Comparative Study of Classifiers on Human Activity Recognition

Mahbuba Tasmin, Sharif Uddin Ruman, N.M. Shihab Islam, Arif-Ur Rahman Chowdhury Suhan, Taoseef Ishtiak, Sifat Jahan

Abdur Raufus Saleheen,, Md. Shahnawaz Zulminan, Rashedur M. Rahman

Department of Electrical and Computer Engineering, North South University

Plot-15, Block-B, Bashundhara, Dhaka, Bangladesh

mahbuba.tasmin@northsouth.edu, sharif.ruman@northsouth.edu, shihab.islam@northsouth.edu, arif.suhan@northsouth.edu, taoseef.ishtiak@northsouth.edu, sifat.jahan@northsouth.edu, abdurraufus.saleheen@northsouth.edu, sajid.ahmed1@northsouth.edu, shahnawaz.zulminan@northsouth.edu, rashedur.rahman@northsouth.edu

*Abstract*— Human activity Recognition (HAR) systems are the key factors in futuristic smart living concept. Human activities are recorded as stream of sensor events in time-series or sequential dataset. HAR classifies the activities from a dataset through state-of-the-art classifier models built for time-series classification. This paper presents a comparative study of classification methods from the Ambient Sensor Dataset from UCI machine learning repository to recognize human activities. Before the classification, we have executed extensive data preprocessing and feature selection techniques to make our dataset ready for training. The attributes from the original dataset has been reduced with respect to different feature selection techniques and statistical analysis for efficient training. Three classifier models: Decision Tree, Random Forest and Nearest Neighbor show different accuracy scores corresponding to different feature selection techniques and without feature selection as well. The research finding highlights the necessity of data preprocessing and significant feature selection for getting better accuracy score for noisy time-series data of HAR activity.

Keywords—Human Activity Recognition, Time Series Data, Activity Classification, Feature Engineering

# Introduction

Technology’s advancement has blessed mankind with an advanced world; some of them ensuring smart living and making salient contribution in making smart home devices, smartphones, wearables and other forms of applications, which has tremendously influenced human lifestyle and is continuing to shape the futuristic lifestyle as well. These technologies have empowered independent lifestyle of an individual, thus significantly reducing dependency on other people [3]. With these smart technologies, the concept of Ambient Assisted Living (AAL) emerged. Ambient Assisted Living [13] presents a system consisting of smart devices, home appliances, wireless networks primarily for healthcare monitoring and smart home living. This futuristic concept helps to provide an independent and quality life primarily targeted for senior citizens. Another sister concept in this regard is Ambient Intelligence (AML). It presents the ability of a computing system to sense its surrounding and interact with people around. Both of these concepts originate from the advancement of Human Activity Recognition system on the infrastructure of networked sensors (Internet of Things). [14].

Data records from different sensor readings has paved the way to identify human activities separately and is leading to smart home systems consequently. Most HAR systems are based on camera or computer vision or wearable sensors [1].

Change detection in time-series data through calculating change in statistical metrics (e.g. Mean and Covariance) [2] assists in classifying activities. In general, activity recognition is a context-aware system [3]. It provides the understanding of the smart home applications to interpret and take action according to input from user interaction. Nevertheless, a real-time indoor HAR system is often limited by the constraints of indoor environments and makes it difficult to build a robust and scalable system.

Computer vision based HAR systems are useful for large sample of data and pedestrian movement. To eliminate the potential privacy issue related to camera based computer vision system, wearable sensors or devices including smartphones [6] are used as the data record and activity sensing infrastructure. This setup is sometime rendered restrained and potential limitations since the user needs to always equip the sensing device while recording data, which does not support seamless activity record process. In addition, the wearable approach requires transition between different positions of the user need to be perceived since the system depends on the target to determine the location of the wearable device with respect to the performed activity [17].

On the other hand, in the indoor environment, intelligent HAR system applies Ambient Intelligence to take action on the basis of the collected information from the surrounding and interacting residents [4, 5]. During recording, embedded sensors collects data readings of the users performing their activities being unaware of the system. Sensor-data is stored in a database and later analyzed to generate target information such as patterns, predictions and transitions [7]

This research work is motivated to classify five distinct activities (Watching TV, Reading, Taking on Phone, Cooking, and Eating) from the dataset of 12 pre-defined activities including unlabeled activity namely “other activity”. The dataset has been acquired from the UCI Machine Learning Repository dataset “*Human Activity Recognition from Continuous Ambient Sensor Data Dataset*” from Washington State University [11]. The motivation is to precisely classify the activities while reducing the number of parameters and selecting important features from original dataset based on statistical approach which will computationally efficient to make models. This originates from the idea to allow human activity recognition with a minimalist model for saving computation power so that real-life applications upon such model will be lighter. The sensors’ signals is preprocessed in original dataset, among them, features with statistically significance values have been selected for training and finally three different classifier models have been used to recognize activity and a comparative study of performance is reported towards the end.

The major contributions of the proposed research are:

1. Data preprocessing of the selected 5-activities dataset through Principal Component Analysis and Linear Discriminant Analysis
2. Feature Selection based on statistical significance and importance score of dataset attributes.
3. Classifier models comparison and choosing the best classifier model along with the most significant features for activity recognition.

The rest of the paper is organized as follows. Section II presents the related works on human activity recognition. Section III presents data source and Section IV presents methodology where data preprocessing and feature selection approaches are discussed with classifier models Section V reports performance evaluation In the following Section VI concludes and gives direction of future works.

# Related Work

The research field of activity recognition considers the combination of embedded sensors, different environmental setups and algorithms to detect activity points. Probabilistic graph based Markov models, conditional random fields, Bayesian network [21, 12] are some of the state-of-the-art classification models for detecting activity from times-series data.

Distinct activities like Walking, Running, Standing, Sitting, Climbing Stairs and Falling) are classified in [13, 18] using accelerometer placed on the body. Recently smartphones with embedded motion detector and orientation sensors (Accelerometer and Gyroscope) are used as wearable device to recognize gesture and motion patterns [19].

In indoor HAR system, large range of activity are observed through embedded sensors at key location of activities. Environment sensors such as motion detector, light sensor, temperature and pressure sensors etc. are used to record stream of sensor data of activities in [4, 5].

In realistic activity recognition tasks, the recognizing activities are performed with interleaved activities [20, 21], embedded errors [19] and concurrent activities are performed by multiple individuals in the setup [6, 20]. Detecting activities in free movement setup, where the residents perform usual daily routines in a smart home environment [3, 6]. These recorded datasets requires manual labelling to segment and analyze the data.

Dedicated HAR architectures recognizes sequential and concurrent human activities using multiple sensor data at a time. Two key approaches are followed in HAR: “Data-Driven” and “Knowledge-Driven” technique [23]. Naïve Bayes (NB) classifiers, Decision Trees, Hidden Markov Models, Bayesian Networks and Support Vector Machine (SVM) classifier had been used as the Data-driven method in [23]. Existing works performed with data-driven technique utilize supervised approach using manually labeled data for training. The unsupervised approaches achieve low performance in comparison with the supervised approach in indoor home environment. Activities are classified with the prior knowledge of pre-recorded data of surrounding. Data-driven techniques are useful for detecting basic distinctive activities, on the other hand unsupervised approach is suitable for creating probabilistic models with good accuracy.

# Data Source

The primary dataset of the project has been collected from UCI Machine Learning Repository [11], *Human Activity Recognition from Continuous Ambient Sensor Dataset*. The dataset is fairly new, published on 20th September, 2019. This dataset recorded multiple sensor data placed at volunteer resident houses where the residents performed their daily activities with no direct contact with the data collector infrastructure. Ambient PIR motion sensors, door/temperature sensors, and Light Switch sensors are used to record activity data as event stream the sensors are located in different corners of resident houses to record event data. The classification task is to predict the activity that is occurring in the smart home and being observed by the ambient sensors. The sensors communicate using the ZigBee Pro protocol [5,6].

