ECS 330: EECS LABORATORY-II

PROJECT REPORT

Project Id: 17 - **SHIRSHAKK PURKAYASTHA**
Faculty Advisor: Dr. Kundan Kandhaway 18247

Github Repository: https://github.com/Shirshakk-P/Smartphone-based-Recognition-of-Human-activities

---

"Smartphone-based Recognition of Human Activities"

OBJECTIVE:

The objective of this project was to develop ML models with a broader aim to classify the different human activities with the help of data accumulated by smartphone sensors.

DATASET:

http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

DATASET INFORMATION:

Dataset Characteristics:

1) Multi-Variate Data 2) Time Series Data

The experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years.
They performed a protocol of activities composed of six basic activities:
>>3 Static Postures (standing,sitting,lying).
>>3 Dynamic Activities (walking,walking downstairs,walking upstairs).

The experiment also included postural transitions that occurred between the static postures:

stand-to-sit	*sit-to-stand*
sit-to-lie	*lie-to-sit*
stand-to-lie	*lie-to-stand.*

All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution.
Data captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using the embedded accelerometer and gyroscope of the device. The experiments were video-recorded to label the data manually.

The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window).
The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity.
The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of 561 features was obtained by calculating variables from the time and frequency domain.

Attribute Information:

The dataset is divided in two parts-

1. **Inertial Sensor Data:**
- Raw triaxial signals from the accelerometer and gyroscope of all the trials with participants.
- The labels of all the performed activities.

2. **Records of activity windows**:
- A 561-feature vector with time and frequency domain variables.
- Its associated activity label.
- An identifier of the subject who carried out the experiment.

Activity Labels:

1. WALKING	2. WALKING_UPSTAIRS
3. WALKING_DOWNSTAIRS	4. SITTING
5. STANDING	6. LAYING
7. STAND_TO_SIT	8. SIT_TO_STAND
9. SIT_TO_LIE	10. LIE_TO_SIT
11. STAND_TO_LIE	12. LIE_TO_STAND

- Features are normalized and bounded within [-1,1].
- The units used for the accelerations (total and body) are 'g'(acceleration due to gravity-> 9.80665 m/sec2).
- The gyroscope units are rad/sec.

LEARNING MODELS:

This project implemented three Machine Learning Models to classify the Human Activity Recognition Dataset, viz:

1. Logistic Regression (LR)
2. K-Nearest Neighbours (KNN)
3. Support Vector Machines (SVM)

COMPARISON METRIC:

The f1-score is chosen as the comparison metric for the three trained models.

CODE SNIPPETS:

```
#Visualization Libraries:
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.graph_objects as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_dark"
sns.set(style="darkgrid")
```

We import the data visualization libraries: Matplotlib and Seaborn ; which would become the backbone of our data interpretation after the classification task.

```
[ ] #Scikit-model requisites:
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import cross_val_score
    from sklearn.metrics import confusion_matrix
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.svm import SVC
    from sklearn.metrics import f1_score
```

We import the necessary scikit-learn libraries to build the classification models and f1_score as the comparision metric for the models.

```
# Get Features from the Dataset:
with open("/content/HAPT Dataset/features.txt", "r") as _:
  features = [x.strip().replace('()', '').replace(',', '').split(' ')[-1] for x in _.readlines()]
print("Number of Features:", len(features))
print("Features:\n",features[:5])
```

```
Number of Features: 561
Features:
 ['tBodyAcc-Mean-1', 'tBodyAcc-Mean-2', 'tBodyAcc-Mean-3', 'tBodyAcc-STD-1', 'tBodyAcc-STD-2']
```

Feature Extraction is done on the dataset by replacing “()” and ”,” with *blank_spaces* in the Features.txt file and then split the dataset at *blank_space* encounters.

```
[ ] #Participant IDs:
    with open('/content/HAPT Dataset/Train/subject_id_train.txt', 'r') as _:
      train_id = pd.Series([int(x.strip()) for x in _.readlines()])
```

```
#Activity Labels:
with open('/content/HAPT Dataset/Train/y_train.txt', 'r') as _:
  y_train = pd.Series([int(x.strip()) for x in _.readlines()])
y_train.value_counts().sort_index()
```

