”SmartMove: Advanced Human Activity Recognition Using Smartphone Sensors”

林威志[a] and 楊恩光[b]

[a]Senior Student, Department of Space Science & Engineering National Central University
[b]Junior Student, Class B, Department of Mathematics National Central University

***Abstract*—This research focuses on the field of Human Activity Recognition (HAR), utilizing machine learning (ML) and pattern recognition techniques to automatically detect and classify various human activities. By analyzing data collected from sensors of smartphones, this study aims to identify activities such as walking, climbing stairs, and cycling. We plan to initially use logistic regression models, expecting to achieve an accuracy of at least 95%. Upon success, we will explore more advanced classification algorithms like Linear SVC, Kernal SVC, Decision Tree, and Random Forest Classifier to determine the most suitable model. The outcomes of this study not only validate the feasibility of autonomous data collection and machine learning solutions but also demonstrate their potential applications in areas such as medical diagnosis and treatment.**

I. Introduction

Our research falls within the domain of HAR, which involves the use of ML and pattern recognition techniques to automatically detect and classify various human activities. HAR enables the extraction of features from sensor data or other sources, allowing the identification of specific activities or movements in progress. The escalating prominence of this field can be attributed to two primary factors. First, the declining costs of sensors. Second, the rise of Internet of Things (IoT), Computer Vision (CV), and ML technologies has significantly contributed to the advancement of this domain. To better understand how to proceed with our research topic, we referred to a paper and online resources. In [1], introduced an effective method for HAR on smartphones. This approach utilizes data mining and ML techniques, employing a model stacking approach to enhance model performance. In [2], provides a comprehensive process for data analysis and model construction, through ML and deep learning (DL) for HAR. While the referenced material focuses on image recognition, the pre-processing methods align with the direction we intend to pursue.

II. Data collection

Our data is sourced from the Physics Toolbox app [3]. By obtaining data from three different instruments to increase the number of features, we hope to improve the accuracy and recall of the model. We placed the smartphone in our pocket for data collection, ensuring consistent coordinate axes across different instruments: +X (left side of the phone), +Y (phone to the bottom), and +Z (back of the phone to the bottom). The positioning of the phone can impact the overall results, so we standardized the placement by using the most convenient method to put in pocket. The data collection involved segmented acquisition, meaning that the time-series data in each file corresponds to a specific activity. This segmentation simplifies the process of data labeling, making it more convenient during the annotation phase.

A. G-force meter

The unit of g-force is m/s^2, which is the ratio of normal force to gravity. Different accelerations lead to varying normal forces, which is one of the reasons for choosing G-force.

B. Linear Accelerometer

The unit of acceleration is m/s^2, and it has already eliminated for the influence of gravity. This data allows for a direct differentiation of activities. The acceleration during walking is depicted in Fig.1, while Fig.2 illustrates the acceleration during climbing the stairs. The acceleration during cycling is displayed in Fig.3.

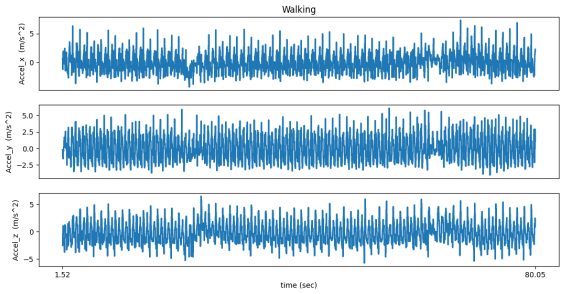

Fig. 1. Acceleration of walking.

C. Gyroscope

The unit of angular velocity is rad/s, which was initially considered indispensable, but after data pre-processing, it was found to have little effect on the model's performance. This point will be further discussed later.

III. Data pre-processing

Each instrument will have values for the x, y, and z axes, as well as a total value. The whole dataset is divided into a window of 2 seconds with 50% overlapping. The final amount of data is presented in Fig.4, and it can be seen that

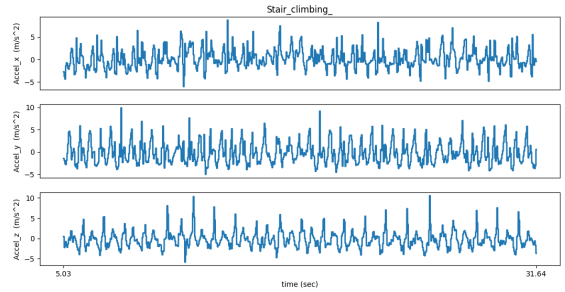

Fig. 2. Acceleration of stair climbing.