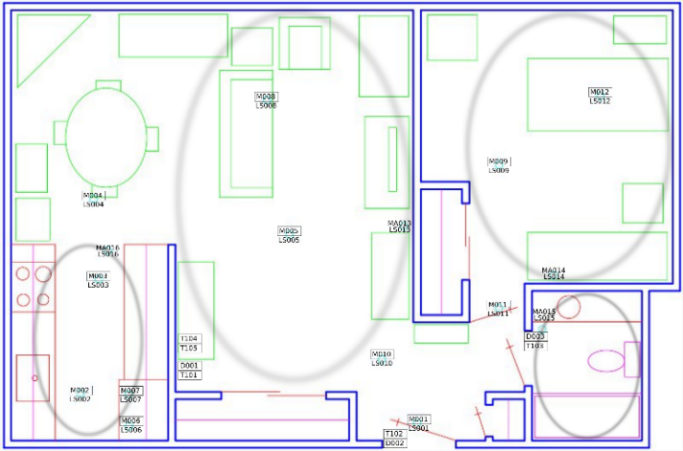
The original format captured from the sensors is provided, as well as the feature vector we generate using a sliding window of 30 sensor events. Each annotated data file (ex: csh101/csh101.ann.txt) has a corresponding feature vector CSV file (ex: csh101/csh101.ann.features.csv). Most of the sensor data files contain labels for two months of the collection period, though some contain labels for extended time periods. [4, 5, 11].

The motion sensors determine the time of motion occurrence in the range of the sensor. The motion sensor reports 1/0 depending on the record of motion activity. The transition period between turning the sensor on and off status is roughly 1.25 seconds. For continuous activity record beyond the threshold time, the sensor will not record 0 until 1.25 seconds after the activity has ceased. One example smart home layout is attached in Figure 1.The key features of the scraped dataset for our purpose is presented in Table 1.

**Table 1: Key features of the Scraped Dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Set Characteristics** | Multivariate, Sequential, Time Series | **Number of Instances** | 4475631 |
| **Attribute Characteristics** | Integer, Real | **Number of Attributes** | 37 |
| **Associated Tasks** | Classification | **Missing Values** | Yes |

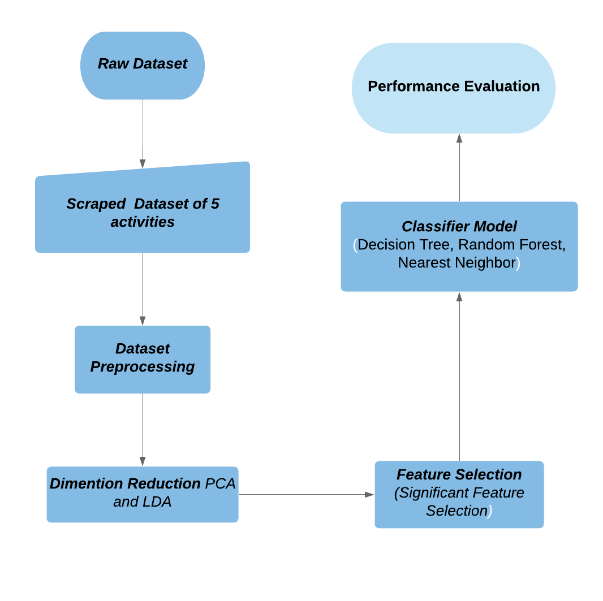
The original dataset is built under the lead of Diane J. Cook from School of Electrical Engineering and Computer Science at Washington State University, and the other creators are Aaron S. Crandall, and Brian L. Thomas. [original dataset paper citation]. Figure 1 shows the illustration…….



**Figure 1: Sensor Layout of One of the Volunteer Resident House**

# **Methodology**

This research work aims to execute activity detection through recognizing the pattern of data collected in “*Human Activity Recognition from Continuous Ambient Sensor Data Dataset*” [4, 11]. The research team has scraped the dataset for this research from the UCI dataset, for the five selected activities (Watching TV, Reading, talking over Phone, Cooking, and Eating). The dataset attributes are presented in table 1. The proposed work is primarily divided into three major segments- Data Preprocessing, Feature Selection and Classifier Model training. Figure 2 presents the basic workflow of this project.



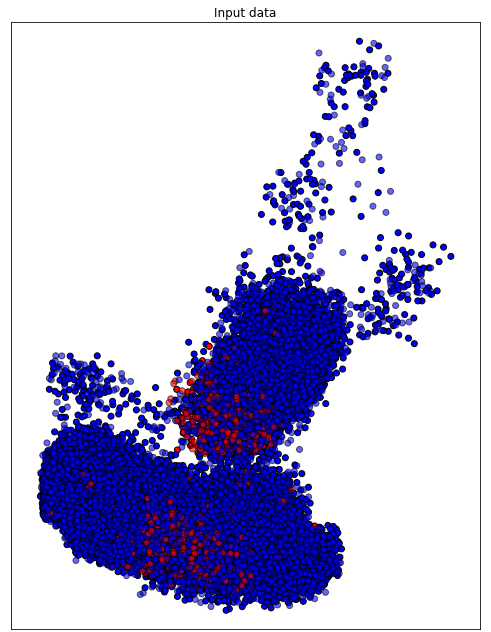
**Figure 2: Workflow of the proposed approach**

## **Data Preprocessing**

The scraped dataset has been standardized and split into training and test set with split ration of 0.3. Figure 3 demonstrates the data points of training and testing set. The blue dots indicate the spread of training set and the red dots indicate the testing set.

## **Dimensionality Reduction**

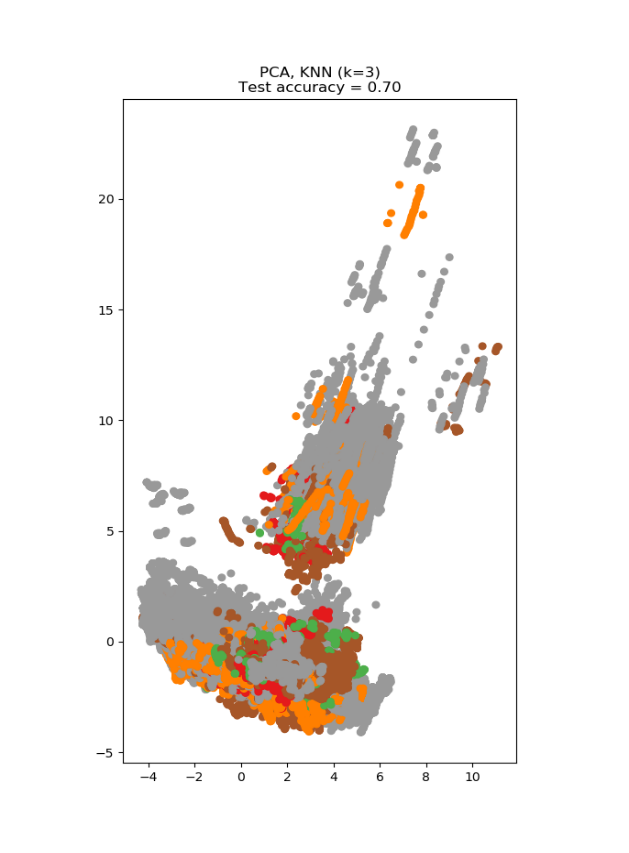
We have applied dimensionality reduction on the dataset to convert the high dimensional data into low-dimensional visually presentable clusters through Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) methods. This approach helped in the end training performance for building a classifier model. Several versions of PCA: PCA 1, PCA 2, PCA 3 have been analyzed; among them PCA 3 has given the greatest number of variabilities upon the dataset.



**Figure 3: Input Data Distribution of Test and Train Split**

### Principal Component Analysis

Principal Component Analysis (PCA) applied to this data presents the most variant combination of principal components reduced from the original dataset. Figure 4 presents the PCA clustering applied on the scraped dataset. The feature of the dataset is standardized first through *StandardScaler( )* function of **LIBRARY NAME**and reduced to dimension of 2. The visual representation of the …….. is demonstrated through Figure 4.



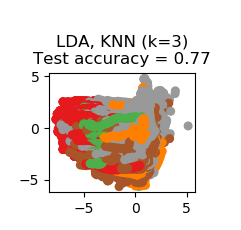
**Figure 4: PCA Presentation with 70% Accuracy**

The PCA variance presents 70% accuracy on the test dataset, which is significantly low since the dimensions are reduced from 37 to principal 2 dimensions.

### Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) identifies attributes with most variance between classes. LDA is a supervised approach that uses known class labels.

The LDA accuracy score outperforms PCA score, with a 77% accuracy score as shown Figure 5. LDA finds centroid of each data point and projects the cluster of data points.



