by home owners in daily life. The system is wireless therefore more adaptable and cost-effective. The mobile monitoring system provides security against intrusion as well as automates various home appliances using SMS. The system uses GSM technology thus providing ubiquitous access to the system for security and automated appliance control.

Hoque *et al., (*2019), in their research 'Design and implementation of an IoT-Based Smart Home Security System' layed out an architecture for a cost-effective smart door sensor that informs a user through an Android application, of door open events in a house or office environment. Their architecture uses an Arduino-compatible Elegoo Mega 2560 microcontroller board along with the Raspberry Pi 2 board for communicating with a web server that implements a RESTful API. They were able to present a low-cost architecture using Radio Frequency based communication in a household to create an IoT-enabled smart home security system. Using affordable components such as microcontrollers from Elegoo and Raspberry Pi and RF signals as a communication channel between these devices, it was possible to develop an IoT system that allows users of a household to view when a particular door has been opened.

3. DESIGN METHODOLOGY

This section describes the process in training the model. The first step is to get the data, which will be used as a raw material for building the model. To this effect, 1,319 records of a dataset that contained activities performed by the elderly in their homes was acquired. The Time Duration, Age of the activity performer, Gender, Height, Mass and BMI are features in the sourced data. Figure 1 shows a sample of the dataset. The design of the system is as shown in the figure 2.

Activities	Activities_Id	Start time	End time	Duration	Status	Age	Gender	Height	Outcome	Mass	BMI
ToiletFlush	14	08:35:46	08:35:47	00:00:01	Y	22	M	1.63	0	67	26.99
Hall-Bathroom door	6	08:46:00	08:46:01	00:00:01	Y	29	G	1.63	0	67	32.34
Hall-Bathroom door	6	10:00:29	10:00:50	00:00:21	E	84	M	1.67	1	69	25.39
ToiletFlush	14	10:01:35	10:01:56	00:00:21	E	89	M	1.49	1	61	25
Hall-Bedroom door	24	10:26:10	10:26:51	00:00:41	E	55	M	1.66	1	69	20.15
Hall-Bedroom door	24	10:26:14	10:26:15	00:00:01	Y	27	G	1.75	0	68	24.34
Pans Cupboard	18	10:26:30	10:26:54	00:00:24	E	70	G	1.54	1	67	19.28
Dishwasher	13	10:26:36	10:26:51	00:00:15	E	55	G	1.51	1	67	21.45
Plates cupboard	9	10:27:46	10:27:58	00:00:12	E	50	M	1.67	1	70	21.45
Freezer	17	10:27:50	10:28:05	00:00:15	E	46	M	1.66	1	57	28.44
Fridge	8	10:29:41	10:29:44	00:00:03	Y	28	G	1.56	0	57	30.63
Fridge	8	10:56:24	10:56:28	00:00:04	Y	20	M	1.55	0	66	23.59
Hall-Toilet door	5	11:08:13	11:08:44	00:00:31	E	51	M	1.68	1	69	18.94
Hall-Bathroom door	6	11:12:29	11:12:32	00:00:03	Y	23	G	1.63	0	69	26.14
Hall-Toilet door	5	11:12:33	11:12:54	00:00:21	E	59	G	1.57	1	70	26.75
ToiletFlush	14	11:15:03	11:15:04	00:00:01	Y	18	G	1.76	0	59	21.05
Hall-Bathroom door	6	11:15:06	11:15:27	00:00:21	E	46	M	1.78	1	68	26.71
Hall-Toilet door	5	11:15:42	11:16:02	00:00:20	E	64	M	1.64	1	57	23.73
Hall-Toilet door	5	11:22:57	11:22:58	00:00:01	Y	28	G	1.78	0	72	24.8
Frontdoor	12	11:36:02	11:36:43	00:00:41	E	86	M	1.7	1	61	20.8
Frontdoor	12	11:36:06	11:36:47	00:00:41	E	61	M	1.69	1	62	22.84
Frontdoor	12	06:18:29	06:18:34	00:00:05	E	69	M	1.82	1	59	23.46
Hall-Bedroom door	24	06:19:34	06:19:55	00:00:21	E	69	G	1.92	1	67	21.43
Hall-Bathroom door	6	06:20:27	06:21:00	00:00:33	E	80	G	1.64	1	59	20.89
ToiletFlush	14	06:20:52	06:20:53	00:00:01	Y	41	G	1.75	0	66	23.04
Frontdoor	12	06:21:17	06:21:22	00:00:05	E	81	M	1.81	1	60	18.47
Hall-Bedroom door	24	08:33:33	08:33:54	00:00:21	E	48	M	1.67	1	57	22.19

