CASAS Smarthome Activity Detection

COMS30051: Applied Data Science 2021 Coursework

Armand Mihai Cismaru
University of Bristol
MEng Computer Science
Bristol, United Kingdom
fz19792@bristol.ac.uk

Edward George Nechitoaia
University of Bristol
MEng Computer Science
Bristol, United Kingdom
dp19681@bristol.ac.uk

Vlad Andrei Bucur
University of Bristol
MEng Computer Science
Bristol, United Kingdom
ot19588@bristol.ac.uk

Matthew Lee
University of Bristol
BEng Engineering Mathematics
Bristol, United Kingdom
nj19257@bristol.ac.uk

Diseng Hu
University of Bristol
MEng Computer Science
Bristol, United Kingdom
zr18100@bristol.ac.uk

***Abstract*—The recent advancements and developments in regular and embedded electronic devices have offered the novel opportunity for smart home sensor systems to be used to unobtrusively monitor human behaviour in natural environments. They can be used to learn and document behaviour patterns and create correlation events that can be linked to changes in areas such as health, daily routine or the productivity of the subject. The demand for related fields like ML (Machine Learning) and its sub-topic DL (Deep Learning) methodologies and approaches has grown in tandem with the rise of this domain and other data-related activity. In particular, here we introduce an approach to activity detection modelling by using a Random Forest classifier and a long short-term memory (LSTM) neural network, respectively. We demonstrate our approach by experimenting on the Center for Advanced Studies in Adaptive Systems Aruba dataset by training and comparing the chosen methods with the testing set. Our evaluations indicate what features are most important to labelling activities and detecting patterns using the different types of sensors. This can be further used for a whole range of analyses, ranging from sleeping patterns to average power consumption. The proposed pipeline can learn from provided annotated sensor data and accurately label incoming new readings. Development has been done using GitHub[1] and Jupyter Notebooks.**

I. Introduction

A. About the dataset

In the last decades, human activity recognition has been the subject of a challenging research area, due to its real-life applicability to different and assisted living (AAL) domains as well as the increasing demand for home automation and convenience services for the elderly [1]. Advancements in technology have made collecting data from homes more accessible and non-intrusive. According to CASAS (Center for Advanced Studies in Adaptive Systems), "since the miniaturization of microprocessors, computing power has been embedded in familiar objects such as home appliances and mobile devices; it is gradually pervading almost every level of society. In the last decade, machine learning and pervasive computing technologies have matured to the point where this power is not only integrated with our lives but it can provide context-aware, automated support in our everyday environments" [2].

Tracking, collecting and modelling activity data can help monitor and understand changes related to seasonal variations, routine patterns or other events, but in a manner that preserves complete anonymity, taking into consideration the strict privacy requirements. Analysing sensor-based time series data can also be used to detect changes in human behaviour related to health events, such as a fall, chronic or incurable disease treatment [3].

The data science application that this report describes proposes a model that can be used to detect and analyse activity based on sensor data. Taking into account that this is a complex pattern detection problem, it was deemed useful to employ different machine learning/deep learning methods in order to compare results and get solid insight. For this purpose, recurrent - Long Short-Term Memory [4] - and Random Forest [5] algorithms have been trained on the CASAS Aruba dataset, which contains extensive, annotated time-series sensor data. The project aims to offer an accurate, reliable detection model and exploratory insight into regular human activity. Its purpose is to offer activity detection that can aid future smart house systems in taking better decisions, help detect changes in the subjects' health and offer insight accessible to non-technical end-users.

The CASAS datasets were introduced by Washington State University. The CASAS Aruba dataset used as a testbed for our project is comprised of one apartment that includes two bedrooms (one with a closet), two bathrooms, a kitchen, dining, office and living. The apartment was equipped with different kinds of sensors (e.g temperature sensors, motion sensors, door sensors). The apartment where the smart home system was installed was occupied by a single elderly woman

[1]Link to the GitHub repository: github.com/vladbucur2000/casas-datascience

whose children and grandchildren visited on a regular basis [6]. Figure 1 below shows the floor plan of the apartment containing the layout of the sensors.

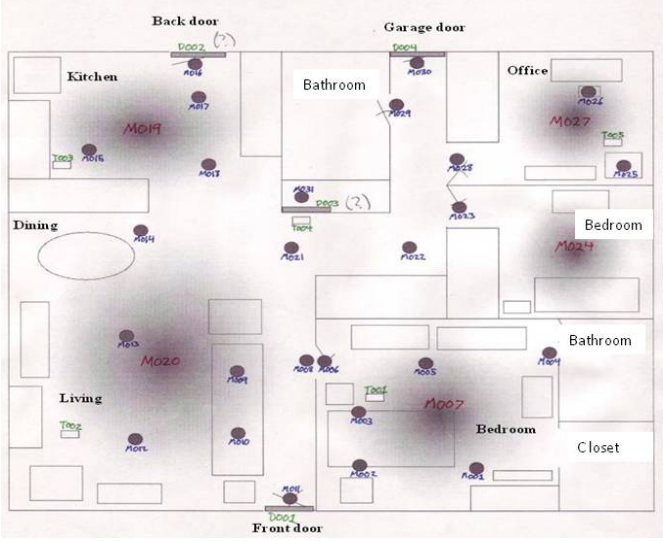

Fig. 1. The house layout of the sensors.

II. Data Preparation

A. Privacy and ethics

One of the first concerns taken into account were the privacy and ethics constraints. The datasets that CASAS offers are ethically sourced from volunteer subjects that consent to having the sensors placed into their own home. The smart house systems are deemed as non-obtrusive, safe and do not collect personal information.

The dataset is provided with the gender of the subject and a few details about family (children and grandchildren visiting) from which we will assume that the person is elderly. Also, the location, Aruba, is known as well as the floor-plan of the apartment. Privacy is ensured as there are no other details provided, such as names, addresses or contact numbers. The dataset is destined for use in research purposes and made publicly available by CASAS of Washington State University.

