Exploratory Data Analysis and Modeling

(Auto Regression on unmanned vehicles data using VAR and ARIMA)

Prahitha Movva
S20180010108

ABSTRACT

The objective of this project is to implement the various techniques learned throughout the course (and otherwise) to extract information from data. The data[1] which I was given is a multivariate time-series. This report discusses how I approached the data and the different methods used during the analysis and prediction phases. I used the normal test to check for normality, the DW Test to check for collinearity, the Granger Causality Test to check for association between the variables, and the ADF Test to check for stationarity. Furthermore, I observed the features of the data using time series analysis and applied regression techniques like VAR(OLS), and ARIMA to predict future outcomes.

Keywords -- Normal test, DW test, Granger causality test, ADF test, OLS, VAR, ARIMA, regression, multivariate, time-series

INTRODUCTION

For this project, my main objective was to perform exploratory data analysis to extract maximum information from the given data and then perform modeling to predict future values. Regression is one task that has to be implemented for this dataset, but I went ahead and built a multivariate time-series model to familiarize myself with time-series forecasting problems too.

To achieve this, I used different tests which give information about the collinearity, causation, and stationarity of the time series. I have also used the least-squares and the auto-regression models which incorporate the time-variant samples very well. I have designed two systems, one using VAR with a lag equal to 26 (for the first sensor data), which means I am training the data on the basis of 26 previous samples. The other uses ARIMA with AR(p)=7, I(d)=1, MA(q)=0 (for the first sensor data).

Finally, I tested the predicted data on the test data set (obtained from dividing the main

data set into two parts, one each for training and testing the model) and then checked if the output is correct, making it a good predictive model. The ARIMA model seemed to be a better choice based on the error-rate and favorable model characteristics[4].

DATASET DESCRIPTION

Data Set Characteristics:	Multivariate, Time-Series	**Number of Instances:**	1672	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	5	**Date Donated**	2018-05-06
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	22988

The unmanned surface vehicles dataset consists of 4 sensor readings all set to measure the humidity and temperature. These values are changing with time and the sensors record this information. Each dataset contains records of the format: {'USV-ID'; 'humidity-value'; 'temperature-value'; 'experiment-id'; 'sensing-time'}

METHODOLOGY

Exploratory data analysis

- Basic preprocessing steps like adding column names, changing the epochs to DateTime format
- Drop the columns which don't add valuable information to the data (like, experiment)
- Partition the data into training and validation sets
- Identify missing values and verify the quality of the data
- Determine likely approaches to modeling, which might yield a predictive function

Device 1:

	humidity	temperature	experiment	time
count	342.000000	342.000000	342.0	3.420000e+02
mean	23.438596	38.391813	1.0	1.522086e+09
std	4.085670	2.391025	0.0	7.194031e+02
min	17.000000	33.000000	1.0	1.522085e+09
25%	20.000000	36.000000	1.0	1.522085e+09
50%	22.000000	39.000000	1.0	1.522086e+09
75%	28.000000	40.000000	1.0	1.522087e+09
max	31.000000	42.000000	1.0	1.522087e+09

From the above image, we can observe that the mean is more than the median value (which is represented by 50% in the index column) for the humidity column.

There is also a notable difference between 50% and 75% for humidity.

The above observations suggest that there are extreme values (outliers) in our dataset. This can be seen below with all the boxplots[5]

Device 2:

	humidity	temperature	experiment	time
count	264.000000	264.000000	264.0	2.640000e+02
mean	38.632576	31.954545	1.0	1.522086e+09
std	1.259730	0.773748	0.0	8.937312e+02
min	34.000000	27.000000	1.0	1.522084e+09
25%	38.000000	31.000000	1.0	1.522085e+09
50%	39.000000	32.000000	1.0	1.522086e+09
75%	39.000000	32.000000	1.0	1.522087e+09
max	43.000000	33.000000	1.0	1.522087e+09

From the above image, we can see that there is a noticeable difference between min-25% and 75%-max pairs. However, the rest doesn't have any significant difference. So we could say that there might not be any outliers and probably the distribution is normal too (which we later see that it is true, using the normaltest)

Device 3:

	humidity	temperature	experiment	time
count	488.000000	488.000000	488.0	4.880000e+02
mean	38.276639	31.239754	1.0	1.522083e+09
std	10.563162	5.308346	0.0	1.113151e+03
min	23.000000	20.000000	1.0	1.522082e+09
25%	28.000000	27.000000	1.0	1.522082e+09
50%	38.000000	31.000000	1.0	1.522083e+09
75%	47.000000	36.000000	1.0	1.522084e+09
max	59.000000	39.000000	1.0	1.522085e+09

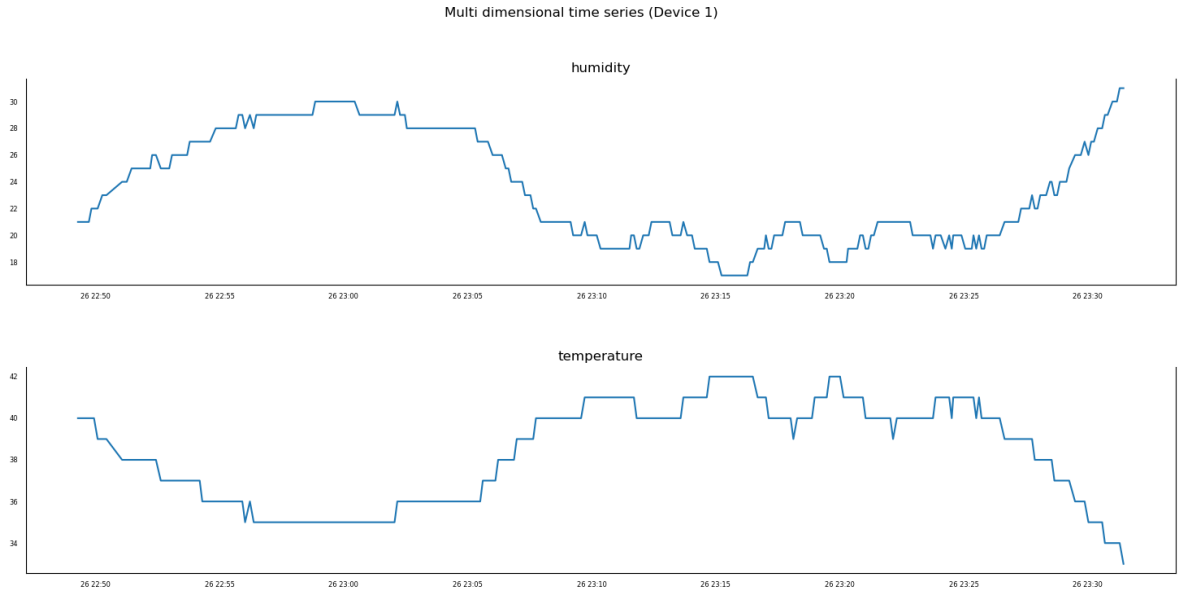
From the above image, we can observe that there are noticeable differences throughout the data, which is why we later see many outlier points and boxes in the boxplot[5]