Figure 1: Dataset Sample

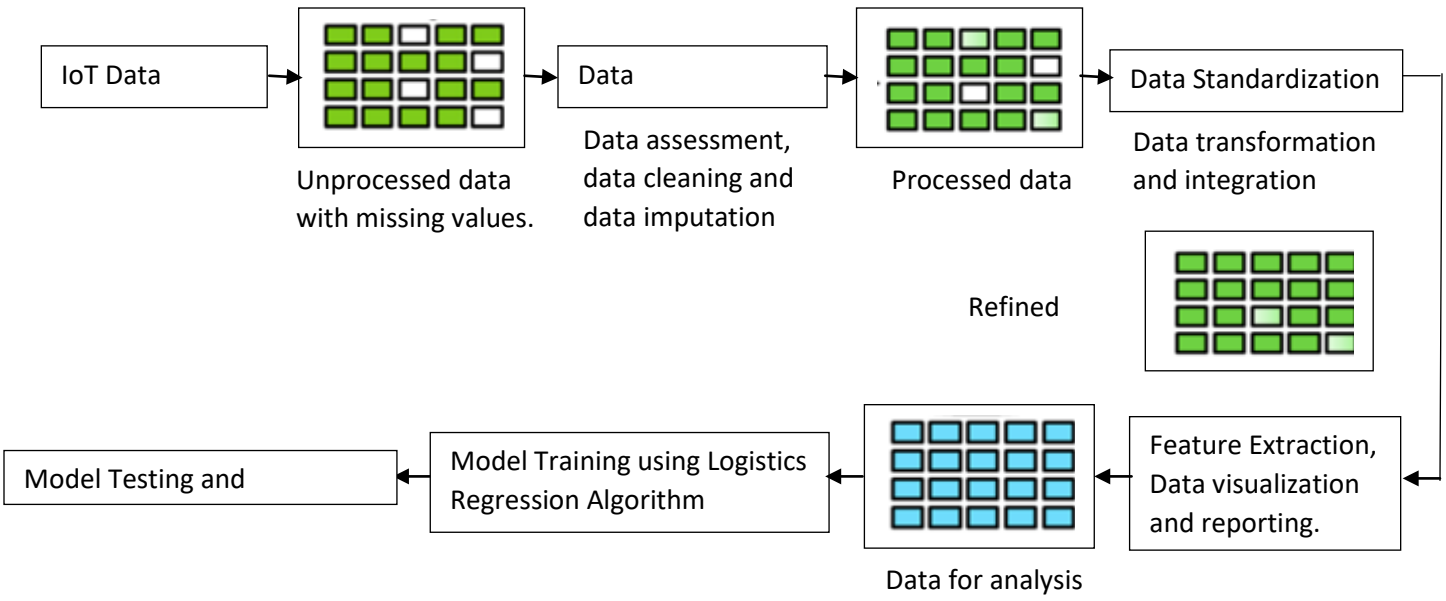

Figure 2: Design of the system

Choice of Algorithm

The desired output is a double state status of the smart home, which should be mutually exclusive. Each state can only exist when the other state is null or void. We want to determine if the home is safe or not safe. Putting this into consideration, there is need for a machine learning algorithm that has a dichotomous output or binary output. An output that is based on classification under only two distinct heads e.g. Yes/No, 1/0, Safe/Not Safe etc. Logistic Regression is one of the best algorithms that is suited to achieve a distinct classification for statistically compiled/mined raw data. Logistic Regression is the most commonly used machine learning algorithm for two-class classification. Looking at the dataset, the "Status" column which states whether the subject of observation is a young or elderly is represented by "Y" or "E" respectively. This means that if the status is Y, then the home is "Not Safe" but if the status is "E" then the home is "Safe".

