Handling Adversarial Attacks on Deep Neural Network through Classification Technique to Smart Home Time Series Data

Mahbuba Tasmin
ID : 1610064042

Sharif Uddin Ruman
ID: 1611557642

N.M. Shihab Islam
ID: 1420339042

Abdur Raufus Saleheen
ID: 1610472642

Sifat Jahan
ID: 1611702642

***Abstract*— Human activity Recognition (HAR) is a challenging time series classification task based on neural network modeling to classify the activity of new unseen subjects from the collected sensor data. It involves predict the movement/activities of a person based on time series data collected from accelerometer of a smartphone or motion sensors in indoor setup. The classification and prediction uses deep domain expertise and signal processing to engineer features from raw data to fit into prospective machine learning model. The exposed vulnerability of deep learning models to adversarial time series examples may lead to false classification result, which is still not widely addressed in the field of HAR activity recognition. In this project, we propose to classify HAR activities from Ambient Sensor Dataset of UCI repository, with added feature of robust architecture of handling adversarial attack on the time series data. A special noise is added to the input time series to reduce the network's confidence when classifying instances at test time. We have prepared and engineered the important features from the raw dataset and applied classifier models on the prepared dataset. The adversarial attack mechanism will be applied in the last phase of the project.**

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

I. DATA SOURCE

The dataset of the project is collected from UCI Machine Learning Repository, *Human Activity Recognition from Continuous Ambient Sensor Data Dataset*. The dataset is fairly new, published on 20th September, 2019.

This dataset represents ambient data collected in homes with volunteer residents with their usual daily activities at home. **Ambient PIR motion sensors, door/temperature sensors, and Light Switch sensors** are placed throughout the home of the volunteer. The sensors are placed in locations throughout the home that are related to specific target activity of daily living.

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**, forming a mesh network with all battery powered sensors as leaf nodes and always-on devices (light switches and ZigBee relays) forming the branches that connect back to the USB gateway on our local SHiB server.

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.

The smart home layout for total **30 volunteer resident houses** and sensor placement from the original formats is found in the included sensor map for each smart home. One example layout is attached below:

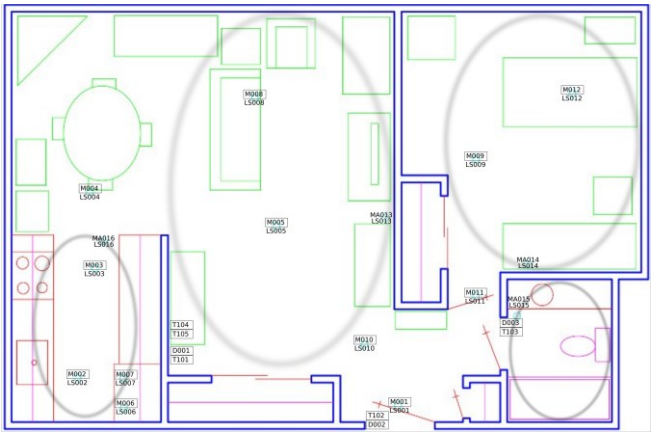

Figure 1: Sensor Layout of One of the Volunteer Resident House

The dataset is collected 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.

The key features of the dataset is presented in the below table:

Data Set Characteristics	Multivariate, Sequential, Time Series	Number of Instances	13956534
Attribute Characteristics	Integer, Real	Number of Attributes	37
Associated Tasks	Classification	Missing Values	Yes
Area	N/A		

Table 1: Key features of the Dataset

II. METHODOLOGY

A. *Data Preprocessing*

From the raw datasets of 30 volunteer resident homes, there is 37 total activities recorded. For the execution purpose, we have selected 5 activities (Watch TV, Read, Phone, Cook, Eat).We have scraped the dataset for these 5 activities and reproduced new set of dataset for our project purpose.

From the collected dataset, we excluded four column attributes based on unique value and data variance of the columns.

B. *Dimensionality Reduction*

We have applied dimensionality reduction through Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) methods to make the data more visually understandable. Principal Component Analysis (PCA) applied to this data identifies the combination of attributes (principal components, or directions in the feature space) that account for the most variance in the data. Here we plot the different samples on the 2 first principal components. Linear Discriminant Analysis (LDA) tries to identify attributes that account for the most variance between classes. In particular, LDA, in contrast to PCA, is a supervised method, using known class labels.