Device 4:

	humidity	temperature	experiment	time
count	578.000000	578.000000	578.0	5.780000e+02
mean	38.072716	25.185225	1.0	1.522089e+09
std	4.562180	3.832083	0.0	2.111060e+03
min	20.000000	21.000000	1.0	1.522085e+09
25%	34.000000	23.000000	1.0	1.522087e+09
50%	38.000000	24.000000	1.0	1.522089e+09
75%	42.000000	26.000000	1.0	1.522090e+09
max	47.000000	37.000000	1.0	1.522093e+09

We observe significant differences in the data, as in device 3. Hence, we could have a similar conclusion.

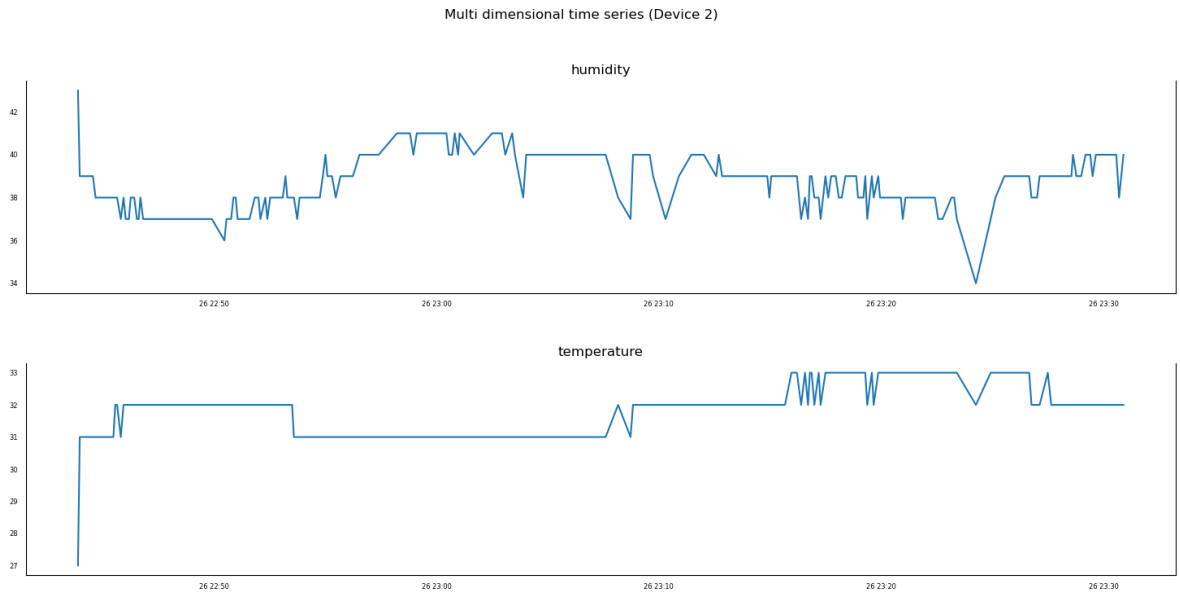
Below we can see the plots for humidity and temperature for all 4 sensors.



We can observe that temperature and humidity are almost inversely related[6]. Also, the

temperature decreases towards the end of the graph, so it could be possible that the vehicle is moving towards a colder place.

[6] Using corr(), gives us a similar result. We get a negative correlation (-0.97), so we can say that they are inversely proportional

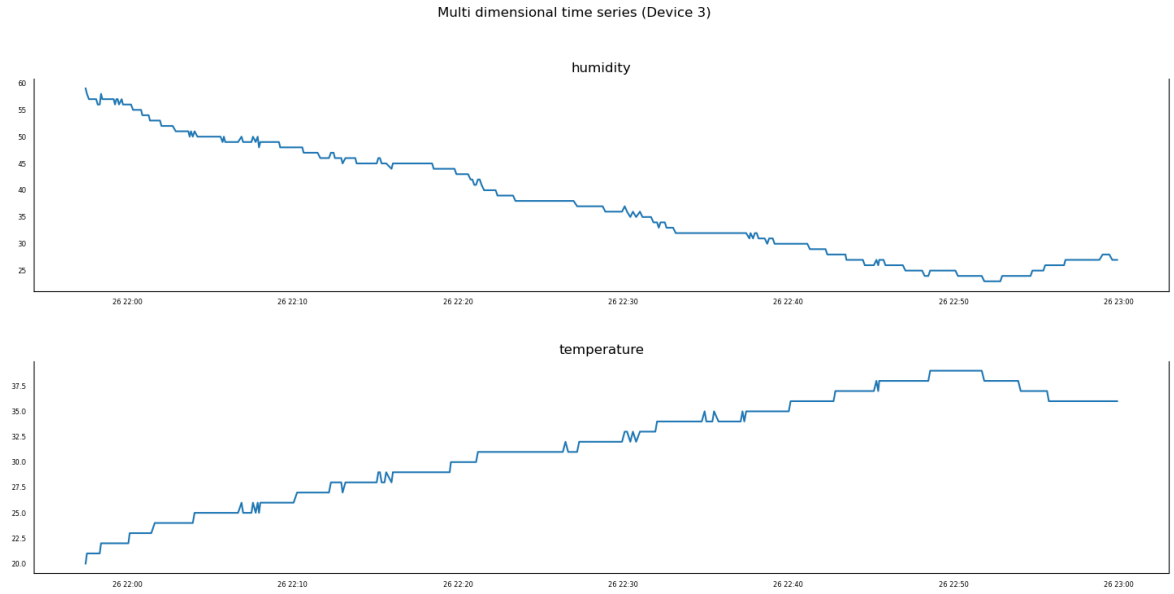


There is no direct relationship between temperature and humidity. However, it is possible that they are inversely proportional with very little correlation[7].

[7] Using corr(), gives us a result of negative correlation (-0.40), which is not very significant but fairly good.

