Human Activity Recognition using Ambient Sensors

Kunal Chhatrapati
Department of Computer Science
California State University, Fullerton
Fullerton, California
kunalchhatrapati@csu.fullerton.edu

Kanika Sood
Department of Computer Science
California State University, Fullerton
Fullerton, California
kasood@fullerton.edu

Purva Surve
Department of Computer Science
California State University, Fullerton
Fullerton, California
purvass@csu.fullerton.edu

***Abstract*— Ambient sensors positioned throughout a residential residence can assist in detecting human activity without the need for wearables or direct communication with the people being watched. Elderly or disabled folks who have trouble utilizing conventional monitoring equipment will mostly benefit from this. The sensors are simple to deploy in various interior settings, making it possible to monitor huge populations or many sites simultaneously. This methodology applies to smart homes, and we suggest gathering ambient data from participants' houses while they go about their regular lives. Based on the activity in the residents' smart homes, the sensors can identify their activities. To group the movements detected by the sensors, we will utilize a variety of classification algorithms based on the UCI dataset.**

Keywords*— *Human Activity Recognition, Machine learning, wearable sensors, Classification, Data Cleaning, IOT Sensors.

I. Introduction

Human Activity Recognition uses machine learning algorithms that recognize and categorize human movements and behaviours based on sensor data. It involves using ML algorithms to analyse data from different sensors to identify patterns and classify them into activities such as sitting, standing, and sleeping [2]. HAR has various applications, including healthcare-assisted living, which can be used to monitor the activities of elderly people or patients with chronic conditions, ensuring they are safe and receiving proper care. Surveillance is another field where HAR is commonly used as it can help identify suspicious behaviour in public spaces and assist law enforcement agencies in maintaining public safety. Computational behavioural science is a growing field that employs HAR to study human behaviour in different contexts, such as studying human interaction with technology and identifying patterns in online user behaviour.

Wearable sensors have been utilized in the healthcare industry to continuously monitor patients' vital indicators, such as heart rate, breathing rate, and blood pressure [3]. This enables medical practitioners to deliver prompt medical inventions and identify early indicators of potential health disorders. Additionally, wearable sensors can be used to monitor eating habits, sleep schedules, and physical activity, which can assist in managing chronic diseases like diabetes and obesity.

Another area where HAR can be applied is assisted living for the elderly. With the aging population, there is a growing demand for technologies that allow older persons to live independently in their homes. Sensor-equipped smart homes can monitor daily activities such as cooking, cleaning, and bathing and alert family members or caretakers to unusual activities or emergencies. For example, sensors can be placed on the refrigerator to monitor when it is opened, indicating whether a person is eating regularly. Motion sensors can be placed throughout the house to monitor movement and detect if a person has fallen or is experiencing mobility issues. In addition, smart homes can be equipped with voice-activated assistants that can help with tasks such as turning on lights or reminding someone to take their medication. These technologies can give older persons a sense of independence while providing peace of mind to their family members and caretakers. Overall, HAR in assisted living can potentially improve the quality of life for elderly people.

HAR can be applied in surveillance, particularly in public locations such as train stations, malls, and airports. Using sensors to monitor human activities in these areas, HAR can identify unusual or suspicious behaviour that may indicate a security threat or criminal activity. HAR can detect if a person is behaving suspiciously. It can also track the movement of people and objects in real-time, allowing security personnel to respond quickly to any potential threats. In addition, HAR can be used to identify patterns of behaviour that may indicate criminal activity[3]. HAR can detect when items are being taken without being paid for or when someone attempts to steal another person's wallet.

Computational behavioural science is an interdisciplinary field that seeks to understand human behaviour through computational methods, including machine learning and statical modelling. One area where HAR can be applied in computational behavioural science is studying human movement and activity. By analysing sensor data, HAR can provide detailed information on how people move and interact with their environment. This information can then be used to identify patterns in behaviour and understand how people respond to different stimuli. HAR can be used to study the behaviour of individuals in specific contexts, such as during social interactions or while engaging in a particular task. By analysing the movements and behaviours of individuals in these situations, researchers can gain insights into human behaviour that can be used to develop more effective products and services [4].

Wearable sensors have various commercial uses, including panic buttons that people can wear to call for help. These sensors are designed to be portable, easy to use, and provide users with a sense of security and safety [5]. One common application of wearable sensors in commercial settings is in the healthcare industry. Wearable sensors can monitor vital signs, such as heart rate and blood pressure, and alert healthcare professionals if a patient's condition changes. This can be especially useful for patients with chronic conditions, such as diabetes or heart disease, who require ongoing monitoring and care. In addition, wearable

979-8-3503-3559-0/23/$31.00 ©2023 IEEE

sensors can be used in workplace safety programs to monitor employees and ensure they follow safety protocols. Wearable sensors and HAR can improve our quality of life in various contexts, such as healthcare, assisted living, security, and behavioural study[6]. To increase the dependability, accuracy, and accessibility of these technologies, more research and development is needed[5].

