

Human Activity Recognition from Machine Learning Techniques

Kommineni Manideep Chowdary¹, Mannava Leela Sekhar Chowdary², Kathi Edwin Sahil Samuel³, Muppalla Lokesh⁴

¹⁻⁴Department of Computer Science and Engineering, KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCE

Abstract—Human Activity Recognition (HAR) is considered a valuable research topic in the last few decades. Different types of machine learning models are used for this purpose, and this is a part of analyzing human behavior through machines. It is not a trivial task to analyze the data from wearable sensors for complex and high dimensions. Nowadays, researchers mostly use smartphones or smart home sensors to capture these data. In our paper, we analyze these data using machine learning models to recognize human activities, which are now widely used for many purposes such as physical and mental health monitoring. We apply different machine learning models and compare performances. We use Logistic Regression (LR) as the benchmark model for its simplicity and excellent performance on a dataset, and to compare, we take Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Network (ANN). Additionally, we select the best set of parameters for each model by grid search. We use the HAR dataset from the UCI Machine Learning Repository as a standard dataset to train and test the models. Throughout the analysis, we can see that the Support Vector Machine performed (average accuracy 96.33%) far better than the other methods. We also prove that the results are statistically significant by employing statistical significance test methods.

Keywords—Machine Learning, Human Activity Recognition, Smartphone Sensor, Logistic Regression, Artificial Neural Network, Grid Search, Statistical Significance Test.

I. INTRODUCTION

Human beings can realize others' psychological state and personality by observing their daily activities. Following this pattern, the researchers want to predict human behavior using machines, and Human Behavior Recognition (HAR) comes as an active research topic. This has become one of the important research topics in machine learning and computer vision. Though motion data collection was hard in previous days, current technological developments help researchers capture the data as they can now use portable devices, including smartphones, music players, smartwatches, or smart home sensors. Especially, motion sensor embedded smartphones, like accelerometers, gyroscope, etc. bring a new era for activity recognition. Researchers use various machine learning and deep learning techniques including Naive Bayes (NB) [1], Decision Tree [2], Support Vector Machine (SVM) [3] [4], Nearest Neighbor (NN) [5], Hidden Markov Model (HMM) [6] Convolutional Neural Network (CNN) [7] etc. to analyze the sensor data and recognize human activity.

HAR is the problem of classifying day-to-day human activity using data collected from smartphone sensors. Data are continuously generated from the accelerometer and gyroscope, and these data are instrumental in predicting our activities such as walking or standing. There are lots of datasets and ongoing research on this topic. In [8], the authors discuss wearable sensor data and related works of predictions with machine learning techniques. Wearable devices can predict an extensive range of activities using data from various sensors. Deep Learning models are also being used to predict various human activities [9]. Nowadays, people use smartphones almost all the time and use many wearable devices. Through these devices, physical and mental health can be monitored by predicting human activity without specialized and costly medical equipment, and nowadays, it is an efficient, cheap, and safe way to do this as the COVID-19 (Coronavirus disease 2019) pandemic is ongoing.

In this paper, we have selected a dataset from the UCI machine learning repository [10] to calculate the accuracy of five machine learning models and to perform some statistical significance tests. We use the Decision Tree (DT), Random Forest (RF), LR, SVM, and Artificial Neural Network (ANN) with a hidden layer to predict human activity from mobile data. In the dataset, the human activities are classified into six categories: walking, walking upstairs, walking downstairs, sitting, standing, and laying, and the data were collected from two different smartphone sensors, which are accelerometer and gyroscope. We ran experiments to see the performance of the models based on classification results and tried to get the highest possible accuracy from these models. Here, we tried to get the highest possible performance by these algorithms by parameter tuning, cross-validation, and finally, comparing the result with two statistical significance tests to get the winner algorithm.

The remainder of this paper is organized as follows. Section II describes the related works, and section III discusses the dataset. Section IV discusses the machine learning models used in this paper, section V describes the experimental methods, and section VI presents the experimental results. Eventually, section VII concludes the paper.