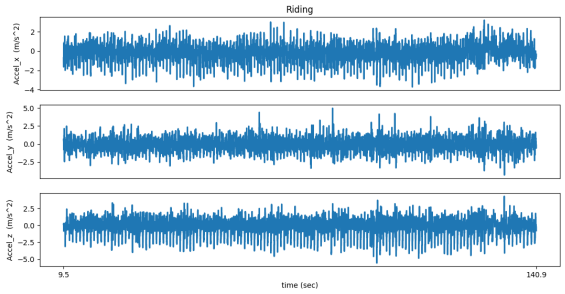

Fig. 3. Acceleration of cycling.

the data is balanced, which is good for future recognition. After that, we can do feature extraction, the features used are determined by the type of action and the sensor modality. G-force meter or accelerometer data, for example, can be used to extract features such as mean, standard deviation, max, min, mad(median absolute deviation), skewness, kurtosis and InterQuartile Range (IQR).

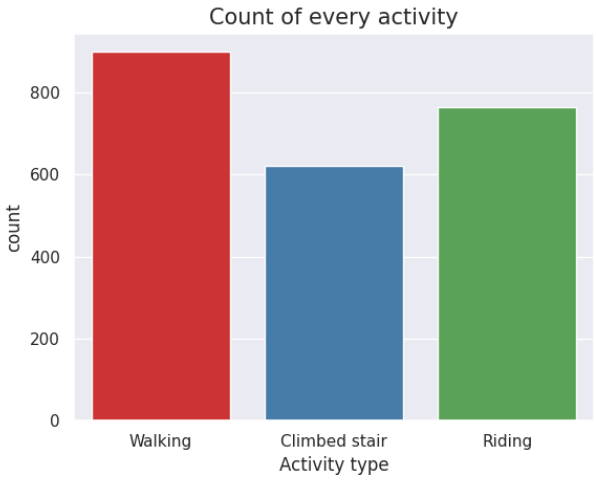

Fig. 4. The data quantities for three activities.

From Fig.5 and 6, it is evident that cycling can be easily distinguished from walking and stair climbing based on g-force and acceleration.

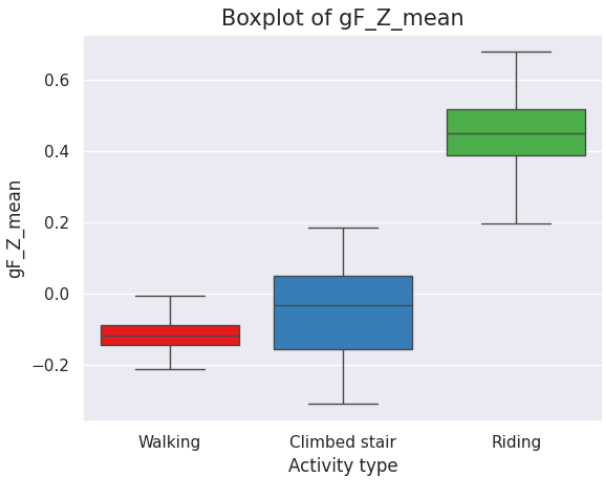

Fig. 5. The boxplot of z-axis g-force's mean in three activities.

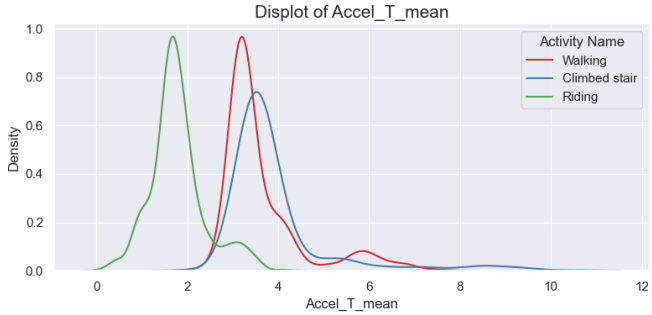

Fig. 6. The distplot of total acceleration's mean in three activities.

After feature extraction, the number of features will increase from the original 12 to 96. However, an excessive number of features can lead to prolonged model training times and overfitting. Therefore, dimensionality reduction is necessary. We chose to assess feature importance using random forests and ultimately selected the top 20 important features as inputs for model training in Fig.7.

IV. Model Deployment

A. Data Splitting

Our dataset consists of a large number of observations and features. To build our machine learning model, we needed to divide the data into a training set and a testing set for training and evaluation purposes.

We chose to use 70% of the data as the training set, reserving the remaining 30% as the testing set. This ensures that we have enough data to train the model while also having an independent dataset to assess the model's performance.