Explaining the concept as it applies to our case study, logistic regression was drawn from linear regression equation. However, in the case of logistic regression the output has to be binary: 0 or 1, which we can relate to our Young or Elderly.

Linear regression equation is represented thus

$Y= B_0+B_1x_1+B_2x_2+\ldots+ B_nx_n$

Here X_1, X_2 $\ldots X_n$ are independent variables that affect the final value of Y, which means their increase or decrease will also increase or decrease the value of Y.

B_1, B_2,$\ldots B_n$ are multiplying Coefficients of the X parameters that will be used to compute the final output Y.

In this research, elderly people are known to be much slower in carrying out activities than younger people. Hence the duration of an activity will tend to indicate whether that activity was performed by a young chap or an elderly person. Relating it to our case, the dependent variable

will be the Status, and our independent variables will be the duration which is computed from End Time subtracted from Start Time.

Mathematically, the linear regression equation is developed to a logistic function used in logistic regression. The logistic regression equation called the logit function is represented thus:

$$\frac{1}{1 + e^{-(y)}}$$

It is also called the Sigmoid Function.

The y in the above equation is the linear regression equation $y=B_0+ B_1x_1+B_2x_2+\ldots+ B_nx_n$.

Logistic regression represents the probability that an event occurred or did not occur. Mathematically,

P= Event occurred

1-P = Event did not Occur.

$$\text{Odds}= \frac{P}{1 - P}$$

Since the Odds must be between 0 and 1 for logistic regression.

$$\frac{P}{1 - P} = B_0+ B_1x_1+B_2x_2$$

$$\text{Ln}\left(\frac{P}{1 - P}\right) = e^{B0+ B1x1+B2x2}$$

A little algebra will arrive at the logit function above, which is $P= \frac{1}{1 + e^{-(y)}}$

A threshold value is chosen such that once the value is above the threshold value the function result is termed positive(+) and mapped to 1, conversely once the function result is below the threshold it is negative(-) and mapped to 0. This is achieved using the principle of *Maximum Likelihood.*

Python programming language was used to build the model using our chosen algorithm. The program used the mathematical procedure offered by logistic regression to output a system that will make these predictions.

Data Analysis

Python environment is started and all the necessary libraries that will aid to achieve our objective are imported. These includes pandas, numpy, seaborn and matplotlib.

Using pandas library, data is loaded into the program and a few rows are outputted to visualize and see if the data spread was normal and that the status and gender observations were not one-sided.

Data Wrangling/Preprocessing

The dataset is prepared to be able to be processed by the python platform. A careful look at our data shows that unnecessary features that will not contribute to the output of the dependent variable, which is the Status of the subject of the observation are in our dataset. Features like BMI, Height, Gender, and Mass were dropped from the dataset.

Some necessary features in the dataset are in the form of string values, these were converted to numerical values for easy processing by the python programming language. Another column called “Outcome” was created to replace the status column. All elderly represented value were matched to value “1” under the outcome and Young were match to value “0” under outcome column. With this we now have

X= Independent variables (Activities, End time, and Start time)

Y= dependent variables (Outcome) i.e. Young or Elderly

Training & Testing

This is perhaps the most important part of the process. The dataset was divided into two significant parts, one part for training and the remaining part for testing the model. The dataset was divided in the ratio of 70%:30%. Where 70% of the data was used for training and 30% was set aside for the testing. The code below performs the splitting of data into training and testing section. Haven done all of these, the dataset is ready to be used to train the logistic regression function.

At this stage of the program, logistic regression function is imported from the sklearn library. The split version of the dataset set aside for training is passed as parameter into the logistic regression function. The program is run and an output is presented by the python program which represents the model. The model is then tested with the 30% dataset that was set aside for this purpose.