We perform Labels and Feature Extraction operations on the Train Dataset.

```
[ ] #Participant IDs:
    with open('/content/HAPT Dataset/Test/subject_id_test.txt', 'r') as _:
      test_id = pd.Series([int(x.strip()) for x in _.readlines()])
```

```
#Activity Labels:
with open('/content/HAPT Dataset/Test/y_test.txt', 'r') as _:
  y_test = pd.Series([int(x.strip()) for x in _.readlines()])

y_test.value_counts().sort_index()
```

Labels and Feature Extraction operation performed on the Test Dataset.

LOGISTIC REGRESSION

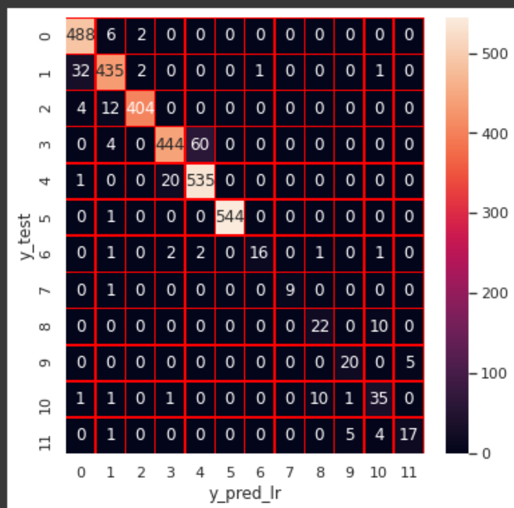
```
[ ] LR = LogisticRegression()
```

```
[ ] #K-fold CV
    "K-value is substiuited by 10, 20 and 30 after a complete run of the code"
    accuraccies = cross_val_score(estimator = LR, X= X_train, y=y_train, cv=30)
    print("Average Accuracies: ",100*np.mean(accuraccies))
    print("Standard Deviation Accuracies: ",100*np.std(accuraccies))

    Average Accuracies:  95.82921792224118
    Standard Deviation Accuracies:  2.8164460759546666
```

Logistic Regression model is built with k-fold cross_validation imbibed within it to restrict the error limits.

The k-fold values are gradually increased from 10 to 30 with a step-size of 10 and their respective Average_Accuracies are measured.

```python
#Confusion Matrix:
y_pred_lr= LR.predict(X_test)
CM = confusion_matrix(y_test,y_pred_lr)

f, ax = plt.subplots(figsize=(6,6))
sns.heatmap(CM,annot = True, linewidths=0.5,linecolor="red",fmt=".0f",ax=ax)
plt.xlabel("y_pred_lr")
plt.ylabel("y_test")
plt.show()
```



Displaying the Confusion_Matrix for the LR-classification model in the heat_map format.

K-NEAREST NEIGHBOURS

```python
KNN = KNeighborsClassifier(n_neighbors = 30) #"K-value is substiuited by 10, 20 and 30 after a complete run of the code"
```

```python
#K-fold CV
"K-value is substiuited by 10, 20 and 30 after a complete run of the code"
accuraccies = cross_val_score(estimator = KNN, X= X_train, y=y_train, cv=30)
print("Average Accuracies: ",100*np.mean(accuraccies))
print("Standard Deviation Accuracies: ",100*np.std(accuraccies))
```

```
Average Accuracies:  91.91573633434099
Standard Deviation Accuracies:  3.896617316349285
```

k-Nearest Neighbours classification model is built with cluster_sizes of (10,20,30) and k-fold cross_validation imbibed.