B. Machine Learning Models

We employed five distinct machine learning models[4] to evaluate the accuracy of classifying human activities into three

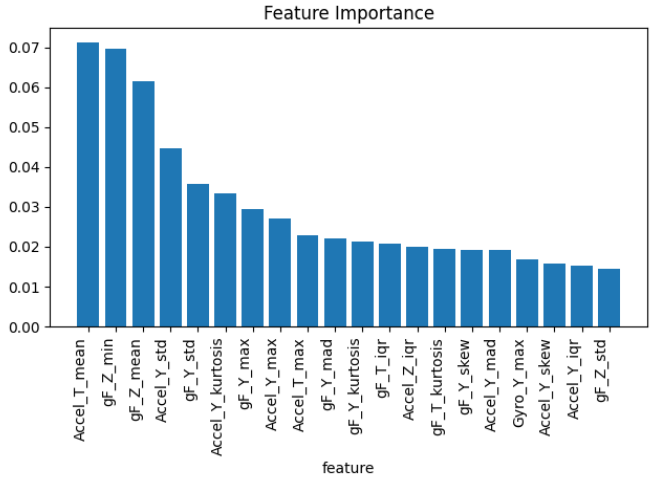

Fig. 7. Feature importance.

categories: walking, climbing stairs, and riding a bicycle. The following models were utilized:

1) **Support Vector Classifier (SVC):**
SVC works by identifying the hyperplane that best separates the classes in the feature space. The optimal hyperplane is the one that maximizes the margin between the classes, defined as the distance between the nearest points of the classes to the hyperplane. These nearest points are called support vectors.
The optimization problem for the SVC can be formalized as follows:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

Subject to the constraints:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for all } i = 1, 2, ..., n \tag{2}$$

Here, $\mathbf{w}$ is the weight vector, b is the bias term, C is the penalty parameter of the error term, and ξ_i are the slack variables that allow for misclassification. The term $\frac{1}{2}\|\mathbf{w}\|^2$ is the norm of the weight vector which we seek to minimize to achieve a larger margin. The constant C controls the trade-off between maximizing the margin and minimizing the classification error.
The dual form of this optimization problem introduces Lagrange multipliers, which provide a way to solve the problem efficiently and also make it possible to apply the kernel trick for non-linear classification:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{3}$$

Subject to the constraints:

$$0 \leq \alpha_i \leq C, \quad \text{for all } i = 1, 2, ..., n \tag{4}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{5}$$

2) **Decision Tree Classifier:**
The core algorithm for building decision trees called the ID3 (Iterative Dichotomiser 3) uses entropy and information gain to construct a decision tree. The decision rules are formed by determining the feature that provides the highest information gain at each step.
Entropy for a binary classification can be defined as:

$$H(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-) \tag{6}$$

where $H(S)$ is the entropy of set S, p_+ is the proportion of positive examples in S, and p_- is the proportion of negative examples in S.
Information gain is calculated as the difference between the entropy before the split and the weighted sum of the entropy of each subset after the split:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \tag{7}$$

where $IG(S, A)$ is the information gain of set S after splitting on attribute A, $Values(A)$ is the set of all possible values for attribute A, S_v is the subset of S for which attribute A has value v, and $|S_v|$ and $|S|$ are the cardinalities of S_v and S, respectively.
The attribute with the highest information gain is chosen to make the decision split at each node in the tree. The process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has the same value of the target variable, or when splitting no longer adds value to the predictions.

3) **Linear Support Vector Classifier (LinearSVC):**
The objective function of LinearSVC can be described as follows:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{8}$$

Subject to the constraints:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \quad \text{for all } i = 1, 2, ..., n \tag{9}$$

Here, $\mathbf{w}$ represents the weight vector, b is the bias term, y_i are the class labels, and $\mathbf{x}_i$ are the feature vectors.
Unlike the traditional SVC, LinearSVC does not use slack variables or a penalty parameter C. Instead, it assumes that the data is linearly separable. The optimization is performed without the slack variables, which implies that it does not allow for misclassification. Hence, LinearSVC is typically used when it is known that the classes can be separated by a linear boundary.

4) **Logistic Regression:**
The model of Logistic Regression is given by the logistic function, which is an S-shaped curve that can take any

real-valued number and map it into a value between 0 and 1, but never exactly at those limits. In the case of binary classification, the predicted probabilities can be formulated as:

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_n x_n)}} \tag{10}$$

where $P(Y = 1|X = x)$ is the probability that the dependent variable Y is 1 given the independent variables X, e is the base of the natural logarithm, and $\beta_0, \beta_1, ..., \beta_n$ are the parameters of the model.
The coefficients β_i of the model are estimated from the training data using the maximum likelihood estimation (MLE) method, which finds the set of parameters that makes the observed outcomes most probable.