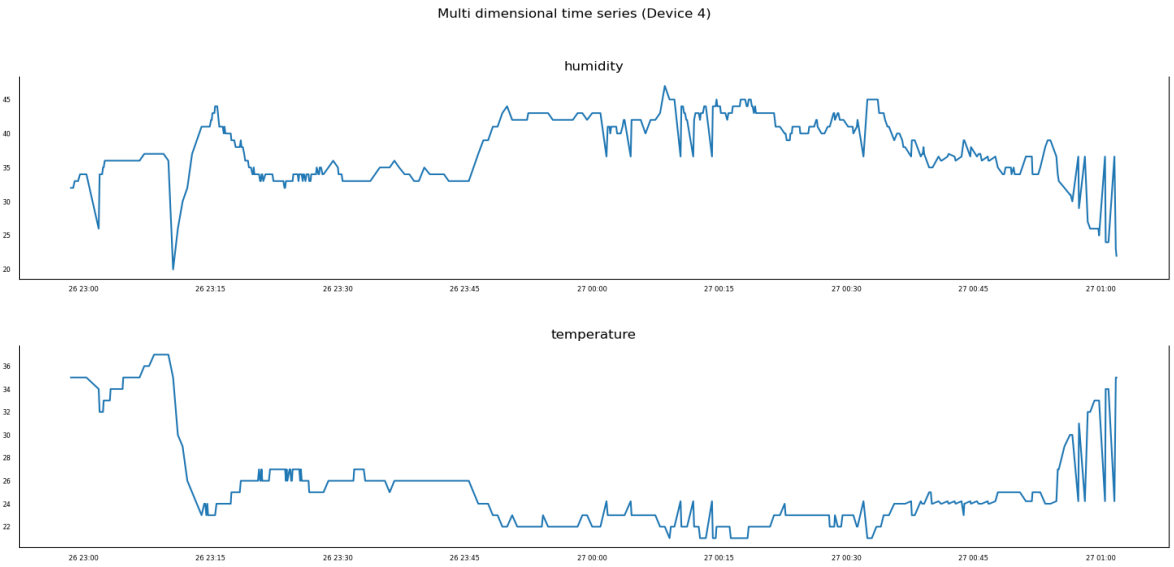
Below, for device3, we can see the humidity and temperature are possibly inversely related[8] and temperature increasing as time progresses. This could either mean that the vehicle/ the engine is getting heated with time or that it is moving to a warmer place.

[8] Using corr(), gives us a similar result. We get a negative correlation (-0.99), so we can say that they are inversely proportional

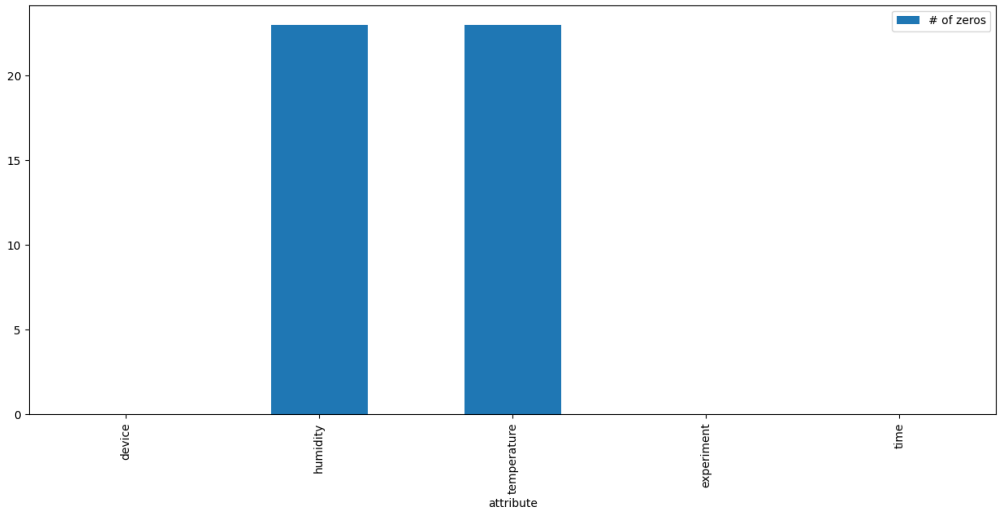


Below, for device4, we can see the humidity and temperature are possibly inversely related[9] and the temperature has a sudden decrease. This could mean that the engine gets very hot initially (as it starts) and probably cools down and then maintains a low temperature.

[9] Using corr(), gives us a similar result. We get a negative correlation (-0.71), so we can say that they are inversely proportional



Below is the graph for the number of zeros observed in the data (particularly, in the device4 data). I've replaced the zeros with the corresponding attribute mean so as to not create much difference in how it influences the predictions.



Additional information:

The normal test (method in python from scipy) returns that only the humidity attribute from device 2 is normally distributed[11] and the rest are not gaussian.

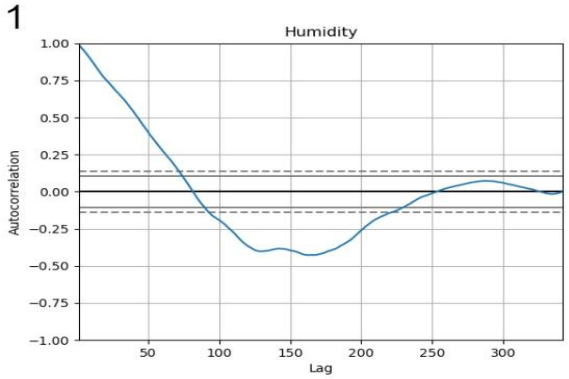
Correlation and autocorrelation plots:

For a stationary[10] time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. Hence, ACF can very quickly give an idea of which lag variables may be good candidates for use in a predictive model and how the relationship between the observation and its historic values changes over time.

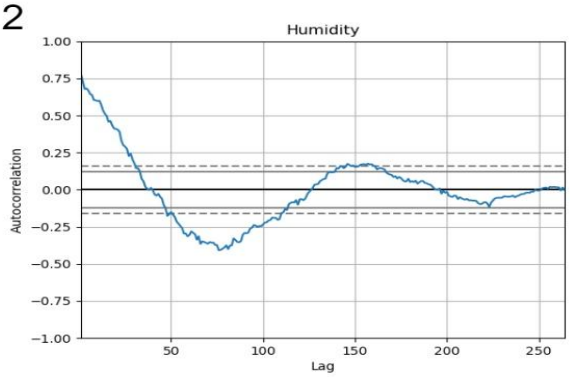
Another helpful test is Granger's causality test. Although the name suggests, it's really not a test of "causality", we cannot say if one is causing the other, all we can say is if there is an association between the variables.

Below are the graphs for autocorrelation for humidity and temperature (marked as 1, 2, 3, 4 for each sensor).

