the data, and does not affect any other statistical characteristics of the data. The plot the daily maximum and minimum temperature recorded from the year 2015 to 2019, is shown in fig. 3



(a) (b)

fig. 3. Plot of the (a) daily maximum, and (b) daily minimum temperatures respectively, for the year 2015 through 2019

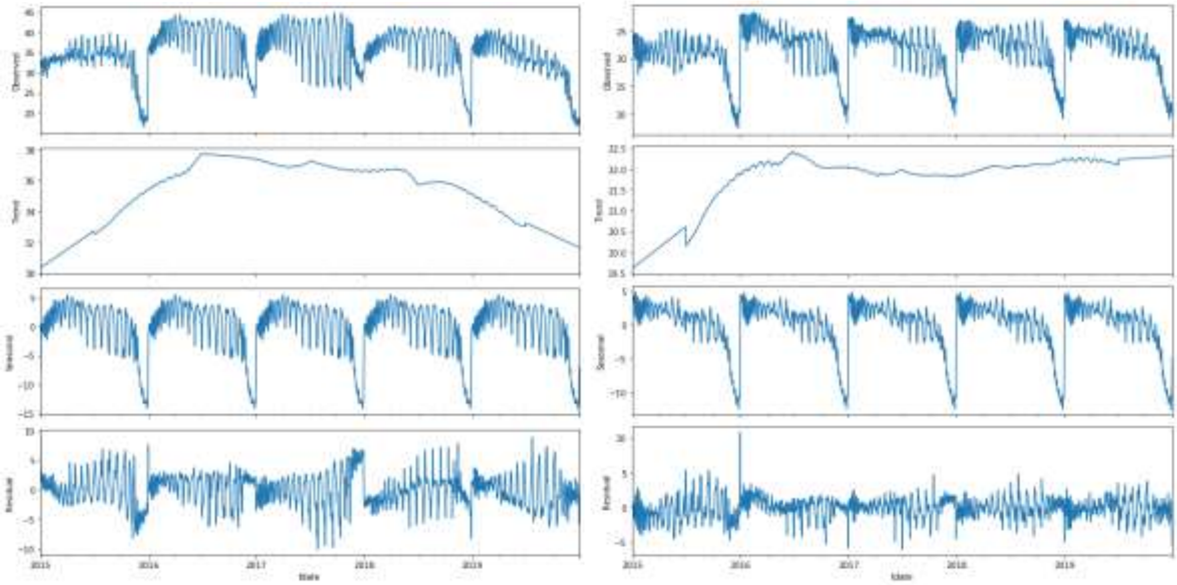
A clear pattern is observed when the data is plotted. The plot shows a seasonal trend, such as low temperatures are observed during the beginning and the end of the year, and high temperatures during the middle of the year. However, the plot is noisy, since it contains all the daily temperatures for five years. Taking a closer look at the data points, we see that there is only a minor temperature change between the current date and the next date. Since, there is a lot of noise in the dataset, we will take a rolling average (also referred as moving average) with a window size of 30 (since we are using a daily dataset), to eliminate short term fluctuations and focus on the long-term trends, as shown in fig. 4



(a) (b)

fig. 4. Moving Average time-series plot for (a) maximum, and (b) minimum temperatures respectively

The data is visualized using a technique called additive time-series decomposition. This decomposes the time-series data into three distinct components: trend, seasonality, and noise, shown in fig. 5. The plot below, clearly shows that the temperature is not uniform, and also exhibits seasonality.



(a) (b)

fig. 5. Additive Decomposition for the (a) daily maximum, and (b) daily minimum temperatures respectively

4.1 Model Fitting:

4.1.1: Model 1: One-Step-Ahead Forecast Model

Since there is no significant change in the temperature of two consecutive days, we create a model which uses the current day's temperature as the prediction for the next day. The graphs in fig. 6 show the of the actual daily maximum and minimum temperature respectively, in "blue" colour, while the forecasts are for the same are plotted with the "orange" colour. The overlap of the plots shows how close the forecasts are to the actual observed temperatures.

