Dyson School of Design Engineering
MEng Design Engineering
DE4-SIOT Sensing & IoT

Jacob Mitchell – CID: 01050663

Presentation URL (publicly accessible link):

https://cl.ly/67015917a584/Screen%20Recording%202019-01-11%20at%2011.57.25.18%20AM.mp4

Code & Data (publicly accessible link):

https://github.com/bacon274/SIOT

1. Introduction and objectives

Damp and Mould growth affects 37% of rented properties in the UK (AWH Solicitors, 2018) and adversely affects the health of inhabitants. Damp and mould have been shown to cause increased likelihood of respiratory problems, allergies, asthma and immune system issues (Platt, Martin, Hunt, & Lewis, 1989)(NHS, 2018).

Mould growth is caused primarily by moisture and condensation. Figure 1 shows the beginnings of mould growth in the area in which the experiment was set up.

Air humidity levels above 55% are considered dangerous (EPA, 2016). Condensation forms when warm, humid air contacts a cold surface such as the area surrounding a window. Windows are a common site for mould growth. The dewpoint is the temperature at which air forms a relative humidity of 100% and condenses. The dew point is affected by the temperature and relative humidity of air (Lawrence, 2015). If the outside air temperature is less than the dewpoint, then it can be assumed that there is high risk of condensation forming around the inside of windows. Figure 2 shows an example of mould growth forming from window condensation.

The intention of this project was to create actionable data to reduce the risk of mould formation.

Figure 1 - Mould growth at target site

Figure 2 - Mould growth from window condensation

The objectives of this project were:

1) Create an IOT sensing unit that measures temperature and humidity of a room in a domestic property.
2) Collect and store this data using robust network communications.
3) Analyse the time series data sets for cross and auto correlation and any other interesting interactions.
4) Communicate important information using an online GUI that live streams data in a useful and easy to understand way.
5) Model/create actuations to change the environmental conditions of the space. Making them less favourable to mould growth.

2. Data Sources and set up

Inside Temperature and Humidity

The data sets for Temperature and Humidity were collected using hardware. They were both measured using the Adafruit SHT31 environmental sensor (Adafruit, 2018)connected via I2C bus to an Arduino Uno (Arduino, 2018). An ESP8266 Wi-Fi shield (Sparkfun, 2018) connected to the Arduino was used for network connectivity.

In the Arduino IDE, an algorithm was built that connected to the local WiFi access point, measure both Temperature and Humidity and publish these data points with the selected sampling frequency. To develop the algorithm in the Arduino IDE the library packages for the environmental sensor, Wi-Fi Shield, and ThingSpeak were installed and example code adapted.

The hardware was placed near a window in a domestic property where it was in range of a wireless internet network and left to gather data over the course of a week.

Outside Temperature

The data for the outside air temperature was collected from a nearby weather station using a read API (Extérieur.2018) This data was compared to the calculated dew point to check if there is a high risk of condensation forming around the window.

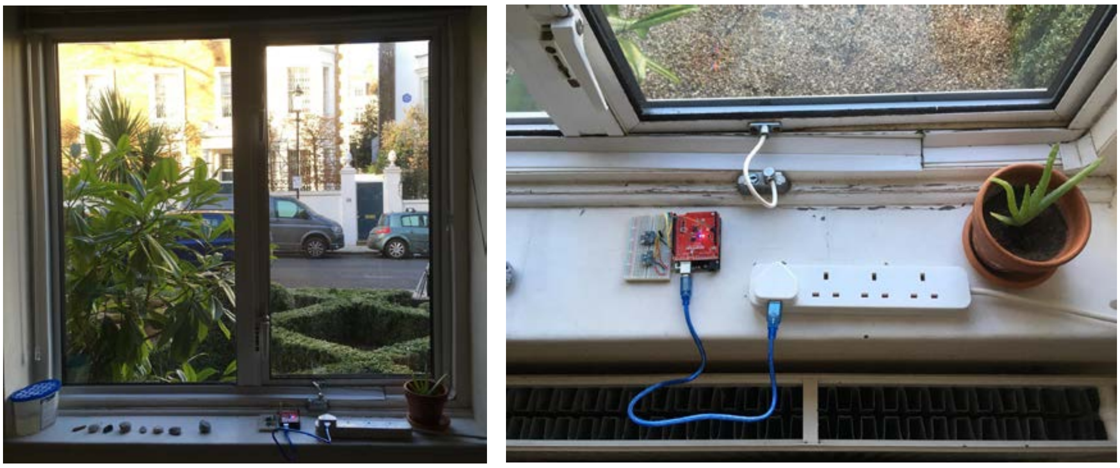

Figure 3 - Data Collection Setup: General Window Area (Left),Upclose of hardware (Right)