C. *Feature Selection*

For a dataset with d input features, the feature selection process results in k features such that k < d, where k is the smallest set of significant and relevant features. One of the feature selection techniques is Wrapper Method, which includes Backward Elimination and Bi-Directional Elimination (Stepwise Selection).
In backward elimination, we start with the full model (including all the independent variables) and then remove the insignificant feature with highest **p-value (> significance level)**. This process repeats again and again until we have the final set of significant features.

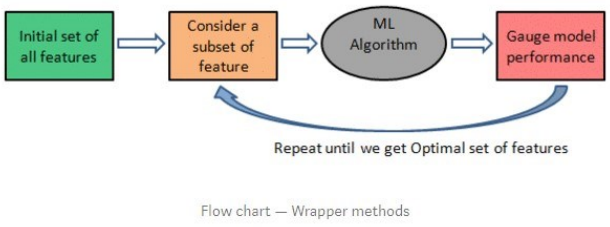

Figure 2: Backtracking Methods

Bi-directional elimination(Stepwise Selection) is similar to forward selection but the difference is while adding a new feature it also checks the significance of already added features and if it finds any of the already selected features insignificant then it simply removes that particular feature through backward elimination.It is a combination of forward selection and backward elimination.

D. *Classifier Comparison*

The classifier comparison presents a set of classifying methods in scikit-learn on our dataset. The point of this comparision is to illustrate the nature of decision boundaries of different classifiers.
Particularly in high-dimensional spaces, data can more easily be separated linearly and the simplicity of classifiers such as naive Bayes and linear SVMs might lead to better generalization than is achieved by other classifiers.
The plots show training points in solid colors and testing points semi-transparent. The lower right shows the classification accuracy on the test set.

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

III. RESULT

In this part of the report, we present the outputs of preprocessing and other parts of the project.

A. *Input Data Visualization*

Figure 3 represents the input data split into test and train set, marked in blue and red points respectively.

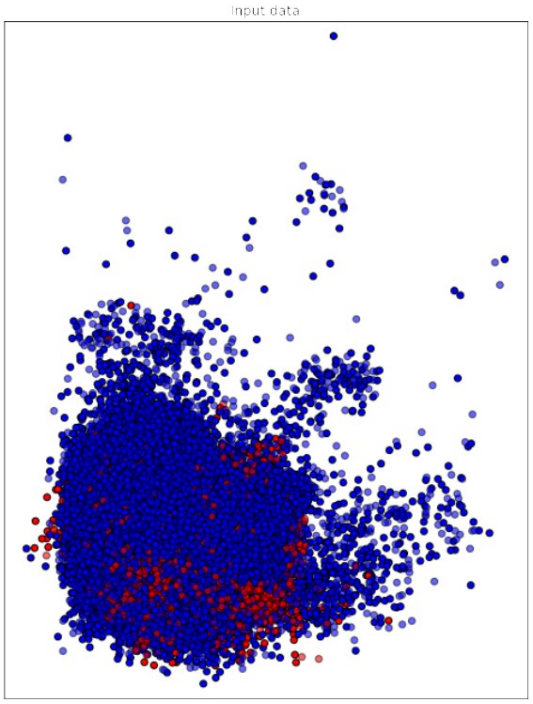

Figure 3: Input Data Distribution of Test and Train Split

B. *Data Visualization Through PCA*

We have applied PCA on the selected six datasets, figures 4 – 9 represent the PCA reduction representation of the datasets.