Testing of model is done to observe how well the model was trained. There are various evaluation metrics that can be used to measure the success of a classifier.

The variable predictions was created in our program which was used to deduct other success rating parameters for the created classifier model. The “x_test” variable was passed as a parameter to the predict the function. The x_test variable is 30% of the independent variable which we set aside earlier on for testing. After executing that line of code, the variable “predictions” can be used to run other test codes.

4. RESULT AND DISCUSSION

Using the confusion matrix as a performance metrics, the model gave an accuracy of 98%. Confusion Matrix metrics is a table that is often used to describe the performance of a classification model on a set of test data, for which the true values are known. The confusion matrix is gotten from the python program with the following code

```
n [31]: from sklearn.metrics import confusion_matrix

n [32]: confusion_matrix(y_test,predictions)

ut[32]: array([[ 75,   3],
               [  0, 150]], dtype=int64)
```

	PY	PN
AY	75 (TP)	3 (FN)
AN	0 (FP)	150 (TN)

Where PN represents Predicted No, PY represents Predicted Yes, AN stands for Actual No, AY stands for Actual Yes, TP represents True Positives, FN represents False Negatives, FP stands for False Positives and TN for True Negatives.

From the confusion matrix, the accuracy of the developed model is calculated using the formula;

Accuracy = TP+TN / (TP + TN + FN + FP)

Accuracy: (75+150)/ (75+150+3+0) =0.98 0r 98%

The Recall of the developed model is calculated with the following formula;

Recall = TP/ (TP+FN) = 75/ (75+3) = 0.96 or 96%

Recall is defined as the measure of the positive predictions of the classifier got right out of the actual positives.

5. CONCLUSION

Testing the model gave an accuracy of 98% and a recall of 96%. This obviously proved that the developed system can predict to about 98% whether the elderly person is the one at home or not. The developed model was then used to implement a simulation system for securing the elderly in the smart home. The screen shots and how the system works are hereby shown:

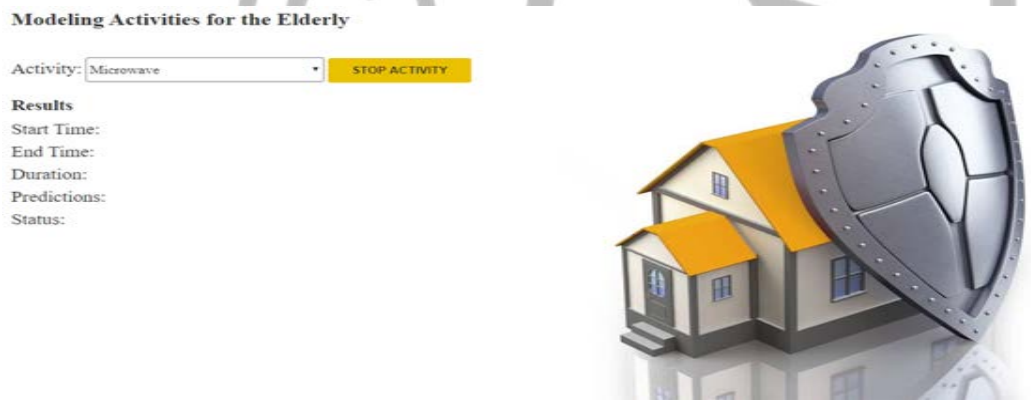

Figure 3: Main Menu of the developed system

The program starts with a user interface which is otherwise referred to as the main menu that direct a user to whatever item he or she wants to process based on the user's choice at that point in time.

The main menu is like a door way to a house. There is no house without door, in other words it is the point of entry into the program. The main menu is shown in figure 3.