**Figure 5: LDA Presentation with 77% Accuracy**

## **Feature Selection**

The research team has selected five distinct activities for fitting into a classification model. Before fitting the dataset into the explored classifiers for activity recognition; several preprocessing techniques have been applied. Statistical analysis of the attributes has been done to reduce the number of features that do not contribute to training. Feature selection approach thus not only reduces the number of training time and computational cost but also reduces the variance of the model to avoid overfitting. This section describes the feature selection techniques applied in the proposed research. Table 2 lists the set of significant column attributes found by the four feature selection approaches. The selected features through different approaches will be used to train a corresponding model. The best performing model’s features will be kept at last for adversarial attack.

### Low Variance Feature Removal:

The low variance feature selection technique removes the features which is found to be constant mostly. The constant value of a feature is not important to find a pattern and can be removed from the dataset. For dataset with large attributes the scikit-learn library automatically identifies the features which have the lowest variance. The heuristic approach before running the feature selection techniques is to use a threshold value to use as cut-off. The feature elimination is run when any features comes beneath this threshold value. On the given threshold the library computes the covariance against each tuple of the dataset and generates the result. A value of 80% is chosen as the threshold.

### L1 Based Feature Selection:

In SVM the parameter C determines the distribution of the vectors. The smaller the value of C is, the few number of features will be selected. L1 model outputs random value when working on large dataset. This feature selection depends on noise level, smallest absolute value of non-zero coefficients, logarithmic number of features and design matrix structure. The design matrix must contain the property of not being too correlated.

### Tree-based Feature Selection:

The tree-based estimators are used to calculate the statistical significance of features and to discard the irrelevant features. Node impurities measure the importance of features in decision tree.

### Feature Selection with Random Forest:

Random forest classifier uses the tree-based strategies to rank the features for improving purity of the node. The interpretability of this approach is very efficient to derive the importance of each attribute in the dataset based on the tree decision. Feature selection is done by Embedded methods in this approach. Such embedded methods are scalable across any dataset for their high accuracy, better generalization and efficient interpretability; including built-in functions for feature selection. Among the huge number of decision trees, random forest executes a random number of feature selection against each tree. To make the approach less likely to overfit, the trees’ chances of correlation gets decreased as every tree does not observe every variable of the whole dataset. Impurity measure is executed through information gain or entropy in this approach. The whole model observes how much impurity is dependent upon removing a feature. Across each tree the average impurity decrease determines the final importance of the variable.

**Table 2: Selected Features through Feature Selection Approach**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original Set of Features** | **Model Obtained Selected Features with Tree-based Feature Selection** | **Model Obtained Selected Features with Random Forest Classifier** | **Top 21 Extracted Features by Extra Tree Classifiers** | **Top 21 Extracted Features by Random Forest Classifier** |
| lastSensorEventHours | lastSensorEventHours | lastSensorEventHours |  |  |
| lastSensorEventSeconds | lastSensorEventSeconds | lastSensorEventSeconds |  |  |
| lastSensorDayOfWeek | lastSensorDayOfWeek |  |  |  |
| windowDuration |  | windowDuration |  |  |
| timeSinceLastSensorEvent |  |  |  |  |
| prevDominantSensor1 | prevDominantSensor1 |  |  |  |
| prevDominantSensor2 |  |  |  |  |
| lastSensorID |  |  |  |  |
| lastSensorLocation | lastSensorLocation |  |  |  |
| lastMotionLocation | lastMotionLocation | lastMotionLocation |  |  |
| complexity |  |  |  |  |
| activityChange |  |  |  |  |
| areaTransitions |  |  |  |  |
| numDistinctSensors |  |  |  |  |
| sensorCount-Bathroom |  |  |  |  |
| sensorCount-Bedroom |  |  |  |  |
| sensorCount-Chair |  |  |  |  |
| sensorCount-DiningRoom |  |  |  |  |
| sensorCount-Hall |  |  |  |  |
| sensorCount-Ignore |  |  |  |  |
| sensorCount-Kitchen | sensorCount-Kitchen | sensorCount-Kitchen | sensorCount-Kitchen |  |
| sensorCount-LivingRoom | sensorCount-LivingRoom |  |  |  |
| sensorCount-Office |  | sensorElTime-Bedroom |  |  |
| sensorCount-OutsideDoor |  |  |  |  |
| sensorCount-WorkArea |  |  |  |  |

## **Feature Importance**

Feature importance calculates the score for each feature in a dataset through the implementation of forests of tree-based approach. Tree based classifiers come with the inbuilt class of feature selection. The Extra Tree classifier and Random Forest classifier have extracted the top important features of the research dataset. Figure 6 presents the feature importance of the forest, along with inter-trees variability. From the achieved importance score, till \*\*\*\* have been considered to train the classifier models. The research team has applied this approach upon 37 basic features of the scraped dataset. Figure 6 & 7 presents the significant features with bar charts and the scores are listed in table 3 & 4 respectively. These significant features along with the features stated in Table 2 have also been considered for the final training of the classifiers. The four different approaches of feature selection and feature importance implemented by the research team has given 4 models against each technique. A comparative analysis has been run at the end of this paper to find the most efficient feature extraction technique for the dataset.

### **Extra Tree Classifier** The ensemble learning approach of Extremely Randomized Trees Classifiers (Extra Trees Classfier) performs the aggregation of de-correkated decision trees’ results in a forest for classification. The decision trees construction differs in this aspect comparing to the construction of Random Forest Classifier. Each tree in the forest gets a random K sample of feature-set. Each decision tree selects the best feature to split the data on mathemtical basis cacluated by the Gini Index [GINI Index paper citation]. The multiple sampling of features aggreagates the multiple de-correlated decision trees. Extra tree classifier performs feature selection in descending order based on the Gini importance with respect to each feature. The research team in this approch has selected the top 21 features in this aspect based on the feature importance score.

Based on the infromation gain the entropy of the data needs to be calculated. The formula for entropy calculation is:-



Here, c represents the unique class labels and pi is the rows vs. output label proportion. Again, each tree’s information gain is measure with the formula:-



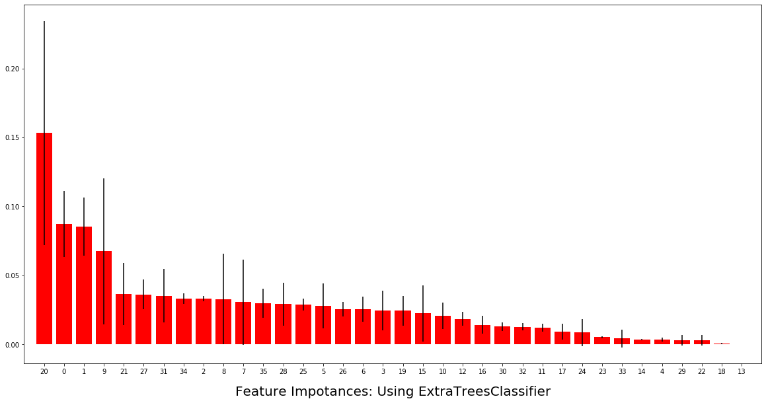
The scikit learn: ExtraTreesClassifier library makes this calculation easy. The acquire results based on the primary dataset has been compiled in Table 3.

**Table 3: Top features extracted from Extra Tree Classifier Approach.**

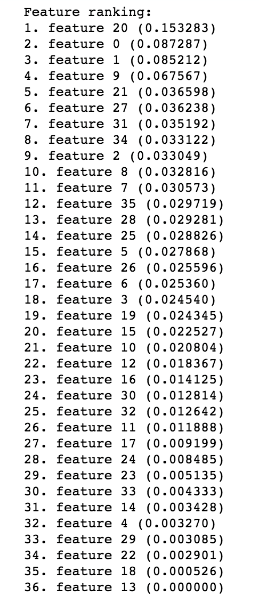
|  |  |
| --- | --- |
| Feature | Score |
| 1. feature 20 (sensorCount-Kitchen) | (0.153283) |
| 2. feature 0 (lastSensorEventHours) | (0.087287) |
| 3. feature 1 (lastSensorEventSeconds) | (0.085212) |
| 4. feature 9 (lastMotionLocation) | (0.067567) |
| 5. feature 21 (sensorCount-LivingRoom) | (0.036598) |
| 6. feature 27 (sensorElTime-Chair) | (0.036238) |
| 7. feature 31 (sensorElTime-Kitchen) | (0.035192) |
| 8. feature 34 (sensorElTime-OutsideDoor) | (0.033122) |
| 9. feature 2 (lastSensorEventSeconds) | (0.033049) |
| 10. feature 8 (lastSensorLocation) | (0.032816) |
| 11. feature 7 (lastSensorID) | (0.030573) |
| 12. feature 35 (sensorElTime-WorkArea) | (0.029719) |
| 13. feature 28 (sensorElTime-DiningRoom) | (0.029281) |
| 14. feature 25 (sensorElTime-Bathroom) | (0.028826) |
| 15. feature 5 (prevDominantSensor1) | (0.027868) |
| 16. feature 26 (sensorElTime-Bedroom) | (0.025596) |
| 17. feature 6 (prevDominantSensor2) | (0.025360) |
| 18. feature 3 (windowDuration) | (0.024540) |
| 19. feature 19 (sensorCount-Ignore) | (0.024345) |
| 20. feature 15 (sensorCount-Bedroom) | (0.022527) |
| 21. feature 10 (complexity) | (0.020804) |

Based on the acquired result, top 21 values have been considered for training in next stage and the rest of the features have been neglected for low feature importance score.