3. Data collection, Storage and Processing

Collection

The code running on the Arduino was set to upload data with a sampling frequency of 1 sample every 5 minutes or (0.0033 Hz). This was selected based on initial observations of the data sampled at a high frequency showing the waveform frequency to be about every 30 mins (0.00056 Hz). Hence an appropriate frequency of (0.0033 Hz) that is 3 times the Nyquist frequency was selected. A high sampling frequency made identifying problems faster and hence troubleshooting was made easier.

Storage

The online platform ThingSpeak (MathWorks, 2018) was used to store the data. ThingSpeak was well suited because Arduino and MATLAB libraries already existed, making uploading, storing and retrieving data fairly straightforward.

A unique channel was created with a few independent data fields (Indoor Temperature, Indoor Humidity, Outside Temperature, Dew Pont and Condensation risk). The ThingSpeak Arduino library uses RESTFUL API that communicates over HTTP to read and publish data to the channel (Mathworks, 2018) from the Arduino using the WiFi Client from the ESP8266. A unique channel API key was used to write data to the correct field. The data was made publicly accessible here or by using the read API key found in the channel information.

Processing

Indoor Temperature and Humidity published to the relevant data fields at the correct sampling rate automatically, however the remaining 3 data fields needed separate files for processing and scheduling in order to be published. For processing of the data MATLAB files were created. These files can be found in the Code section of the GitHub page; however the files were hosted on the ThingSpeak Platform. This was so they could be scheduled to run at the same sampling frequency as the other fields. File scheduling was achieved using the built in ThingSpeak App: Time React which triggers files being run at set time intervals (matching the established sampling rate).

Dew Point

The dew point was calculated using equation (1) (Sensirion, 2018):

$$Dp(T,H) = \frac{\lambda(\ln\left(\frac{H}{100}\right)+\frac{\beta T}{\lambda+T})}{\beta-(\ln\left(\frac{H}{100}\right)+\frac{\beta T}{\lambda+T})} \tag{1}$$

Where β = 17.62 and λ = 243.5 °C, T = Indoor Temperature °C , H = Indoor Humidity %

T and H are obtained by subscribing to the fields created using a read API. The output Dp was then published to the Dew Point field using ThingSpeak write functions and the API publishing key.

Condensation risk

This file subscribes to both the Dew Point field but also the outside Temperature API. If the outside Temperature is less than or equal to the Dew Point, then there is a risk of condensation forming around the window. This assumes that the window is around the same temperature as the outside. The logical output of 0 being high risk and 1 being safe was published to another field.

4. Basic characteristics of the end-to-end systems

The overall system is shown in Figure 4. Hardware collects the data and connects to a local WiFi Access Point. The data is published to the ThingSpeak Channel. In the ThingSpeak platform scheduled analysis is performed to populate the remaining fields but also to actuate email notifications in the case of high-risk events. The figures and graphics are displayed on a SquareSpace web domain.

