NEAT Activity Detection using Smartwatch at Low Sampling Frequency

Ankita Dewan
Dept of Computer Science
IIT Ropar
Ropar, India
2017csz0012@iitrpr.ac.in

Venkata M. V. Gunturi
Dept of Computer Science
IIT Ropar
Ropar, India
gunturi@iitrpr.ac.in

Vinayak Naik
CSIS and APPCAIR
BITS Pilani Goa Campus
Goa, India
vinayak@goa.bits-pilani.ac.in

Kousik Kumar Dutta
Dept of Computer Science
IIT Ropar
Ropar, India
2019csm1018@iitrpr.ac.in

***Abstract*—Our paper aims to build a classification model to discern the typical NEAT (Non-Exercise Activity Thermogenesis) activities done in a home setting. The concept of NEAT is broadly defined as the energy spent in everything which is not sleeping, eating, or a traditional form of physical exercise. We focus on the following NEAT and non-NEAT activities in this paper - cooking, sweeping, mopping, walking, climbing up, climbing down, and non-NEAT activities (e.g., watching television and working on a desk). This aim is to build a classification model which can work with data sampled at a low frequency of 1Hz. However, building such a classifier is non-trivial because the NEAT activities are not easily separable in low-frequency data. The state-of-the-art in the area of human activity recognition either uses multiple physical devices (e.g., accelerometers on arms, waist, and feet) for data collection or use data that is sampled at high frequency (20Hz or above). In contrast, our model performs NEAT activity recognition using data sampled at 1Hz and from a single smartwatch worn on the dominant hand. Thus, making it more energy-efficient and easily usable for widespread use. We evaluate our proposed model using actual data collected on a smartwatch, and we compare it with alternative models. Our results indicate that the proposed model is able to achieve much higher accuracy than the alternative approaches.**

***Index Terms*—Activity Recognition, Smart Watch sensors, Low frequency**

I. Introduction

The central goal of this paper is to develop a classifier that can discern the typical activities done in a home setting using the data coming from the accelerometer and gyroscope sensors ([1]) embedded in a smartwatch. More precisely, the classifier should be able to distinguish among the following seven activities: (a) cooking, (b) sweeping, (c) mopping, (d) walking, (e) climbing up, (f) climbing down, and (g) non-NEAT activities. The non-NEAT activity encompasses activities that involve little to no physical movement, e.g., watching television, working on a desk/computer, and remaining stationary. Our aim is to develop a classifier which is robust enough to work with data collected from a smartwatch at a low sampling frequency such as 1Hz. This granularity is much less than the 20Hz (or more), which is typically used in several current state-of-the-art activity recognition models.

Building such an activity classifier is of value in a health monitoring mobile application that can record the time spent in non-exercise activity thermogenesis (NEAT) [2]. NEAT is broadly defined as the energy consumed in everything which is not sleeping, eating, and traditional physical exercise. In other words, NEAT focuses on activities that are not considered as "exercise" in a conventional sense. Examples of the NEAT activities include daily house activities such as cooking, cleaning (mopping and sweeping in Figure 1), walking around the house, etc. Getting insights into the total daily time spent in NEAT activities can help a user plan and execute a healthy lifestyle in the long run [3], [4]. NEAT activities have also shown benefit in management (and also prevention to some extent) of obesity, diabetes, cardiovascular disease and mental health problems [3]–[6]. Such an application could also be of great use during a pandemic where several people are constrained to home environments due to lock downs. In such situations, doing home chores (e.g., cooking and cleaning) may perhaps be the only form of exercise that many people may get in a day. Thus, an application which can keep track of NEAT activities (via home chores) could be of value in keeping track of total time spent in doing physical activities in a home setting.

(a) Sweeping

(b) Mopping

Fig. 1: A Few NEAT activities, image courtesy Google (NDTV.com and Amazon.in).

We believe that an ideal solution for NEAT activity recognition should have the following two features. Firstly, it should use a single and readily available hardware (e.g., smartphone or smartwatch). Secondly, it should be energy efficient as NEAT activities are likely to spread across the entire day. It would be reasonable to assume that any user would like to record their data without charging the device multiple times for the whole day to get meaningful results.

In this work, we develop a smartwatch-based solution for NEAT activity recognition. Regarding energy efficiency, one can achieve energy efficiency by reducing the sampling rate of the sensors as sampling rate and battery consumption are known to be directly proportional to each other [7]. Table I details the battery consumption for different sampling rates.