II. RELATED WORKS

Since HAR becomes an important research topic, the researchers use different Machine Learning methods, such as Naive Bayes, Logistic Regression, Decision Trees, Support Vector Machines, Nearest Neighbor, Hidden Markov Model, CNN, RNN (Recurrent Neural Network), etc. for recognizing human activities. D. Singh et al. used RNN in their work for human activity recognition. They use smart home sensors to collect data and applied a Long Short Term Memory (LSTM) RNN. ANN is used for prototyping data acquisition module [11]. A Anjum et al. [12] use smartphone sensors over detectable physical activities, including walking, running, climbing, cycling, driving, etc. They compare Naive Bayes, Decision Tree, KNN (K Nearest Neighbours), and SVM to calculate accuracy. They achieve a greater than 95% true positive rate and less than 1.5% false positives rate.

Z Chen et al. [13] propose a robust HAR system based on coordinate transformation and PCA (CT-Principal Component Analysis) and online SVM. CT-PCA scheme is used to reduce the effect of orientation variations. Their presented OSVM is independent, which uses a tiny portion of data from the unseen placement. Sun et al. [14] propose a HAR approach with varying position and orientation through smartphone accelerometer data. They use their acceleration magnitude as the fourth data dimension. They also use generic SVM and location-specific SVM in the model.

Deep Learning approaches require a considerable amount of training data; therefore, many researchers mainly use generic machine learning approaches [15]. Hence, in our paper, we focus on the generic approaches to analyze and detect human activity.

III. DATASET

A. Data Exploration and Exploratory Visualization

The dataset is taken from UCI machine learning repository [10]. The dataset contains the information from 30 volunteers within the age range: 19-48. Each volunteer performs six activities:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Laying

Hence, the dataset has six labels to predict. The dataset consists of 561 feature vectors with time and frequency domain variables. These features come from the following data:

- Gravitational acceleration for x, y, and z axes
- Body acceleration data files for x, y, and z axes
- Body gyroscope data files for x, y, and z axes

We also have the ID of an individual volunteer for each record. A total of 10299 records and training and test dataset splits are 70% and 30%. The 6 classes are converted to numerical values sequentially: [1, 2, 3, 4, 5, 6].

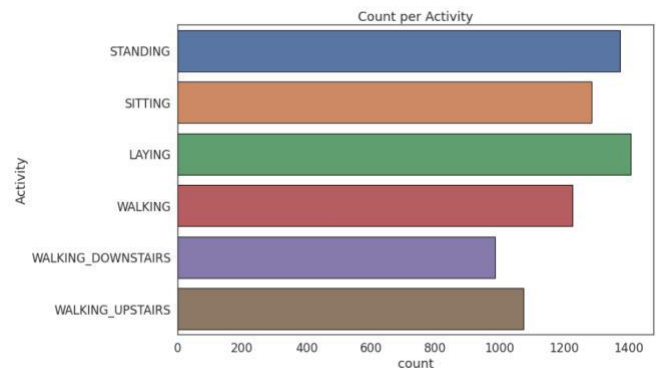


Fig. 1: Different human activities with their count in the training dataset.

Figure 1 shows the total number of different human activities in training data, and figure 2 shows the distribution of two types of activity (static: sitting, standing, and laying – dynamic: walking, walking downstairs, and _walking upstairs) using tBodyAccMag-mean() feature, which is taken from the accelerometer of smartphones. It is clear that the two types of activities are easily separable. There are no missing and duplicate values.