Human Activity Recognition (HAR) is a process that involves continuously collecting data related to physical activities. Most research in this field utilizes datasets obtained from everyday routines by deploying ambient sensors, such as passive infrared (PIR) motion sensors, door/temperature sensors, and light switch sensors in homes[1]. These types of sensors have proven to be valuable in capturing essential movements and understanding human physical activities. The dataset used by our algorithms contains the ambient data collected continuously during the volunteers' normal routines using ambient PIR motion sensors, temperature sensors and light switch sensors. The PIR motion sensors are used to measure the infrared radiating light from the object placed in the space. The sensors communicate using the ZigBee Pro protocol, forming a mesh network with all battery powered sensors as leaf nodes and always-on devices forming the branches that connect back to the USB gateway.

In this research paper, we train our models using sensor data and apply various machine learning techniques, such as Support Vector Machines (SVM), k-Nearest Neighbours (KNN), Naïve Bayes, Decision Trees and Random Forest. Considering the complexity, time, and performance metrics analysis of these ML models, we can conclude that the Random Forest ML technique is well-suited to address the problem, followed by KNN and Decision Trees. The paper is structured as follows: Section II presents the background work, where we discuss the overview of the HAR techniques and its usage in the industry. In the next two sections, we discuss about the dataset, data preprocessing techniques utilized to refine and clean the data and the various machines learning techniques used to provide the solution of the problem statement. Section V, VI and VII collectively discuss about the solution methodology, the result and the conclusions we draw of our analysis. The final section, Section VIII discusses the future scope. We have also attached the references at the end of the research paper.

II. Background

Due to its capacity to gather information on human activities unobtrusively, ambient sensors have attracted considerable attention in Human Activity Recognition (HAR). A form of sensor technology known as an ambient sensor can detect and record a wide range of physical and environmental factors, including temperature, humidity, light, and sound in the immediate area. The goal of HAR using ambient sensors is to recognize and categorize particular movements or behaviors of a person based on sensor data gathered from the immediate surroundings[2]. Since ambient sensors do not need to be physically attached to the human body, they are more practical and less obtrusive than other sensor technologies, such as wearable sensors.

There are numerous potential use cases for ambient sensor applications in HAR. For instance, ambient sensors in smart homes may recognize human activities like cooking, cleaning, and bathing and modify the environment of the home accordingly to enhance energy effectiveness, comfort, and safety. Ambient sensors can be used in healthcare to track patients' movements and activities and spot any anomalies that could point to a possible health problem[3]. In surveillance, ambient sensors can detect anomalous activity or conduct in public locations like train stations, malls, and airports.

However, using ambient sensors for HAR is not without its difficulties. It can be difficult to extract useful information from sensor data that is noisy and unclear, which is acquired from the environment. Furthermore, various environmental conditions, such as adjustments in illumination and sound levels, might impact the HAR system's accuracy. Researchers are investigating numerous methods, including signal processing, feature extraction, and machine learning algorithms, to address these issues and increase the accuracy of HAR utilizing ambient sensors[6]. Efforts are also being undertaken to create low-cost, energy-efficient ambient sensors that can function dependably in various environmental circumstances. Using ambient sensors in HAR can transform several industries [7], including healthcare, smart homes, and surveillance. More study is required to address the issues with this technology and improve its accuracy and dependability.

III. Dataset and Data Preprocessing

Data preparation is transforming raw data into a format that can be interpreted. Since we cannot work with raw data, this is important for data mining. Before implementing data mining or machine learning techniques, the data quality should be assessed. The initial stage in the process is to predict the class of the supplied data points. The objective, label, and categories are other names for the classes. Since it affects whether information can be used for its intended purpose in each setting, data quality is also important. So how do you evaluate a dataset's worth? There are various aspects to consider when it comes to data quality.

We use the data created by Brian L. Thomas, Aaron S. Crandell, and Diane J. Cook[1] available on UCI website. The dataset consists of 13956534 instances with 37 attributes with target feature labelled as activity with 41 possible values, making it a multi-class classification problem. The data pre-processing steps that we used are follows:

A. Visualization

Firstly, before starting any data pre-processing step, we first visualize the data. To better understand and comprehend our dataset we use histogram which provides the graphical distribution of numerical data across all the feature set. The below Fig. 3 shows the representation.

B. Data Preprocessing

1) Data collection: To gather sensor data from ambient sensors, one must have the sensors in place and collect data over time. The length of time will depend on the project's specific objectives and the activities being monitored.