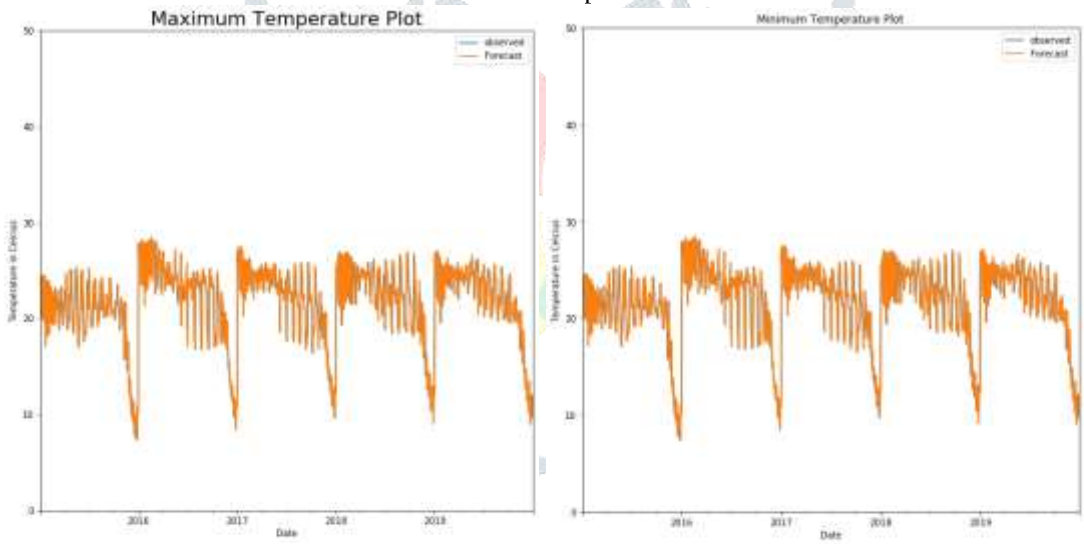


fig. 6. One-Step-Ahead Model Forecasts for (a) daily maximum, and (b) daily minimum temperatures respectively.

4.1.2: Model 2: SARIMA Model

Grid search technique is used to iteratively explore different combinations of 7 parameters classified into two categories, namely:

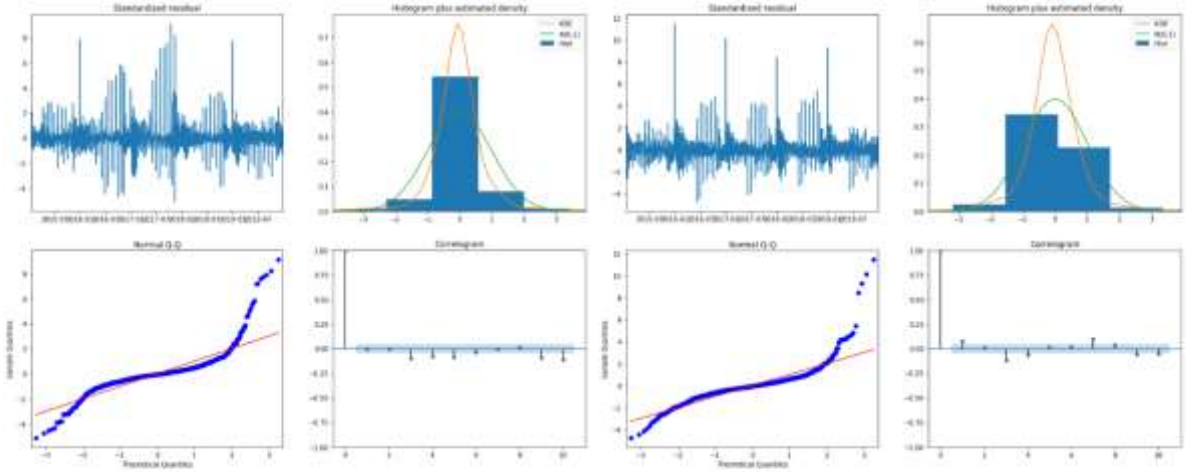
1. Trend Elements:
    - p - Trend Autoregression Order
    - q - Trend Difference Order
    - d - Trend Moving Average Order
2. Seasonal Elements:
    - P - Seasonal Autoregression Order
    - Q - Seasonal Difference Order
    - D - Seasonal Moving Average Order
    - m - The number of time steps for a single seasonal period

For each combination of parameters, a new SARIMA model is fitted, and its overall quality is assessed before selecting the right combination of parameters. The AIC measures indicates the extent to which the model fits the data while taking the overall complexity of the model into account; a lower AIC score indicates a more predictive model. Following wee the results obtained from the Grid Search technique:

- The output of the Grid Search suggests that SARIMAX(1, 1, 1)x(1, 0, 1, 12)12 yields the lowest AIC value of 7781.363925913243, for the daily Maximum temperature.