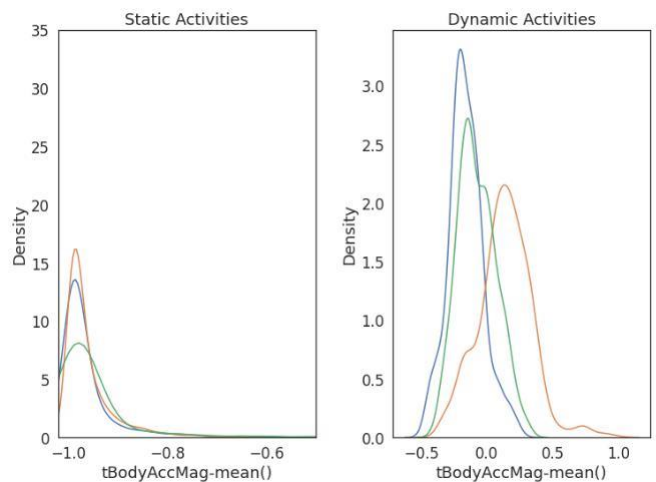


Fig. 2: Static and dynamic activity pattern

IV. MACHINE LEARNING MODELS

A. Support Vector Machine

Support Vector Machine (SVM) is used as one of the selected algorithms. SVM is a discriminative classifier determined by a separating hyperplane. If training data is fed to the SVM, it gives an optimal hyperplane that classifies new data. This hyperplane is simply a line dividing a plane into two parts in two-dimensional space wherein each class lay on either side [16]. There are SVM variants, and one is Support Vector Classifier (SVC) for more than two classes. In this paper, we use SVC with 'linear,' 'rbf' [17] and 'sigmoid' kernel. SVM usually provides good accuracy when the number of features

is large. In our dataset, the number of features is large; hence we selected this algorithm.

TABLE I: Parameters for DT

Parameters	Values	Description
Criterion	[Gini, Entropy]	Measure the quality of the data split. Gini - Gini impurity and Entropy - information gain.
Max Depth	[0, 2, 4, 6, 8]	The maximum depth of the tree.

B. Decision Tree

A Decision tree is a tree-like formation of decisions and possibilities. It differentiates instances by sorting from root to leaves. Here, we check impurity using Gini and gain information from entropy. These measures the quality of the split. We use various max depths (0 to 8), which is the maximum depth of the tree.

TABLE II: Parameters for RF

Parameters	Values	Description
Criterion	[Gini, Entropy]	Measure the quality of the data split Gini - Gini impurity and Entropy - information gain
N Estimators	[10, 30, 60, 90, 120, 150]	Number of trees in the generated forest
Max Depth	[0, 2, 4, 6, 8]	The maximum depth of the tree

TABLE III: Parameters for SVM

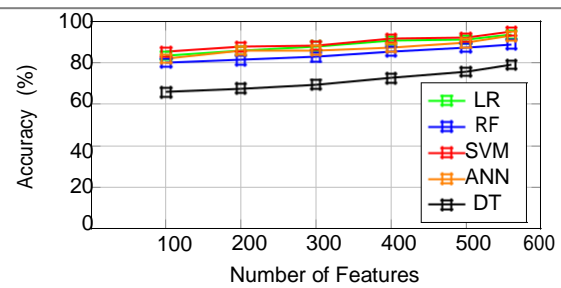
Parameters	Values	Description
Kernel	[Linear, RBF, Sigmoid]	The type of kernel to be used in the model
C	[0.1, 0.5, 1, 2, 5, 10, 100]	Penalty parameter C of the error term.

TABLE IV: Parameters for ANN

Parameters	Values	Description
Hidden layer sizes	[(10,), (50,), (100,)]	Number of hidden nodes
Alpha	[1e-4, 1e-3, 1e-2]	L2 regularization parameter
Learning rate	[1e-3, 1e-2, 1e-1]	Step size
Beta1	[0.1, 0.5, 0.9]	Adam specific parameter
Beta2	[0.1, 0.5, 0.9]	Adam specific parameter

C. Random Forest

Random Forest (RF) model is an ensemble of some decision trees. It combines the learning methods and increases the overall result. Here criterion measures the quality of a split. We use Gini for impurity checks, and the information is gained from entropy. We use N estimators and max depth as the parameter. N estimators are the number of trees in the generated forest, and max depth is the tree's maximum depth.


Fig. 3: Increasing accuracy of the five algorithms by increasing features.

D. Artificial Neural Network

An artificial Neural Network having a single hidden layer and the sigmoid activation function is used for our experiment. Adam, a stochastic gradient-based optimizer [18], is used for weight optimization. The maximum epoch is 200, and the dataset is shuffled between each epoch. The ANN is selected because it is currently used in many learning problems.

E. Benchmark Model

We used Logistic Regression (LR) as a benchmark model, and it focuses to maximize the probability of the data. This is a basic model without much complexity like neural networks. The performance of LR improves if the data lies on the correct side of the separating hyperplane. We used the kernel trick and 'lbfgs' optimizer to compare the performance. Some previous research has some good results with that combination of LR [19]. Hence, we used this algorithm because LR, and its variants are the right choice for benchmarking.