2) Data cleaning: Once the data is collected, the next step is to clean it by removing any outliers or noise. This can be done using interpolation, smoothing, or filtering techniques. The aim is to ensure the data is as accurate and reliable as possible.

lastSensorEventHours	24	sensorCount-Hall	1
lastSensorEventSeconds	85689	sensorCount-Ignore	7973
lastSensorDayOfWeek	7	sensorCount-Kitchen	8054
windowDuration	15372	sensorCount-LivingRoom	8148
timeSinceLastSensorEvent	2826	sensorCount-Office	1
prevDominantSensor1	9	sensorCount-OutsideDoor	5078
prevDominantSensor2	9	sensorCount-WorkArea	7487
lastSensorID	9	sensorElTime-Bathroom	1917368
lastSensorLocation	9	sensorElTime-Bedroom	1822448
lastMotionLocation	9	sensorElTime-Chair	835435
complexity	2289	sensorElTime-DiningRoom	1401111
activityChange	280836	sensorElTime-Hall	1
areaTransitions	27	sensorElTime-Ignore	1164046
numDistinctSensors	1	sensorElTime-Kitchen	1681464
sensorCount-Bathroom	7821	sensorElTime-LivingRoom	1835571
sensorCount-Bedroom	7954	sensorElTime-Office	1
sensorCount-Chair	2908		
sensorCount-DiningRoom	7418		

Fig. 1. Feature variables

We then find the number of distinct values in the datafile for each feature variable. Fig. 1 is the list of all feature variables with their unique instances. To demonstrate it we use python3 environment along with python machine learning libraries such scikit, pandas. We use nunique() function to find the distinct values. Now next we check the data file for any missing or null values using isna() function which returns the list of number of missing or NaN values present in each column of the dataset.

The next step of data cleaning is to fill the missing values will filler data in our case since all the data was of numeric type we use '0' to fill in the missing values. We replace our null values with '0' using fillna() function available with pandas libraries. We then use StandardScalar to normalize the dataset with each feature having a mean of 0 and variance of 1. Fig. 2 represents the output of dataset after Scalar transformation. While executing StandardScalar transformation, the output can have NaN or infinite numbers. We solve this issue by using numpy.nan_to_num() function which converts NaN and infinte values with finite numeric values.

```
   lastSensorEventHours  lastSensorEventSeconds  lastSensorDayOfWeek  \
0             -0.617668               -0.678311             0.530148
1             -0.617668               -0.678182             0.530148
2             -0.617668               -0.678118             0.530148

   windowDuration  timeSinceLastSensorEvent  prevDominantSensor1  \
0        0.143004                 -0.249845             0.882335
1        0.163719                  0.385888             0.882335
2        0.163719                 -0.249845             0.882335

   prevDominantSensor2  lastSensorID  lastSensorLocation  lastMotionLocation
0             0.797014      0.356224            0.356224           -0.008142
1             0.797014     -1.071979           -1.071979           -0.008142
2             0.797014     -1.071979           -1.071979           -0.008142

   ...  sensorElTime-Chair  sensorElTime-DiningRoom  sensorElTime-Hall  \
0  ...            0.929737                -0.815276                0.0
1  ...            0.929737                -0.815247                0.0
2  ...            0.929737                -0.815224                0.0

   sensorElTime-Ignore  sensorElTime-Kitchen  sensorElTime-LivingRoom  \
0            -0.417670             -0.181235                -0.809102
1            -0.577072             -0.179447                -0.806904
2            -0.577072             -0.178084                -0.805230
...
1  0.667429
2  0.667429
```

Fig. 2. Output after Standard Scalar transformation

3) Data labeling: The data collected will likely contain information about various activities. This data needs to be labeled with the corresponding activity for analysis. This can be done by manually observing the data and assigning activity labels or by using machine learning algorithms to automate the labelling process. We convert our activity column into categorical data, find unique categorical values present for activity target feature and assign unique codes using LabelEncoder(). We loop through the categorical column of activity and use fit_transform method of the encoder object to encode the values of the activity feature column. We then label the activity feature to be used for further analysis.

4) Feature extraction: Once the data is labelled, extracting relevant features from the sensor data is next. This could include statistical measures such as mean, median, standard deviation, maximum, and minimum.

5) Feature scaling: After feature extraction, the data is often scaled to ensure that all features are on a similar scale. This can be done using techniques such as standardization, where the data is transformed to have a mean of zero and a standard deviation of one.

6) Feature encoding: To use categorical data for analysis, it must be encoded into numerical values. This can be done using techniques such as label encoding. Using label encoding we assigned a unique numerical value to each of the categories.

