Human Activity Classification Using Basic Machine Learning Models

Bikram Khanal* , Pablo Rivas† , *Senior, IEEE*, Javier Orduz ‡
School of Engineering and Computer Science
Department of Computer Science, Baylor University
Email: {*Bikram_Khanal1, †Pablo_Rivas, ‡Javier_OrduzDucuara}@Baylor.edu

***Abstract*—Human activity recognition (HAR) is the object of interest for many researchers in machine learning. In principle, providing accurate and reasonable information on an individual's activities and movements for pervasive computing is a very challenging problem. Recent advances in HAR have led to advanced tracking of highly complex human behaviors. This is progressively driving humans and computers to become seamlessly integrated through devices and software. The impact of this type of research has numerous applications in different sectors. This paper presents our initial experiments on evaluating the performance of popular machine learning algorithms in predicting human behaviors accurately. Our experiments suggest that some models can accomplish high recognition accuracy and low computational cost.**

***Index Terms*—Human activity recognition, machine learning, classification.**

I. Introduction

Computers and humans converge into one inseparable entity in daily life [1]. We use computers or their services every day; e.g., mobile phone connects to operators, and they use computers, we use the internet on computers (or personal devices, which are similar to computers), any computational device link to the internet can place an order of our products or services, these tasks execute on computers. In general, those machines contain small accessories made with different materials and techniques. We call those areas microelectronics since the scale of the region are at micro or nano-meters, namely, μm or nm, respectively.

The developed microelectronic devices have facilitated remarkable features on the ubiquitous phone and wearable gadgets. The low cost of these powerful computing devices, small in size, and portability allow users to interact efficiently. Motivated by the recent studies, activity recognition using smartphones equipped with a rich set of embedded sensors, such as the accelerometer, GPS, microphone [2], has been introduced. HAR works date back to '90s [3]. There is substantial research to extract the knowledge from the data acquired by pervasive sensors [4]. However, there are still numerous arguments with the correct recognition and classification of human activities that motivate the expansion of technologies to improve accuracy under more realistic conditions [5]. Researchers heavily include smartphones and wearable gadgets for their advantageous characteristics aiming towards thriving human activity recognition. It is a relatively simple task to classify activities that differ highly in magnitude in user, but recognizing activities with similar body movements is challenging. For instance, it is comparatively simple to recognize running from sleeping, but it is arduous to classify sitting from sleeping..

Despite significant research efforts, HAR remains a challenging problem. It involves using different sensing technologies to automatically collect and classify users' activities for numerous medical, sports, and leisure applications. Activity recognizing devices has to acknowledge and understand their users' activity dynamically. Although this information can be provided by obtaining the current and updated information about the user, accurately recognizing the user's activities is tedious. There does not exist any hard and fast ML algorithm affirming to classify human activities or activity. When we look at HAR literature, most studies primarily collect sensory data and apply different classification algorithms offline on those collected data. Offline processing of data can exploit the overlap between training and test datasets [6]. We use offline processing in those applications where online processing is not necessary. For instance, if the operation follows a person's daily routine, we do not have to process the data in real-time.

We can collect the data throughout the day, upload it to the server at the end, and process it afterward. However, we can process the user's real-time activity if the job tracks every instance during certain sports activities. This paper limits up to offline processing due to time limitations, interested participants, and sensor-based computing devices. Notably, we are interested in classifiers' performance in offline processing that we can implement in the future for real-time processing. We aim to evaluate three algorithms, Perceptron, Logistic Regression, and Support Vector Machine, for a time-series classification task in detail. We have hand-picked *Human Activity Recognition Using Smartphones* time-series dataset from *UCI Machine Learning Repository achieve* for our experiment. This dataset is suitable for

TABLE I
PERFORMANCE OF DIFFERENT CLASSIFIER IN HUMAN ACTIVITY RECOGNITION

Classifier	Average Accuracy (5-Fold) Cross validation	Training Time (in sec)
KNN	87% (k = 10)	16.517
Complex Decision Tree	91.8 %	7.6294
SVM	93.5 % (quadratic Kernel)	4.947
Linear Discriminant	80.2 %	1.257

classification and clustering and contains multivariate time-series dataset.