V. EXPERIMENTAL METHODS

A. Data preprocessing

Dataset has a train and test portion. There are no duplicate and missing values. Train and Test data are stored in the data frame initially. We tried to remove some features to reduce the feature space. We tried to select features by computing ANOVA F-value [20] for the dataset. We selected top 100, 200, 400, and 500 features with top ANOVA F-values. Nevertheless, each time we increased the features, the accuracy increased significantly for all five algorithms and the trend is depicted in figure 3. Therefore, we use all the features for training and testing. The learning is divided into two parts. Initially, the best parameters for all the algorithms are learned. Then, in the second step, the algorithms are compared. The dataset is evaluated by a K-fold cross-validation set, and we used k=5 for cross-validation. The data splitting is done

TABLE V: Parameters for LR

Parameters	Values	Description
Regularizer	[L1, L2]	Specifies the type of regularization
C	[0.1, 0.5, 1, 2, 5, 10, 100]	Penalty parameter C of the error term.

TABLE VI: Best parameters for the selected models

Models	Parameters
DT	Criterion: Gini, Max Depth: 4
RF	Criterion: Gini, N Estimators: 90, Max Depth: 6
SVM	C: 1.0, Kernel: Linear
ANN	Alpha: 0.01, Beta1: 0.9, Beta2: 0.9, Hidden layer sizes: (50,), Learning rate: 0.001
LR	C: 2, Regularizer: L1

TABLE VII: Accuracies for all algorithms

Model	Average Accuracy
DT	80.3%
RF	90.05%
SVM	96.33%
ANN	93.8%
LR	96.01%

by stratified sampling. We use 70% training data for cross-validation purposes without touching the test data.

B. Implementation

A cross-validation dataset is used for an exhaustive grid search to find the best hyperparameters. The models learn from the dataset for all combinations of the listed parameters. Table I (for DT), Table II (for RF), Table III (for SVM), Table IV (for ANN), and Table V (for LR) have the parameter list that we use to get the best combination of parameters. The parameters are chosen from a range of values that work well in many problems. Accuracy is used as the scoring method. We use the python Scikit-learn library for the implementation. A detailed explanation of all parameters can be found in [21] [22] [23] [24] [25].

C. Refinement

After finding the best parameters, we evaluated the five algorithms on test data. Test data is divided into 10 random sets, and every set consists of 50% of the data. Hence, we have 10 runs for every algorithm and calculated the mean and variance for these runs. The learning process to find the best parameters used only training data. Hence, we got an unbiased estimation for cross algorithm comparisons.

The best parameters for the five algorithms are found after an exhaustive grid search. All combination of the given parameters is used to find the best parameters After 5-fold cross-validation (stratified), we got the best parameters in table VI.

TABLE VIII: Average scores per class for DT

Class	Precision	Recall	F1-score
1	0.76	0.86	0.81
2	0.86	0.79	0.84
3	0.93	0.88	0.91
4	0.85	0.88	0.87
5	0.87	0.83	0.85
6	0.83	0.84	0.84

TABLE IX: Average scores per class for RF

Class	Precision	Recall	F1-score
1	0.83	0.96	0.89
2	0.93	0.84	0.88
3	0.98	0.96	0.97
4	0.92	0.97	0.94
5	0.94	0.91	0.93
6	0.91	0.92	0.92

TABLE X: Average scores per class for ANN

Class	Precision	Recall	F1-score
1	0.86	0.97	0.91
2	0.96	0.86	0.91
3	1.00	0.97	0.99
4	0.92	0.99	0.95
5	0.98	0.94	0.96
6	0.95	0.95	0.93

D. Refinement with Statistical Significance Test

We used hypothesis testing for this paper. A hypothesis is examined by estimating and inspecting a random sample of the population selected to be analyzed. We examined specific hypotheses using a random population pattern: one is the null and another one is the alternative hypothesis. The null hypothesis related to the equality between population parameters, and as an example, a null hypothesis assumes that the population mean return is the same to zero. Alternatively, the alternative hypothesis is efficaciously the reverse of a null hypothesis, and as an instance, the population mean return is not identical to zero. Therefore, they can not happen simultaneously, and only one hypothesis can be correct. We tested all hypotheses using the following four-step process:

At first, the we have to state the two hypotheses so that only one can be accurate. Then, there is a plan to outline how the data are deter-mined. After that, the plan is carried out, and the sample data are analyzed.