(a)



(b)

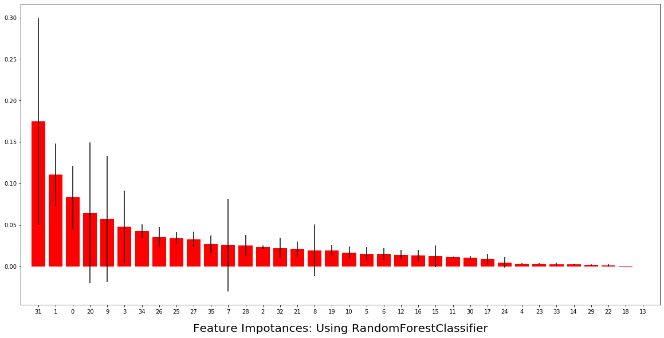
****

**Figure 6: Extra Tree Classifier based feature selection results: (a) Importance Score, (b) Feature ranking**

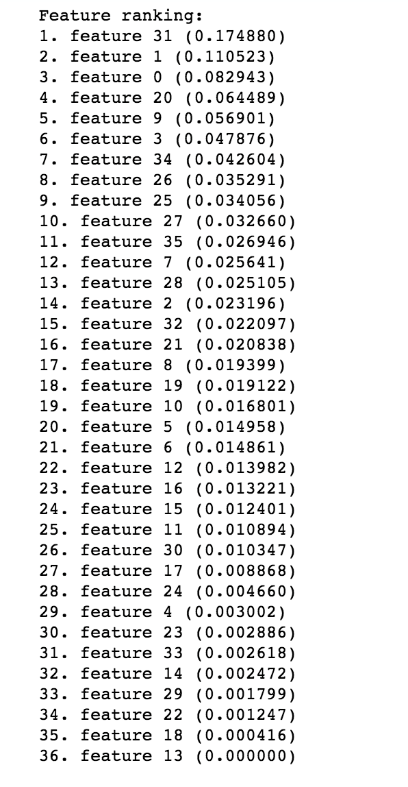
### **Random Forest Classifier**

### The “random forest” concept of this classifier works with several decision trees. Every node in the tree splits the dataset in to sub-set conditioning a single feature. It ensures the similar response values come to the same set. Impurity, here chooses the locally optimal condition by Gini impurity [gini impurity paper] or by the information gain or entropy [information gain/entropy paper]. While training, thus, the computation of decrease in weighted impurity is measured. The average impurity decrease is measured with this approach and the features’ importance are ranked thereby.

(a)



(b)



**Figure 7: Random Forest Classifier based feature selection results: (a) Importance Score, (b) Feature ranking**

**Table 4: Top 10 Significant feature score in Random Forest Classifier**

|  |  |
| --- | --- |
| Feature | Score |
| 1. feature 31 (sensorElTime-Kitchen) | (0.166521) |
| 2. feature 1 (lastSensorEventSeconds) | (0.116857) |
| 3. feature 20 (sensorCount-Kitchen) | (0.085449) |
| 4. feature 0 (lastSensorEventHours) | (0.080503) |
| 5. feature 3 (windowDuration) | (0.045838) |
| 6. feature 9 (lastMotionLocation) | (0.043050) |
| 7. feature 34 (sensorElTime-Kitchen) | (0.041152) |
| 8. feature 25 (sensorElTime-Bathroom) | (0.038173) |
| 9. feature 26 (sensorElTime-Bedroom) | (0.036752) |
| 10. feature 27 (sensorElTime-Chair)X | (0.030283) |
| 11. feature 28 (sensorElTime-DiningRoom) | (0.027393) |
| 12. feature 8 (lastSensorLocation) | (0.024743) |
| 13. feature 35 (sensorElTime-WorkArea) | (0.024461) |
| 14. feature 21 (sensorCount-LivingRoom) | (0.024006) |
| 15. feature 2 (lastSensorDayOfWeek) | (0.022917) |
| 16. feature 32 (sensorElTime-LivingRoom) | (0.022379) |
| 17. feature 19 (sensorCount-Ignore) | (0.020214) |
| 18. feature 10 (complexity)X | (0.017101) |
| 19. feature 7 (lastSensorID)X | (0.015514) |
| 20. feature 15 (sensorCount-Bedroom) | (0.014889) |
| 21. feature 12 (areaTransitions) | (0.014055) |

## **Classifier Comparison**

The classifier comparison presents a set of classifying methods in scikit-learn on the research dataset. The point of this comparison is to illustrate the nature of decision boundaries of different classifiers. After feature selection is done, two versions of datasets are generated based on the Tree-based and Random-forest based feature selection. The L1-based and Low-variance approach do not reduce the dimension significantly and hence we have discarded those results.

We have tested on **Nearest Neighbor, Decision Tree and Random Forest Classifiers** to run on the two datasets.

# **Performance evaluation**

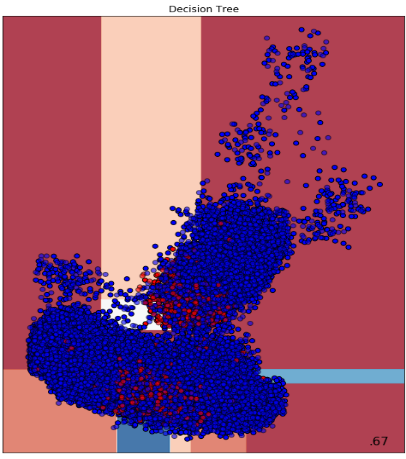
In this section, three state-of-the-art classifier models for activity recognition in HAR system are applied on the primarily sparse dataset and the dataset after applying different feature selection methods. The applied three models are: Decision Tree, Nearest Neighbor and Random Forest. The results clearly show an increase of 5%-10% on accuracy varying on models when dataset is preprocessed doing feature selection and significant features are selected. Later in this section, the confusion matrices with respect a particular classifier have been presented and the generated statistical metric score have been compared and for each model performance.

**Table 5: Performance evaluation summary across different feature-set selection techniques and classifiers.**

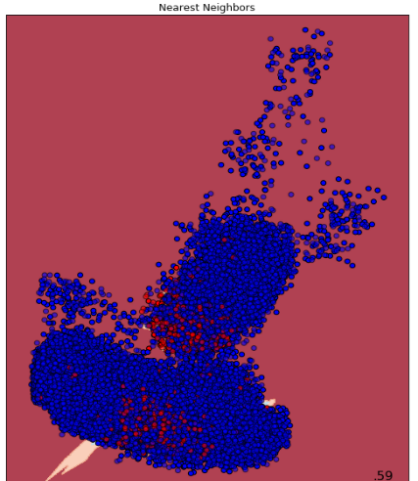
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classifier | **Original Set of Features** | **Model Obtained Selected Features with Tree-based Feature Selection** | **Model Obtained Selected Features with Random Forest Classifier** | **Top 21 Extracted Features by Extra Tree Classifiers** | **Top 21 Extracted Features by Random Forest Classifier** |
| Nearest Neighbors | 59 | 62 | 61 | 66 | 65 |
| Decision Tree | 67 | 70 | 70 | 74 | 73 |
| Random Forest | 67 | 70 | 70 | 74 | 73 |
| Naive Bayes |  |  |  | 74 | 72 |
| Neural Net |  |  |  | 74 | 73 |

## **Classifiers peroformance without Feature Selection**

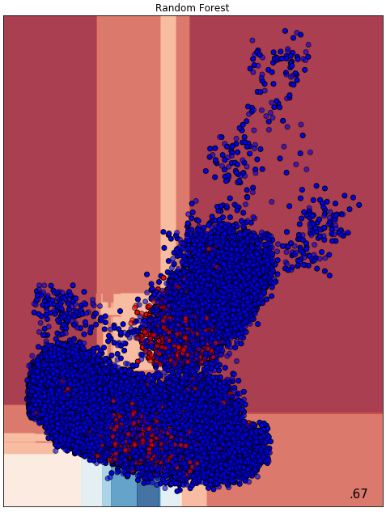
At first step, the research team trained the classifier models without applying feature selection on the dataset. The Figures 9-11 represent the model accuracy of Decision Tree, Nearest Neighbor and Random Forest respectively on the primary dataset.