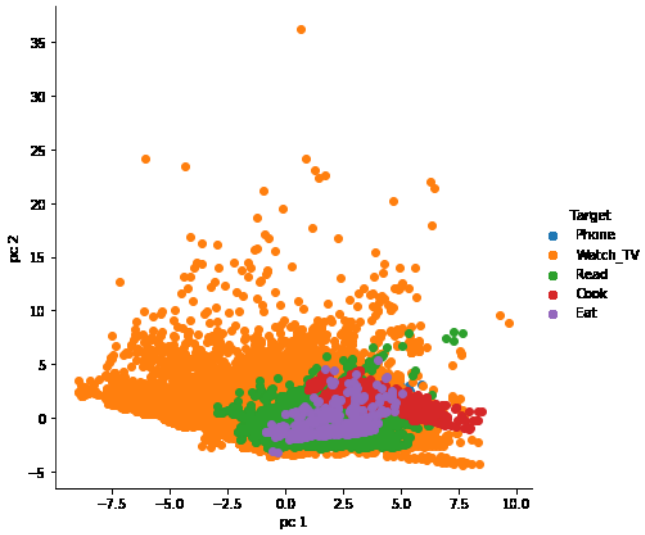

Figure 4: PCA reduction of Dataset 1

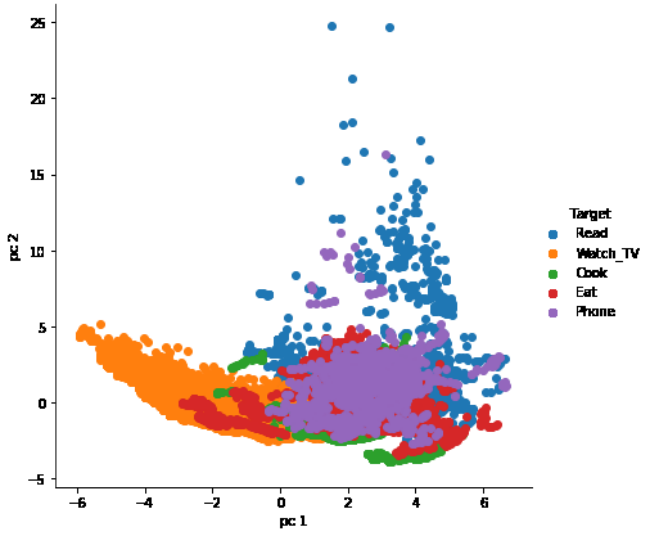

Figure 5: PCA reduction of Dataset 2

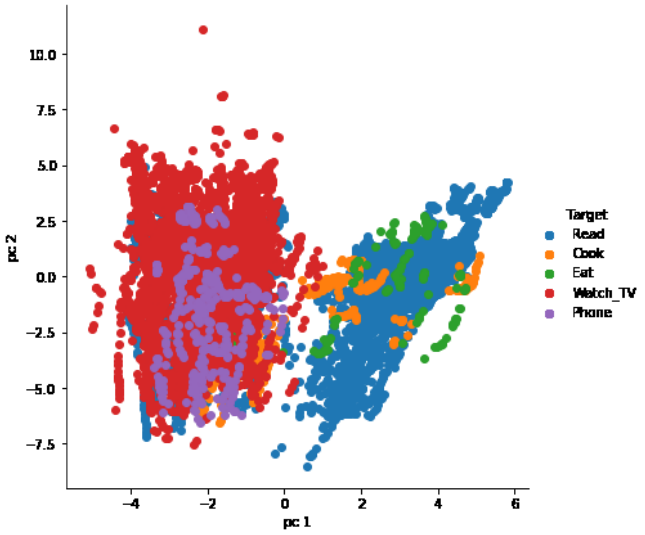

Figure 6: PCA reduction of Dataset 3

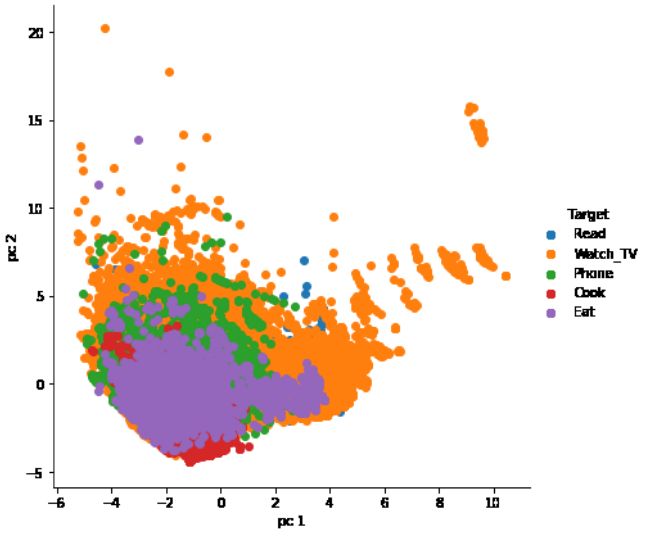

Figure 7: PCA reduction of Dataset 4

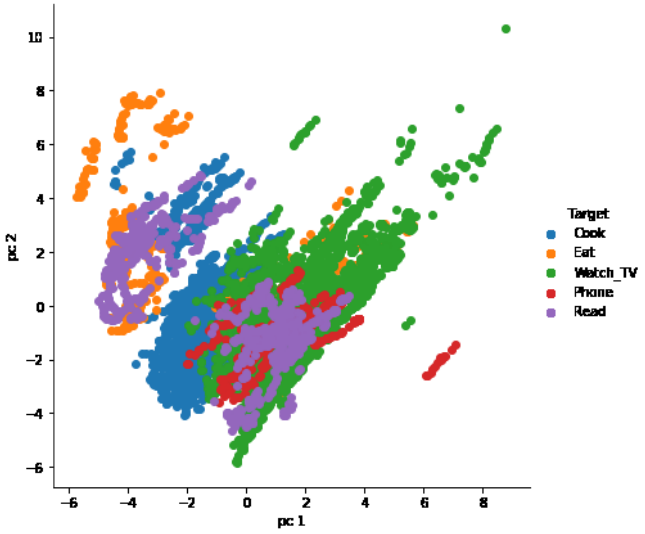

Figure 8: PCA reduction of Dataset 5

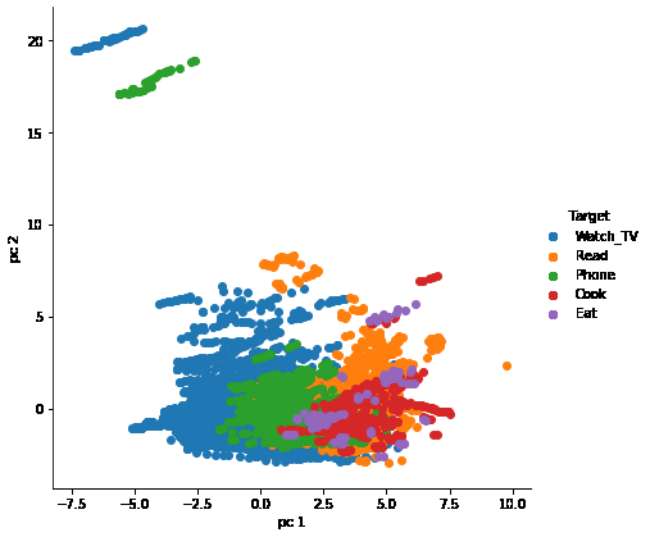