TABLE XI: Average scores per class for SVM

Class	Precision	Recall	F1-score
1	0.90	0.97	0.94
2	0.96	0.89	0.92
3	1.00	1.00	1.00
4	0.95	1.00	0.97
5	0.99	0.98	0.98
6	0.98	0.95	0.97

TABLE XII: Average scores per class for LR

Class	Precision	Recall	F1-score
1	0.89	0.98	0.93
2	0.98	0.86	0.91
3	1.00	1.00	1.00
4	0.95	1.00	0.97
5	1.00	0.97	0.99
6	0.97	0.95	0.96

TABLE XIII: Welch's t-test values for the top three models

Models	SVM(t-value)	SVM(p-value)	ANN(t-value)	ANN(p-value)	LR(t-value)	LR(p-value)
SVM (t-value j p-value)	0	1	7.44	6.71e-07	2.31	.0324
ANN (t-value j p-value)	-7.44	6.71e-07	0	1	-5.88	1.42e-05
LR (t-value j p-value)	-2.31	.0324	5.88	1.42e-05	0	1

TABLE XIV: 5 by 2-fold Cross Validation t paired test

Models	SVM(t-value)	SVM(p-value)	ANN(t-value)	ANN(p-value)	LR(t-value)	LR(p-value)
SVM (t-value j p-value)	0	1	2.72	4.1e-02	3.38	.0195
ANN (t-value j p-value)	-4.65	5.5e-03	0	1	-6.11	1.7e-03
LR (t-value j p-value)	-3.81	.0124	2.73	4.0e-02	0	1

Finally, the result is analyzed and take a decision whether to reject the null hypothesis, or state that the null hypothesis can be possible with the given data [26].

method because the algorithms' variances are different. We can find output from the t-test that shows whether the algorithms have significant changes in their performance measurements.

The t-test show the importance of the differences between groups. It can explain whether the differences measured in means, could have happened by chance. The two-tailed t-test is done pairwise for every algorithm with $H_0: \mu_0 = \mu_1$ and significance level $\alpha = 0:05$ [28] to produce a ranking between the algorithms and for finding the winner algorithm. Here, H_0 is the null hypothesis, and μ_0 and μ_1 are the means of two groups of population. Also, the algorithms are compared using

5 times 2-fold cross-validated paired t-test [29] as suggested by the researchers in [30].

VI. EXPERIMENTAL RESULTS

We used 5-fold stratified cross-validation and grid search to find the best hyperparameters. Then run the models with the best parameters on test data and finally run the statistical significance test.

A. Evaluation Metrics

Accuracy is calculated by dividing the number of correct predictions by the total number of predictions. This metric is fundamental as we want to know whether a human is walking or sitting correctly. Accuracy is the primary metric to distinguish between our models. In equation 1, we get an accuracy score using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Additionally, we use Precision, Recall, and F1 score as the evaluation metrics. The equation of these three metrics is given in equations 2, 3, and 4.

$$P \text{recision} = \frac{TP}{TP + FP} \quad (2)$$

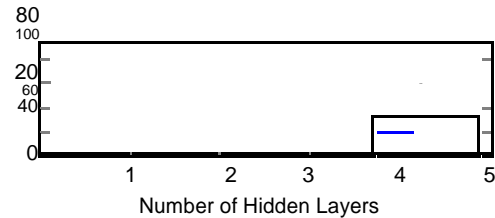


Fig. 4: Decreasing accuracy of ANN by increasing the number of hidden layers.

$$F1 = \frac{2 \cdot P \text{recision} \cdot \text{Recall}}{P \text{recision} + \text{Recall}} \quad (4)$$

B. Model Evaluation & Validation

Using the derived best parameters, the best model for each algorithm is formulated. Then the models are evaluated on the test data. Test data is divided into 5 random folds (stratified), and each fold contains the whole dataset. We got the average accuracy for the five different models using the data.

In Table VII, we can see the accuracies for the algorithms. Decision Tree and Random forest have low accuracy compared

to the other five algorithms. ANN performs well but has less accuracies compare to the other two algorithms. We can see the average precision, recall, and F1 score per class from tables VIII, IX, X, XI, and XII. There are 6 classes, and all scores per class are in the tables. We also understand that SVM and LR perform better than DT, RF, and ANN by inspecting the results from the tables mentioned above. We also tried multiple hidden layers with more hidden nodes, but the accuracy dropped and the phenomenon is depicted in figure 4.