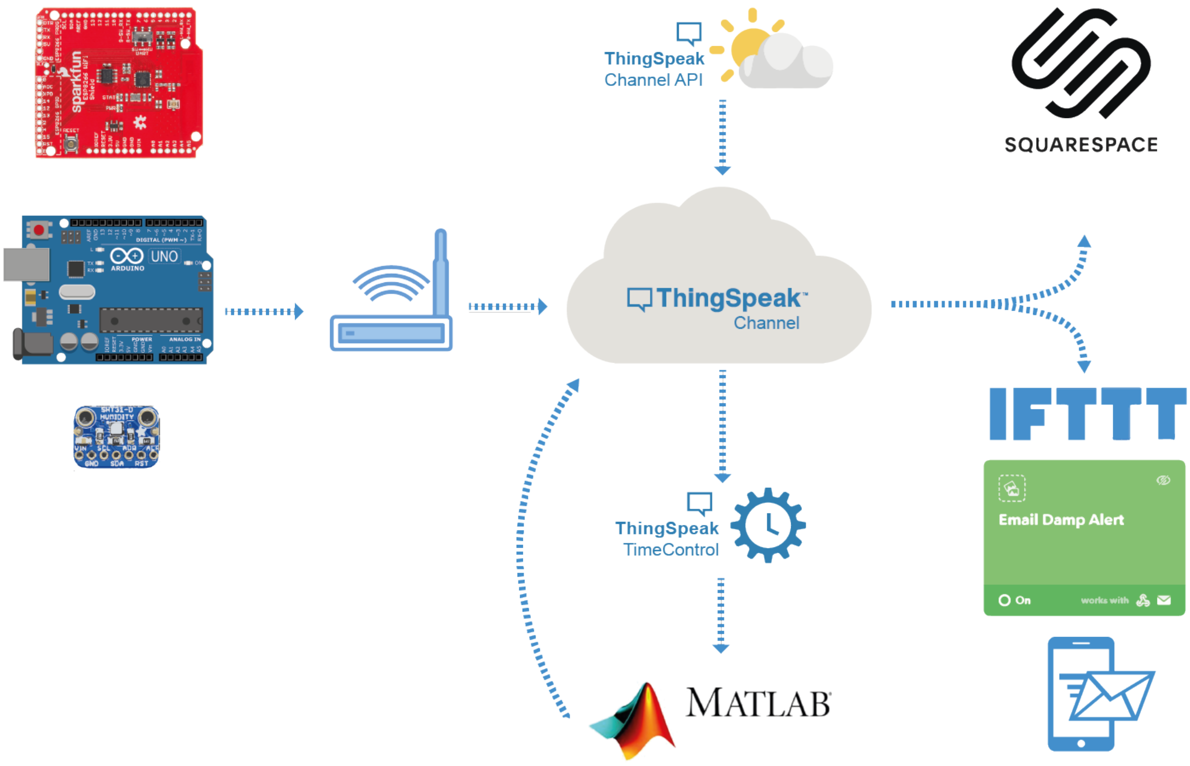

Figure 4 – end to end system diagram

5. Front End Platform

A web dashboard was created and can be accessed at www.jacobmitchelldesign.com/iot. The page was hosted using a SquareSpace domain. It uses live data and HTML objects from the ThingSpeak platform embedded in the web page. It includes a dashboard of information and graphics including the current snapshot information of the room, a list of advice and actions that can be taken and historical data from the last 24 hours. In addition, an email notification was set to alert the user if dangerous criteria was met.

Email notification:

When dangerous criteria are met an email is automatically sent to the user. Figure 5 shows an example email. The email contains details of the criteria, a timestamp of the event and actionable advice. The automatic email was created in MATLAB code hosted on the ThingSpeak Platform. It is triggered by the Time Control App once a day and checks the humidity and condensation risk values of that day. The email service was hosted on IFTTT (IFTTT, 2018) and is triggered by the MATLAB code to send an alert by email. The link provided takes the user to the website above. The following sections detail the website contents.

Figure 5 – Email Notification

Current State:

The most recent data points for condensation risk, room air temperature and humidity are displayed with dynamic graphics. The light will turn bright red if the outdoor temperature exceeds the dew point calculated for the room. This means that there is high risk of condensation forming around the window area. The temperature gauge shows the indoor room temperature. As shown in section 6 temperature and humidity have an inverse relationship, so increasing temperature will reduce