Figure 9: PCA reduction of Dataset 6

C. Subplot

Representation of PC1 and PC2 components from PCA analysis on the dataset.

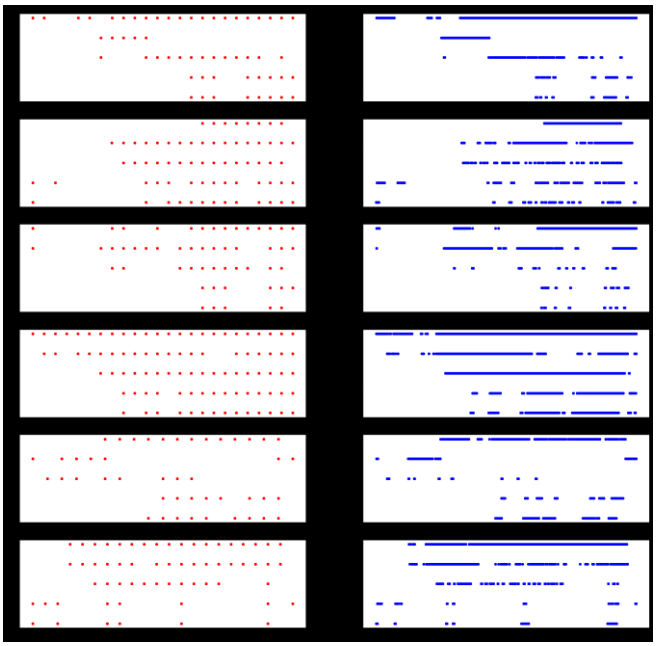

Figure 10: Subplot representation of PC1 and PC2 Components.

D. Classifier Comparison Output

We have applied Nearest Neighbors, Decision tree and Random forest classifier on the test set and the accuracy of the classifiers is shown in down-left corner of each of the pictures.