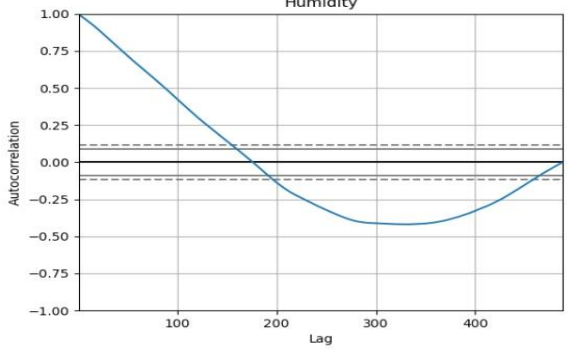
1


Humidity
Autocorrelation
Lag

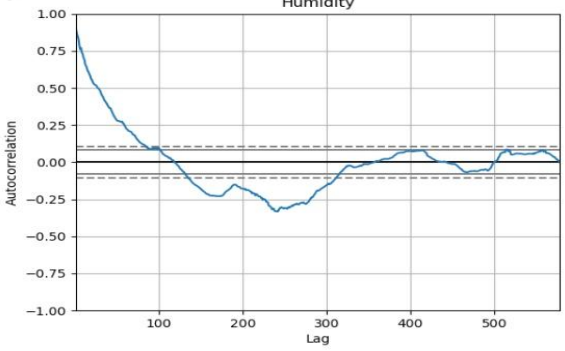
2


Humidity
Autocorrelation
Lag

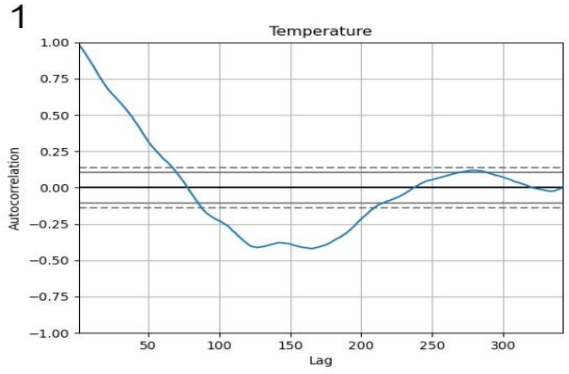
3


Humidity
Autocorrelation
Lag

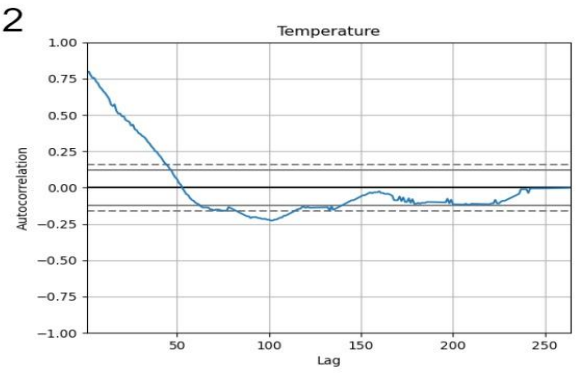
4


Humidity
Autocorrelation
Lag

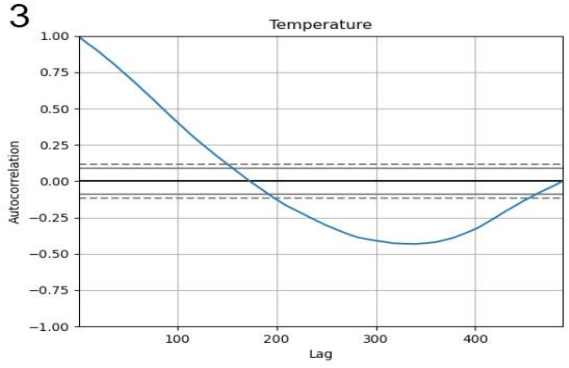
1


Temperature
Autocorrelation
Lag

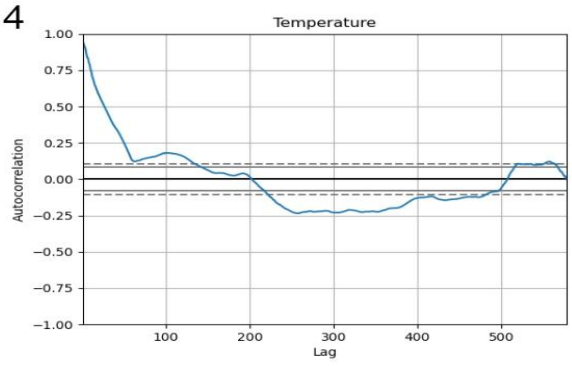