Algorithm 3:

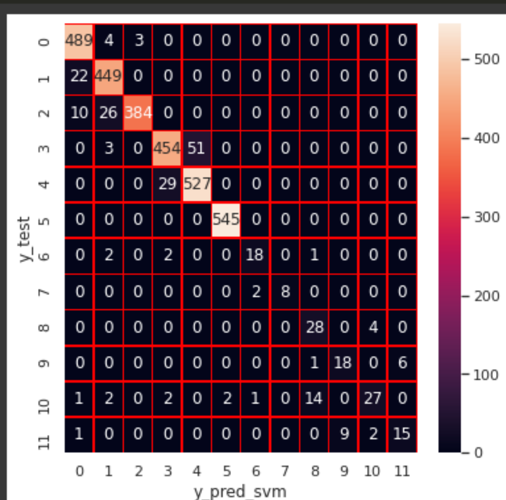
SUPPORT VECTOR MACHINES

```
[ ] SVM = SVC(random_state=42)
```

```
[ ] #K-fold CV
    "K-value is substiuited by 10, 20 and 30 after a complete run of the code"
    accuraccies = cross_val_score(estimator = SVM, X= X_train, y=y_train, cv=30)
    print("Average Accuracies: ",100*np.mean(accuraccies))
    print("Standart Deviation Accuracies: ",100*np.std(accuraccies))

    Average Accuracies:  94.77352768050442
    Standart Deviation Accuracies:  4.173446610891643
```

Support Vector Machine classification model is built with k-fold cross_validation imbibed with k-values of (10,20,30).

```
#Confusion Matrix:
y_pred_svm= SVM.predict(X_test)
CM = confusion_matrix(y_test,y_pred_svm)

f, ax = plt.subplots(figsize=(6,6))
sns.heatmap(CM,annot = True, linewidths=0.5,linecolor="red",fmt=".0f",ax=ax)
plt.xlabel("y_pred_svm")
plt.ylabel("y_test")
plt.show()
```



Displaying the Confusion_Matrix for the SVM classification models in the heat_map format.

Model Comparisions:

Models are compared on the basis of their **f1-score**.

```
#Model Comaparision:
data = [['Logistic Regression',f1_score(y_test, y_pred_lr, average='weighted')],
        ['K-Nearest Neighbors Algorithm (K-NN)',f1_score(y_test, y_pred_knn, average='weighted')],
        ['Support Vector Machines (SVM)',f1_score(y_test, y_pred_svm, average='weighted')]]

F1_Score = pd.DataFrame(data, columns = ['Ml Algorithms', 'f1_score'])
F1_Score.sort_values(by = 'f1_score', ascending=0)
```

	Ml Algorithms	f1_score
0	Logistic Regression	0.938662
2	Support Vector Machines (SVM)	0.936055
1	K-Nearest Neighbors Algorithm (K-NN)	0.885910

The three classification models are compared with the f1-score as the comparision_metric and the results are plotted.

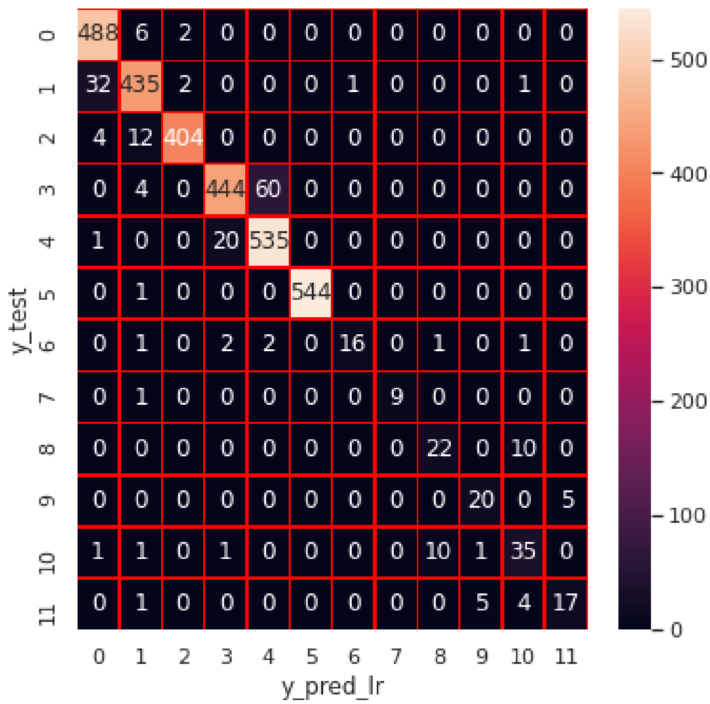
Maximum accuracy is achieved by the Logistic_Regression model with k-fold cross_validation value of 30.