B. Cleaning

The first step of the preparation consisted of cleaning the data of any typos, trailing white-spaces and wrongly labelled sensors. The cleaning has considered all the possible errors in the dataset, being designed to cover different datasets of similar formats. There have been instances of skewed sensor states across the dataset, for example 50 instances of OFFc, 23 instances of ON5, 22 instances of OFcF and so on. Each state has been individually curated and assigned to its correct state. For temperatures, there were 4 examples of readings that missed the decimal dot, resulting in temperatures of 245 degrees Celsius. Given the size of the dataset, they were treated as statistical error. For wrong sensor codes or invalid date-times, it was impossible to rightfully correct the issue so the corresponding entries have been completely eliminated from the dataset in order to ensure robustness and purity of the data.

TABLE I
Example of the Aruba dataset.

2010-11-04 00:03:50.209589	M003	ON	Sleeping begin
2010-11-04 00:03:57.399391	M003	OFF	
2010-11-04 00:15:08.984841	T002	21.5	
2010-11-04 00:30:19.185547	T003	21	
...			
2010-11-04 05:40:43.642664	M003	OFF	Sleeping end
...			
2010-11-04 05:40:51.303739	M004	ON	Bed_to_Toilet begin
...			
2010-11-04 05:43:30.279021	M004	OFF	Bed_to_Toilet end

C. Pre-processing

1) Data for the Random Forest Model: Random Forest is a model that assumes samples are independent and identically distributed variables (i.i.d). As seen in the example, not all entries have activity annotations. For consistency purposes, transition labels have been added between the end of an activity and the start of the next one. For its duration, all entries take the name of the activity (e.g. for a sleep period, instead of having *"Sleeping start"* and *"Sleeping end"*, with the entries in between not being annotated, now all entries in the period take *"Sleeping"* as annotation). For the periods between activities, there were added labels that represent the transition between the previously annotated activities, like *"Transition_Relax_Sleeping"*.

The timestamps of the sensor entries have been stored in the categorical "Datetime" format, making them unsuitable for further data exploration and modelling. The choice was to use it to create 2 new numerical features and 1 new categorical feature for the events as follows: the day of the week (Weekday), the aggregated time of the day (Seconds), which is represented in seconds relative to midnight, and the part of the day (Daytime) which can be one of the four keys: Morning Night, Morning Day, Afternoon and Evening. For example, Timestamp "2010-11-04 00:03:50.209589" has now become Weekday "3", Seconds "230" (past midnight) and Daytime "Morning Night".

In order to make the data consistent and suitable for the random forest model, we had to preprocess it so that a single row explains the state of the whole house at a specific point in time. We modified the structure of the data so that every sensor has become a feature of the dataset. For door and motion sensors their states have been binary encoded (ON/OFF & OPEN/CLOSE) while the temperature sensors' readings take floating point numerical values.

At the end of the preprocessing, the final DataFrame has 45 columns/features. The last steps of preprocessing consisted of splitting the dataset into instances and target activity labels. In the end, the Standard Scaler method was used to resize the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1.

Table II present a snippet of the i.i.d preprocessed dataset.

TABLE II
PRE-PROCESSED DATA FOR RANDOM FOREST MODEL.

Weekday	Seconds	M001	...	T005	...	Morning
3	230	0.0	...	23.018	...	1
3	237	0.0	...	23.018	...	1
...						
5	85379	1.0	...	23.50	...	0
...						
2	42689	1.0	...	17.25	...	1
2	42692	0.0	...	17.75	...	1

TABLE III
THE FREQUENCY OF ACTIVITIES IN THE DATASET

Activity	Number of occurrences
Relax	2918
Transition_Relax_to_Relax	1921
Meal_Preparation	1606
...	...
Transition_Leave_Home_Meal_Preparation	1
Transition_Meal_Preparation_Respirate	1
Transition_Relax_Leave_Home	1

2) Data for recurrent models: If the preprocessing of the data for the random forest model consisted in splitting the timestamps into more features, the data for the recurrent models does not take their explicit value into consideration, but the order of the data instances remains important. Before doing any custom preprocessing, we applied the same technique of adding the transition labels to the empty labeled data. The data needs to be segmented for each activity so the final goal of the preprocessing would be to have an array of arrays. Each array would represent the chronological progress regarding the sensors states of the current activity [7].

To achieve this, we created a dictionary in which the sensors are the keys and the values are different numbers representing all of their possible states. For example, M001ON becomes 1, M002OFF becomes 2 and so on, in chronological order. If the activity "Sleeping" has the following representation sequence: M001ON, M002ON, M003ON and M001OFF, these different instances become only one sequence: [1, 3, 5, 2]. As we can see from this example, numbers 1 and 2 represent the same sensors but in different states. This data format preserves the time-series property of the dataset and encodes it powerfully enough to create patterns. Some arrays do not have the same length because some activities take longer (they are recorded in more time-steps).

For feeding the dataset into the RNN, we need the sequences/arrays to have the same length, so we apply a padding (adding 0s to shorter arrays to make them match a fixed size for all). Furthermore, a problem occurs when compressing multiple instances into only one preprocessed instance. This problem relates to the frequency of the labels, as there are activities that appear only once (see Table III). The model is going to overfit these activities due to their presence in only one training or testing set. To mitigate this issue, we decided to remove these (8 such activities had been removed - only transition activities).

III. DATA EXPLORATION

A. Structure of the dataset

The Aruba dataset was selected among all available CASAS datasets motivated by the fact that it spans for a time period of 7 months and is annotated. There is a number of 11 activity classes as follows: Meal Preparation (1606), Relax (2910), Eating (257), Work (171), Sleeping (401), Wash Dishes (65), Bed to Toilet (157), Enter Home (431), Leave Home (431), Housekeeping (33), Resperate (6).