Frequency	Battery
10 Hz	93%
5 Hz	86%
1 Hz	78%

TABLE I: Total Battery consumed in 5 hours

Frequency	1Hz	5Hz	10Hz
KNN	74	85	89
Multi-layer Perceptron	71	82	89
SVM	78	87	92
Logistic	61	71	81
Random Forest	47	53	61
Naive Bayes	60	67	73
XGBoost	81	89	94
Our Model	87	94	96

TABLE II: Accuracy of different classifiers on 1Hz, 5Hz, and 10Hz sampled data (2 sec windows 50% overlap.)

While low sampling frequency ensures good battery life, it is essential to note that *activities focused in this paper cannot be easily discerned in low-frequency data*. Table II illustrates the total accuracy (for our activities of interest) obtained by different classification techniques on data sampled at 10Hz, 5Hz, and 1Hz. As the table shows, the performance of the classifiers drops down significantly as the sampling frequency is reduced to 1Hz.

Our Contributions:

1) This paper proposes a novel classifier which can accurately identify the following seven different types of activities done in a typical home setting: (a) cooking, (b) sweeping, (c) mopping, (d) walking, (e) climbing-up, (g) climbing-down, (h) non-NEAT. Such a classifier can help in developing a solution for recording NEAT activities done by a person in a day.
2) The classifier proposed in this paper splits the seven-class classification problem into a series of two-class classification problems, which are organized in a hierarchy. At the higher levels of this hierarchical model, we classify the data into meta-level categories (e.g., windows having "no-leg motion" vs. "leg-motion," etc.). Following this (at lower levels), we classify the data into one of the previously mentioned seven activities. The proposed model works with the data coming from the accelerometer and gyroscope sensors available in a typical smartwatch.
3) Our proposed classifier can work with data sampled at low frequency (1Hz). Thus, it is energy efficient.
4) We evaluated (trained and tested) our proposed classifier experimentally on real data collected by four volunteers. We also compared our approach with the alternatives to establish the superiority of the proposed approach.

Scope of the paper: In this paper, the "non-NEAT" motion class encompasses the following activities: (a) watching television, (b) working on desk/computer, and (c) remaining stationary. In the future, we would expand the "non-NEAT" motion class to include other activities such as driving a car, etc.

II. Basics Concepts and Problem Definition

A. Basic concepts in motion classes

1) **Cooking:** This motion class takes place in the kitchen. To collect the data, there were dedicated start and stop buttons in the user's smartwatch. The user presses the start button on entering the kitchen and, the stop button on exit. All the sub-activities of typical cooking activities such as "making an Indian flatbread" (which include actions such as smoothing and flattening of dough, toasting it on a pan), "making curry" (which include actions such as peeling, chopping, washing, and stirring the vegetables), etc., are included in this class.
2) **Sweeping:** This motion class takes place in the entire house. In this activity, the user sweeps the floor using a broomstick (figure 1a). Moreover, the person performing the activity may also walk and move objects intermittently.
3) **Wet Mopping:** In this motion class, the person mops the floor using the wet-mopping stick (figure 1b). This motion class also includes the sub-activity of wringing the mop after it is dipped into the cleaning solution.
4) **Walking:** This motion class largely happens when the user is walking around in home (but not sweeping or mopping).
5) **Climb up/Down:** This class includes the activity of climbing up or down the stairs. Note that volunteers were also allowed to hold the side railing while performing this activity.
6) **non-NEAT:** This class majorly includes activities, which involve few or no physical movements. During this activity, volunteers were allowed to work on a computer, watch television, or largely remain stationary.

In all the above-mentioned classes, the volunteers were wearing the smartwatch in their dominant hand, i.e., usually the right hand.

B. Problem Definition

Input: Our raw data consists of a set of time series ($\mathcal{T}$) of smartwatch sensor data (accelerometer and gyroscope sensors). For any time point p in a time series $t_i \in \mathcal{T}$, we have sensor values for the accelerometer and gyroscope sensors in the smartwatch. Each time series $t_i \in \mathcal{T}$ corresponds to one of the following seven motion classes: (a) Cooking, (b) Sweeping, (c) Moping, (d) Walking, (e) Climbing up, (f) Climbing Down and (g) non-NEAT.

The input set of time series $\mathcal{T}$ is processed into a set of overlapping windows $\mathcal{W}$. Each window $w_i \in \mathcal{W}$ is β in length. Two temporally adjacent windows overlap by an amount θ. We vary both β and θ in our experiments. Each $w_i \in \mathcal{W}$ is assigned a unique class label corresponding to one of the seven previously mentioned activities and is represented using a set of features.