C. Correlation analysis: Once the data has been pre-processed and cleaned, the next step that we perform was to analyse the relation between input features and target feature i.e. activity name. And to analyse the relationship we used the correlation matrix shown in Fig. 4. We select the features that had an impact on the output of target features and then created and trained model using those features. The features selected from dataset are lastSensorEventSeconds,LastSensorLocation, activityChange and activity. The activity data feature is categorical while all other are of numeric datatype. We also visualise the relationship between the relevant features using pair plots as shown in Fig 5.

D. Handling Outliers: Once the data is pre-processed and cleaned, we used the Inter Quartile Range (IQR) method to identify the outliers in the dataset. Outliers are the extreme values that vary from the general observations of the dataset and it is important to identify and remove the outliers in order to reduce the errors. IQR is a statistical method that measures the spread of the dataset. It is basically calculated using the difference of third quartile (Q3) and first quartile (Q1). The third quartile is about the 75 percentile of the dataset value and first quartile is the 25 percentiles of the dataset. The IQR is calculated using below "equation (1)".

$$IQR = Q3 - Q1 \tag{1}$$

After calculating the IQR, we remove the outliers from the dataset by removing the datapoints less that are less than the difference of first quartile Q1 and greater than the difference of the value of the third quartile Q3 [13].

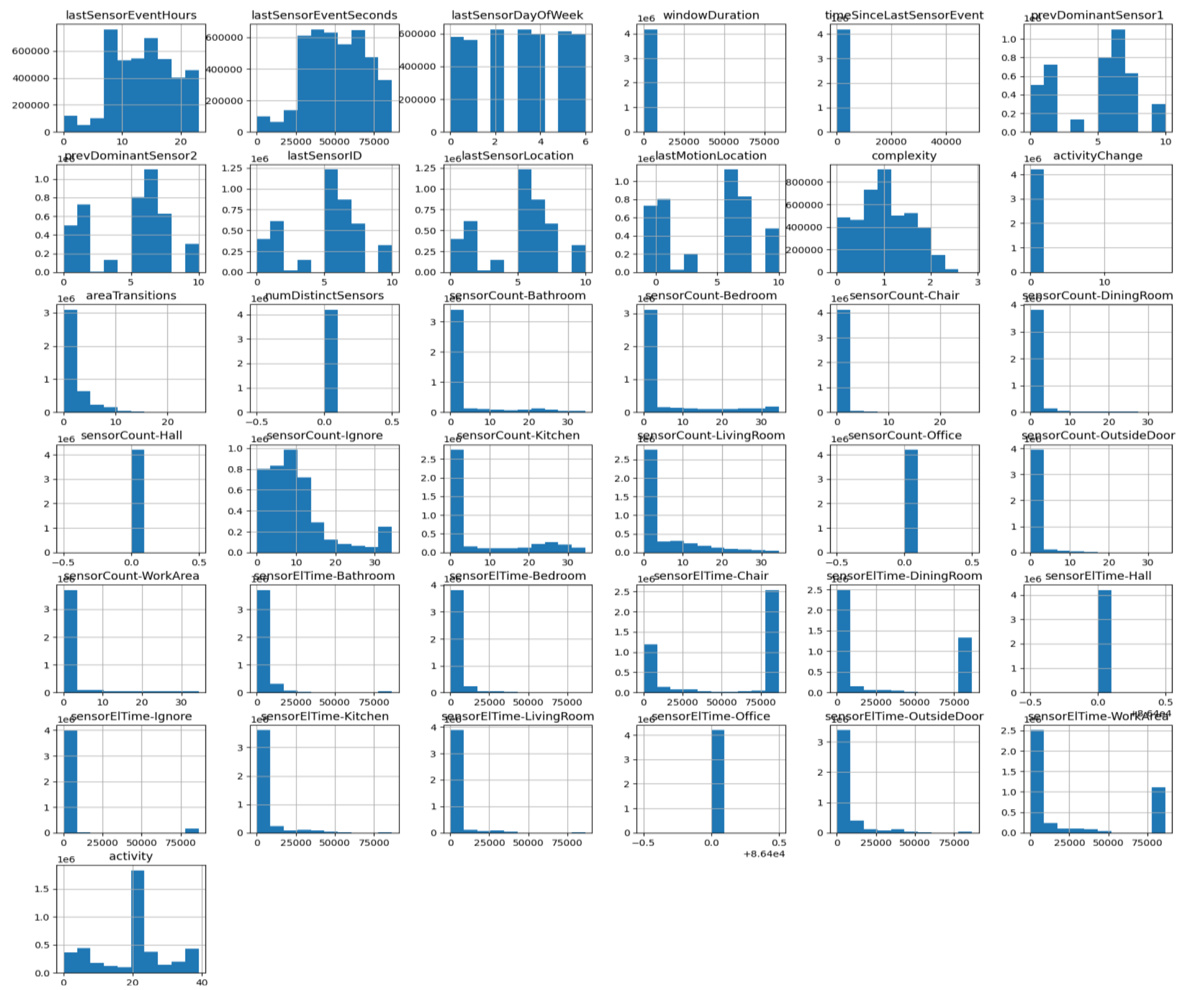


Fig. 3. Histogram for dataset

IV. ML Classification Techniques

In this section, we present the techniques used for building the model. We build classification models using five different ML techniques and evaluate the results. We compare the model based on the time, complexity and performance metrics to understand which models' best suits to solve our problem. We used below classifiers to train our model:

A. Classifiers

1) K-Nearest Neighbours – This algorithm is easy to implement and faster as it does not have any training period. It learns from the training dataset while making the predictions. The new data can be added at any point without affecting the model, the model changes and evolves with new additions. In our model, we had used five different values for starting from K=1 to K=5, however, we got good accuracy for K=5 and K=1. We decided to select the K=5 value to further evaluate our model as K=5 value provides more generalization capability to our model and would be able to predict new data point more accurately.

2) Naïve Bayes – This algorithm works best for binary or multi-class classification problem with categorical data. For our model, we had used gaussian bayes model which assumes the input features follow the normal distribution. For our model, we used Gaussian Naïve Bayes [8] which is a variation of Naive Bayes that assumes the data is normally distributed.

3) Support Vector Machine – This supervised learning technique can be used for both classification and regression and works well for high dimensional dataset. It has good generalization performance as it can provide results for new data.

4) Random Forest – It is an ensemble learning technique which uses multiple decision trees to make the predictions. The prediction is made by aggregating the results of individual classifier [9]. Since this method uses a bagging, it increases the accuracy of the model. It is mainly used for classification problems and can handle continuous data.

5) Decision Tree – This algorithm is easy, fast and operates well on large datasets [9]. It requires less computational power and are easy to visualize. The predictions on the target feature are made on the basis of the decision rules inferred from the input variables.