**Figure 9: Decision Tree Accuracy 67%**

****

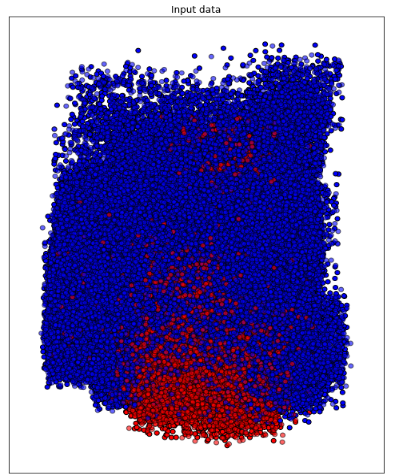
**Figure 10: Nearest Neighbor Accuracy 59%**

****

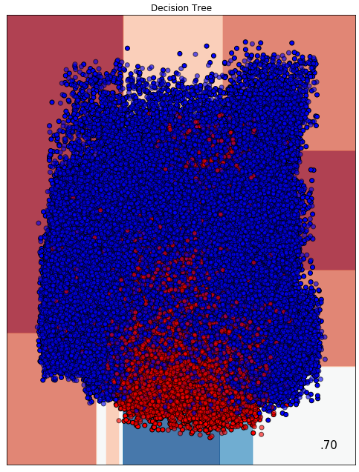
**Figure 11: Random Forest Accuracy 67%**

## **Classifiers peroformance on Tree-based Classifier Feature Selection**

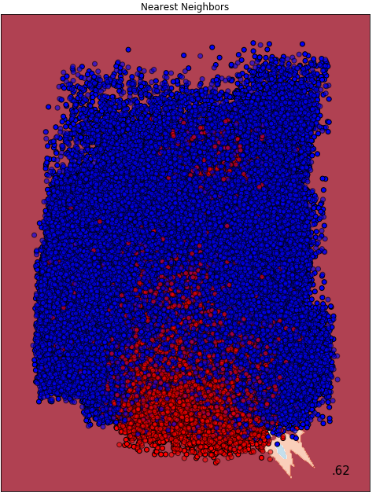
The tree-based classifier feature selection selects top few attributes and produces a new set of important features. The significant features detected with this approach are enlisted in Table 2. Figure 12-15 represent the new dataset distribution based on the selection and the same classifier models’ (Decision Tree, Nearest Neighbor and Random Forest respectively) accuracy scores. The distribution pattern in Figure 12 is significantly different and denser compared to the input distribution of the dataset without feature selection in Figure 3.



**Figure 12: Input Data Distribution of Tree-based Feature Selection**

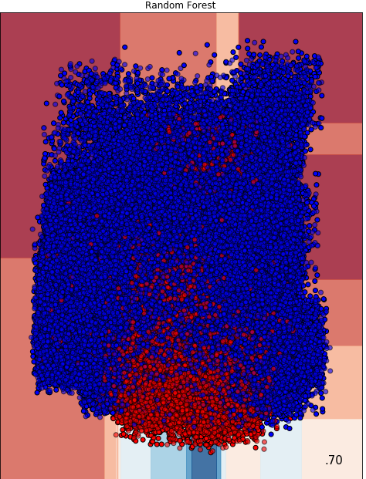


**Figure 13: Decision Tree Accuracy 70%**



**Figure 14: Nearest Neighbor Accuracy 62%**

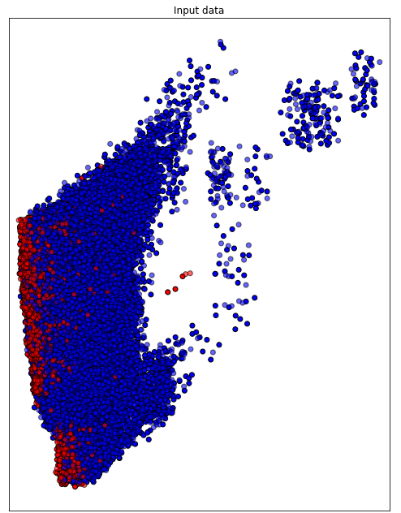
w



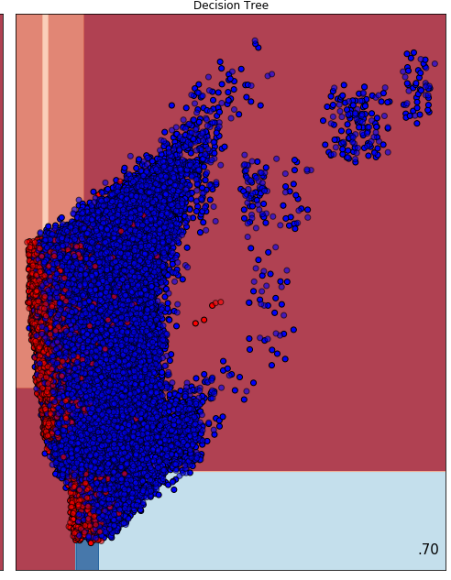
**Figure 15: Random Forest Accuracy 70%**

## **Classifiers peroformance on Random Forest Classifier Feature Selection**

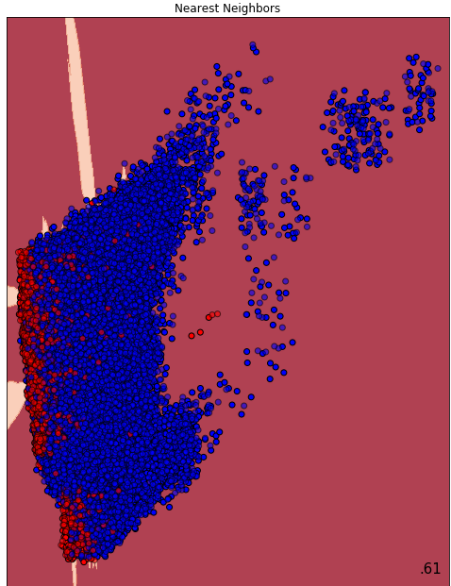
The Random-forest selection classifier feature selection selects top few attributes and produces a new set of features mentioned in Table 2. Figures 16-19 represent the new dataset distribution based on the selection and the following classifier models (Decision Tree, Nearest Neighbor and Random Forest respectively) accuracy scores.



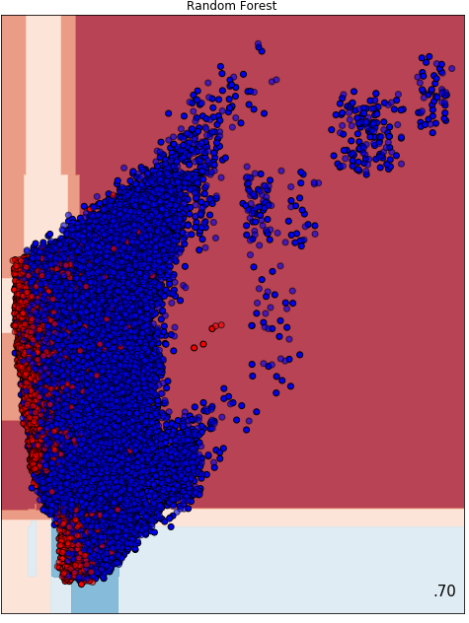
**Figure 16: Input Distribution of Random-Forest Based Feature Selection**

**.**

**Figure 17: Decision Tree Accuracy 70%**

****

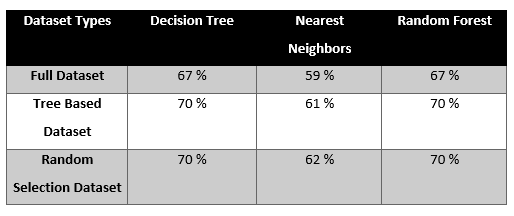
**Figure 18: Nearest Neighbor Accuracy 61%**

****

**Figure 19: Random Forest Accuracy 70%**

Table 5 summarizes the performance of the three different classifier models on the 3 datasets.