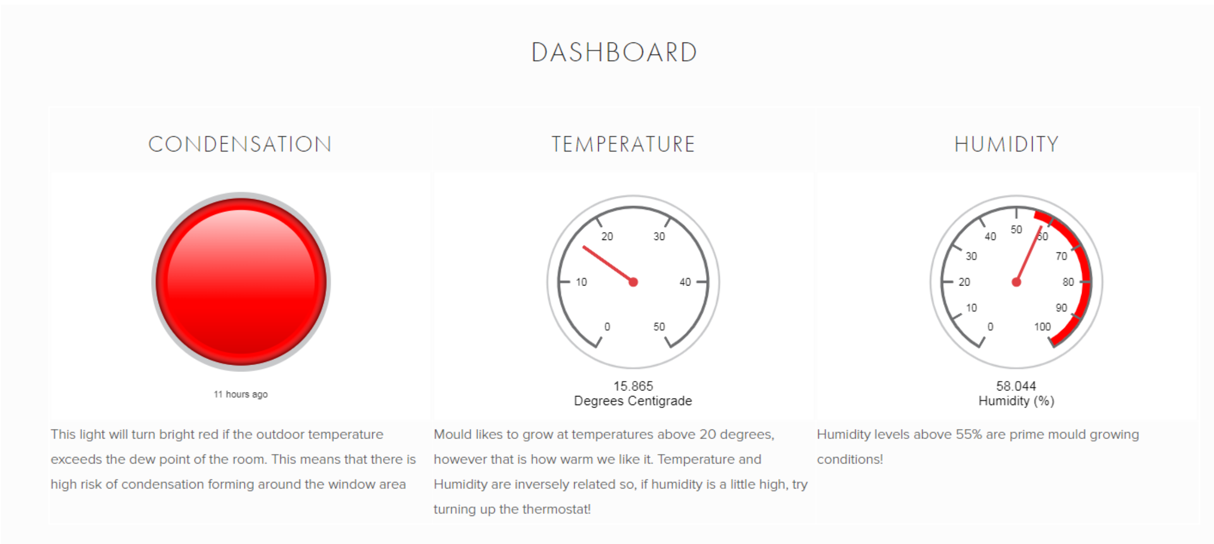

Figure 6 – Live data graphics

humidity. The humidity gauge shows the air humidity in the room and the red bar shows whether it is the dangerous region.

Actions:

Possible actions to reduce the risk of mould growth are shown here. In a potential future system with smart hardware such as a thermostat, electronically controlled windows or a smart dehumidifier could be incorporated to change the environmental conditions. These potential actuations are discussed in more detail in Section 8.

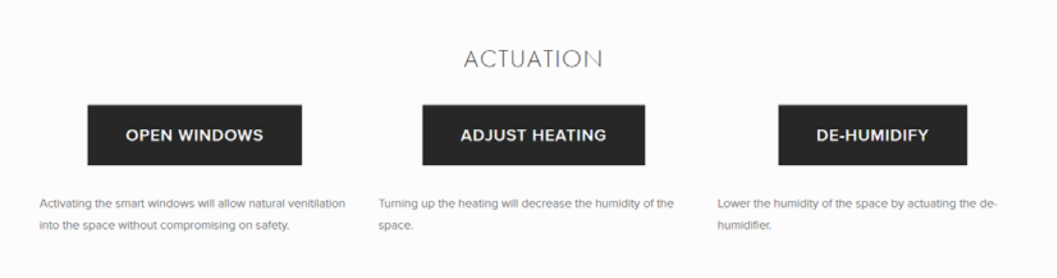

Figure 7 – Actions for the user to take

Historical Data:

Data for condensation risk, temperature and humidity over the past 24 hours are displayed for the user to check up on the conditions. It was assumed that this kind of interface would only be used around once a day as a means of checking up on the state of the conditions.