In the nearest neighbor method, the accuracy comes as 59%,

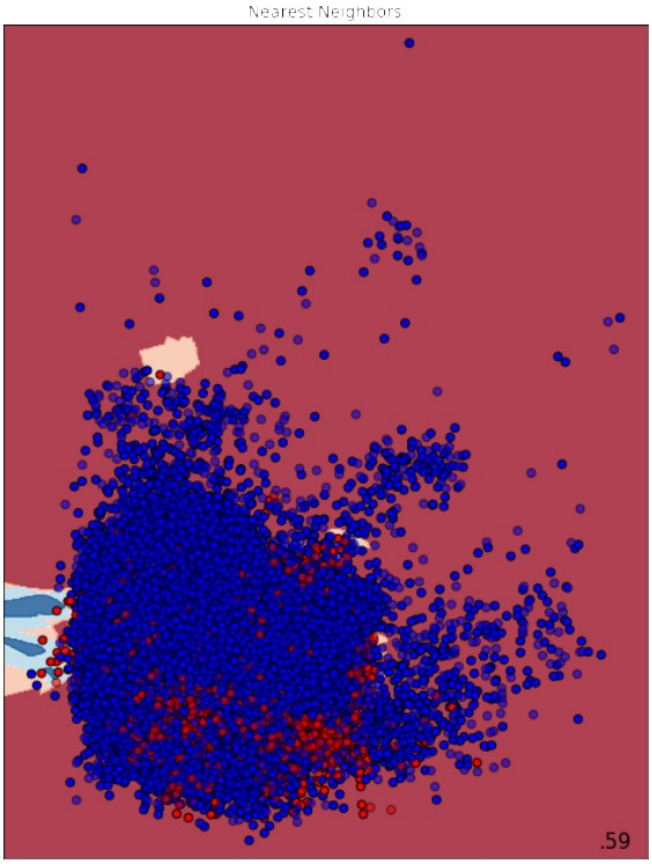

Figure 11: Nearest Neighbors Classifier

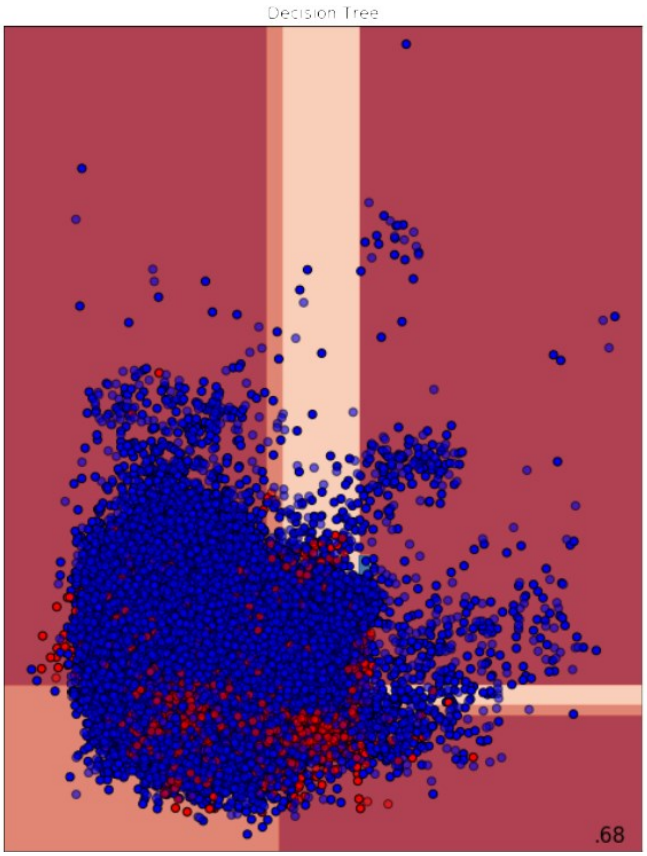

Figure 12: Decision Tree Classifier

In the Decision Tree method, the accuracy comes as 68%,

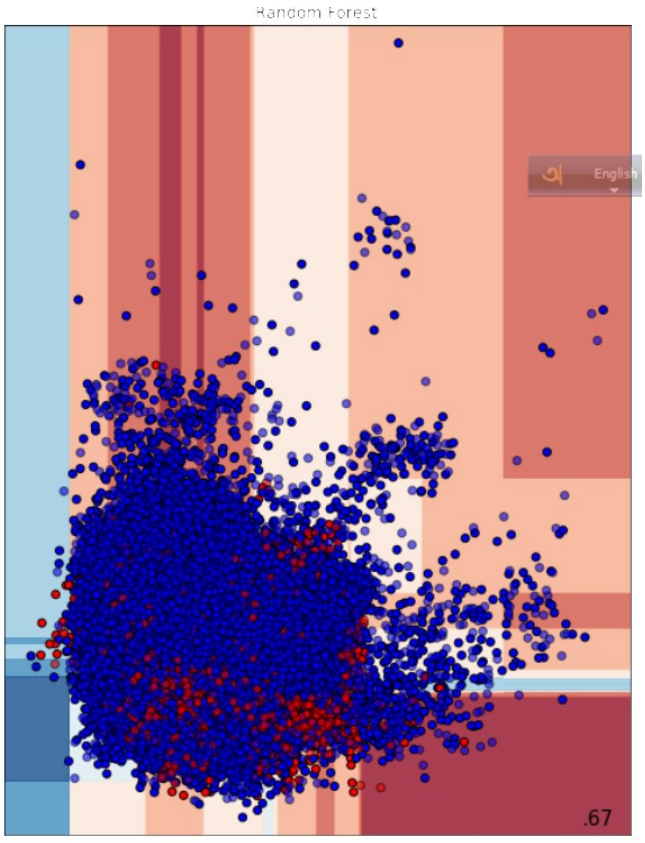

Figure 13: Random Forest Classifier

In the Random Forest method, the accuracy comes as 67%,

E. ***Confusion Matrix***

Accuracy Score : 0.9067392431311561
Classification Report :

	precision	recall	f1-score	support
0	1.00	0.94	0.97	2323
1	0.79	0.16	0.27	1230
2	0.57	0.27	0.36	1339
3	0.86	0.85	0.85	8033
4	0.92	0.99	0.96	25655
accuracy			0.91	38580
macro avg	0.83	0.64	0.68	38580
weighted avg	0.90	0.91	0.89	38580