- The output of The Grid Search suggests that SARIMAX(1, 1, 1)x(1, 0, 1, 12)12 yields the lowest AIC value of 7781.363925913243, for the daily Minimum temperature.

After model fitting, the model diagnostics are plotted to check the behaviour of the same, shown in fig. 7.



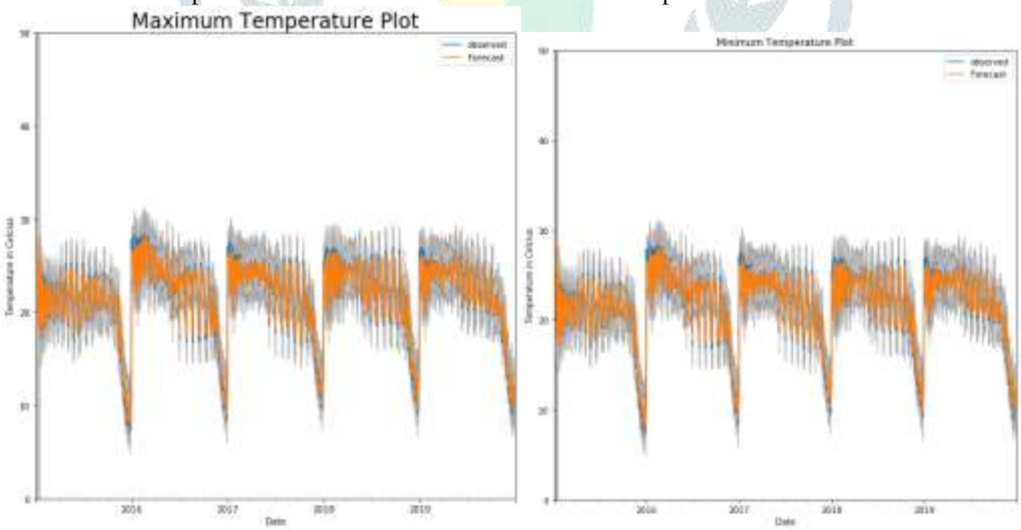
(a) (b)

fig. 7. The SARIMAX model diagnostics for (a) daily maximum, and (b) daily minimum temperatures respectively

The Seasonal ARIMA must have uncorrelated residuals and that they are normally distributed with zero mean, else the model requires further improvement. The model diagnostics for the maximum and minimum temperatures show in fig. 7 respectively, suggest that the model residuals are normally distributed based on the following observations:

1. The graph in the top-left of the two figures, show the residuals over time. In these graphs do not display any obvious patterns or seasonality, and appear to be white noise. This is corroborated by the autocorrelation plot on the bottom right graphs of the two figures, which show that the residuals exhibit low correlation with lagged versions of itself.
2. The plots on the top-right of the two figures show that the KDE line (in red), follows a similar trend as observed in the N(0,1) line (in green). The N(0,1) is the standard notation for a normal distribution, where the mean is 0 and standard deviation is 1. The absence of skew indicates that the residuals are normally distributed.
3. The qq-plot on the bottom-left of the figures indicate that the ordered distribution of residuals (blue dots), follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). This is another indication that the residuals are normally distributed.

The observations discussed suggest that the model produces a satisfactory fit that can successfully forecast future values. The plots shown in fig. 8, show the forecasts generated by the SARIMA model. Similar to the One-Step-Shifted model, the SARIMA model forecasts are represented by “orange” colour and the observed (actual) temperature are represented in “blue” colour. From the plots it is evident that the overlap of the colours indicates that the model has performed well and that the forecasts are accurate.



(a) (b)

fig. 8. SARIMA Model Forecasts for (a) daily maximum, and (b) daily minimum temperatures respectively. The observed and forecasts overlap showing that the forecasts are very close of the actual observed temperatures for the day.