II. Related Work

Activity recognition mainly focuses on the users or their surrounding environment. Researchers especially observe human actions to understand the varieties of activities that humans function within a time interval. One of the crucial components in HAR is the classification algorithm used to classify different movements and actions based on the users' input data. Since the 2000s, there have been several studies in activity recognition. From the survey, researchers noticed that Decision Trees, K-Nearest Neighbour(KNN), Naive Bayes(NB), Support Vector Machine (SVM), and Neural networks are extensively practices [7]. Baerhoven et al. [8] experimented with the classification of 20 activities using accelerometer data. Using a five biaxial accelerometer, they collected data from walking, sitting, running, watching TV, and climbing stair sets. They used the collected data to train different classifiers, listed in Table 1, in the WEKA [9] toolbox. In an experiment, Ravi et al. collected data from a single user, and their result reveals that applying a 3-axis accelerometer to detect human activities (we list these activities in the methods section) gives an accuracy of 90.61% [10], [11]. Following a similar experiment, Tahamia et al. collected data from 20 users randomly. Their result concluded that we could improve the classification accuracy by changing the parameters like the number of splits on KFold, types of a kernel in SVM, number of Neighbours in KNearest neighbors, and learning rate for respective classifier [12]. We present the results from these experiments that are important in our research in Table 1.

Mustafa Kose and Ozlem used intelligent phone sensors to recognize human activity in real-time. They asked five volunteers to carry their smartphones in their pockets during the data collection and active exercises. All subjects performed the predefined task, running, walking, sitting, and standing. They used clustered KNN and Naive Bayes classifiers to train their model. From their result, researchers found that clustered KNN gives an accuracy of 92.13% [12].

Algorithms like KNN are suitable to be implemented in smartphones because they need fewer computation resources. HAR gadgets manufacturers often use semi-supervised classification algorithms that use multiple classifiers to classify unlabeled data. Since supervised classification algorithms need intensive computation to generate models, we estimate most computations in servers. We migrate the obtained model from servers into smartphones and gadgets for the classification of input data. We can improve the training time by doing parallel training in clusters rather than sequential training [13]. Haitao et al. used the fusion of $SVM\&HMM$ for activity recognition. They used motion features, structure features, and polar coordinates features that could make some very similar activities get the distinction significantly [14]. Their result shows that the $SVM\&HMM$ has a recognition accuracy of 98%. We will now discuss some methods we implemented to evaluate the algorithms mentioned earlier on the chosen dataset.

III. Methods

This section presents the proposed methodology, including data extraction, the used classifiers, and the performance evaluation using Logistic Regression and Support Vector Machine.

A. Data Acquisitions

We fetched the used dataset from UCI Machine Learning Repository available at https://archive.ics.uci.edu/ml/machine-learning-databases/00240/. Detailed information, including videos about data collection methods and experiments, is available at [15]. This dataset contains a 10299 number of instances with 561 features. They selected 30 volunteers to perform the following activities:

1) Walking.
2) Sitting.
3) Standing.
4) Lying.
5) Walking Upstairs.
6) Walking Downstairs.

They used Gyroscope and Accelerometer to record these activities as data. They randomly split the dataset into 70% training data and 30% test data. Each data point in the dataset corresponds to one of the six activities from above.

B. Data Visualization

We used standard Machine Learning libraries, matplotlib and seaborn, to visualize the data. We are primarily interested in the activities performed by each volunteer.