5) **Random Forest Classifier:**
A Random Forest classifier consists of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \ldots, K\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.
Given a set of training vectors x_i, $i = 1, \ldots, n$, and a vector of class labels y_i, $i = 1, \ldots, n$, Random Forest generates a forest of classification trees, where each tree is grown as follows:
   a) If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
   b) If there are M input variables, a number $m \ll M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
   c) Each tree is grown to the largest extent possible. There is no pruning.

The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Random Forests achieve high accuracy by ensuring that the trees are both accurate and diverse.

For each model, a confusion matrix was generated to illustrate its performance in terms of true positives, false positives, true negatives, and false negatives for each activity class. The diagonal elements of the confusion matrix represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix, the better, indicating many correct predictions.

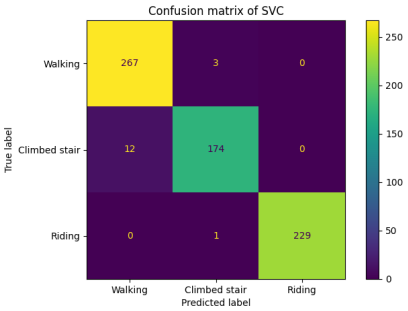

Fig. 8. Confusion matrix of SVC.

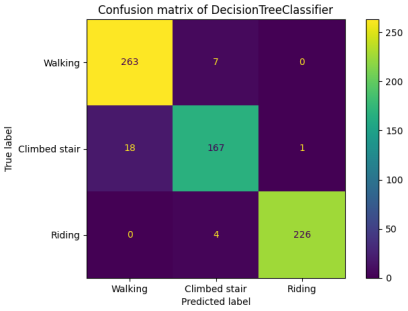

Fig. 9. Confusion matrix of Decision Tree Classifier.

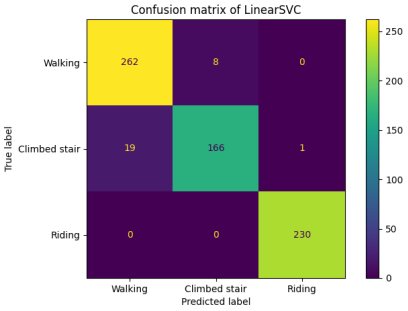

Fig. 10. Confusion matrix of LinearSVC.

Fig. 11. Confusion matrix of Logistic Regression.

C. The accuracy for models

The accuracy of each predictive model was calculated using the standard formula for accuracy in the context of classification models. The formula is defined as the ratio of the sum of true positives (TP) and true negatives (TN) to the

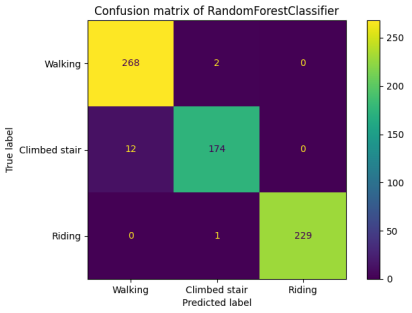

Fig. 12. Confusion matrix of Random Forest Classifier.

total number of observations, which includes true positives, true negatives, false positives (FP), and false negatives (FN). Mathematically, the accuracy (Acc) is expressed as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

This metric provides us with an overall measure of how well our model correctly identifies or excludes conditions. High accuracy indicates that the model is able to correctly label the positive cases as well as the negative cases, which is crucial for the reliability of the predictive model in practical applications.

V. Results

The evaluation of the machine learning models on the task of classifying human activities—walking, stair climbing, and riding—yielded insightful results.

ML Model Results

Model	Hyperparameter Tuing	Accuracy	Recall	Walking	Stair climbing	Riding
Logistic Regression	GridSearchCV	0.9563	0.9527	0.8537	0.1463	0.0
Linear SVC	GridSearchCV	0.9592	0.9543	0.8415	0.1585	0.0
Kernal SVM	GridSearchCV	0.9767	0.9733	0.9756	0.0244	0.0
Decision Tree	GridSearchCV	0.949	0.9447	0.8902	0.1098	0.0
Random Forest Classifier	None	0.9781	0.9751	0.9634	0.0366	0.0

Fig. 13. Accuracy for detecting walking status.