2

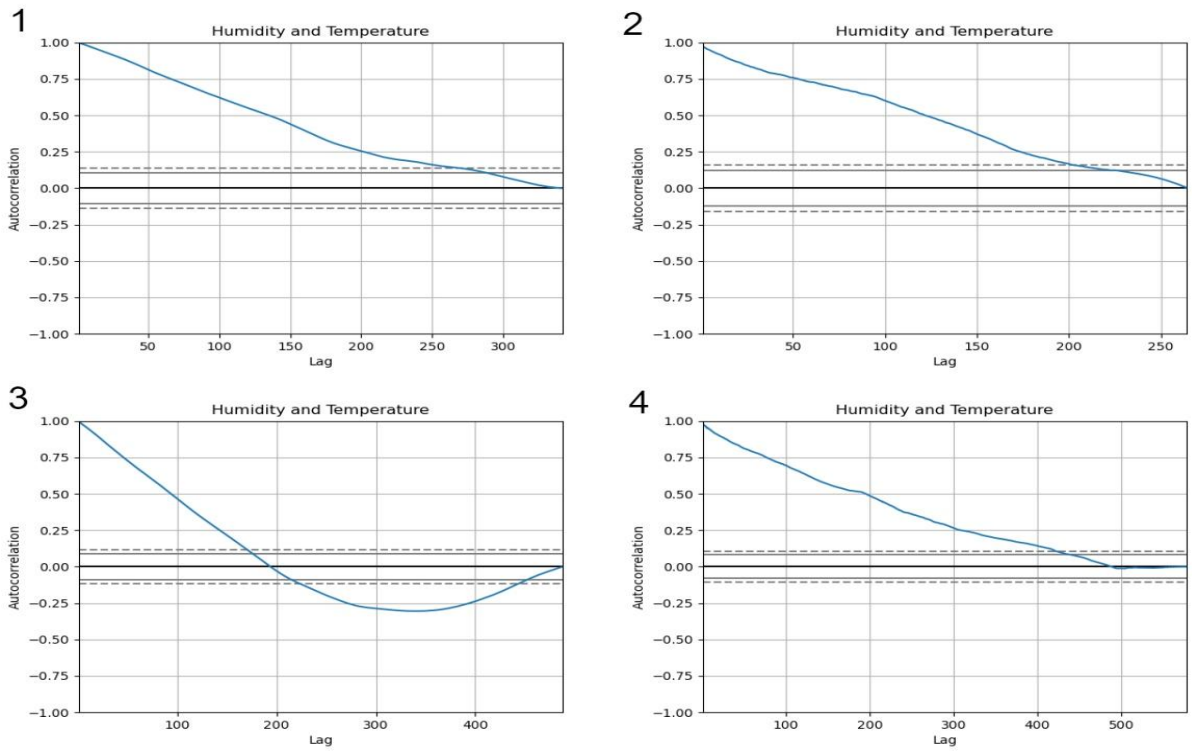

Temperature
Autocorrelation
Lag

3

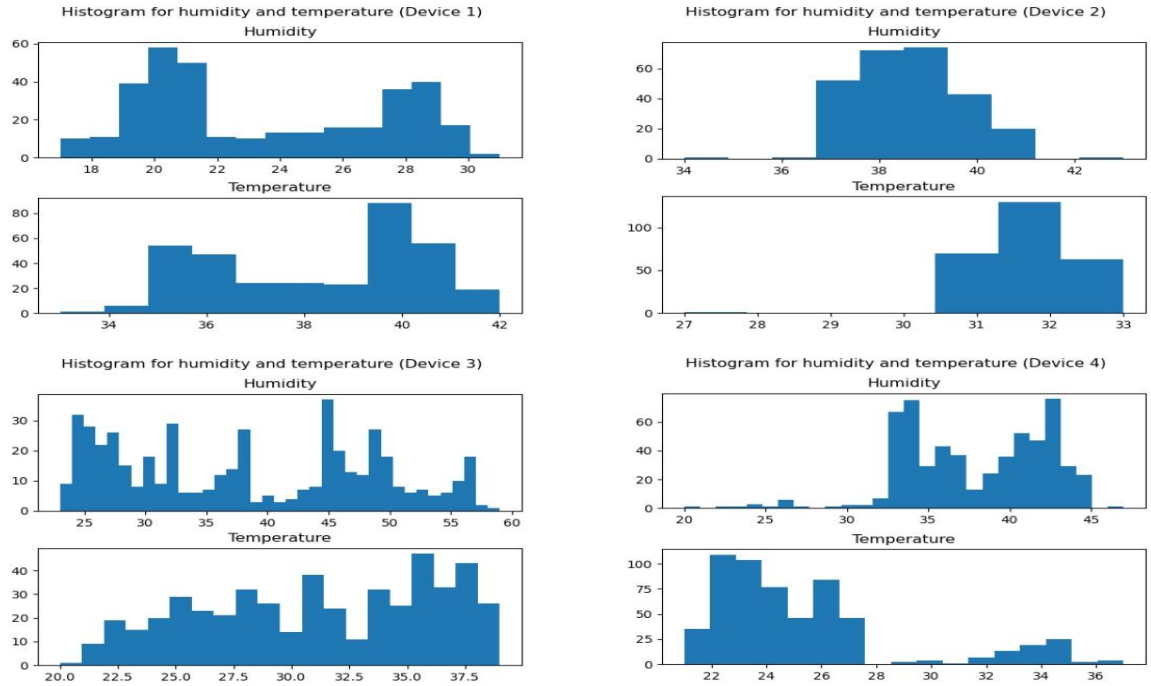

Temperature
Autocorrelation
Lag

4

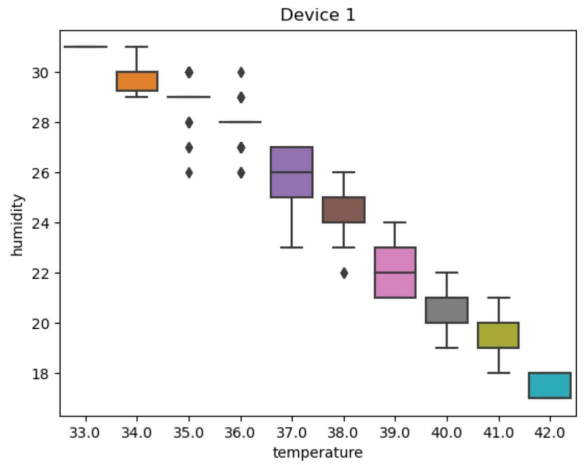

Temperature
Autocorrelation
Lag

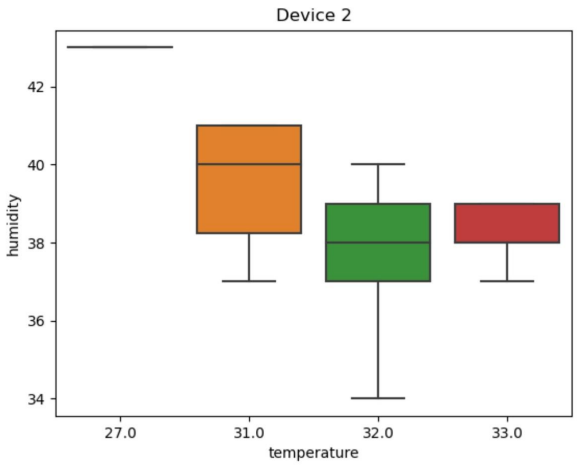


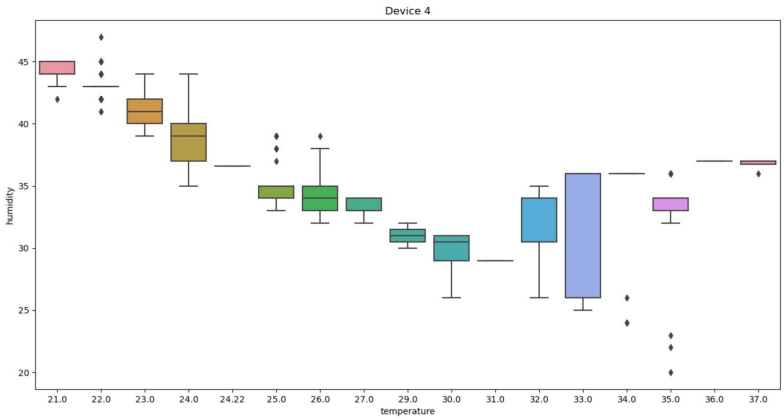
[11] Histograms for humidity and temperature of each sensor:

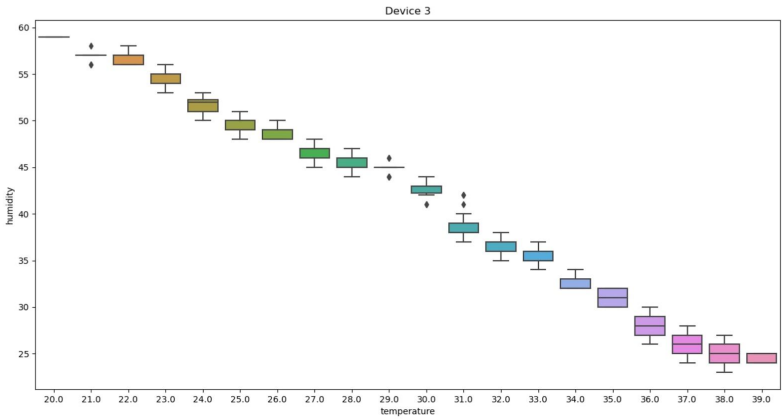


[5] Boxplots showing outlier points:









Statistical Modeling

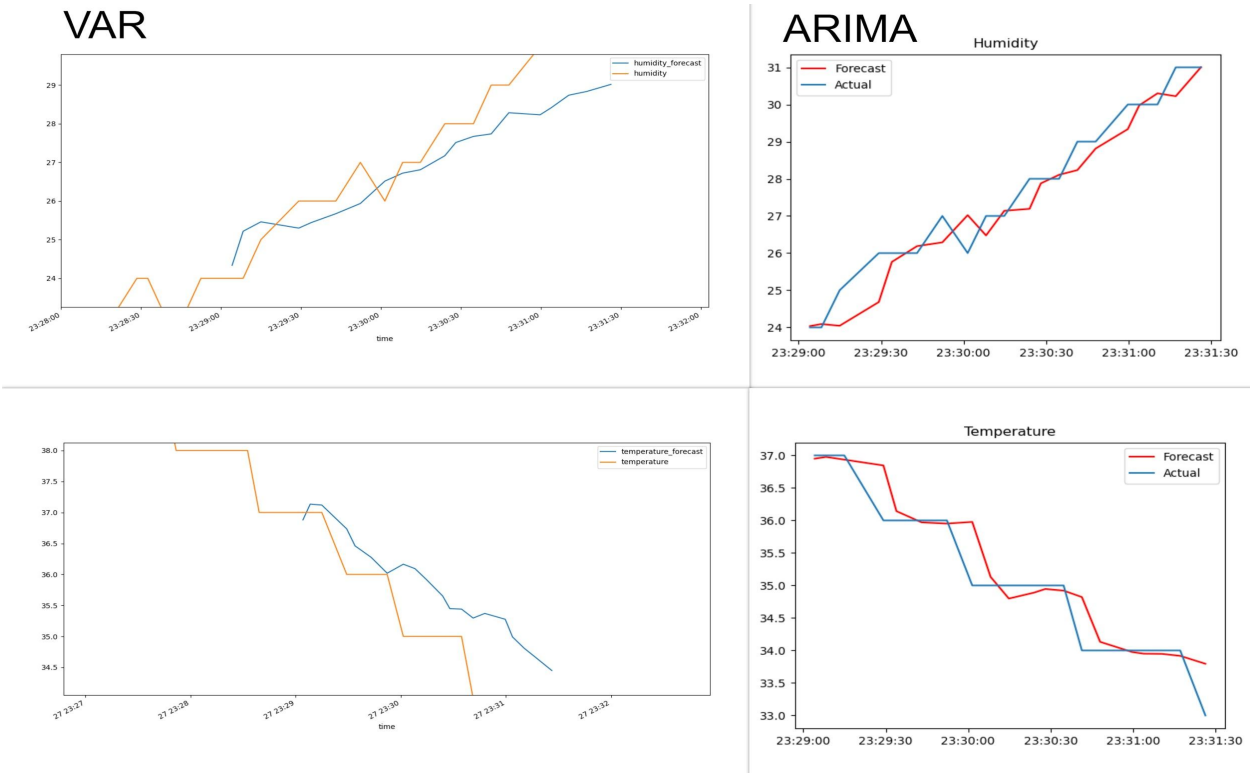
For time series modeling, data needs to be stationary — meaning if there is a trend in the data we need to get rid of it. To check whether data is stationary there is a test called Augmented Dickey-Fuller (ADF) Test.

The ADF test was implemented for all 4 sensors. It generates a tuple consisting of 6 parameters: the ADF test statistic, p-value, number of lags used, number of observations used, critical values at 1%, 5%, 10%, and the maximized information criterion. I mainly used the p-value to determine if the series is stationary or not. If p-value<0.05, then I considered the series to be stationary. Once I got which variables are non-stationary, the next step I did was to make them stationary. For this, there are many ways but I went ahead with differencing them to eliminate the non-stationarity.

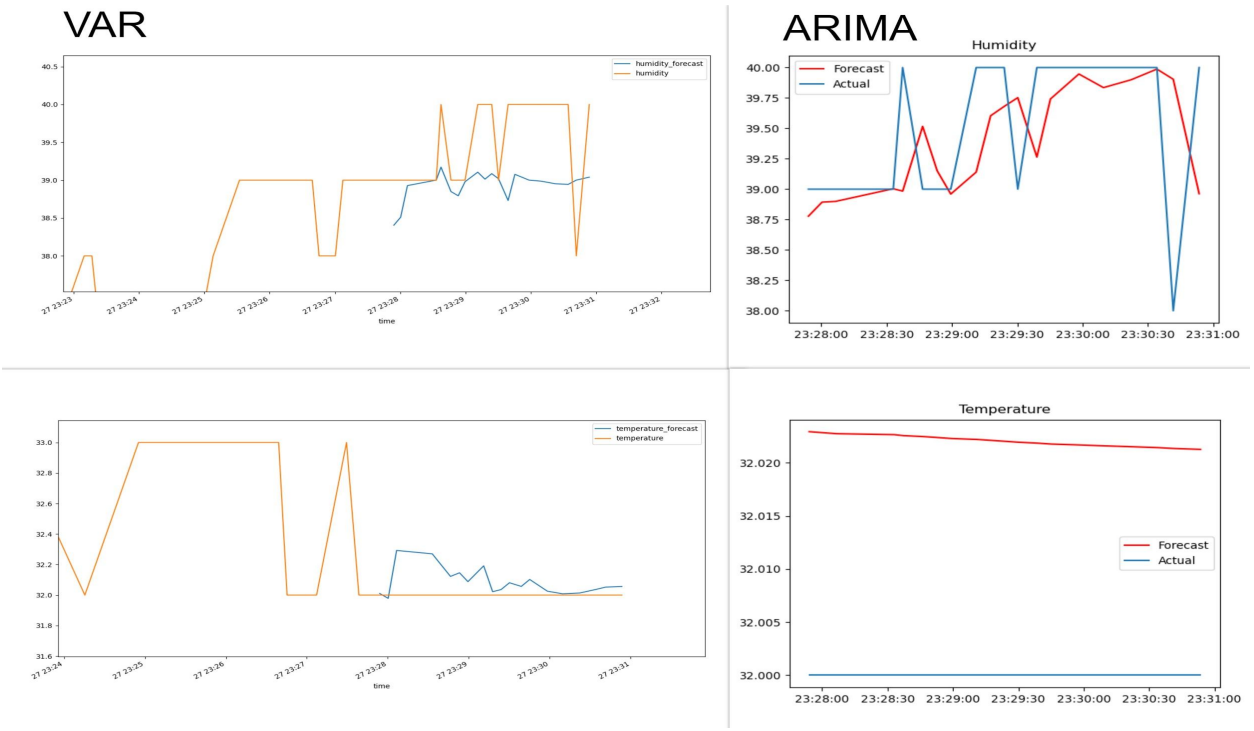
Further, I chose auto-regression models as they incorporate the time-variant samples very well.