The dataset is quite simple and contains a set of three or four features for each entry: the timestamp of the sensor reading, sensor ID, state and corresponding start/end of an activity (if present, otherwise null). Table I shows an example of the Aruba dataset, with annotations that indicate whether an activity has started or ended.

B. Statistical analysis

To determine what models and features would be suitable for use in our pipeline, we needed to explore the data. The Aruba dataset contains an extensive collection of entries, collected from three types of sensors: Motion, Door Closure and Temperature. The ∼1.72 million entries have been collected using 39 sensors over the course of 7 months.

Annotation provides an invaluable asset to this dataset, being the cornerstone of our work. Even though they represent a mere 0.38% of the entries in our dataset, they are pivotal to the training data that will be fed to the ML models. The activity annotations represent start/end markers for known recorded activities. By filling the entries between them with their corresponding activity labels, we learn that in 45.3% of the 7 months' time the occupant is involved in an activity of any type, while the rest is represented by transitions between activities.

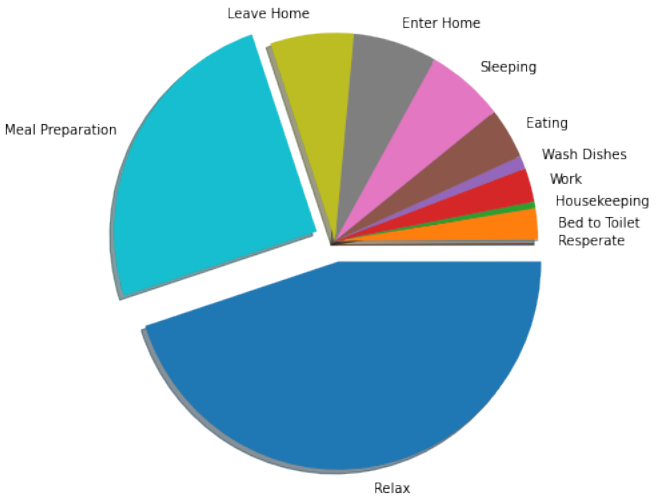

Fig. 2. The proportion of recorded activities.

Figure 2 illustrates the composition of the dataset based on activities. *'Relax'* and *'Meal Preparation'* are the most prevalent, representing 70% of the recorded activities of the dataset.

Each sensor reading holds its state, whether ON/OFF for motion sensors, OPEN/CLOSE for door sensors or floating point numbers for temperature readings. The categorical states are evenly divided, while temperatures vary between 16.0 and 43.0 degrees Celsius with a standard deviation of 3.02 and a mean of 23.01. The temperature data contains 4 skewed entries, probably due to the omission of the decimal dot, but including them in the analysis does not impact the analysis given the large size of the dataset. On the other hand, there are no outliers as the temperatures recorded match the climate of the location of the experiment.

The timestamp feature column is in the date time format, providing lots of detail about the time relativity of every recorded reading. By breaking it into a set of features aforementioned in the *Data Preparation* section, we can extract much more meaning and insight from the data. Figure 3 shows how the activities recorded are distributed across weekdays.

We can correlate the fact that the occupant is an elderly person with the apparent balance of the activities, especially on working days vs. weekends. It seems that Tuesday is the most active day relative to the mean, with Thursday on the opposite end. In terms of the quality of the date-time feature, the dataset contains robust and correct data, but any wrong entry would be discarded.

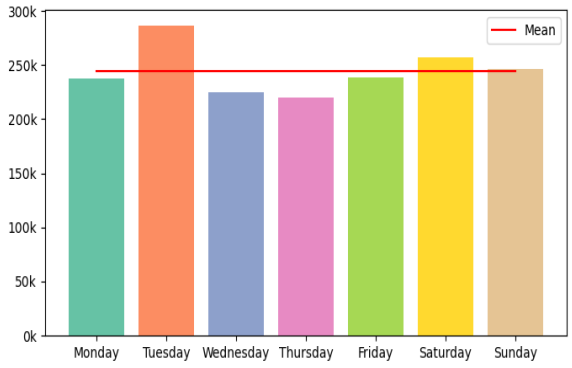

Fig. 3. The distribution of cumulative activities by weekday.

C. Potential of well-processed data

The motivation of this project is to build a reliable model for classifying sensor activity data. But what purpose would it serve if the activity data did not have real-life applicability? Ranging from learning one person's habits to monitoring activities influencing the subjects' health, activity pattern recognition can serve a consistent array of uses.

Analysing the preprocessed data that would be fed into the Random Forest model has led to interesting insights such as sleep statistics over months' time or the analysis of average activity in different parts of the day. Figure 4 shows how daily sleeping hours fluctuate over the course of the whole experiment duration. Sudden spikes might be the result of the subject leaving the house or resting after more active periods. Given that the recording starts in November and ends in July, and Aruba is located in the Caribbean, it is interesting to see how the hours trend is declining, probably being correlated to rising temperatures during spring.

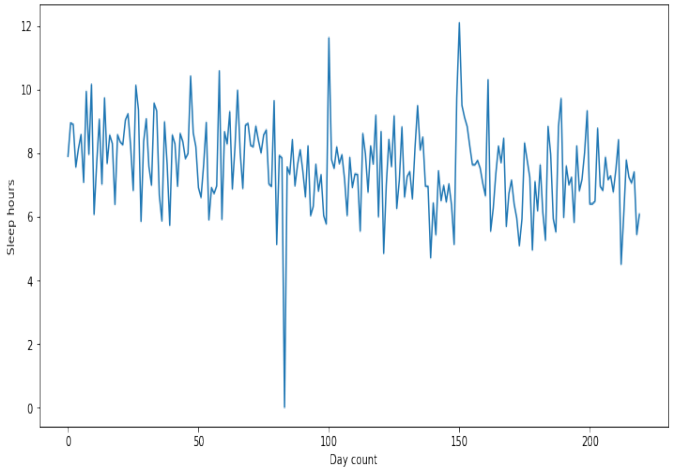

Fig. 4. Daily sleeping hours over the duration of the experiment.