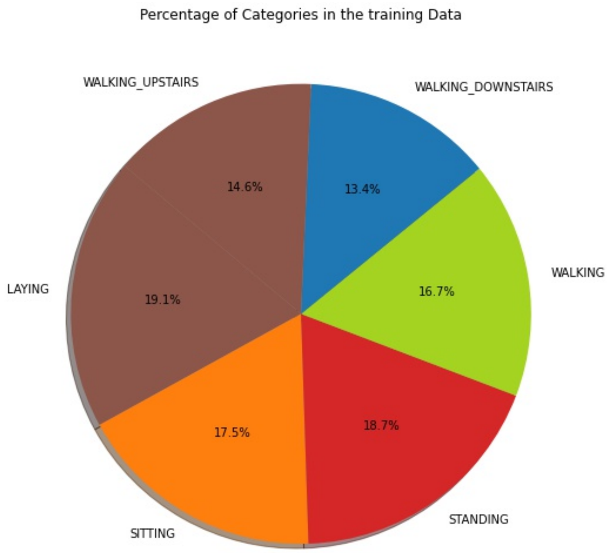

Fig. 1. Pie-Chart of activities in training dataset

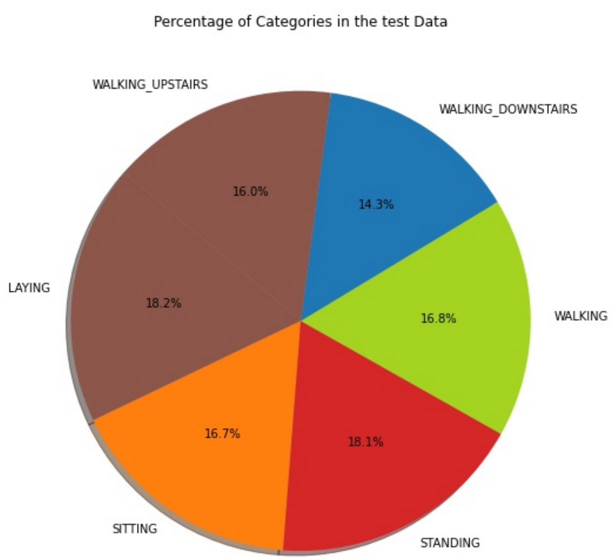

Fig. 2. Pie-Chart of activities in test dataset

Figure1,2 shows that the category distribution is uniform in the training and test dataset. The difference between the "Laying class (19.1%) and "waling Downstairs (13.14%)" might create some misclassification. We will consider this difference while analyzing the result. Here onwards, we will only visualize the training dataset because the readings are uniform for both datasets.

We removed few distinct noises from the definitions of the training and testing dataset features. These feature definitions have an inconvenient naming for further calculations and training to remove special characters. We have filled the null values in the dataset with 0 to make a calculation straightforward.

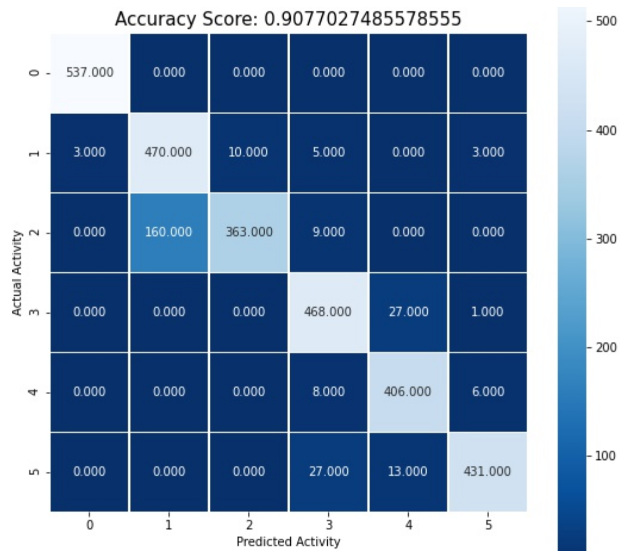

Fig. 3. Confusion Matrix of perceptron

TABLE II
BALANCED REPORT OF PERCEPTRON

	precision	**recall**	**f1-score**	**support**
LAYING	0.99	1.00	1.00	537
SITTING	0.75	0.96	0.84	491
STANDING	0.97	0.68	0.80	532
WALKING	0.91	0.94	0.92	496
WALKING dOWNSTAIRS	0.91	0.97	0.94	420
WALKING UPSTAIRS	0.98	0.92	0.95	471
accuracy			0.91	2947
macro avg	0.92	0.91	0.91	2947
weighted avg	0.92	0.91	0.91	2947

C. Algorithms Implementation

As we mentioned in our introduction, we will implement classical machine learning in this cleaned and ready-to-train dataset, particularly perceptron, Logistic regression, and Support Vector Machine.