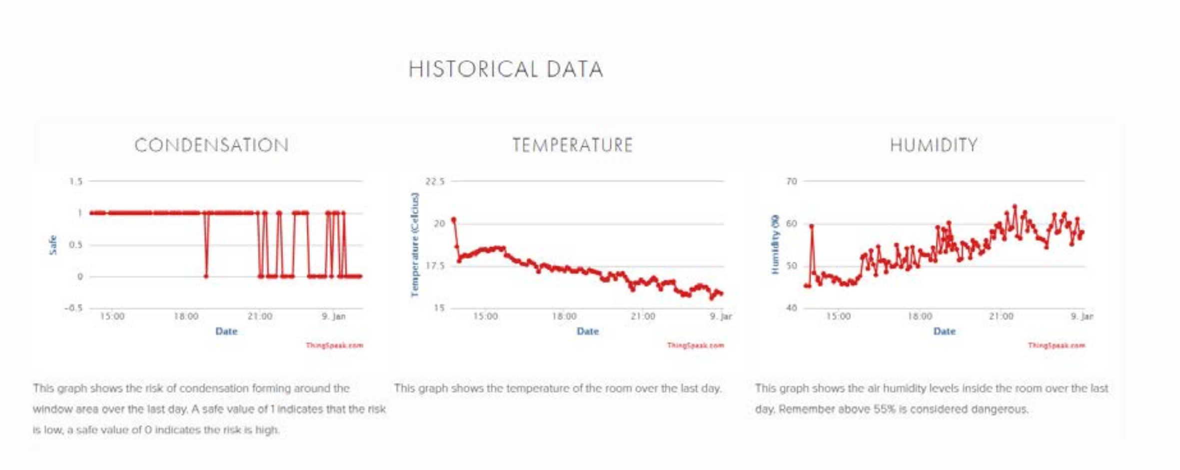

Figure 8 – Historical data

6. Data analytics, inferences and insights

Analysis of the Temperature and Humidity data collected over the course of a week was performed with MATLAB. The *Analysis.mlx* file can be found on the GitHub page which performs and displays all analysis steps.

Accessing data:

The data was acquired using the API read key for the data published to the ThingSpeak channel.

Cleaning data:

The data sampling on the Arduino was only able to publish the variables separately and due to the 15 second publishing restriction the data sources were sampled at slightly different times. In addition, a failsafe in the code caused repeat publishing of the data if it did not publish successfully. However, the code sometimes called an error despite uploading successfully. Therefore, sometimes repeat points were seen. Repeat values were removed by checking the time difference between points. NaN values were also removed and both data set timestamps were merged into one variable so that they could be cross compared.

Visualising data:

The data sets are plotted in Figure 9 and 10. Periodicity is clear in the Temperature plot on a daily or bi-daily basis. The time scale is restrictive, but the trend of the data indicates periodicity on a larger scale, with a peak being reached around Jan 02 2019. It is however impossible to identify this trend without more data. Humidity showed periodicity as well however is a bit less consistent.

Figure 9 shows at some points a clear inverse relationship between Temperature and Humidity. When Temperature rises, Humidity falls. However, at some points during sharp rises they seem to positively correlate.

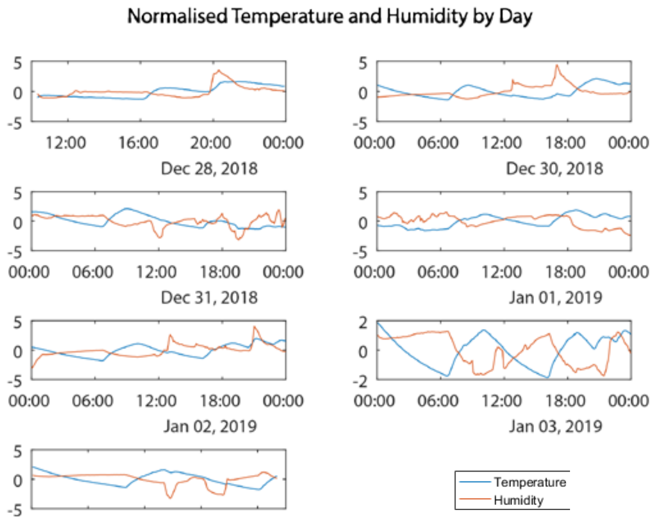

Figure 9 – Normalised Temperature and Humidity by Day

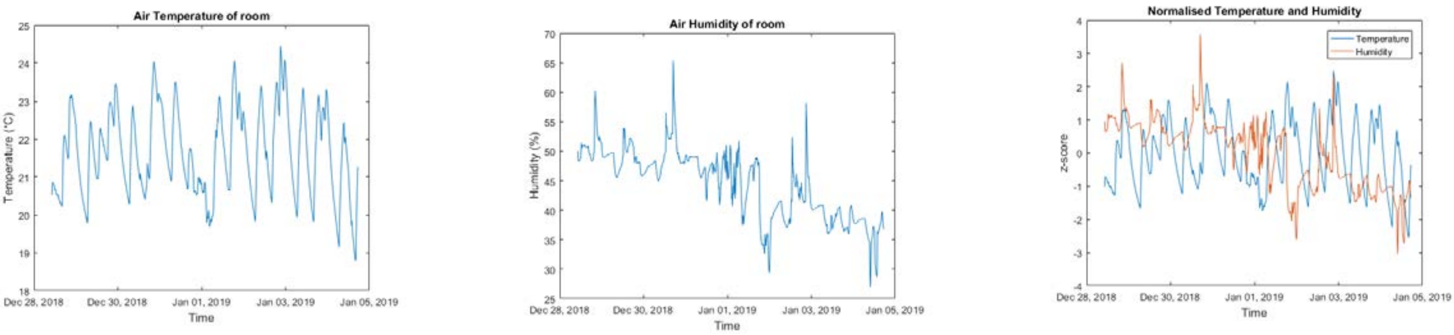

Figure 10 – Temperature, Humidity and both Temperature and Humidity Normalised (Left to Right)