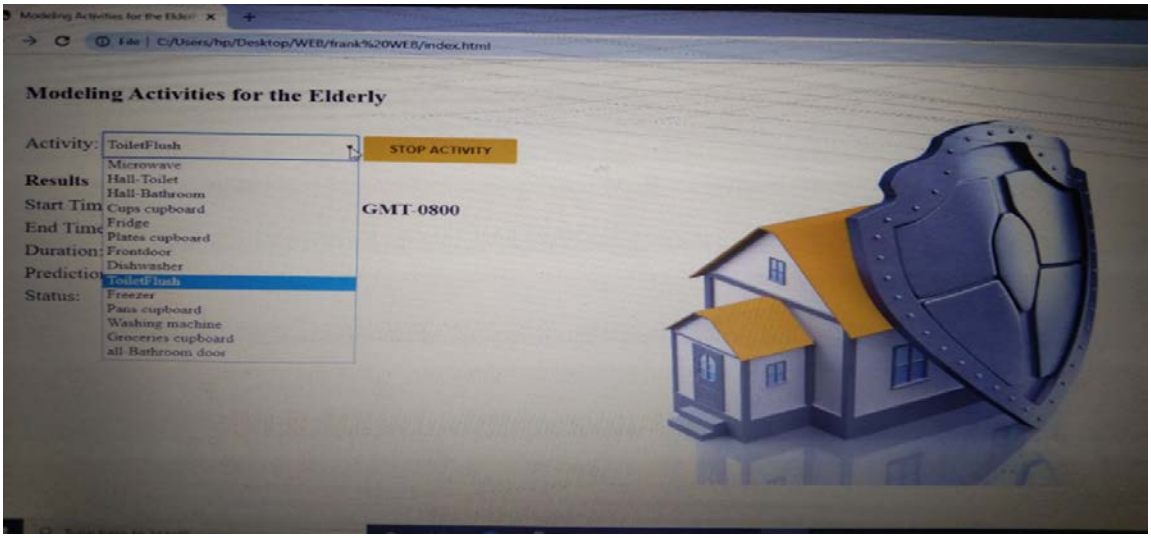

Figure 4: Activities of the smart home.

Figure 4 shows the activities modeled in the smart home. These includes ToiletFlush, Microwave, Hall-Toilet, Hall-Bathroom, Cups cupboard, Fridge, Plates Cupboard, frontdoor, Dishwasher, Freezer, Pans cupboard, Washing machine, Groceries Cupboard and Hall-Bathroom door. There are fourteen (14) activities which are incorporated in our dataset. The menus on this window include Activity (from which an activity to be performed by the elderly is selected), Results which displays the Start Time, End Time, Duration of activities. It also displays the result of the prediction made (which is either Young or Elderly) and the status of the home (which is either Trigger Alarm or Safe). The prediction made is based on the input values which are Activity (the activity to be performed), Start Time (shows the time and date of when the activity started when selected from the drop list.) and End Time (shows the time and date of when the activities is terminated from the Stop activities button). The stop activity button from the GUI terminate the duration of the running activities. The duration label records the difference in duration between the start time and end time for each activity in terms of seconds and used to make prediction based on the developed model.

Figure 5: Screenshot showing Prediction as Young and Status as Trigger Alarm.