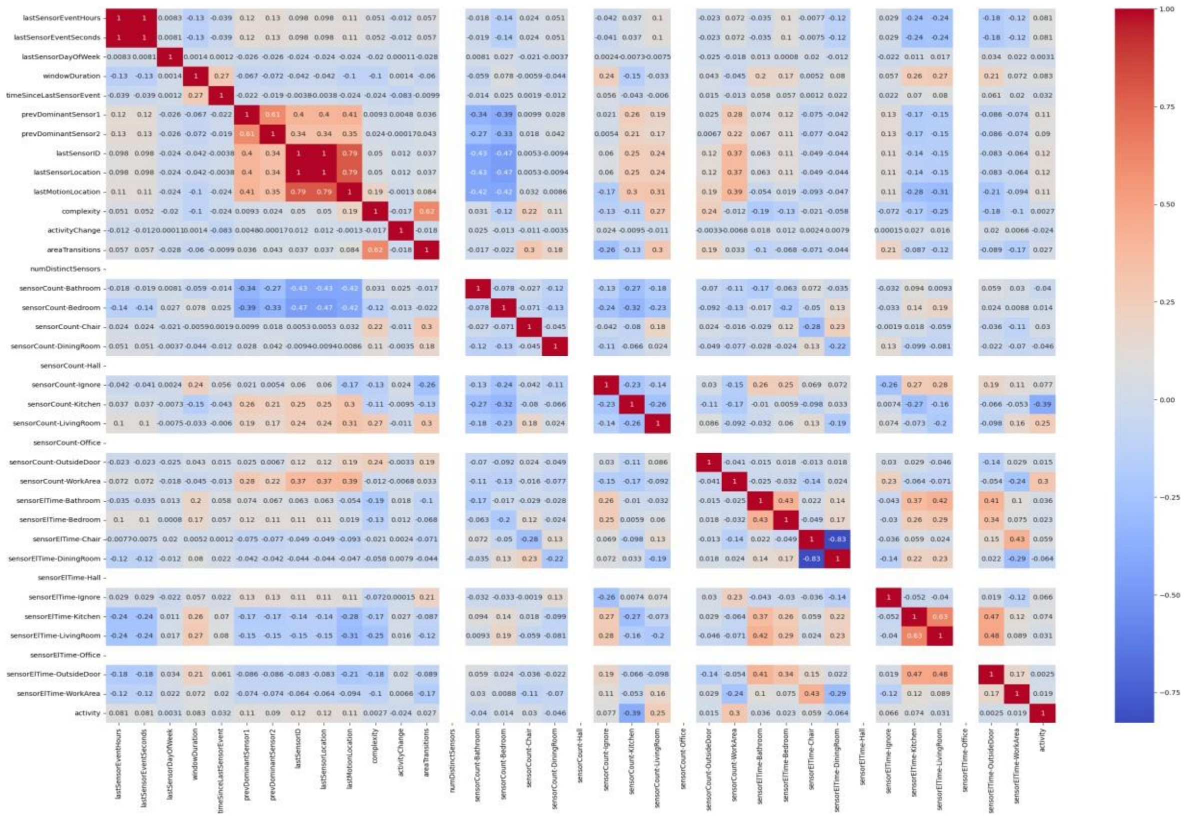


Fig. 4. Correlation matrix

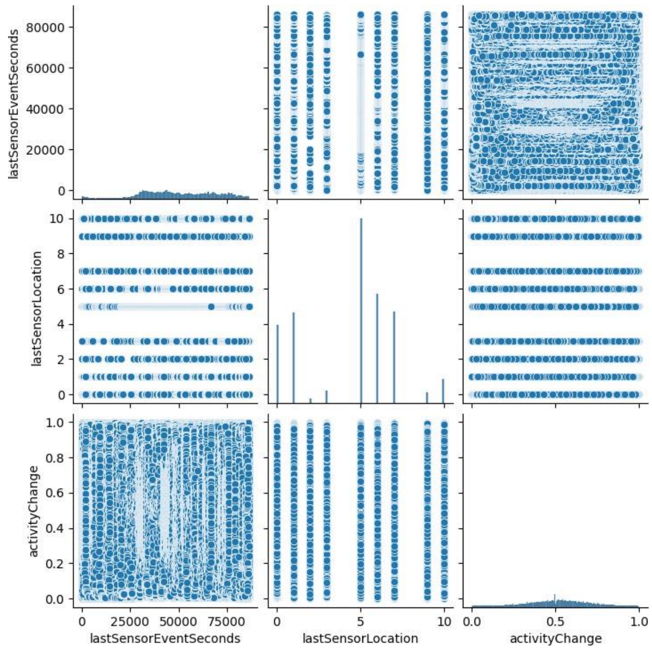


Fig. 5. Pairplot of reduced feature set

V. Methodology

In this section, we will present the techniques we use for building our prediction model.

A. Firstly, we visualize our dataset to better understand and interpret it. We train our model using KNN technique to make the predictions. With raw data of 13956534 instances and 37 feature attributes [1] we get an accuracy of around 22%. Visualizing the dataset help us to understand the data distribution of various attributes and its weightage, which provided insights for data processing. Along with visualization we use the correlation matrix to find highly correlated features in the dataset[10] We then remove the feature attributes which were highly correlated to reduce the dimension of the data and selected the important features that had a direct impact on the prediction output the model.

B. Once our data is pre-processed, we use train and test split approach to build our models. In train and test approach, the entire dataset is split into two subsets, namely training set and testing set as shown in Fig. 6. The training dataset includes all the attribute feature and target feature. The model is trained using training dataset and then later tested using test set. The input element is provided to the test set to make the predictions. The predictions are then compared to the expected values in dataset to get the model performance. For our model, we divide our data into 80% training data and 20% test data. We use same approach to build, train and test model using 5 classifiers namely KNN, SVM, Naïve Bayes, Random Forest and Decision Tree.

C. Once the model is ready, we save our individual models and analyse them using performance metrics.

The performance metrics used to evaluate and visualize the results of the models are as below:

1) *Accuracy:* It defines the percentage of correct predictions made by the classifier to the total number of the predictions [11] .The accuracy value of the model should be close to 100%.

2) *Precision:* It measures how many correct predictions made by the classifiers are actually correct. The precision score >0.5 suggests that model is fair.

3) *Recall:* It is also termed as the sensitivity of the model and explains how many actual correct predictions were made by the classifier [11].

4) *F1 score:* This score is a combination of Precision and recall score. The score will be in between 0 to 1[12].

5) *Confusion matrix [12]* – It is a matrix that compares the predictions of each class such as True positive, False positive, True negative and False negative for each feature set. The confusion matrix for the Random Forest model which provided good results and Naïve Bayes model with fair results as compared to other models is in Fig. 7.