Objective: To learn a classification model (using $\mathcal{W}$ as training data) which can recognise the previously mentioned seven activity classes.

III. Summary of Related Work

Research literature most relevant to our problem consists of the work done in the areas of human activity recognition and transportation detection. We shall now briefly summarize the current research literature and discuss their relevance.

The field of human activity recognition has been studied for more than a decade, and researchers have approached it from multiple directions (e.g., [8]–[22]. Overall, these works developed classification models which can discern a wide variety of activities. Examples include both simple activities (e.g., walking, sitting standing, etc.) and complex activities (e.g., car driving, vacuum cleaning, intravenous injection, etc.). Work done in this area can be broadly classified into the following two categories: (a) sensor data was collected from multiple devices (including customized hardware) [8]–[10], [12], [14], [15], [21], [23], [24] and (b) sensor data was collected from a single device (e.g., smartphone, smartwatch, etc.) and [16], [18]–[20], [25].

Solutions developed in the first category use data from multiple physical sensors (e.g., [12], [14], [15], [23], [24], [26]) in their classification model. For instance, [14] proposed a model that required data from three different accelerometers (one on the right wrist, one in the breast pocket, and one on the back hip) to perform activity recognition in a hospital setting. Similarly, the approach proposed in [26] requires data from five accelerometers (located on the various positions on the body) and a heart monitor to predict human activities such as rowing, cycling, walking, etc. [12] using data from five accelerometers (placed on both legs, arms, and hip) to recognize body postures (sitting, standing, etc), movement (walking, running, etc), hand gestures (chop, throw, punch, etc) and hand postures (holding phone or raising a hand). [15] developed a classification model to discern eighteen home activities using data from multiple sensors located on the wrist, chest, and ankle. While these approaches show impressive results, they are not relevant in our problem setting as any solution which uses multiple devices to perform NEAT activity recognition would not be easily adopted by the consumers. In other words, it is imperative to limit to at most one easily accessible device (e.g., commonly available smartphone or smartwatch) for the data collection and prediction.

Solutions developed in the second category (i.e., solutions which use a single device) can be further divided into the following two sub-categories: (i) data is collected at high frequency (e.g., 10Hz or above) [16], [18]–[20] and; (ii) data is collected at low frequency [25]. Solutions which are based on high-frequency data are not suitable for NEAT activity recognition. This is because such solutions would also consume the battery faster. High-frequency data collection and battery consumption are known [7] to be directly proportional to each other. An ideal solution for NEAT activity recognition should be energy efficient as NEAT activities are likely to spread throughout the day, and the consumer would like to collect data for an entire day (at a time) to get meaningful insights into his/her NEAT activities.

[25] works with a very low frequency of 1hz. The device used is a belt-worn smartphone to identify activities. The activities distinguished here are sitting, standing, lying, walking, posture change, and gentle motion. While our work shares some everyday activities (e.g., walking) with this work, we focus on several non-trivial activities (e.g., cooking, mopping, and sweeping), which are essential for a solution for NEAT activity monitoring.

Work was done in the area of transportation mode detection (e.g., [27]–[34]) focused on developing classification models for identifying different kinds of transportation modes. Works have focused on several common transportation modes such as walking, running, car, bus, metro rail, trams, etc. It is important to note that NEAT activity recognition is fundamentally different from transportation mode detection since it involves more nuanced activities, such as cooking, sweeping, and mopping.

IV. Proposed Approach

A. Preprocessing and Features

The raw sensor values are in the form of time series data $\mathcal{T}$, which are divided into a set of overlapping windows ($\mathcal{W}$) for the purpose of training. Two temporally adjacent windows w_i and w_{i+1} in $\mathcal{W}$ can have a degree of overlap defined by the parameter θ. We set the value of θ to be 0 and 0.5. When $\theta = 0$, there is no overlap. When $\theta = 0.5$, there is 50% overlap among the consecutive windows. This overlapping of 50% essentially creates another data point between two otherwise non-overlapping (but temporally adjacent) windows. This, in turn, helps in learning a more robust model by reducing the effect of outlier data. In fact, our experiments also show that all models obtain higher accuracy in the case of overlapping windows (details in Section V). Each window is of length ω. We considered ω to be 2, 4, and 6 seconds.