Auto-correlation:

Both time series data sets were tested for auto-correlation. Temperature was found to have a seasonality of 0.48 days. This makes sense as the programming on the thermostat was set to operate twice daily. Humidity was found to have seasonality of 0.25 and 1.25 Days.

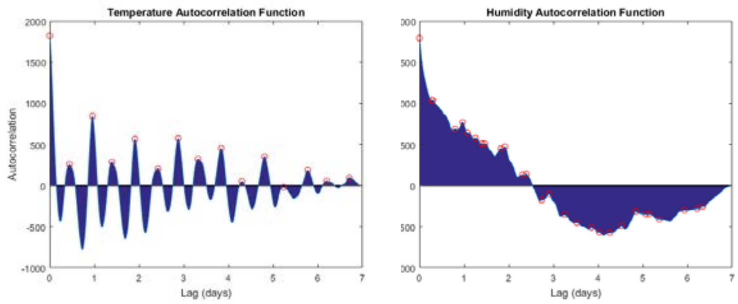

Figure 11 – Autocorrelation of Temperature, Humidity (Left, Right)

Cross-correlation:

A cross correlation was undertaken with both entire time series datasets. The correlation was found to have zero lag difference. The Pearson coefficient was found to be -0.066 with a p-value of 0.0044. This indicates a slight negative correlation between Temperature and Humidity which is significant (as p-value<0.05).

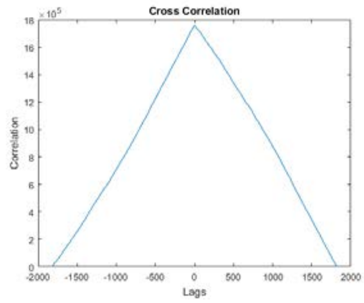

Figure 12 – Cross Correlation of Temperature and Humidity

Regression:

In order to understand this correlation better a regression model was fitted to Temperature and Humidity data taken from one day. This day was chosen because it appeared the least noisy (Figure 13). Temperature was plotted against Humidity (Figure 14) and regression model fitted.

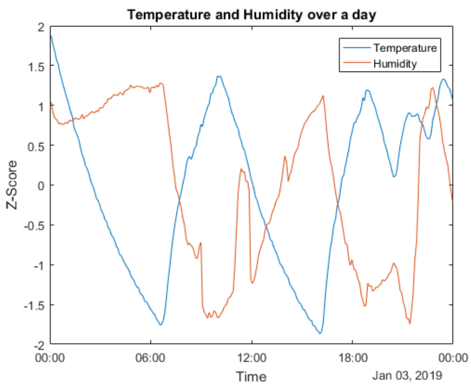

Figure 13 – Time series data from one day

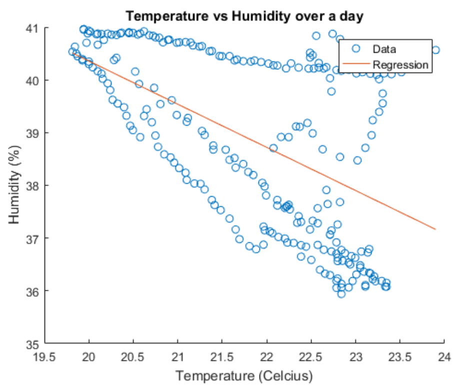

Figure 14 – Temperature vs Humidity over one day

$$H = -0.82T + 56.72 \quad (2)$$

$$R^2 = 0.29 \quad (3)$$

An inverse relationship was found between temperature and humidity (2). The r-squared value (3) is acceptable. However, by looking at Figure 14 it can be seen that T and H have different relationships throughout the day, judging by the lines of visibly different gradients. However, this data is taken on one day and assumes a fixed body of air. Changes to the conditions are inherent from human interaction in the space. Overall, this is an interesting relationship as it means that in order to reduce Humidity, Temperature can be increased.