1) **Perceptron**
We implemented the sklearn perceptron module with ten iterations and no shuffling. The initial training gave an accuracy of 86% on perceptron. We implemented the 10 Fold cross-validation to improve the model performance and repeated (9 times) 10 Fold cross-validation. The accuracy improved to 90% with 6 10 Fold and the repeated cross-validation shows the accuracy of 98.5%. The resulted of our experiment is shown in Figure 3 an Table II.
We observe from the confusion matrix that the perceptron has activity 1 (Sitting) and activity

TABLE III
BALANCED REPORT OF LOGISTIC REGRESSION

	precision	recall	f1-score	support
LYING	1.00	1.00	1.00	537
SITTING	0.98	0.88	0.92	491
STAND-ING	0.90	0.98	0.94	532
WALKING	0.94	0.99	0.97	496
WALKING dOWN-STAIRS	0.99	0.96	0.98	420
WALKING UPSTAIRS	0.97	0.94	0.96	471
accuracy			0.96	2947
macro avg	0.96	0.96	0.96	2947
weighted avg	0.96	0.96	0.96	2947

2 (Standing); it has 160 misclassifications on these classes. However, the repeated 10-fold cross-validation shows the result to be consistent above 98%. This improvement is significant.

2) **Logistic Regression**
Yasar S. Abu-Mostafa defines Logistic Regression [16] as:

$$h(x) = \theta(w^T x), \tag{1}$$

where $h, h \in H$ is a hypothesis, $h(x)$ is a prediction model, **w** is a weight vector and **x** is an input vector. θ is so called logistic function defined as [16]:

$$\theta(s) = \frac{e^s}{1 + e^s}. \tag{2}$$

We can interpret the output of Logistic Regression as a probability. Thus, we can use these properties of Logistic Regression to predict, given the input vector x_i, the probability that the given data can be one of 6 activities. This notion is crucial because sometimes the tracking devices can predict running as walking or sitting as sleeping. The probability bound of Logistic Regression will help update each activity's possibilities weights and reclassify correctly.

a) **Performance**
We used a linear model from the sklearn library to implement the Logistic Regression. We changed a penalty to $l2$, iterations to 300, and the rest of the parameters to default. We were able to get an accuracy of 96% and the training time was below a minute. We used a confusion matrix to plot the error. Figure 4 shows the confusion matrix for Logistic regression and we present the balanced score in Table III.

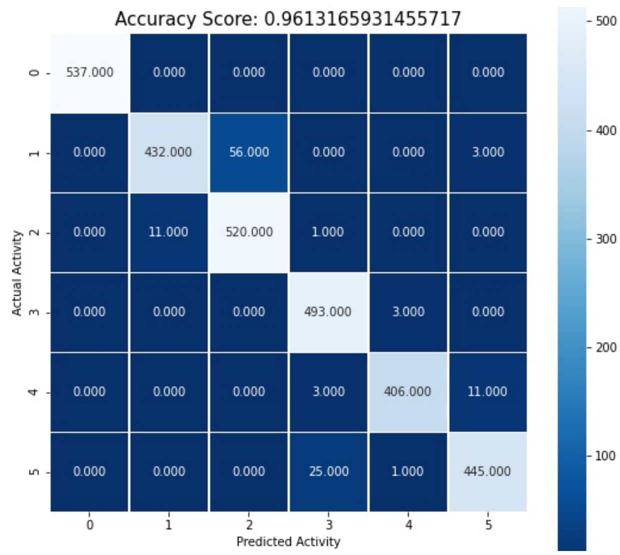

Fig. 4. Confusion Matrix of Logistic Regression

Figure 4 shows a maximum error of 57 instances between activity 1 (refer to data Acquisitions sections for activity name) and activity 2. There is also an error of 25 between actual activity five and predicted activity 3. The accuracy improved to 98.5% with 10 Fold cross-validation. The cross-validation is similar to perceptron; thus, we leave that section as an explanation to the reader.