Apart from being split into the days of the week, the data has been added the feature that keeps track of the time of the day that the activity takes place, as follows: 12AM-6AM Morning Night; 6AM-12PM Morning Day; 12PM-6PM Afternoon; 6PM-12AM Evening. The combined average of these insights is an useful way to determine a person's behaviour patterns relative to the 2 dimensions of time, and can be used to predict things like energy usage in a house, or be used by smart appliances, like a coffee machine that knows when to turn on and make a coffee by itself.

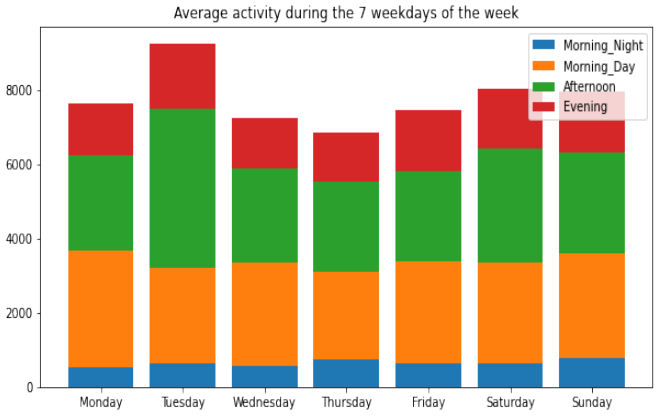

Fig. 5. Avg. activity relative to time.

Figure 5 explains the average activity in the Aruba apartment based on their place in time. Without knowing the age

range of the occupier we could spot that it is someone who does not leave the house during the day and has little activity during Evenings and Nights, also while being consistent throughout working days and weekends. Our focus has to avoid inferring anything about the person's privacy, so this will be useful for general activity patterns and medical purposes.

D. Feature correlation

The original dataset contains features comprised of mostly categorical data, the only exception being the Temperature sensor readings. The timestamps are considered as categorical as they are packed into a pre-defined data-time format. To be able to identify correlations between features that could be plugged into a ML model, the data had to be converted into continuous form (this is detailed in the previous section of the report).

Calculating the correlations between the target variable and the features yield interesting results. This was done for the Random Forest data, as the batch processed for the LSTM algorithm aggregates the original sensor codes and their readings altogether from the start, as detailed above, so this analysis would not be appropriate.

As the features are a mixture of continuous and categorical variables, a set of different techniques have been employed to describe the relationship between different pairs of variables, as follows [8]:

- Data from contingency tables was used to extract the correlation between categorical variables. For each pair, its corresponding confusion matrix has been used to calculate Cramer's V coefficient that uses the Chi-squared statistic. The findings show a relatively modest correlation between the motion sensor features and the labels, with a mean of 0.20. The same goes for door sensors at a mean of 0.24. The highest coefficient of the motion sensors has been recorded from M009 at 0.47 and from D004 at 0.38 for doors. Meanwhile, Daytime seems to have the highest relevance to the target at 0.42 while Weekday comes at 0.18.
- For categorical and continuous variables it is more difficult to determine if there is an outright relationship. 'Seconds' comes as having close to no correlation to the target variable in tests like Point-Biserial Correlation or Kruskal-Wallis H-test (used from *scikit.stats* [9]), even though it is the feature that places an action in the time dimension of a given day. For example, we might observe a greater incidence of the 'Sleeping' activity close to midnight *(Seconds=0)* or 'Preparing_meal' around midday or evening, thus relevant to our analysis.
  Similarly, temperatures yield low correlation scores as they might be tied to the season of the year, hour of day or other factors. But, by observing temperature anomalies in certain times of the year we might suppose that the subject might be cooking or being away from home.

Given the complexity of the features that define a certain activity, or transition from one activity to another, we assumed that a combination of all of them is important to making an accurate prediction model. For the first round of Random Forest training all the features were used.

IV. Data Modelling

The main purpose is to model activity recognition by taking 2 machine learning approaches: the Random Forest Model and the Recurrent Neural Network. In order to achieve our main goals, we have made use of various tools specific to the data science area of expertise. For implementing and training the models, we used Python. For the random forest, we used scikit-learn library[10]. This library offered us tools to split the data, apply stratified cross validation, instantiate the random forest model and finally, training and testing. For the recurrent model, we used scikit-learn for data management and Keras[11] with Tensorflow[12] for designing the architecture, training and testing. The main development environment is Google Colaboratory, which provides us with powerful cloud machines.

Figure 6 shows the development pipeline. The yellow color shows the end parts of the pipeline (the input and the output). The green color represents the "Data Engineering" part, where it prepares the data for training. The purple color represents the "Machine Learning" step, which includes both training and validation. It has been designed to work fully automatically until the last step, where the results are manually treated and visualised.