RESULTS

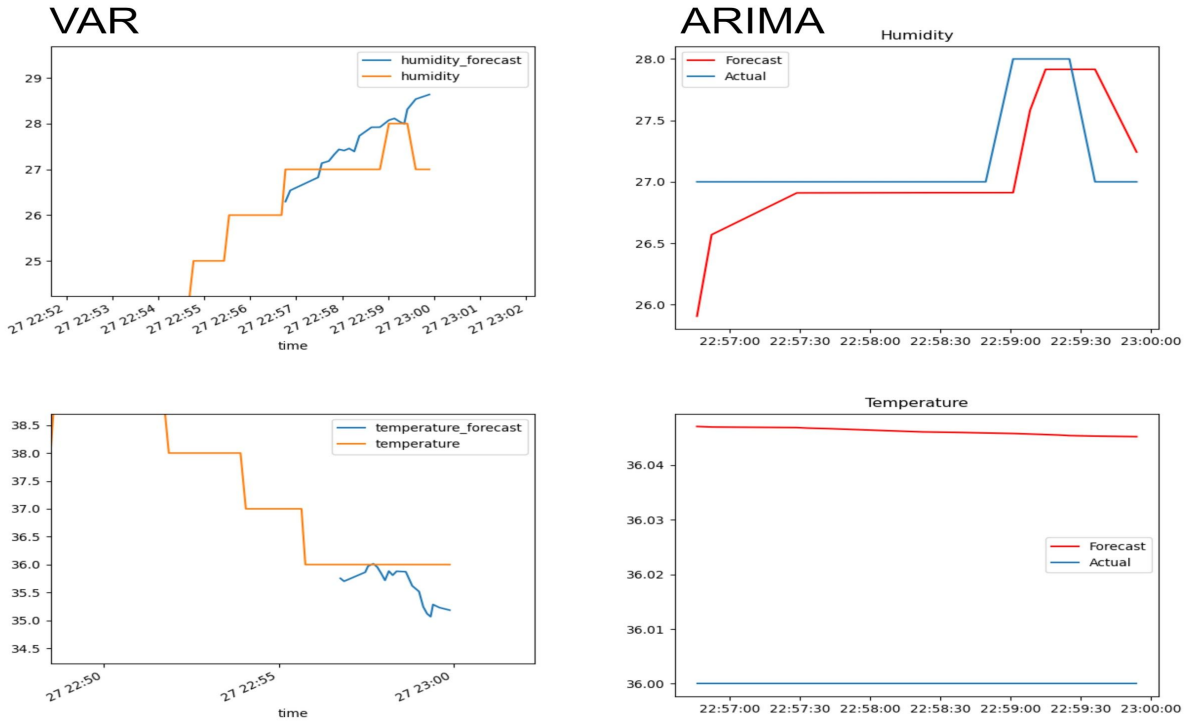
Device 1:



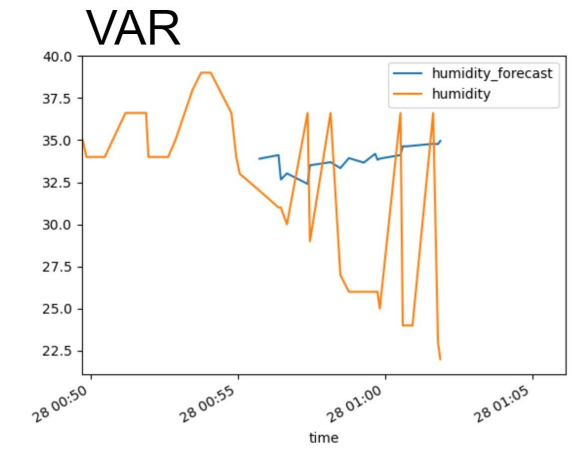
Device 2:

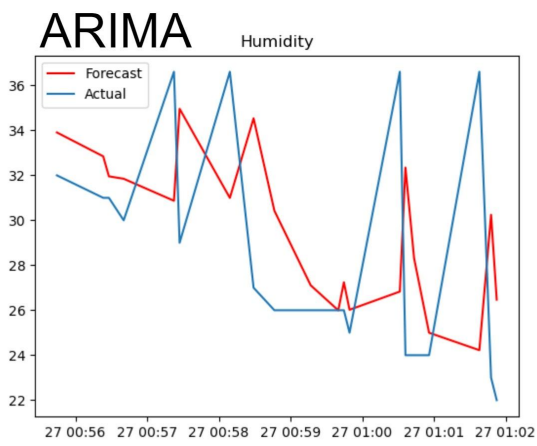


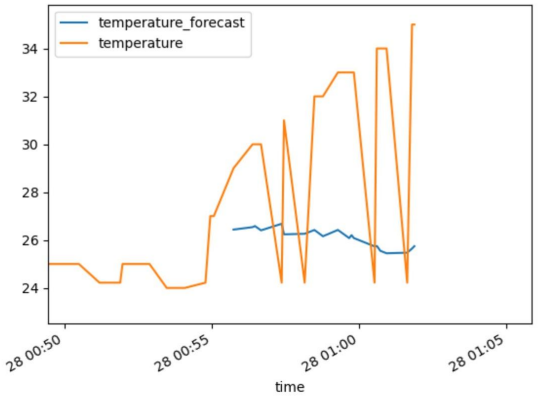
Device 3:

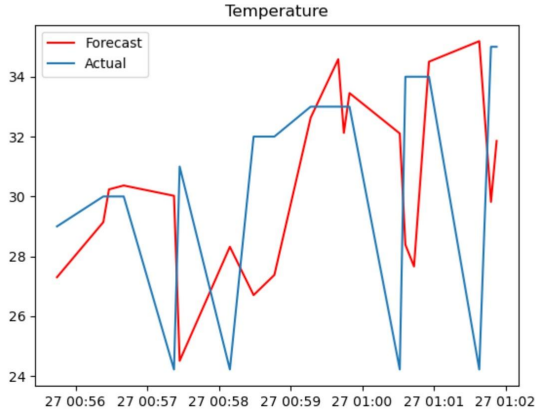


Device 4:









CONCLUSION

On comparing the error rate and MSE for both the models, the ARIMA performed significantly better than the VAR model. There could be multiple reasons for this but the two main reasons I figured were:

- The VAR model performs better when the series is not differenced[3], which is not a very good choice since the series we have is not stationary. And to make it stationary, we have to take the difference.
- Multiple researches have shown that ARIMA performs better than VAR even in similar conditions[4]

Real-world time series forecasting is challenging for a whole host of reasons not limited to problem features such as having multiple input variables, the requirement to predict multiple time steps, and the need to perform the same type of prediction for multiple physical sites.