**Table 5: Accuracy Score of Classifier Models on Three Datasets**

****

## Evaluation Metrics

The performance of the three classifier model on 3 datasets is evaluated through four key metrics of accuracy: *precision, recall, f1-score and support*. We have applied the classifier models on the raw dataset as well as the feature selected two other datasets. The confusion matrix and evaluation metrics are presented here on the basis of the different datasets and the result of classification models on those datasets. The activities are coded into numerical values in here, hence the below graphs will show activity id comparisons.

Accuracy gives the sum of correct classifications to the total number of instances.

Accuracy =

Where TP i.e. true positive is the category of positive attributes correctly classified as positive attributes, TN i.e. true negative is the set of negative samples identified as negative samples. FP i.e. false positive is the category of negative attributes classified as positive attributes. And FN i.e. false negative are the positive samples being classified as negative samples.

The formula for Precision =

This performance metric presents the proportion of positive attributes those were classified correctly. Recall presents the proportion of actual positive classes those were identified in proportion to all samples in the actual class

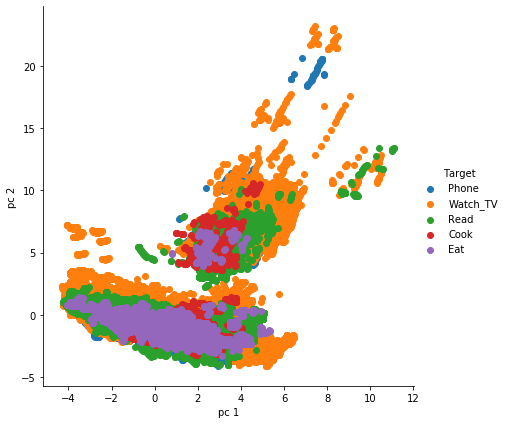
The formula for Recall =

F1-score is a performance metric that measures the weighted harmonic mean of precision and recall. It is used to evaluate the classification accuracy of an algorithm.

F−Score=2\*

### Full Dataset Metric Scores

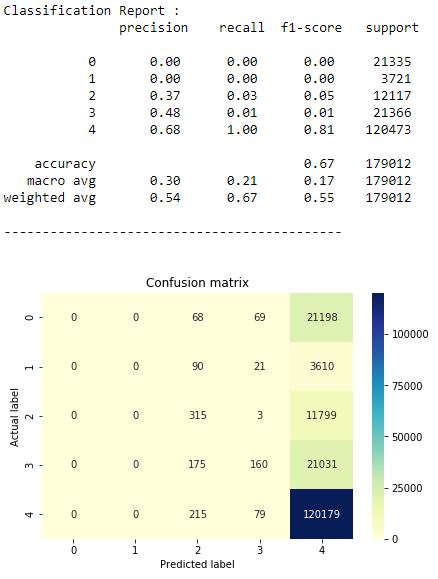
In this sub-section, three confusion matrix and respective statistical scores are presented on the basis of the three classifier models applied on the dataset without significant feature selection. Figure 20 presents the PCA clustering of the full dataset.



**Figure 20: PCA Clustering of Full Dataset**

#### Decision Tree

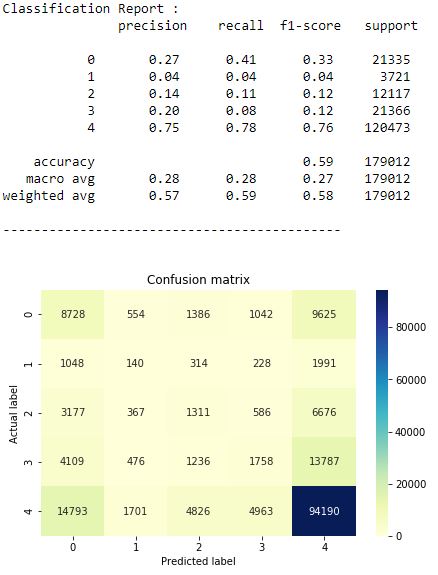
The evaluation metrics show that decision tree worked fairly well for activity 2 and activity 4, with a weighted average accuracy of 67% in figure 21. For activity 0, 1 and 3, values are zero or near to zero. This represents that decision tree didn’t work so well on the full dataset.



**Figure 21: Decision Tree Confusion Matrix and Evaluation Scores on Full Dataset**

#### Nearest Neighbor

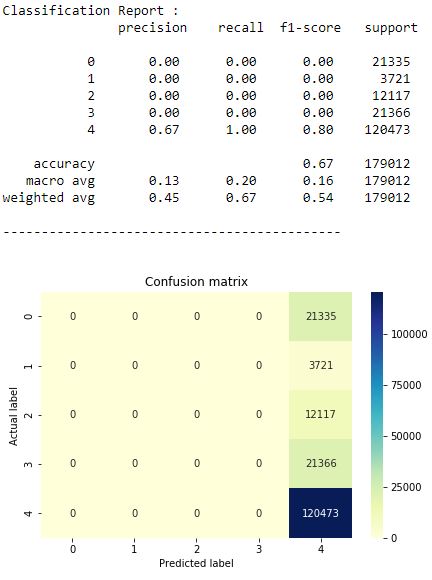
The evaluation metrics show that compared to decision tree, nearest neighbor worked fairly well for all 5 activities as shown in figure 22. The f1-score ranging from 0.76 for activity-4 to the least 0.04 for activity-1 presents a moderate performance of the model on the dataset. The overall accuracy stands to 59% in this case.



**Figure 22: Nearest Neighbor Confusion Matrix and Evaluation Scores on Full Dataset**

#### Random Forest

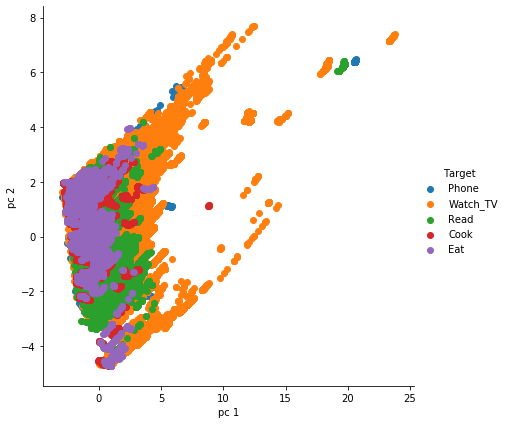
In this case, the evaluation metric showed quite surprising results in terms of zero f1-score for all 4 activities and 81% for activity-4. All the evaluation metrics through Random forest classifier only could classify activity 4 fairly well. The overall accuracy is 67% as presented in figure 23.



**Figure 23: Random Forest Confusion Matrix and Evaluation Scores on Full Dataset**

### Random Forest Feature Select Dataset

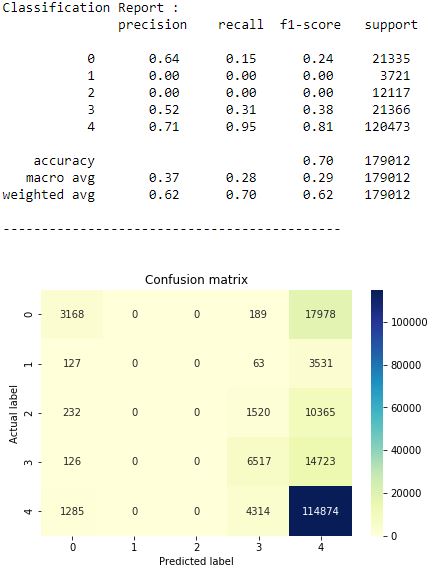
In this sub-section, the dataset in execution consists of the selected statistically significant attributes from Random Forest Feature Select approach on the full dataset. Three classier models have been executed and evaluation metric scores of each one of those is discussed below. Figure 24 presents the PCA clustering of this feature selected dataset.