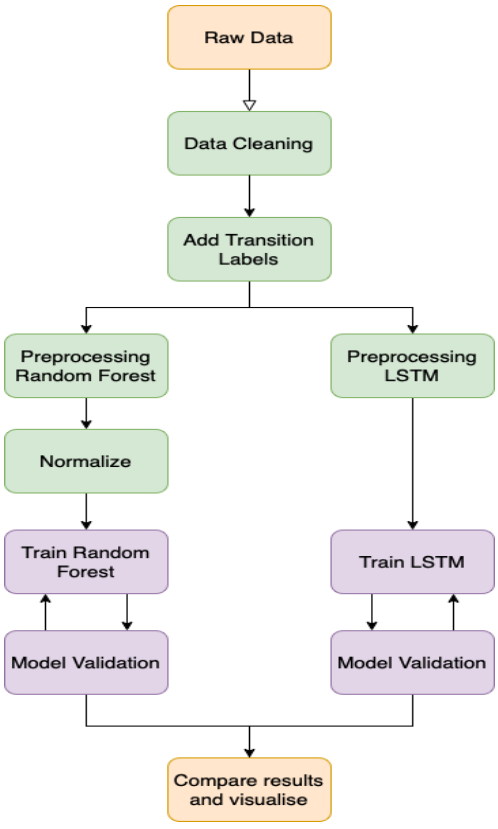

Fig. 6. Modelling Pipeline

A. *Random Forest*

Taking into account the complexity of the dataset (the high dimensionality of the feature space), our choice for modelling was a Random Forest.

Random Forest is an ensemble method of decision tree algorithms. A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a motion sensor activates or deactivates), each leaf node represents a class label (decision taken after computing all features, e.g. model recognises that the activity is "Sleeping") and branches represent conjunctions of features that lead to those class labels. The decision tree is built and trained using the CART algorithm [13]. The structure of a decision tree is suitable for our kind of dataset. It will include multiple layers for verifying the current state of the house. The edges represent queries for the day of the week, the time of the day, for the state of the sensors etc.

Random forest trains multiple decision trees on segments of the initial data (bagging), and chooses the most frequent predicted label. The main advantages of using a Random Forest are that it can handle high dimensional data efficiently, it improves the model accuracy and minimizes the overfitting of the data by having more than a single decision tree.

The model random forest configuration is:

- Decision Trees (n_estimators): 100
- max_features: auto
- max_depth: None
- criteria: gini

The pre-processed CASAS Aruba dataset was split into train and test sets (70% - 30%), for testing the accuracy of our model after the training. Besides the initial train-test split, another metric function was necessary in order to keep track of the performance. During training we have used a Stratified K-fold cross-validation with K = 3 splits. This is a method of cross-validation which helps us choose an equal number of labels when splitting the data. This way both the train and test set will have the same proportion number of each label.

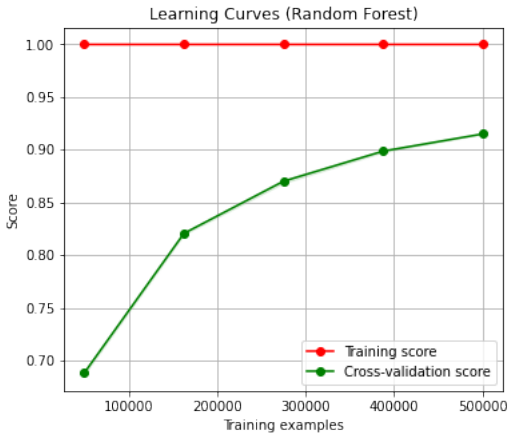

Fig. 7. Learning Curve - Random Forest training

In Figure 7, there is the Learning Curve representation for 500,000 elements of our dataset. It can be observed that the model is overfitting by looking at the training accuracy. However, the cross-validation accuracy was increased along with the number of training examples, getting closer to the training accuracy curve, scoring up to 90% accuracy. It is expected that using a bigger dataset would produce a decreasing training line and an increasing cross-validation line converging to each other, which is often found in complex datasets [14].

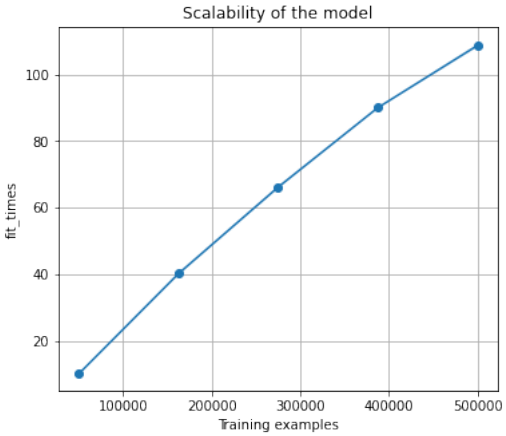

Fig. 8. Scalability of the model - Random Forest training

In Figure 8, it can be seen that the training fit time increases linearly with the number of training examples.

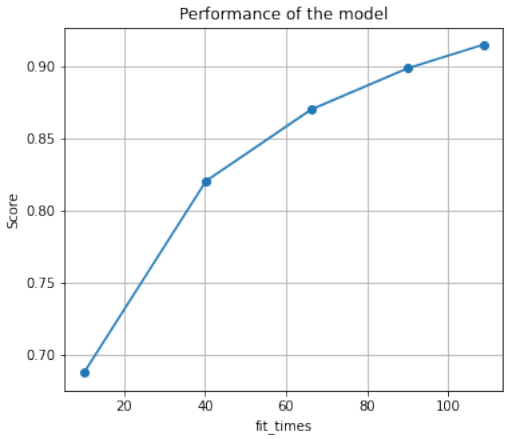

Fig. 9. Performance of the model - Random Forest training

In Figure 9, it can be observed that the line seems to converge. The highest increase in accuracy is in the beginning when the fitting time of the model changes from around 15 to 40. As the fitting times and the training examples grow respectively, the accuracy increase turns to be gradually smaller than the previous one.