REFERENCES

[1] Harth, N. Anagnostopoulos, C. (2018) Edge-centric Efficient Regression Analytics. In: 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, CA, USA, 02-07 Jul 2018

[2] Harth, N., Anagnostopoulos, C., (2017) Quality-aware Aggregation & Predictive Analytics at the Edge. IEEE International Conference on Big Data (IEEE Big Data 2017), December 11-14, 2017, Boston, MA, USA

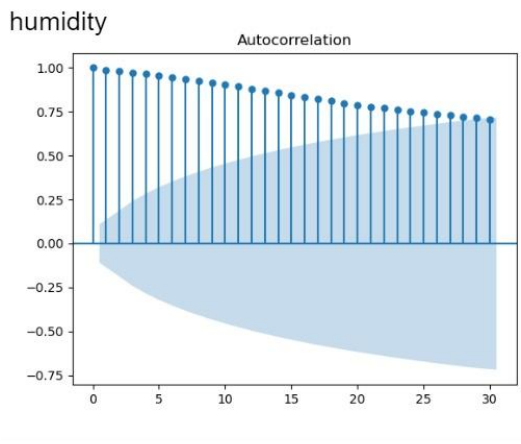
[3] Why using the VAR model on undifferenced data is better than having a differenced model to obtain stationarity: VAR (diff vs undiff)

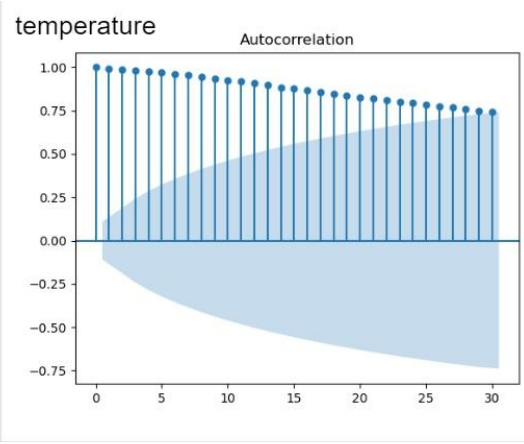
[4] Comparing the univariate ARIMA, multivariate ARIMA, and VAR methods for better results during forecasting: ARIMA vs VAR

[10] A stationary time series is one whose properties do not depend on the time at which the series is observed. Time series with seasonality, are not stationary, because seasonality will affect the value of the time series at different times.

Below are the plots that were used for choosing the lag values for VAR

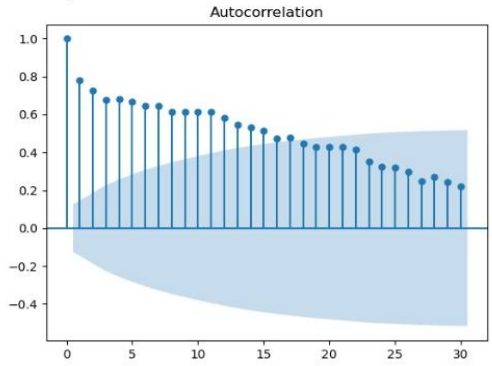
[12] Device 1



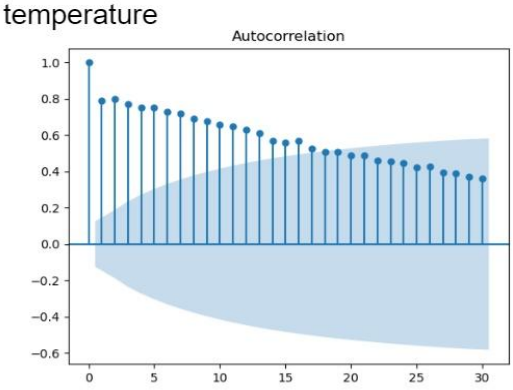


[13] Device 2

humidity

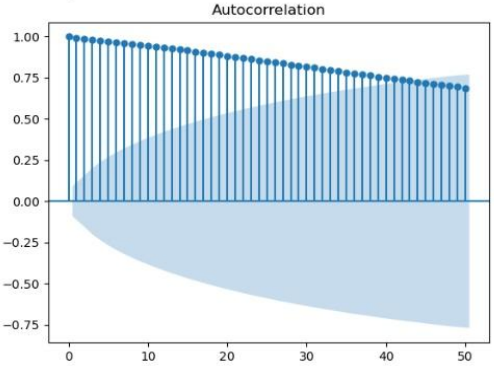


temperature

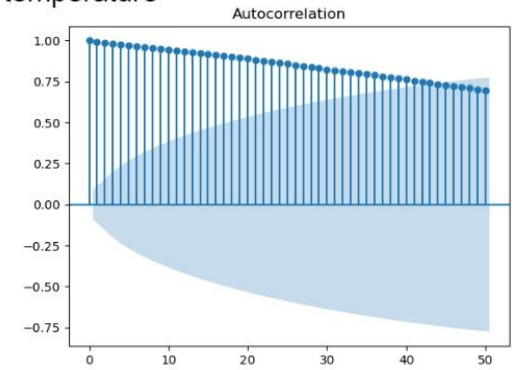


[14] Device 3

humidity

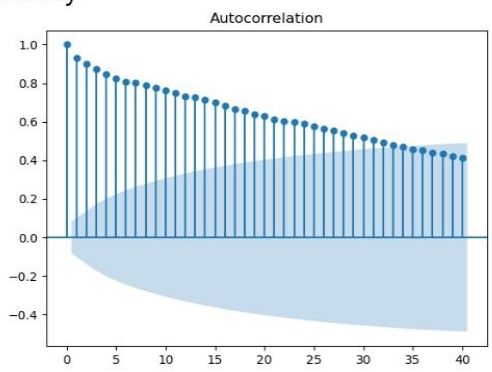


temperature



[15] Device 4

humidity



temperature