Accelerometer Individual Axis Features: Accelerometer in a device sensor gives us the rate of change of velocity of that device. Accelerometer gives the output in all three dimensions, i.e., x-axis, y-axis, and z-axis. Therefore, for a specific window $w_i \in \mathcal{W}$, there will be three time series of acceleration values consisting of a_x, a_y, and a_z. We apply five statistics features on each three time series in a window – (1) mode, (2) max, (3) median, (4) lower quartile, and (5) standard deviation. Hence, we have 15 combinations possible, i.e., each axis (3) paired with each of the statistical features (5). Please note that we have considered the raw

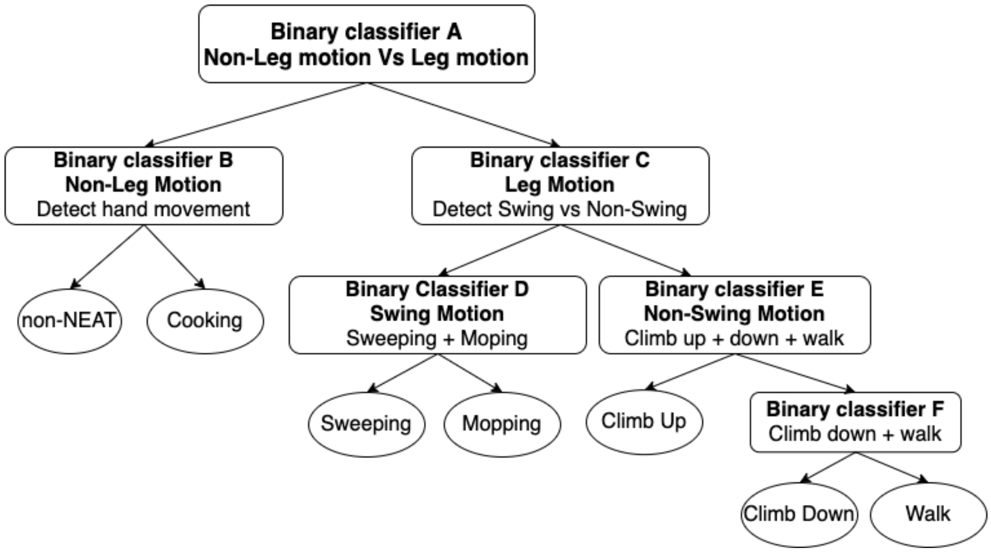


Fig. 2: Proposed hierarchical model for distinguishing the NEAT and Non-Neat activities.

sensors values, and no filter was applied before sending it for calculating the statistical features.

Accelerometer Magnitude Features: This feature is found using the formula $a_{mag} = \sqrt{a_x^2 + a_y^2 + a_z^2}$. So given the individual axis of the accelerometer, we can find out its magnitude for a given time instant. For this feature as well, we find out the same five statistical features (mode, max, median, lower quartile, and standard deviation) for each window $w_i \in \mathcal{W}$. There is no filter used before finding the magnitude here as well.

Gyroscope Individual Axis Features: Gyroscope sensor gives the angular motion speed of the device worn or carried by a person. Just like an accelerometer, a gyroscope also gives a three-dimensional output for the x-axis, y-axis, and z-axis. We denote the resulting time series as g_x, g_y, and g_z. Along with accelerometer values, the orientation of the device given by the gyroscope sensor is also useful in detecting the user's motion. We find the previously mentioned five statistical features for each of the axes given in a time series window having a total of 15 statistical features.

Gyroscope Magnitude Features: Just like accelerometer magnitude, we also find out the overall magnitude of the gyroscope sensor along the individual axis as $g_{mag} = \sqrt{g_x^2 + g_y^2 + g_z^2}$. After computing the magnitude, we preprocess it by finding out the five statistical features (mode, max, median, lower quartile, and standard deviation) for each g_{mag} over the window length. Henceforth, we get five features from this category as well.

B. Proposed Model

We demonstrate our proposed hierarchical learning model for distinguishing the seven NEAT activities, viz., Cooking, Sweeping, Mopping, Walking, Climbing up, Climbing down, and non-NEAT in Figure 2.

As shown in Figure 2, our proposed hierarchical model is a combination of various binary classifiers (A, B, C, D, E, and F), i.e., at each level, we are doing the classification only between two classes. The two classes were formed so that the most alike ones are clubbed together; they were separated from the rest of the same ones. We first check if the action involves a change in the location, i.e., leg motion. If it does not, then we separate it at the root node itself. That is why our topmost level classifier "A" is distinguishing "Cooking" and "non-NEAT" from the rest of the classes. We merge the Cooking and non-NEAT class into one class and the rest into another. On the second level on the left-hand side, we separated the non-leg motion data and made a classifier "B", which distinguishes further into non-NEAT and Cooking. We know that there is a hand movement in cooking with minimal leg motion, unlike non-NEAT with minimal or no hand movement involved. Hence, those are distinguishable among themselves.