3) **Support Vector Machine**
Support Vector Machine is a classification technique that finds the decision boundary to separate different classes and maximizes the margin [12]. We will experiment with SVM with the varying kernel, "linear, ploy, RBF or sigmoid," and account for the best performing kernel trick.

a) **Performance**
We used the sklearn library to implement the SVM and used the confusion matrix for error measurement. We tasted with regularization values of $0.5, 1, 1.5, 2.0$ on sigmoid kernel and RBF. SVM performed uniform with both transformations on all regularization values with an accuracy of 95%, but the result reduced the accuracy to 79% on a regularization value of 0.5. Figure 5 shows the error measures of SVM in the confusion matrix and we present the balanced report in Table IV.

The SVM has an accuracy of 95%. The most misclassified activities are activity one and

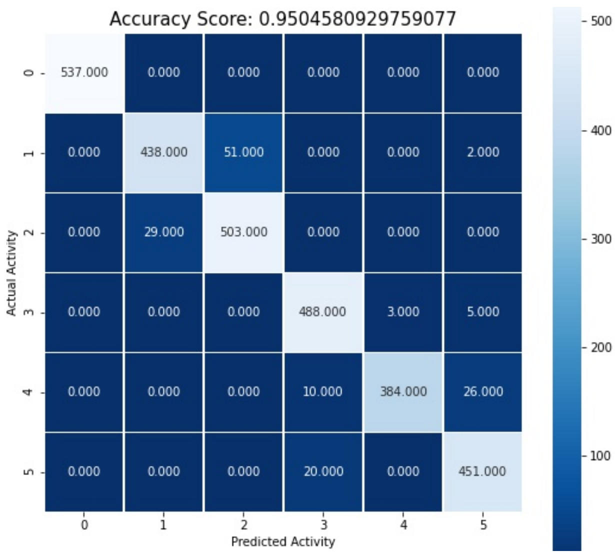

Fig. 5. Confusion Matrix of Support Vector Machine

TABLE IV
BALANCED REPORT OF SUPPORT VECTOR MACHINE

	precision	recall	f1-score	support
LYING	1.00	1.00	1.00	537
SITTING	0.95	0.89	0.92	491
STAND-ING	0.91	0.95	0.93	532
WALKING	0.94	0.98	0.96	496
WALKING dOWN-STAIRS	0.99	0.96	0.98	420
WALKING UPSTAIRS	0.99	0.95	0.94	471
accuracy			0.95	2947
macro avg	0.95	0.95	0.95	2947
weighted avg	0.95	0.95	0.95	2947

activity 2, and activity four and activity 5. SVM has minor misclassification of every activity as activity 6. We trained our model by changing the transformation degree between $1-6$. However, there was an improvement of 0.037% on 5^{th} degree transformation, we decided to choose a $3-degree$ transformation on regularization of 1 with $Sigmoid$ kernel. The Support Vector Classifier(SVC) shows slightly lower accuracy on repeated 10 Fold cross-validation but still above 98%.

4) **Result**
We tested our dataset on other classification algorithms and accounted for the performance. We present all the implemented algorithms and their performance accuracy in Table V.

TABLE V
PERFORMANCE OF DIFFERENT CLASSIFIER IN HUMAN ACTIVITY RECOGNITION

Classifier	Accuracy %	Standard Deviation
Random Forest	92	0.006
Ridge	96	0.005
SVC	95	0.009
Decision Tree	86	0.013
Gaussian NB	77	0.027
K-Neighbour (10)	90	0.012
Perceptron	91	0.043
Logistic Regression	96	0.008

IV. DISCUSSION

We limited our experiment to few classification algorithms. On Support Vector Classification (SVC) and Logistic Regression, both algorithms have an error of less than 5%. The result is satisfying given the computation required. We believe the research can further expand by applying k-means clustering to the data set before using any classification algorithms. Once we obtain the cluster from k-means, and if we know the overlapping labels, we can perform feature transformation on overlapping data and make them separable in Z space given by the equation:

$$Z = \phi(x) = (1, x_1^2, x_2^2, ..., x_n^2), \tag{3}$$

where n is the number of features we need to transfer, the transformation mentioned above limits the transformation of 2 degrees, but it can further be generalized by Legendre Polynomial Transformation. We can perform feature reduction to improve accuracy. In this paper, we limit ourselves to evaluate different Machine Learning Algorithms' performance and leave the accuracy improvement for further research.