Figure 10 shows the feature importance for all the 45 features of the Random Forest Classifier. The features are labeled from 0 to 44: Weekday (0), Seconds (1), Daytime (2), M001 (3), M002 (4), etc. The most important one is

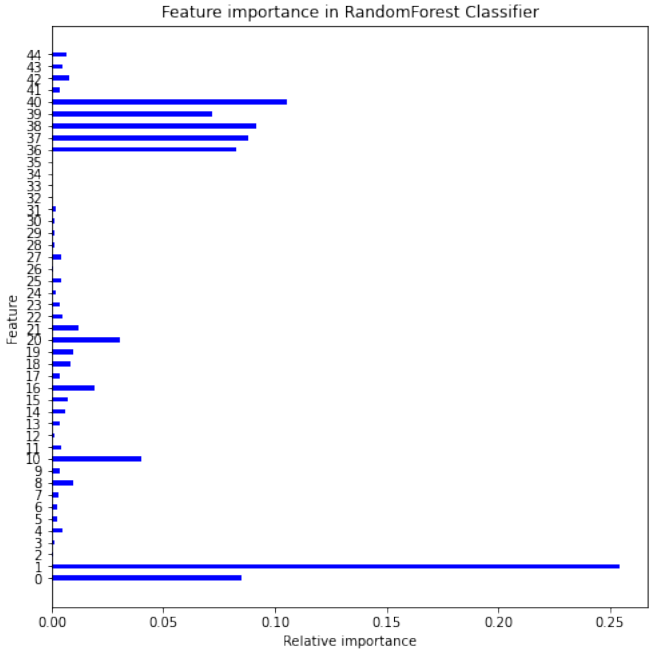

Fig. 10. Feature Importance - Random Forest classifier

"Seconds", showing that there is a high correlation between the time of the day and each activity. It can be observed that Weekday (feature 0) has high importance, showing that some activities are specific for some weekdays. A cluster of sensors between 36 and 40 seems to have a lot of impact on the model performance. These correspond to the temperature sensors, which are located in: Bedroom, Living, Kitchen, the main hallway that connects all the rooms and the Office. The accuracy on the test dataset that we hold out initially is 95%.

In order to improve the accuracy of our model and to avoid overfitting, we removed the low-important features. The relative importance threshold used is 0.25. The extracted features are: all the temperatures sensors, 2 motion sensors, "Seconds" and "Weekday". Then we followed the same procedure as above for training. The accuracy obtained on the test set is 98%, giving a significant improvement of 3% from the previous model.

B. Long short-term Memory

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning (DL). Unlike standard feedforward neural networks, LSTM has feedback connections[15]. It can process not only single data points (such as images), but also entire sequences of data (such as speech, video, or in our case, chronological sensors activity). As the name suggests, LSTM is able to retain memory of the past instances for longer or shorter periods. These periods are decided by the model itself while training during "Backpropagation" (the process of updating the trainable variables).

As opposed to random forest, LSTM does not have only input/output flow (variable "x" and "c" in Figure 11). There

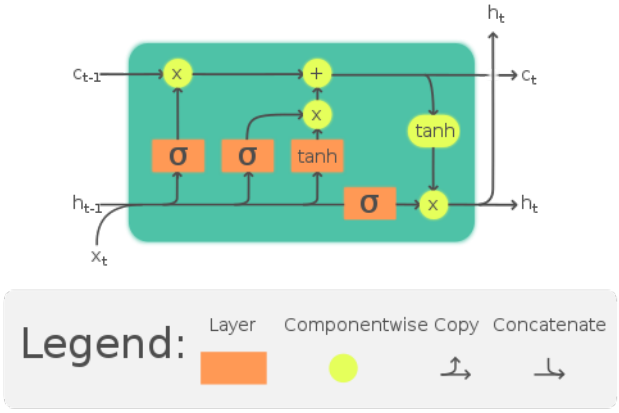

Fig. 11. The Long Short-Term Memory (LSTM) cell

is a third variable - the forget gate (variable "h" in Figure 11), which controls if any past information should be retained or kicked out of the network data flow.

We have decided to fully take advantage of the LSTM architecture and the time-series format of the Aruba dataset to train a robust and powerful model.

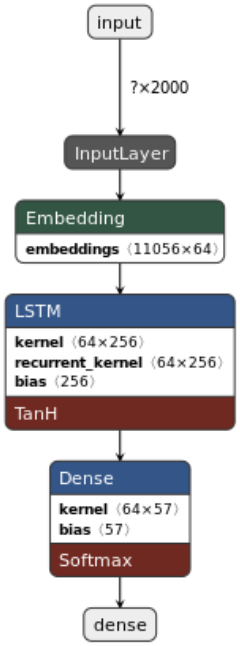

Fig. 12. Model Architecture - LSTM

Figure 12 shows our model architecture, from the input layer to the output layer. The "Embedding" Layer prepares the input sequences to be fed to the LSTM layer. The LSTM layer is the main computing layer, being made of 64 cells, a "Sigmoid" function between the recurrent steps and a "Hyperbolic Tangent(TanH)" activation function. The output layer is implemented with a plain "Dense layer" with *size = the number of activities* (57). In the end, the "Softmax" method

is applied on the dense layer to get the sparse probabilities of each activity.

Similar to the Random Forest model, during the training of the LSTM we have used Stratified K-fold cross-validation with K = 2 splits. This way, we could keep track of the training accuracy. The model was trained during 100 epochs (it uses the whole dataset 100 times while training) in mini bathes of 64 (it runs 64 instances before updating the parameters and averages the results) .

The training time for the configuration explained above was 8 hours on using the machine provided by the Google Colaboratory environment. The reported accuracy on the test set is 86,6%.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3}$$

TABLE IV
CLASSIFICATION METRICS - LSTM

Activity	Precision	Recall	F1-score	Support
Relax	0.97	0.99	0.98	2918
...	...	...	...	...
Leave_Home	1.00	0.33	0.50	3
...	...	...	...	...
Transition_Eating_Relax	0.43	0.17	0.24	105

Finally, the other metrics used to analyse the relevance of the data are the precision, recall, F1-score and support. Precision (1) is the fraction of relevant instances among the retrieved instances, while recall (2) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. F1-score is a metric that combines the precision's and recall's metric knowledge. The support simply provides the number of labels. As expected, the model performs better on activities that have bigger support (has more appearances) in the training set, by predicting more accurate results. In Table IV, the activity "Relax" has support of 2918 and both precision and recall are close to 1.0. The model is accurate when it comes to predicting this label.