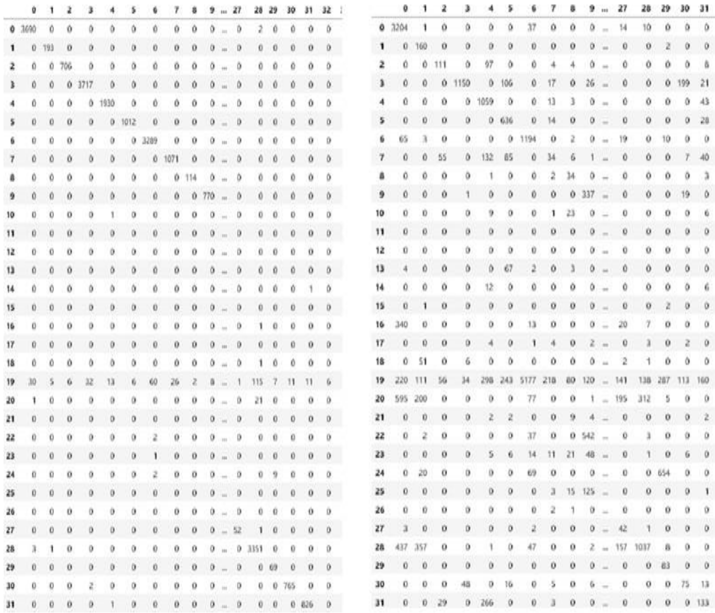
Fig. 6. Train test split approach

6) *AUC ROC curve* – It is used to visualize the results of the model. It is a probability curve that plots the area under the ROC curve, which distinguishes between the classes [11]. The value of AUC should be close to 1.0 which suggests that model is perfectly able to distinguish true and false cases.

We also use precision-recall curve to visualize the performance of the model. We have considered time as a performance metric to evaluate the performance of the model. According to which the model using SVM classifier took a lot of time to train, test and provide result for test dataset.

VI. Results

We build and train the models on raw dataset with all feature variables and then with reduced feature set and observed the results.



(a) (b)

Fig. 7. Confusion matrix, (a) Confusion matrix of Random Forest, (b) Confusion matrix of Naïve Bayes

A. Full dataset

The data used for building and training model is pre-processed for outliers and null values. The model was fed the full feature set of 36 variables. Firstly, all the models took an average time of 5-10 mins to produce results. The accuracy for models was less than 95% and for Naïve Bayes, the accuracy was very less close to 22%.

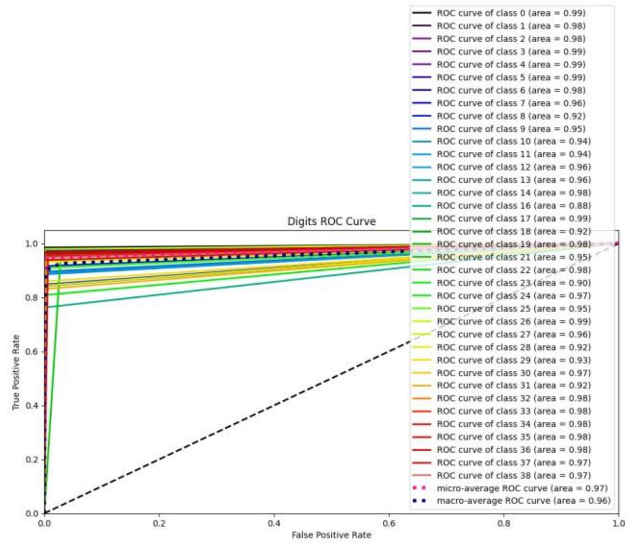


Fig. 8. ROC curve for Random Forest

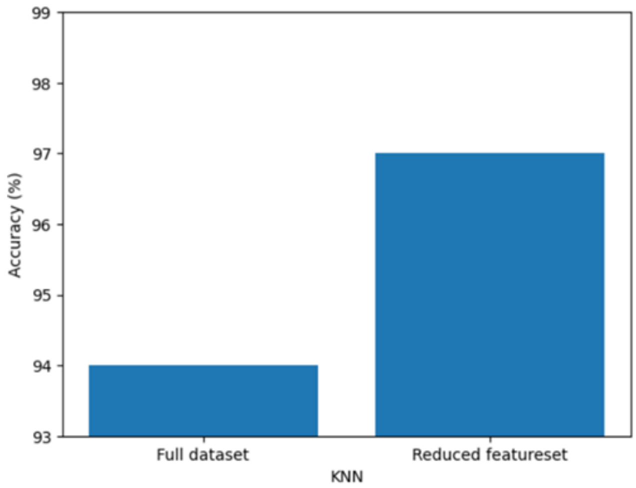
B. Reduced feature dataset

By finding the correlation between the feature variable, we identify the relevant feature that had an impact on target activity feature. The analysis of models using reduced feature set is as shown in Table I.