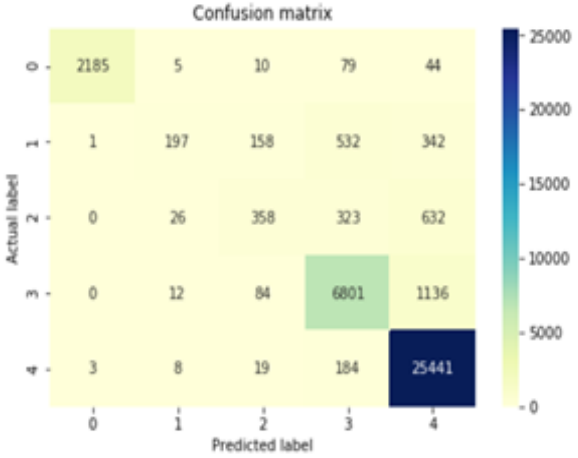

Figure 14: Confusion Matrix of KNN

Accuracy Score : 0.9841109383100052
Classification Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2323
1	0.96	0.97	0.96	1230
2	0.95	0.93	0.94	1339
3	0.98	0.97	0.97	8033
4	0.99	0.99	0.99	25655
accuracy			0.98	38580
macro avg	0.97	0.97	0.97	38580
weighted avg	0.98	0.98	0.98	38580

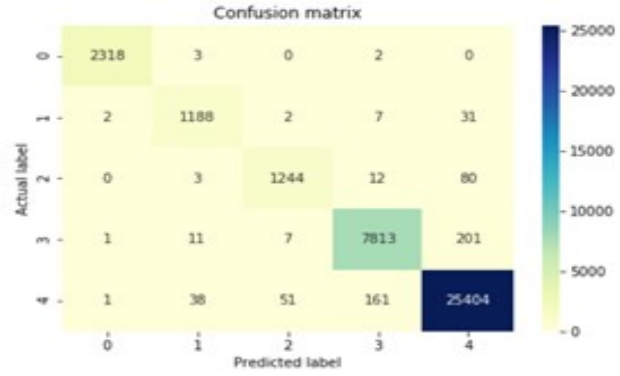

Figure 15: Confusion Matrix of Decision Tree

Accuracy Score : 0.8223950233281493
Classification Report :

	precision	recall	f1-score	support
0	0.96	0.84	0.89	2323
1	0.91	0.08	0.15	1230
2	1.00	0.09	0.16	1339
3	0.88	0.49	0.63	8033
4	0.80	1.00	0.89	25655
accuracy			0.82	38580
macro avg	0.91	0.50	0.55	38580
weighted avg	0.84	0.82	0.79	38580