V. RESULTS

	Duration of dataset (in Years)	Minimum Temperature Forecasts		Maximum Temperature Forecasts	
		RMSE (in °C)	MAPE (in %)	RMSE (in °C)	MAPE (in %)
One-Step-Ahead Forecast Model	1-5	1.54945	4.31231	2.08036	3.45033
SARIMA Non-Dynamic Forecast Model	1	1.63574	4.56325	2.19826	3.49698
	2	1.52562	4.37548	2.15487	3.49274
	3	1.45361	4.26647	2.0338	3.34563
	4	1.42591	4.13007	1.6792	3.18894
	5	1.14461	3.96653	1.52387	3.32273

Table 1. Results Table

Note:
RMSE: Root Mean Square Error
MAPE: Mean Absolute Percentage Error
Actual Temperature (°C) = Forecasted Temperature (°C) ± RMSE (°C)
Actual Temperature (°C) = Forecasted Temperature (°C) ± MAPE (%)

Since, One-Step-Ahead simply shifts the current day's parameters to generate the following day's forecast, its performance is therefore unchanged irrespective of the number of years of historical data being taken into consideration.

VI. CONCLUSION

The proposed system performs well, producing a Maximum Absolute Percentage Error of 4.56% and a maximum Root Mean Square Error of 2.198 ° C, which is well under the traditional weather forecast minimum error threshold. Currently weather forecasts are usually 90% accurate, i.e., approximately a 10% error, while the proposed system, offers a maximum of 4.56 % of error, thereby reducing the error by 54.4%.

While the One-Step-Ahead successfully predicts the following day's maximum and minimum temperature with a satisfactory accuracy, it is observed that the SARIMA Non-Dynamic Forecast model, using 5 years of data outperforms the One-Ste-Ahead Model's forecasts, in predicting the minimum as well as the maximum temperatures. The results indicate that an increase in the size of the dastaset results in the reduction of the RMSE and consequently a reduction MAPE. The results suggest that using five years' worth of data, the SARIMA model perform the best amongst all the models.

VII. ACKNOWLEDGMENT

I would like to express my sincere gratitude and thank my mentor, Dr. R. C. Jaiswal, for his tremendous support and guidance over the course of this research. He has given this paper the insight and the expertise needed to make it a presentable one. His advice, constructive criticism and encouragement at various stages of the research, have been of great value.

REFERENCES

[1] N. Singh, S. Chaturvedi and S. Akhter, "Weather Forecasting Using Machine Learning Algorithm," 2019 International Conference on Signal Processing and Communication (Icsc), Noida, India, 2019, Pp. 171-174, Doi: 10.1109/Icsc45622.2019.8938211.

[2] S. E. Haupt, J. Cowie, S. Linden, T. Mccandless, B. Kosovic And S. Alessandrini, "Machine Learning for Applied Weather Prediction," 2018 IEEE 14th International Conference On E-Science (E-Science), Amsterdam, 2018, Pp. 276-277, Doi: 10.1109/Escience.2018.00047.

[3] Z. Xinxiang, Z. Bo And F. Huijuan, "A Comparison Study Of Outpatient Visits Forecasting Effect Between Arima With Seasonal Index And SARIMA," 2017 International Conference On Progress In Informatics and Computing (Pic), Nanjing, 2017, Pp. 362-366, Doi: 10.1109/Pic.2017.8359573.

[4] Development of High Resolution Daily Gridded Temperature Data Set (1969-2005) for the Indian Region. Atmospheric Science Letters Atmos. Sci. Let. (2009) Doi: 10.1002/Asl.232.

[5] Gay, Warren. (2014). Dht11 Sensor. 10.1007/978-1-4842-0769-7_1.

[6] Jaiswal R.C. And Vipul Phulphagar, "Arduino Controlled Weight Monitoring With Dashboard Analysis," International Journal For Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC value: 45.98; sj Impact Factor:6.887, Volume 5 Issue XI November 2017

[7] Jaiswal R.C. And Sumukh Ballal, "Scalable Healthcare Sensor Network," International Journal Of Emerging Technologies And Innovative Research (www.Jetir.Org | UGC and ISSN Approved), ISSN:2349-5162, Vol.6, Issue 2, Page No. Pp350-354, February-2019

[8] Jaiswal R.C. And Nehal Borole, "Autonomous Vehicle Prototype Development and Navigation Using RTOS," International Journal for Research In Applied Science & Engineering (IJRASET), Open Access, Peer Reviewed and Referred Journal, ISSN: 2321-9653; IC Value: 45.98; sj Impact Factor:7.177, Volume 7 Issue XI, Pp. 510-514, Nov 2019.