However, the relationship clearly varies as shown in the difference of effectiveness of the predictive model demonstrated in Figure 15. Therefore it is therefore not reliable enough to predict values of Humidity from Temperature.

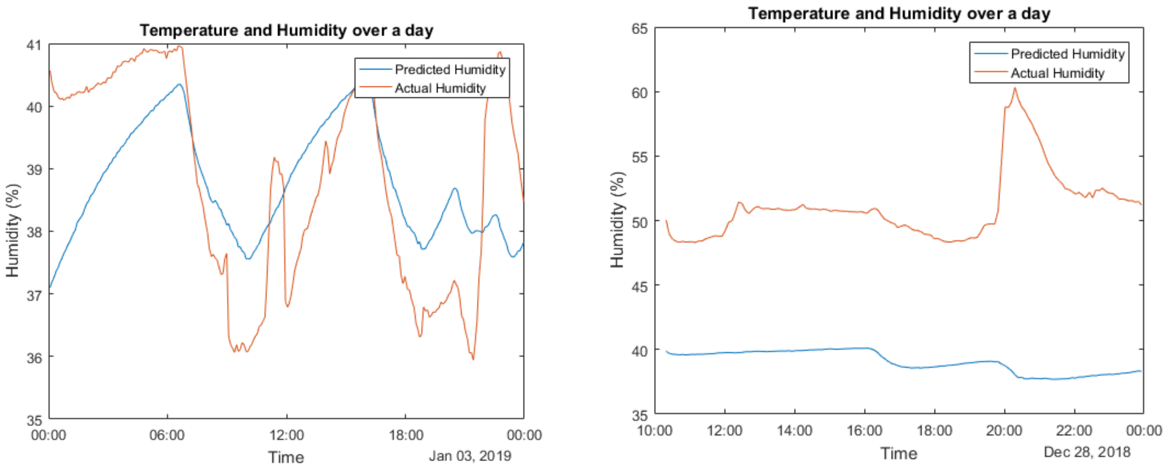

Figure 15 – Predicted Humidity using model on Day 6 and Day 1 (Left to Right)

ARIMA:

An ARIMA model was created in order to forecast the next day Temperature and Humidity based on past data (Figure 16). This used the seasonality calculated from auto-correlation and seasonal moving averages in order to make a forecast for the next day. Predicted Temperature was compared to the actual results (Test Data) and clear good fit can be seen. An ARIMA Model was attempted for Humidity however it was found to be more challenging to forecast.

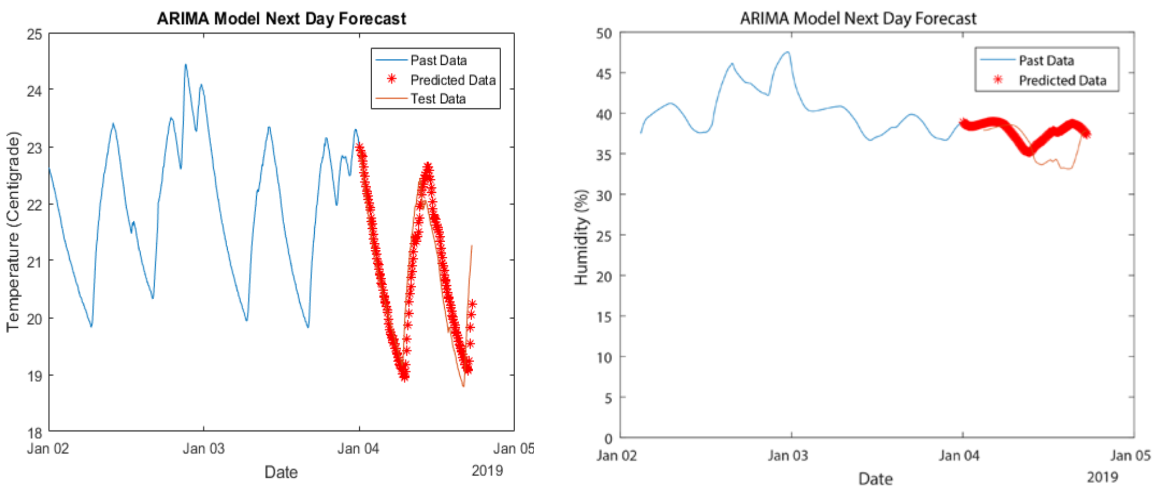

Figure 16 – ARIMA Forecasts

7. Discussion

This system provides a robust means of collecting and analysing Temperature and Humidity data. This is displayed in a user-friendly web platform that conveys useful information clearly.