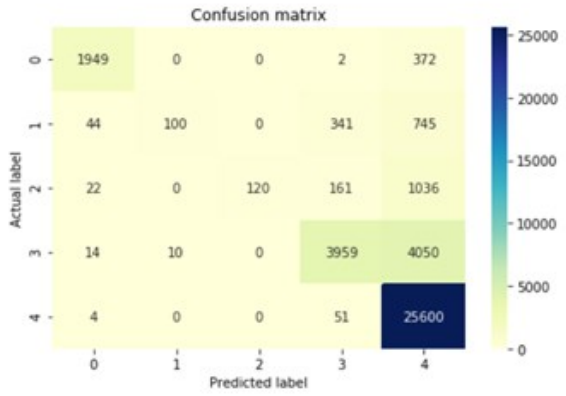

Figure 16: Confusion Matrix of Random Forest

F. ***Backward Elimination Output***

Variance Threshold is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples. With this technique, 5 features have been .found which can be reduced. Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. The two columns that are found most significant through this technique are "lastSensorEventSeconds, sensorElTime-Bedroom."

	coef	std err	t	P>|t|	[0.025	0.975]
const	6.402e-11	3.55e-12	18.035	0.000	5.71e-11	7.1e-11
x1	0.0515	0.017	3.028	0.002	0.018	0.085
x2	-2.103e-05	4.74e-06	-4.432	0.000	-3.03e-05	-1.17e-05
x3	-0.0165	0.003	-5.931	0.000	-0.022	-0.011
x4	-3.727e-06	1.46e-05	-0.255	0.799	-3.24e-05	2.49e-05
x5	-0.0002	7.95e-05	-2.937	0.003	-0.000	-7.76e-05
x6	0.0177	0.003	7.012	0.000	0.013	0.023
x7	0.0103	0.002	4.486	0.000	0.006	0.015
x8	-0.0238	0.002	-13.595	0.000	-0.027	-0.020
x9	-0.0238	0.002	-13.595	0.000	-0.027	-0.020
x10	-0.0261	0.004	-6.118	0.000	-0.034	-0.018
x11	0.4485	0.016	27.341	0.000	0.416	0.481
x12	0.0450	0.021	2.145	0.032	0.004	0.086
x13	0.0022	0.002	0.899	0.369	-0.003	0.007
x14	6.307e-14	8.49e-15	7.427	0.000	4.64e-14	7.97e-14
x15	-0.0228	0.004	-5.968	0.000	-0.030	-0.015
x16	-0.0272	0.003	-8.212	0.000	-0.034	-0.021
x17	-0.0905	0.003	-34.644	0.000	-0.096	-0.085
x18	0.0667	0.001	66.164	0.000	0.065	0.069
x19	3.908e-17	2.52e-18	15.507	0.000	3.41e-17	4.4e-17
x20	0.0066	0.001	5.101	0.000	0.004	0.009
x21	0.0442	0.001	40.071	0.000	0.042	0.046
x22	0.0027	0.001	2.399	0.016	0.000	0.005
x23	4.959e-17	5.55e-18	8.931	0.000	3.87e-17	6.05e-17
x24	-0.0009	0.005	-0.191	0.849	-0.011	0.009
x25	0.0214	0.001	17.513	0.000	0.019	0.024
x26	-3.671e-05	5.43e-06	-6.754	0.000	-4.74e-05	-2.61e-05
x27	6.864e-06	5.43e-06	1.265	0.206	-3.77e-06	1.75e-05
x28	4.84e-06	1.84e-07	26.352	0.000	4.48e-06	5.2e-06
x29	9.05e-06	1.23e-06	7.345	0.000	6.63e-06	1.15e-05
x30	5.531e-06	3.07e-07	18.035	0.000	4.93e-06	6.13e-06
x31	5.495e-06	1.96e-07	28.003	0.000	5.11e-06	5.88e-06
x32	-9.561e-05	9.06e-06	-10.552	0.000	-0.000	-7.79e-05
x33	7.301e-05	1.06e-05	6.898	0.000	5.23e-05	9.38e-05
x34	5.531e-06	3.07e-07	18.035	0.000	4.93e-06	6.13e-06
x35	-3.97e-06	1.19e-06	-3.322	0.001	-6.31e-06	-1.63e-06
x36	-1.495e-05	8.93e-07	-16.750	0.000	-1.67e-05	-1.32e-05

Omnibus:	1972.542	Durbin-Watson:	0.063
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8908.507
Skew:	0.167	Prob(JB):	0.00
Kurtosis:	5.698	Cond. No.	2.79e+19

Figure 17: Backward Elimination