**Figure 24: PCA Clustering of Feature Selected Dataset**

#### Decision Tree

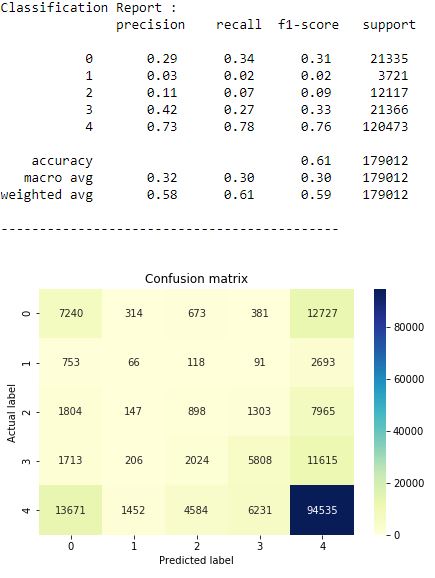
Unlikely the first iteration of Decision Tree on full dataset, the model performs better in the Random Forest Feature Selected Dataset. The overall accuracy here is 70%, while the F1-score goes 81% as maximum for activity 4. Precision, Recall and F1-score- each metric has 1 pair of zero values for activity 1 & 2. Figure 25 presents the confusion matrix for this classifier model on the feature selected dataset.



**Figure 25: Decision Tree Confusion Matrix and Evaluation Scores on Random Forest Feature Select Dataset**

#### Nearest Neighbor

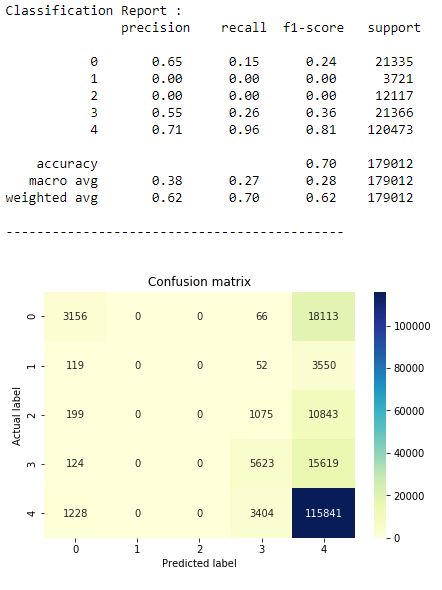
Nearest Neighbor classifier on Random Forest feature select dataset produces overall accuracy of 61%, while the F1-score, Precision and Recall metrics here produces values more than zero for all the activities. The highest metric score goes to activity-4 in all three metrics, as presented in Figure 26.



**Figure 26: Nearest Neighbor Confusion Matrix and Evaluation Scores on Random Forest Feature Select Dataset**

#### Random Forest

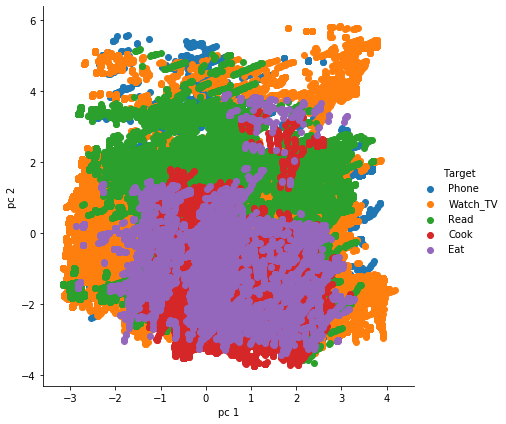
Random forest classifier on this dataset produces an overall accuracy of 70%. The outputs here have a strong similarity with the Decision Tree result on the same dataset, with 1%-2% difference for metric scores recorded at each activity. The findings are presented in figure 27.



**Figure 27: Random Forest Confusion Matrix and Evaluation Scores on Random Forest Feature Select Dataset**

### Tree based Feature Select Dataset

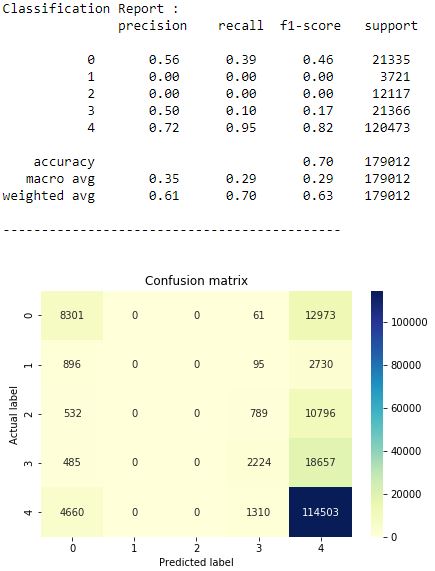
In this sub-section, the dataset in execution consists of the selected statistically significant attributes from Tree Based Feature Select approach on the full dataset. Three classier models have been executed and evaluation metric scores of each one of those is discussed below. Figure 28 presents the PCA clustering of this dataset.



**Figure 28: PCA Clustering on Tree Based Feature Select Dataset**

#### Decision Tree

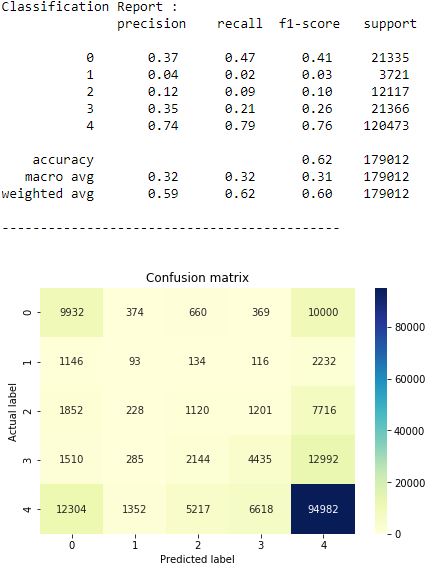
The overall accuracy of Decision Tree on this dataset is 70%, while the F1-score goes 82% as maximum for activity 4. Precision, Recall and F1-score- each metric has 1 pair of zero values for activity 1 & 2, similar to the decision tree result on random forest based feature selected dataset. The confusion matrix is presented in figure 29.



**Figure 29: Decision Tree Confusion Matrix and Evaluation Scores on Tree Based Feature Select Dataset**

#### Nearest Neighbor

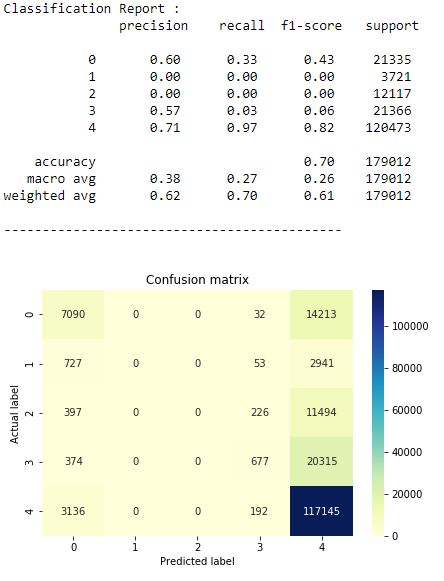
Nearest Neighbor Classifier on this dataset achieves 62% overall accuracy. All the metric score are consistently measured for all activities. Nearest neighbor showed the consistent output of being able to classify correctly all the activities across all three datasets. Figure 30 presents the evaluation metric score and confusion matrix here.



**Figure 30: Nearest Neighbor Confusion Matrix and Evaluation Scores on Tree Based Feature Select Dataset**

#### Random Forest

Random forest classifier on this dataset achieves an overall accuracy of 70%, as shown in figure 31. In this dataset too, activities 1 & 2 have achieved zero metric score across all three evaluation metrics on Random Forest.



**Figure 31: Random Forest Confusion Matrix and Evaluation Scores on Tree Based Feature Select Dataset**

On the basis of these nine analysis, Nearest Neighbor has proved to be the overall best classifier for these three datasets, on the classifying ability of five listed activities.

# Conclusion

In this paper, we present a comparative model approach to classify five selected activities from the dataset. The classifier models show significant changes after application of precise data preprocessing and feature selection approach. The accuracy score is increased by 10% using Decision Tree from the raw dataset when feature selection is applied. From the three classifier state-of-the-art model for human activity recognition, we have utilized here Decision Tree, Random Forest and Nearest Neighbor. Only the Nearest Neighbor classifier has persistently detected all five activities with varying metric score in the three datasets of this research. On the raw dataset without prior feature selection based on feature significance calculation, Nearest Neighbor achieved 59% overall accuracy, which increased to 61% in Tree based feature selected dataset and 62% in Random Forest based dataset. Hence, this research paper presents that for human activity recognition systems, data preprocessing and feature selection greatly affects the classification performance and consequently the AAL and AML structures on the basis of HAR. State-of-the-art classifier models have presented varying accuracy score on the basis of how well the dataset have been preprocessed for running machine learning model on the dataset.