On the other hand, if we take "Leave_Home" which appears on the training set only 3 times has a precision of 1.00, but the recall value is low. The high precision and low recall show the tendency of the model to not label all the activity of this kind correctly in a future inference, thus the low F1-score.

Figure 13 presents this situation visually. The full-line circle represents the ground truth (dataset) and the dotted circle represents the model results (the predictions). The green color around the circles and at the intersection represents the "Truth Positives". The intersection shows that the model misses a lot of "Relax" examples. "False Negatives" will tend to appear more often during inference. In other words, the model is underfitting, having a narrow perspective when it comes down to detecting this kind of label. Similar to this, Transition_Eating_Relax has both precision and recall under 0.5, but the support is a little bit higher, but not high enough if you compare it to "Relax". The same problem appears as the model will not be able to correctly classify future instances of this type. The model is underfitting for this type of activity.

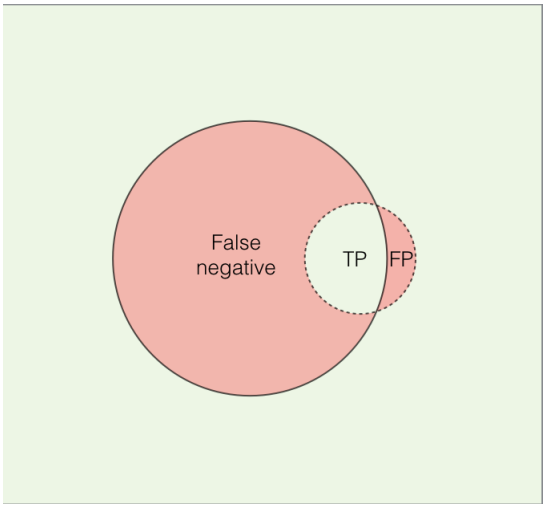

Fig. 13. High Precision / Low Recall

These are evidence that having a high discrepancy in number of instances between labels can affect future inferences, thus the overall accuracy of the model. These can be easily solved by gathering more data for longer periods of time and from multiple houses.

V. FINDINGS

Interesting findings can be observed in the random forest's feature importance.

The most significant feature is "Seconds". This shows the difference of time-frame for different activities. For example, Sleeping is mostly done during the night (0 seconds is midnight).

The next important feature is "Weekday". This shows the existence of considerable differences between activity types based on the day of the week. Some days, the person spends more time in the kitchen while other days, she is sleeping more. Taking into account that the tracked activity is based on the lifestyle of a grandmother, some days might be "visit days" from children and grandchildren so the indoor activities are emphasised (the house is full of people, triggering all the sensors, producing noise in the dataset).

Temperature sensors represent the final important set of features. If the previous 2 features offered us a temporal insight, these final ones offer us a spatial insight. The temperature sensors are situated in specific places around the house. As stated before, the places are: Bedroom, Living, Kitchen, Office and the main hallway that connects all the rooms. Taking into account that one person can be in only one place at the same

time and that a person generates heat, it is possible to guess what room the person is in. Furthermore, if we know what room the person is in, it is possible to guess what activity she is performing. If she is in the Kitchen, she is most probably preparing a meal, if she is in the bedroom, she is most probably sleeping, and so on. One interesting sensor is T004, which is in the hallway. This sensor shows when the person is changing the room between activities. This sensor seems to show an increase in value during activities that are of type *"Transition"* and *"Relax*.

However, the temperature sensors activate during the "Sleeping" activity, which can happen due to the drop in temperatures caused by the lower outside temperatures during the night.

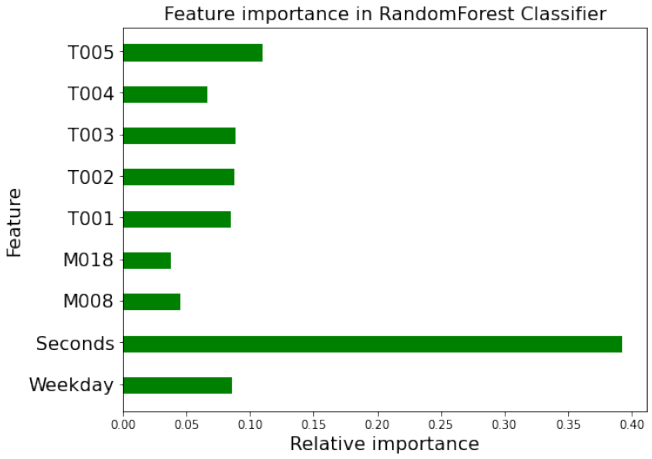

Fig. 14. Feature Importance

Figure 14 shows the feature importance of the second trained random forest model (the one trained only on the extracted important features from the first model). The distribution is respected, showing the same proportions, as in the previous model.

The training time for the Random Forest is around 3 minutes and the training time for the LSTM NN is around 480 minutes. The Random Forest is less computationally expensive than the LSTM, the training for the Random Forest model being 160x faster. However, the time for training the LSTM Neural Network can be improved by enabling the GPU.

Furthermore, there is one issue regarding future inferences using the LSTM model. The tracking system will gather the data unlabeled but LSTM needs the data segmented. So, in order for the LSTM to be used in production, a segmentation system will need to be put in place. Such a system could be a new deep learning model, that will segment the data based on the moment of the day and the past activity. For example, the tracking system gathers information for a whole day. Then, the information is used as input in the segmentation system which outputs the same information, but segmented. Then the segmented data will make use of the LSTM and inference the activity type.