The prediction about condensation forming on the windows has yet to be proven. Subjectively, it was noted that indeed condensation was seen at times that were calculated as high risk. However, there has been no evidence collected that it is an accurate prediction.

The usefulness of this system is questionable when a lot of this data is obvious. A person can see that condensation is forming or feel that the air is humid. What is more interesting is the automated actuation, which unfortunately was not covered by this project. This work so far does however leave the possibility of adding automated actuation in very easily. Utilising the ThingSpeak React App and setting up an additional Arduino with a WiFi shield and some hacked appliances would have been a great way of demonstrating the concept. However, it was not achieved within the time constraints of the project.

For the analysis section, the data should have been collected over a longer time frame in order to pick up on longer seasonal trends.

Finally, the relationship between Temperature and Humidity is a well-established one. It would have perhaps been more interesting to explore a novel relationship between other factors such as light level.

8. Avenues for future work and potential impact

The current system only alerts the user to high risk events; however, an interesting and more useful development would be to create automated actuation of hardware to change the environmental conditions in the home. For instance, connecting to smart windows, smart dehumidifiers or a smart thermostat could be used to reduce humidity or condensation automatically. This is likely to be more effective than simply alerting the user. The current platform is very open for expansion in this area.

The predictive model produced in section 6 could be developed for humidity in order to predict when high risk events are likely to occur before they do. This way interventions could be scheduled before conditions become high risk.

The GUI can only really be viewed effectively on a full screen computer web browser. Therefore, producing a mobile phone app would make the platform more accessible and convenient. In addition, the hardware could be compacted and integrated into a consumer product in order to make the project scalable and able to reach more people. The software would also have to reflect this, with a more complex website allowing multiple users accessing their own data.

If this system did get scaled up, it has the potential to greatly improve people's understanding about mould growth in their homes and potentially save lives.

9. References

Adafruit. (2018). SHT31 sensor. Retrieved from https://www.adafruit.com/product/2857

Arduino. (2018). Arduino. Retrieved from https://www.arduino.cc/

AWH Solicitors. (2018). Mould and damp repairs your social landlord is responsible for. Retrieved from https://awhsolicitors.co.uk/services/housing-disrepair/mould-damp/

EPA. (2016). Mould information. Retrieved from https://www.epa.gov/mold/mold-course-chapter-2#Chapter2Lesson3

Extérieur

. (2018). Retrieved from https://thingspeak.com/channels/25998

IFTTT. (2018). Ifttt. Retrieved from https://ifttt.com/applets/DTBA4bqQ-email-damp-alert

Lawrence, M. G. (2015). *The relationship between relative*

humidity and the dewpoint

temperature in moist air. (). Retrieved from https://pdfs.semanticscholar.org/e873/a898ba9373af4e12907841411f3e9d83cb9a.pdf

Mathworks. (2018). Rest. Retrieved from https://uk.mathworks.com/help/thingspeak/rest-api.html

MathWorks. (2018). ThingSpeak. Retrieved from https://thingspeak.com

NHS. (2018). Mould and health Retrieved from https://www.nhs.uk/common-health-questions/lifestyle/can-damp-and-mould-affect-my-health/

Platt, S. D., Martin, C. J., Hunt, S. M., & Lewis, C. W. (1989). *Damp housing, mould growth, and symptomatic health state.* (). doi:10.1136/bmj.298.6689.1673

Sensirion. (2018). Dew point calculation. Retrieved from

http://irtfweb.ifa.hawaii.edu/~tcs3/tcs3/Misc/Dewpoint_Calculation_Humidity_Sensor_E.pdf

Sparkfun. (2018). Esp8266. Retrieved from https://learn.sparkfun.com/tutorials/esp8266-wifi-shield-hookup-guide/all