Hence, we did not include that model for comparison. We used the kernel trick implementation for logistic Regression, but the accuracy is not significantly sufficient, and it takes more time than simple LR. Therefore, for simplicity, we did not include kernels in the LR. Both SVM and LR have closer accuracy and precision-recall scores per class. Nevertheless, we cannot decide the best model without a statistical significance test.

C. Statistics for Justification

$$\text{Recall} = \frac{TP}{TP + FN}$$

- (3) Test data is divided into 10 random folds (stratified), and each fold contains 50% of the data. Then we run the top three

best models with the best parameters. After getting the result from 10 runs for each of the three models, we ran a two-tailed Welch's t-test on the results from the top three models. We got the result for the three models and shown in table XIII. From Table XIII, we understand that SVM is the winner algorithm. Our hypothesis $H_0: \mu_0 = \mu_1$ has $\alpha = 0.05$ and from this table we see that every p-value is less than alpha. Therefore, we can reject the hypothesis. So, we can conclude that the results are significantly different. Hence, SVM is the best algorithm here. We also tested with a 5 by 2-fold cross-validation t paired test to confirm the winner algorithm. In several research papers [31] [32] [33], we found this as a recommended test. In this test, the whole dataset (training and test data) is divided into two random folds and run 5 times. Then the results are collected, and the t statistic is calculated. In table XIV, the statistic is shown. It is clearly visible that the hypothesis is rejected every time, and the results are significantly different according to the test. Table 9 shows that the t-value j p-value pair is different for the same pair if the order is different. Because 5 times 2-fold cross-validation makes the outcome different. We can conclude that the SVM algorithm is the winner for this dataset.

VII. CONCLUSION

In conclusion, it is evident that SVM performs better than the other two algorithms, though LR is the closer in case of accuracy. ANN cannot compete with SVM for this dataset. In most of the classification problems, ANN and deep ANN performs well. For this dataset, deep neural network architectures or convolutional neural network might increase the performance. We think we will explore the path in the near future. We also want to explore the performance of other machine learning algorithms in the future. In this experiment, we understand that simple machine learning algorithms can do well with proper parameter tuning, and we can determine the significance of the result by the statistical testing.

REFERENCES

- [1] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in International conference on pervasive computing. Springer, 2004, pp. 1–17.
- [2] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in International conference on pervasive computing. Springer, 2004, pp. 158–175.
- [3] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware activity recognition and anomaly detection in video," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 1, pp. 91–101, 2012.
- [4] K. M. Chathuramali and R. Rodrigo, "Faster human activity recognition with svm," in International Conference on Advances in ICT for Emerging Regions (ICTer2012). IEEE, 2012, pp. 197–203.
- [5] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," Journal of medical Internet research, vol. 14, no. 5, p. e130, 2012.
- [6] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 838–845.
- [7] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human activity recognition with convolutional neural networks," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2018, pp. 541–552.
- [8] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," IEEE communications surveys & tutorials, vol. 15, no. 3, pp. 1192–1209, 2012.
- [9] T. Plotz and Y. Guan, "Deep learning for human activity recognition in mobile computing," Computer, vol. 51, no. 5, pp. 50–59, 2018.
- [10] "Mlrepo." [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/humanactivityrecognitionusingsmartphones>
- [11] S. Oniga and J. Sut'o, "Human activity recognition using neural networks," in Proceedings of the 2014 15th International Carpathian Control Conference (ICCC). IEEE, 2014, pp. 403–406.
- [12] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensors," in 2013 IEEE 10th consumer communications and networking conference (ccnc). IEEE, 2013, pp. 914–919.
- [13] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, "Robust human activity recognition using smartphone sensors via ct-pca and online svm," IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3070–3080, 2017.
- [14] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in International conference on ubiquitous intelligence and computing. Springer, 2010, pp. 548–562.
- [15] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," IEEE transactions on intelligent transportation systems, vol. 16, no. 5, pp. 2406–2417, 2015.
- [16] S. Patel, "Chapter 2 : Svm (support vector machine) - theory," May 2017. [Online]. Available: <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>
- [17] "Radial basis function kernel," Dec 2020. [Online]. Available: https://en.wikipedia.org/wiki/Radial_basis_function_kernel
- [18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.