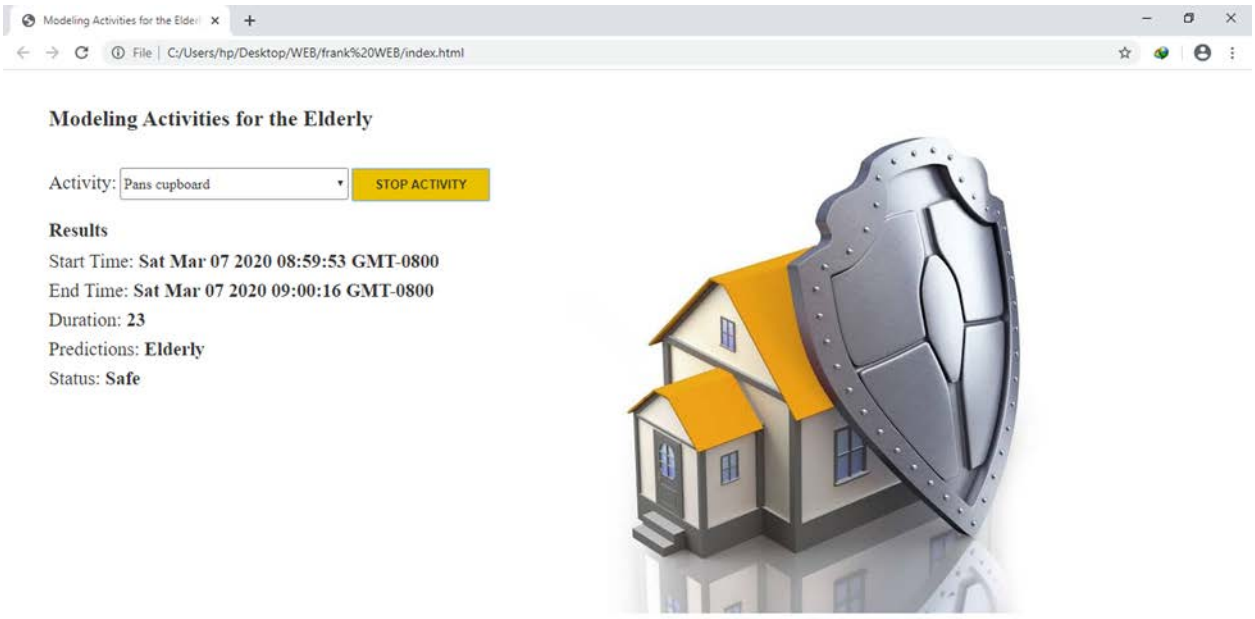

Figure 6: Screenshot showing Prediction as Elderly and Status as Safe.

REFERENCES

- Sintef O. V., Norway, FriessEU P. (2014). "Internet of Things– From Research and Innovation to Market Deployment". *River publishers' series in communications.*
- Satpathy L. (2006). Smart housing: Technology to aid aging in place. New opportunities and challenges. *M.S. dissertation, Mississippi State University.*
- Resul Das, Gurkan Tuna & Ayse Tuna (2015). Design and Implementation of a Smart Home for Elderly and Disabled. *International Journal of Computer Networks and Applications* (IJCNA). 2(6). 242-246.
- M. Shamim Hossain, Md. Abdur Rahman & Ghulam Muhammad (2017). Cyber-physical cloud-oriented multi-sensory smart home framework for elderly people: An energy efficiency perspective. *Journal of Parallel and Distributed Computing. Vol. 103. 11-21.*
- Radosveta Sokullu, Mustafa Alper Akkaş & Eren Demir (2020). IoT supported smart home for the elderly. *Internet of Things.* Volume 11. https://doi.org/10.1016/j.iot.2020.100239. (https://www.sciencedirect.com/science/article/pii/S254266052030072X)
- N.K. Suryadevara, S.C. Mukhopadhyay & R. Wang, R.K. Rayudu (2013). Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Engineering Applications of Artificial Intelligence.* 26(10). 2641-2652. ISSN 0952-1976. https://doi.org/10.1016/j.engappai.2013.08.004. (https://www.sciencedirect.com/science/article/pii/S0952197613001553).
- Deen, M.J. (2015). Information and communications technologies for elderly ubiquitous healthcare in a smart home. *Personal and Ubiquitous Computing.* 19. 573–599. https://doi.org/10.1007/s00779-015-0856-x
- Raad, M.W. & Yang, L.T. (2008). A ubiquitous smart home for elderly. *Information Systems Frontiers.* 11(5).https://doi.org/10.1007/s10796-008-9119-y
- Himanshu Thapliyal, Nathan Ratajczak, Ole Wendroth & Carson Labrado (2018). Amazon Echo Enabled IoT Home Security System for Smart Home Environment.

- Minal S. Khandare & Anjali Mahajan (2010). Mobile Monitoring System for Smart Home. *2010 3rd International Conference on Emerging Trends in Engineering and Technology (I*EEE).
- Mohammad Asadul Hoque & Chad Davidson (2019). Design and Implementation of an IoT-Based Smart Home Security System. 7(2), 85 – 92.