ML Model Results

Model	Hyperparameter Tuing	Accuracy	Recall	Walking	Stair climbing	Riding
Logistic Regression	GridSearchCV	0.9563	0.9527	0.8417	0.9167	0.8417
Linear SVC	GridSearchCV	0.9577	0.9536	0.0	1.0	0.0
Kernal SVM	GridSearchCV	0.9767	0.9733	0.0	1.0	0.0
Decision Tree	GridSearchCV	0.9475	0.943	0.0	0.9583	0.0417
Random Forest Classifier	None	0.9796	0.9758	0.0	1.0	0.0

Fig. 14. Accuracy for detecting climbing stairs status

ML Model Results

Model	Hyperparameter Tuing	Accuracy	Recall	Walking	Stair climbing	Riding
Logistic Regression	GridSearchCV	0.9563	0.9527	0.0052	0.0105	0.9843
Linear SVC	GridSearchCV	0.9563	0.9552	0.0	0.0	1.0
Kernal SVM	GridSearchCV	0.9767	0.9733	0.0366	0.0	0.9634
Decision Tree	GridSearchCV	0.9548	0.951	0.0	0.0157	0.9843
Random Forest Classifier	None	0.9767	0.9728	0.0	0.0	1.0

Fig. 15. Accuracy for detecting biking status

The Logistic Regression model, utilizing GridSearchCV for hyperparameter tuning, demonstrated a high accuracy of 0.9563 and recall of 0.9527, suggesting its adeptness in identifying true positives. However, its precision in detecting walking activity was notably lower, which may indicate challenges in distinguishing this activity from others.

The Linear SVC model, with a similar tuning approach, displayed comparable accuracy to the Logistic Regression model but with a marginally lower recall. It performed with high precision for walking and stair climbing but showed limitations in recognizing riding activity. This could point to a potential weakness in the model's ability to differentiate between activities with similar feature representations.

The Kernel SVM model stood out with the highest accuracy (0.9767) and recall (0.9733), showcasing superior performance in capturing the variance across different activities. Despite this, a minor drop in precision for stair climbing activity was observed, hinting at some confusion between activities.

The Decision Tree model, while slightly lower in accuracy and recall compared to the Kernel SVM, showed perfect precision in identifying the walking activity. However, its performance on stair climbing and riding activities was less impressive, possibly due to the model's tendency to overfit on the dominant class.

Lastly, the Random Forest Classifier, without hyperparameter tuning, attained an accuracy on par with the Kernel SVM (0.9767) and a near-equivalent recall (0.9728). It demonstrated perfect precision in classifying riding activity but had room for improvement in walking and stair climbing. The lack of hyperparameter tuning suggests that default parameters may already be well-suited for the dataset, or there might be potential for further optimization.

In summary, while all models exhibit commendable performance metrics, Kernel SVM and Random Forest Classifier particularly excel in overall accuracy and recall. The choice between models would hinge on the specific application requirements, such as the trade-off between precision and recall, computational constraints, and the cost associated with misclassification.

VI. Discussion

In this project, we encountered several challenges, such as selecting appropriate features and choosing the best model for classification tasks. The process of feature selection is crucial as it directly influences the model's ability to distinguish between different activities accurately. Similarly, the choice of model is significant; a model that performs well on one type of activity classification may not necessarily be the best choice for another.

Looking forward, we aspire to develop a mobile application capable of real-time detection of the user's current state. The goal is to not only recognize distinct activities but also to accurately detect mixed states, providing nuanced judgments in complex scenarios. Such an application would require robust real-time processing and a highly accurate classification algorithm, which could be achieved through further research and development in machine learning techniques and user interface design.

The prospect of creating a versatile and reliable mobile application presents an exciting avenue for future work, with

the potential to make a substantial impact in fields such as personal fitness, health monitoring, and even smart home automation.

VII. Conclusion

In conclusion, this project successfully demonstrated the application of various machine learning models, including SVC, Decision Trees, Linear SVC, Logistic Regression, and Random Forest, in the classification of human activities. Our findings highlight the strengths and limitations of each model, providing valuable insights into their practical applications. The Logistic Regression and Random Forest models, in particular, showed promising results in terms of accuracy and reliability. Future work could explore the integration of these models into a real-time mobile application, expanding their utility in dynamic settings. Additionally, further research into optimizing model parameters and exploring hybrid models could enhance their effectiveness in complex classification tasks.

References

[1] Sakorn Mekruksavanich, Narit Hnoohom and Anuchit Jitpattanakul, "Smartwatch-based Sitting Detection with Human Activity Recognition for Office Workers Syndrome," ECTI-NCON 2018.

[2] Human Activity Recognition (HAR):Fundamentals, Models, Datasets

[3] Physics Toolbox Sensor Suite

[4] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R (Second Edition)*, Springer, 2021.

Appendices

Demo video : Data Collection & Accuracy