ML MODEL 1: LOGISTIC REGRESSION

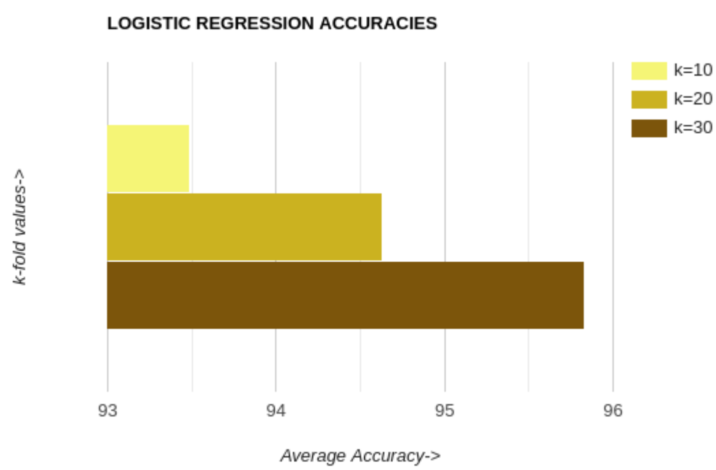
We trained the dataset on a *Logistic Regression* model with k-fold cross-validation values of (10,20,30). The performance of the LR Model on the corresponding k-fold values is given by:

k-fold Value	AVERAGE ACCURACY	LR SCORE
k = 10	93.4862	93.8962
k = 20	94.6312	93.8962
k = 30	95.8291	93.8962

CONFUSION MATRIX:



GRAPHICAL COMPARISON:



OBSERVATION:

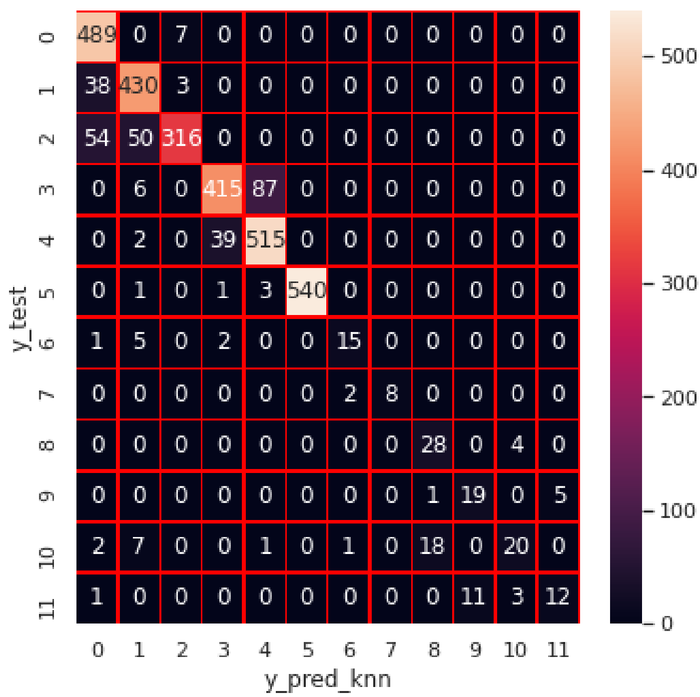
A LR Model with k-fold value of 30 gives the maximum average accuracy.

ML MODEL 2: k-NEAREST NEIGHBOURS

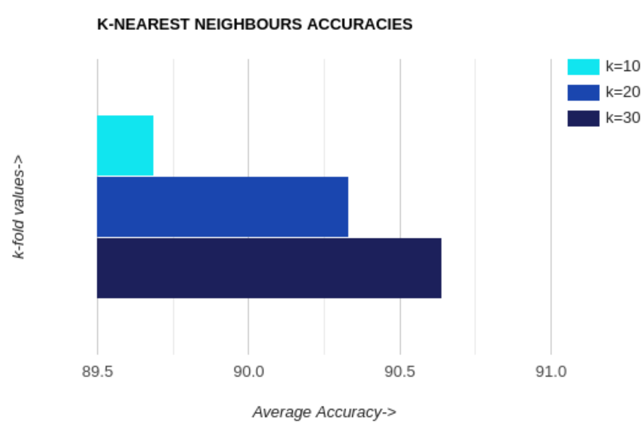
We trained the dataset on a *k-Nearest Neighbours* model with k-fold cross-validation values of (10,20,30). Also cluster-sizes are chosen as (10,20,30). The performance of the kNN Model on the corresponding k-fold values is given by:

k-fold Value	Cluster Size	AVERAGE ACCURACY	kNN SCORE
k = 10	10	**89.2717**	88.7729
k = 20	20	**90.3313**	88.7729
k = 30	30	**90.6408**	88.7729

CONFUSION MATRIX:



GRAPHICAL COMPARISON:



OBSERVATION:

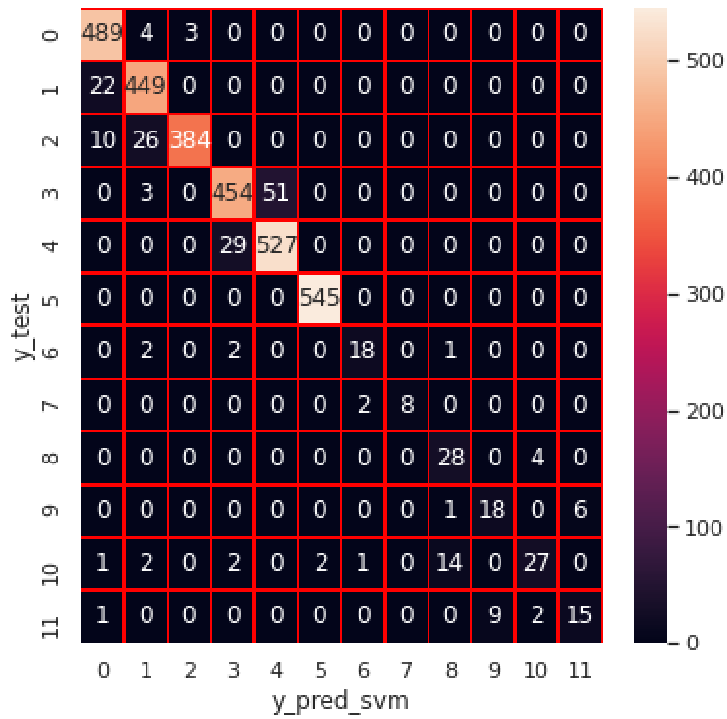
A Cluster size of 30 with a k-fold value of 30 gives the maximum average accuracy.

ML MODEL 3: SUPPORT VECTOR MACHINES

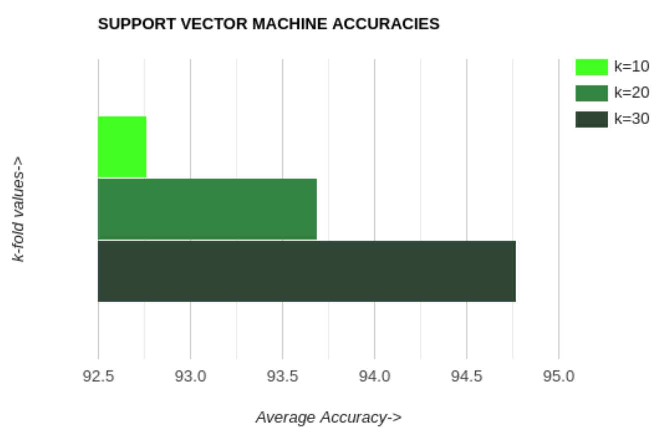
We trained the dataset on a *Support Vector Machine* model with k-fold cross-validation values of (10,20,30). The performance of the SVM Model on the corresponding k-fold values is given by:

k-fold Value	AVERAGE ACCURACY	SVM SCORE
k = 10	92.7650	93.6748
k = 20	93.6909	93.6748
k = 30	94.7735	93.6748

CONFUSION MATRIX:



GRAPHICAL COMPARISON:



OBSERVATION:

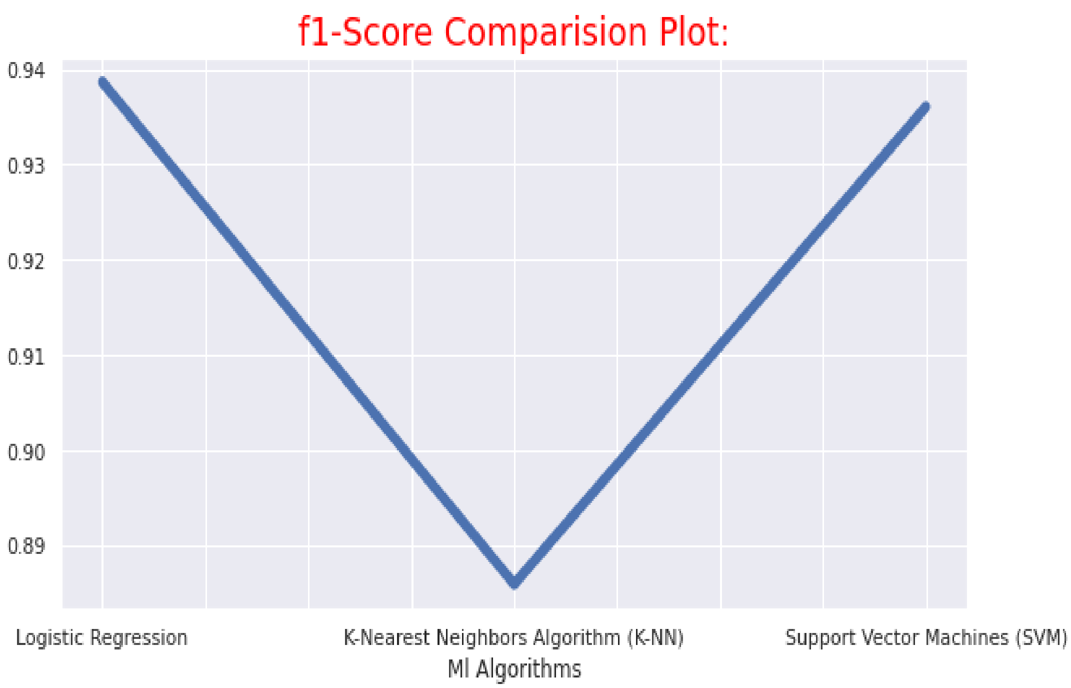
A SVM Model with k-fold value of 30 gives the maximum average accuracy.

f1-SCORES:

The corresponding f1-Scores of the three trained models is given by:

Serial No.	ML MODEL	f1-SCORE:
1.	LOGISTIC REGRESSION	0.938662
2.	k-NEAREST NEIGHBOURS	0.885910
3.	SUPPORT VECTOR MACHINES	0.936055

f1-SCORE PLOT:



CONCLUSIONS:

The highest classification accuracy is achieved by the LOGISTIC REGRESSION Model (k-fold value of 30) and the lowest by the k-NEAREST NEIGHBOURS Model (k-fold value of 10) in our predictions.

REFERENCES:

1. Human Activity Recognition on Smartphones With Awareness of Basic Activities and Postural Transitions.(Jorge-Luis Reyes-Ortiz, Luca Oneto, Alessandro Ghio, Albert SamÃ¡, Davide Anguita and Xavier Parra).
Artificial Neural Networks and Machine Learning â€" ICANN 2014. Lecture Notes in Computer Science. Springer. 2014.

2. A Public Domain Dataset for Human Activity Recognition Using Smartphones.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz).
21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

3. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz).
Journal of Universal Computer Science. Special Issue in Ambient Assisted Living: Home Care. Volume 19, Issue 9. May 2013

4. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.(Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz).
4th International Workshop of Ambient Assited Living, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012.
Proceedings. Lecture Notes in Computer Science 2012, pp 216-223.

5. Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments. (Jorge Luis Reyes-Ortiz, Alessandro Ghio, Xavier Parra-Llanas, Davide Anguita, Joan Cabestany, Andreu CatalÃ).
21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

*****