The future work includes preparing neural network approach to classify the activities and on the basis of the model, we aim to produce a robust time-series model to handle adversarial attack.

The end goal of the research work is to inject adversarial attack on the model to confuse the network and identify the actual activity after the injection. To suggest more amicable work based on such data, this research team could be explored in variety of fields in health, administration and security issues where such dataset generation and model implementation will be useful for activity recognition.

# REferences

1. G. A. Oguntala et al., "SmartWall: Novel RFID-Enabled Ambient Human Activity Recognition Using Machine Learning for Unobtrusive Health Monitoring," in IEEE Access, vol. 7, pp. 68022-68033, 2019. doi: 10.1109/ACCESS.2019.2917125
2. D. Cook, N. Krishnan, and P. Rashidi. Activity discovery and activity recognition: A new partnership. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 43(3):820-828, 2013
3. Reisburg B, Finkel S, Overall J, Schmidt-Gollas N, Kanowski S, Lehfeld H, Hulla F, Sclan SG, Wilms HU, Heininger K, Hindmarch I, Stemmler M, Poon L, Kluger A, Cooler C, Bergener M, Hugonot-Diener L, robert PH, Erzigkeit H. The Alzheimer’s disease activities of daily living international scale (ADL-IS) International Psychogeriatrics. 2001;13(2):163–181
4. S. Aminikhanghahi, T. Wang, and D. Cook. Real-time change point detection with application to smart home time series data. IEEE Transactions on Knowledge and Data Engineering, to appear
5. S. Aminikhanghahi and D. Cook. Enhancing activity recognition using CPD-based activity segmentation. Pervasive and Mobile Computing, 53:75-89, 2019
6. A. Alberdi, A. Weakley, A. Goenaga, M. Schmitter-Edgecombe, and D. Cook. Automatic assessment of functional health decline in older adults based on smart home data. Journal of Biomedical Informatics, 18:119-130, 2018
7. Brdiczka O, Crowley JL, Reignier P. Learning situation models in a smart home. IEEE Transactions on Systems, Man, and Cybernetics, Part B. 2009;39(1)
8. B. Minor, J. Doppa, and D. Cook. Learning activity predictors from sensor data: Algorithms, evaluation, and applications. IEEE Transactions on Knowledge and Data Engineering, 29(12):2744-2757, 2017
9. Logan B, Healey J, Philipose M, Tapia EM, Intille S. A long-term evaluation of sensing modalities for activity recognition. Proceedings of the International Conference on Ubiquitous Computing; 200
10. Gu T, Chen S, Tao X, Lu J. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. Data and Knowledge Engineering. 2010
11. P. Alinia, C. Cain, R. Fallahzadeh, A. Shahrokni, and H. Ghasemzadeh. How accurate is your activity tracker? A comparative study of step counts in low-intensity physical activities. Journal of Medical Internet Research, 5(8):e106, 2017
12. K. Feuz, and D. Cook. Modeling skewed class distributions by reshaping the concept space. AAAI Conference on Artificial Intelligence, 2017
13. J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita,“Transition-Aware Human Activity Recognition Using Smartphones,” Neurocomputing, vol. 171, pp. 754–767, Jan. 2016
14. Q. Ni, A. B. García Hernando, and I. P. de la Cruz, “The Elderly’s Independent Living in Smart Homes: A Characterization of Activities and Sensing Infrastructure Survey to Facilitate Services Development.” Sensors. vol. 15, no. 5, pp. 11312–62, May 2015.
15. “Welcome to CASAS.” [Online]. Available: http://casas.wsu.edu/datasets/. [Accessed: 24-Apr-2018].
16. Lara, Oscar & Labrador, Miguel. (2013). A Survey on Human Activity Recognition Using Wearable Sensors. Communications Surveys & Tutorials, IEEE. 15. 1192-1209. 10.1109/SURV.2012.110112.00192.
17. Davis-Owusu, Kadian & Owusu, Evans & Bastani, Vahid & Marcenaro, Lucio & Hu, Jun & Regazzoni, Carlo & Feijs, Loe. (2016). Activity recognition based on inertial sensors for Ambient Assisted Living.
18. B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. B¨ula, and P. Robert, “Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly,” Biomedical Engineering, IEEE Transactions on, vol. 50, no. 6, pp. 711–723, 2003.
19. Assessing the quality of activities in a smart environment. Cook DJ, Schmitter-Edgecombe M Methods Inf Med. 2009; 48(5):480-5
20. N. A. Capela, E. D. Lemaire, and N. Baddour, “Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients,” PloS one, vol. 10, no. 4, p. e0124414, 2015.
21. K. Davis, E. Owusu, C. Regazzoni, L. Marcenaro, L. Feijs, and J. Hu,“Perception of human activities a means to support connectedness between the elderly and their caregivers,” in Proceedings of the 1st International Conference on Information and Communication Technologies for Ageing Well and e-Health. SCITEPRESS, 2015, pp. 194–199.
22. Gu T, Wu Z, Tao X, Pung HK, Lu J. epSICAR: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. Proceedings of the IEEE International Conference on Pervasive Computing and Communications; 2009. pp. 1–9
23. N. D. Rodríguez, M. P. Cuéllar, J. Lilius, M. D. Calvo-Flores, "A survey on ontologies for human behavior recognition", ACM Comput. Surv., vol. 46, no. 4, pp. 1-43, 2014
24. Nweke, Henry & Wah, Teh & al-garadi, Mohammed & Alo, Uzoma. (2018). Deep Learning Algorithms for Human Activity Recognition using Mobile and Wearable Sensor Networks: State of the Art and Research Challenges. Expert Systems with Applications. 105. 10.1016/j.eswa.2018.03.056.
25. Banos, O., Garcia, R., Holgado-Terriza, J. A., Damas, M., Pomares, H., Rojas, I., Saez, A., & Villalonga, C. (2014). mHealthDroid: a novel framework for agile development of mobile health applications. In International Workshop on Ambient Assisted Living (pp. 91-98): Springer.
26. Y. Chen, L. Yu, K. Ota, M. Dong, "Robust activity recognition for aging society", IEEE J. Biomed. Health Inform., vol. 22, no. 6, pp. 1754-1764, Nov. 2018
27. Cornacchia, M., Ozcan, K., Zheng, Y., & Velipasalar, S. (2017). A Survey on Activity Detection and Classification Using Wearable Sensors. IEEE Sensors Journal, 17, 386-403.
28. Ha, S., Yun, J. M., & Choi, S. (2015). Multi-modal Convolutional Neural Networks for Activity Recognition. In 2015 IEEE International Conference on Systems, Man, and Cybernetics (pp. 3017-3022).
29. B. Chen, Z. Fan, and F. Cao, “Activity Recognition Based on Streaming Sensor Data for Assisted Living in Smart Homes,” in 2015 International Conference on Intelligent Environments, 2015, pp. 124–127.
30. K. Robertson, C. Rosasco, K. Feuz, M. Schmitter-Edgecombe, and D. Cook, “Prompting technologies: A comparison of time-based and context-aware transition-based prompting.,” Technol. Health Care, vol. 23, no. 6,pp. 745–56, Jan. 2015.
31. J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A Survey,” Pattern Recognit. Lett., Feb. 2018.
32. A. Jordao, A. C. Nazare, J. Sena, W. R. Schwartz, "Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art" in arXiv:1806.05226, 2018, [online] Available: <https://arxiv.org/abs/1806.05226>
33. K. Wang, J. He and L. Zhang, "Attention-Based Convolutional Neural Network for Weakly Labeled Human Activities’ Recognition With Wearable Sensors," in IEEE Sensors Journal, vol. 19, no. 17, pp. 7598-7604, 1 Sept.1, 2019. doi: 10.1109/JSEN.2019.2917225
34. Chung, Seungeun et al. “Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning.” Sensors (Basel, Switzerland) vol. 19,7 1716. 10 Apr. 2019, doi:10.3390/s19071716.