The accuracies on the training set for both models are around 98%. The big difference appears on the test set, the random forest has 98% accuracy while the LSTM has 86%, and thus random forest performs much better (see Figure 15). In practice, neural networks will require far more data, more powerful machines, and more training time to be effective. This can be one of the reasons why there is a difference of more than 10% between the accuracy obtained on the test set on the two models.

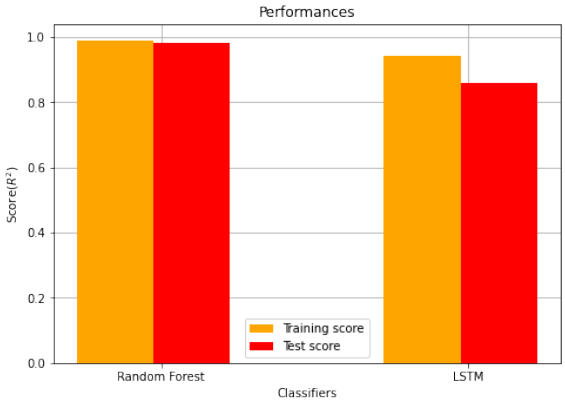

Fig. 15. Feature Importance

VI. Discussion and Conclusion

To deliver effective services to its residents, the IoT ecosystem necessitates sophisticated activity learning technology. In this paper, two models were designed for offering an accurate and reliable activity detection model. The models are built based on the data format found in the CASAS Aruba dataset. The accuracies obtained on the test sets offer the confidence on extending to further production applications.

While researching, some ideas arise as possible future work. Some extensions are:

- Lifestyle Tracker
- Health Application
- Mental State Tracker and Alert

In terms of privacy concerns, the ultimate goal is to have a fully-automated activity recognition system. The system will be interacting with the sensors by using a secured protocol, with an encrypted connection. This way, the data emitted by the sensors will be tracked on the activity log that will be used as input for the detection models. The activity log would be continuously encrypted as new data arrives. The models work as black boxes, so there will be no security issues. The activity predictions would be encrypted as well.

Moreover, having an encrypted connection between the sensors and the modelling pipeline will ensure privacy for the user, by not disclosing any data that can put the person in danger. (e.g. any pattern found in the sensors can lead to private information disclosed - you can detect when the person is not home).

A lifestyle tracker could be really helpful for casual users. It could help them track their indoor habits and manage

their life from a different perspective. The end product could take the form of a mobile application that will show what activities have been done, for how long, at what time. Using the temperature sensors, it might be possible to track the effort (temperatures above average could signal more effort put into performing a specific task). With this information, the users can make significant changes in their life (e.g. sleepless, relax more).

On a medical level, a health application could really help the average end-user to not only track their activities, but to see the medical implication of their habits. The outputs of the model would be correlated to a medical model that will take real time activity log as input and would output a medical report. For example, if the person is spending a lot of time in the office working, there is a high predisposition to burn-out and becoming an "workaholic".

Mental state is of the most important thing in a human's life. One in four people in the world will be affected by mental or neurological disorders at some point in their lives[16]. The most common mental disorders are: Anxiety Disorders (e.g. social, phobias, panic) and Mood Disorders (e.g. depression, bipolarity). An end-product would use the activity recogniser to track a patient's lifestyle. The main goal would be to automatically anticipate escalation of the disorder and keep the patient safe by alerting the responsible authorities. Furthermore, on a more direct approach, with less intimacy, a psychologist could directly track the habits and activity of its patients.

Furthermore, the model can be synced and connected to multiple devices (e.g smartwatch, smartphone, smart fridge) and other types of sensors (e.g. sound sensors, video cameras). This way, the accuracy of the product will be close to state of the art.

References

[1] Carsten Röcker, Martina Ziefle, and Andreas Holzinger. "Social inclusion in ambient assisted living environments: Home automation and convenience services for elderly user". In: *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. Citeseer. 2011, p. 1.

[2] Washington State University CASAS: Center for Advanced Studies in Adaptive Systems. *About CASAS: Center for Advanced Studies in Adaptive Systems*. 2008. URL: http://casas.wsu.edu/about.

[3] Diane Cook Gina Sprint, Roschelle Fritz, and Maureen Schmitter-Edgecombe. "Using Smart Homes to Detect and Analyze Health Events". In: *IEEE Reviews on Biomedical Engineering* 12 (2018), pp. 319–332. ISSN: 99164-2752.

[4] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[5] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[6] Center for Advanced Studies in Adaptive Systems. *CASAS Aruba Dataset description*. 2010.

[7] Daniele Liciotti et al. "A Sequential Deep Learning Application for Recognising Human Activities in Smart Homes". In: *Neurocomputing* (2019). ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.10.104. URL: http://www.sciencedirect.com/science/article/pii/S0925231219304862.

[8] Outside Two Standard Deviations Blog. "An overview of correlation measures between categorical and continuous variables". In: *Medium.com* (2018).

[9] The SciPy community. *Statistical functions (scipy.stats)*. 2022. URL: https://docs.scipy.org/doc/scipy/reference/stats.html.

[10] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[11] François Chollet et al. *Keras*. https://keras.io. 2015.

[12] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[13] Bhumika Dutta. "A Classification and Regression Tree (CART) Algorithm". In: - (2021). URL: https://www.analyticssteps.com/blogs/classification-and-regression-tree-cart-algorithm.

[14] scikit-learn. *Plotting Learning Curves*. 2022.

[15] Wikipedia. *Long short-term memory — Wikipedia, The Free Encyclopedia*. [Online; accessed 19-April-2022]. 2022. URL: http://en.wikipedia.org/w/index.php?title=Long%5C%20short-term%5C%20memory&oldid=1082693417.

[16] Gregory Härtl. "The World Health Report 2001: Mental Disorders affect one in four people". In: (2001).