V. CONCLUSION

In this paper, we experimented with various classifiers measuring their performance on six different human activities. The performance was measured using standard accuracy as the main comparison element. We focused our attention on the SVM classifier and Logistic Regression, which repeatedly reported accuracy of above 98%, using 10-fold cross-validation. Our experimentations suggest that implementing either logistic regression or SVMs for HAR is viable and has great potential for success. Further, our initial findings suggest that optimizations, such as applying feature transformations can increase classification performance. These initial experiments are rooted in the belief that we must first attempt to solve machine learning problems using established, simple algorithms as a baseline before deciding to use other experimental deep learning algorithms. Consequently, further work should consider deep learning methodologies and ensemble methods to verify the dataset's properties and the response of other experimental models.

ACKNOWLEDGMENT

The work presented here was supported in part by the Baylor AI lab in Baylor University's Department of Computer Science.

REFERENCES

[1] U. R. Alo, H. F. Nweke, Y. W. Teh, and G. Murtaza, "Smartphone motion sensor-based complex human activity identification using deep stacked autoencoder algorithm for enhanced smart healthcare system," *Sensors*, vol. 20, no. 21, p. 6300, 2020.

[2] A. Subasi, K. Khateeb, T. Brahimi, and A. Sarirete, "Human activity recognition using machine learning methods in a smart healthcare environment," in *Innovation in health informatics*. Elsevier, 2020, pp. 123–144.

[3] L. Zhao and W. Chen, "Detection and recognition of human body posture in motion based on sensor technology," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 5, pp. 766–770, 2020.

[4] 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, 2013.

[5] A. Gupta, K. Gupta, K. Gupta, and K. Gupta, "A survey on human activity recognition and classification," in *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020, pp. 0915–0919.

[6] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/1/2059

[7] H. Badem, A. Caliskan, A. Basturk, and M. E. Yuksel, "Classification of human activity by using a stacked autoencoder," in *2016 Medical Technologies National Congress (TIPTEKNO)*, 2016, pp. 1–4.

[8] K. Aminian, B. Najafi, C. Büla, P.-F. Leyvraz, and P. Robert, "Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes," *Journal of biomechanics*, vol. 35, no. 5, pp. 689–699, 2002.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10–18, Nov. 2009. [Online]. Available: https://doi.org/10.1145/1656274.1656278

[10] F. Samie, L. Bauer, and J. Henkel, "Hierarchical classification for constrained iot devices: A case study on human activity recognition," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8287–8295, 2020.

[11] A. Narayanan, F. Desai, T. Stewart, S. Duncan, and L. Mackay, "Application of raw accelerometer data and machine-learning techniques to characterize human movement behavior: A systematic scoping review," *Journal of Physical Activity and Health*, vol. 17, no. 3, pp. 360–383, 2020.

[12] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Evaluation of supervised classification algorithms for human activity recognition with inertial sensors," in *2017 IEEE SENSORS*, 2017, pp. 1–3.

[13] A. Sanjel, "Tyro: A first step towards automatically generating parallel programs from sequential programs," One Bear Place #97148, 11 2020, an optional note.

[14] H. Wu, W. Pan, X. Xiong, and S. Xu, "Human activity recognition based on the combined svm amp;hmm," in *2014 IEEE International Conference on Information and Automation (ICIA)*, 2014, pp. 219–224.

[15] J.-L. Reyes-Ortiz, D. Anguita, A. Ghio, and X. Parra, "Human activity recognition using smartphones data set," 2012. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.php

[16] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook New York, NY, USA:, 2012, vol. 4.