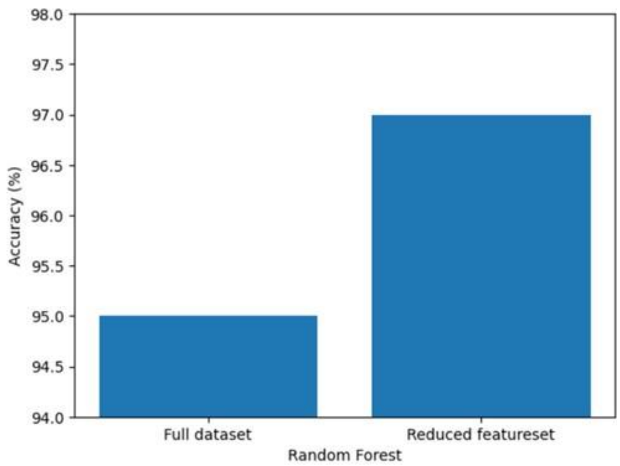
TABLE I. ANALYSIS OF ML MODELS

Classifiers	Performance metrics			
	Accuracy	***Precision***	***Recall***	***F1 score***
KNN	0.960	0.961	0.961	0.960
Naïve Bayes	0.349	0.609	0.349	0.345
Random Forest	0.975	0.975	0.975	0.975
Decision Tree	0.946	0.947	0.947	0.946

Based on the analysis of above performance metrics, we achieved good results for Random Forest, Decision Tree and KNN model. The area under ROC is greater than 90% for all the models, Fig. 8 represnts the ROC for Random Forest. The Gaussian Naïve Bayes model has results less than 50% for all the performance metrics. The SVM model took a lot of time to produce results.



(a)



(b)

Fig. 9. Comparison of Random Forest and KNN with full feature and reduced feature set, (a) Random Forest, (b) KNN

VII. CONCLUSION

In this project, we build models that predicts the activity of the user based on the user's data captured via motion and thermal sensors. The model learns and trains on the sensor data and provide the output for target feature using various ML techniques such as KNN, Naïve Bayes (Gaussian), Random Forest, SVM and Decision tree. Since it is a multiple classification problem, SVM model took a lot of time to provide output as compared to other ML models. Taking into consideration the complexity, time and the analysis of ML models based on performance metrics we can conclude Random Forest ML technique is well suited to solve the problem followed by KNN and Decision Tree. Although our data is imbalanced and contains outliers, there is no apparent consensus on which model would be the best. We have also provided some visualization techniques to analyse and examine the efficiency of all the models for further improvement.

VIII. FUTURE WORK

In the future we will try to refine the existing model and get better accuracy. Also, we would like to collect more relevant data instances and train algorithm to get accurate output for the model. The existing data consisted of null values and outliers which affected the performance of the model. We would also add more values to the target feature (i.e., activity name) to more precisely identify the activity of the user in a particular space and not just generalize it to a common dining or kitchen activity. We can also work on the user interface of the model where the UI will highlight the layout of the house with which the output activity is associated. The house layout user interface will help to better understand and visualize the output of the model. Along with it, we can also add the behavior analysis of the user based on user's day to day activities.

REFERENCES

[1] "UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+from+Continuous+Ambient+Sensor+Data# (accessed 2023)

[2] Š. Aida and J. Kevrić, "Human Activity Recognition Using Ambient Sensor Data," IFAC-PapersOnLine, vol. 55, no. 4, pp. 97–102, 2022, doi: https://doi.org/10.1016/j.ifacol.2022.06.016.

[3] A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning, doi: https://doi.org/10.3390/s21186037.

[4] Survey Paper on Deep Learning for Human Activity Recognition https://www.researchgate.net/publication/35278510

[5] M. Z. Uddin and A. Soylu, "Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning," Scientific Reports, vol. 11, no. 1, Aug. 2021, doi: https://doi.org/10.1038/s41598-021-95947-y.

[6] Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques, Security and Communication Networks, vol. 2020, Article ID 2132138, 12 pages, 2020 https://doi.org/10.1155/2020/2132138

[7] A survey on wearable sensor modality-centered human activity recognition in healthcare https://irep.ntu.ac.uk/id/eprint/40527/1/1356942a853/Cang.pdf

[8] V. Păpăluță, "How to impliment a Gaussian Naive Bayes Classifier in python from scratch?," Medium, Feb. 13, 2020. https://towardsdatascience.com/how-to-impliment-a-gaussian-naive-bayes-classifier-in-python-from-scratch-11e0b80faf5a

[9] Oracle.com, 2023. https://blogs.oracle.com/ai-and-datascience/post/random-forests-decision-trees-and-ensemble-methods-

explained#:~:text=The%20random%20forest%2C%20first%20described (accessed May 11, 2023)

[10] R. Vidiyala, "Performance Metrics for Classification Machine Learning Problems," Medium, Jul. 26, 2020. https://towardsdatascience.com/performance-metrics-for-classification-machine-learning-problems-97e7e774a007

[11] "Evaluation Metrics For Classification Model | Classification Model Metrics," Analytics Vidhya, Jul. 20, 2021. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/#:~:text=There%20are%20many%20ways%20for

[12] R. Vickery, "8 Metrics to Measure Classification Performance," Medium, Dec. 08, 2021. https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa

[13] Handling Outliers in Machine learning

https://medium.com/analytics-vidhya/how-to-handle-outliers-in-machine-learning-5d8105c708e5