In the second level on the right-hand side (Classifier "C"), we separate the hand swing motion from the non-swing motion. If there is a detection of swing motion, we move further to the left subtree and categorize it into Sweeping + Mopping. If there is a detection of non-swing hand motion, we move further to the right subtree and categorize it into the climb up + down + walk. Later, Classifier "D" classifies the data into sweeping and mopping. In the last level, we have distinguished climb up from Climb down + walk using classifier "E" because of similarity in hand motion between climb down and walk class. Later, classifier "F" classifies into climb down and walk.

Implementation of the Learning Model: We have seven different class labels – 1 being the cooking class, 2 being the sweeping class, 3 being the mopping class, 4 being the walking class, 5 and 6 are climb up and climb down respectively, and 7 is the non-NEAT class. We take the training data, perform the masking, and send it through all the available classifiers. The classifier, which gives the best accuracy among all, is chosen as the final classifier for that level. At the root level, we first mask our training data into "17" vs "23456". The data labels of "17" correctly classified are directly given to the classifier "B" for further classification between 1 and 7. Once the "23456" data is passed through the top-level Classifier "A" and correctly classified, it is masked again into "23" vs. "456". Here, the wrongly classified ones are thrown away. The data labels of "23" correctly classified are directly given to a classifier "D" for further classification between 2 and 3. Once the "456" data is passed through the second level classifier "C" and correctly classified, it is again masked into "5" vs "46", and the wrongly classified ones are thrown away. At the last levels, "4" and "6" are given to a classifier "F" for further classification. The best models with the best accuracy on training data are stored for the testing phase at each level.

After the data points reach a leaf node, they are no longer trained further. In the test phase, the data points in the form of windows are passed through each level classifier, and the final labels are matched with the actual labels. Hereafter, the confusion matrix is formed. Note that we discard those window points that do not pass through the correct labels during the training phase. For example, in the training phase, if the data points belonging to Cooking or non-NEAT are classified as "Leg motion", then those data points are discarded before making a new masked training data of "Swing motion" vs. "Non-Swing motion." We did this for all the lower-level classification since it ensures the quality of the final model without running into an overfitting problem. Moreover, those discarded data points were anyway not crucial in the training phase.

We used the following classical Learning algorithms at each level - (1) K-Nearest Neighbour with k=5), (2) Multi-Layer Perceptron (3-hidden layers), (3) Support Vector Machine (with RBF Kernel), (4) Random Forest, and (5) XGB. XGB has the maximum accuracy.

V. Experimental Analysis

Dataset Used: We have collected the data from a real-time home environment. We collected all of our data using a Fossil sports smartwatch, running an Android OS and powered by 'Wear OS by Google', where all four volunteers used to wear a watch on their dominant hand (mostly right hand). The smartwatch application contained a list of activities and a designated start and stop button at the top. Before starting any activity, the volunteer noted down the ground truth labels by clicking on the designated button created in the smartwatch application for that activity. So the volunteers were asked to press the activity button (cooking, sweeping, mopping, walking, climbing up, climbing down, and non-NEAT) before pressing the start button. The stop button was pressed once the activity was meant to be stopped. During the data collection, the volunteer was asked to keep the mobile phone with him in his/her pocket or hand since, after every few minutes, we were transferring the data from the smartwatch to a smartphone using Bluetooth. The data transfer limit and timing from the smartwatch to the smartphone can be configured and is purely at our discretion. As stated previously, we were not using the "GPS" information or any kind of "tag" information from the watch or phone. We collected around four to five hours of data, but due to the problem of imbalanced activity classes, we could consider only three and half hours of data. This imbalance was natural since the time spent for a person to climb up and down was much lesser than the time spent in the kitchen (cooking) or doing any other household activity like sweeping or mopping or even for instance sitting idle / watching TV / working on a laptop. Henceforth, we took equal instances from each class so that the problem of miss-classification due to dominant class does not occur.

TABLE III: Total Training Data

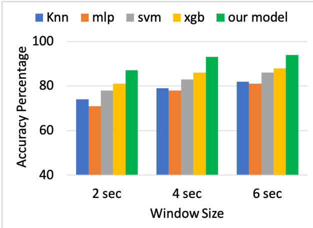
Window Length	1Hz (50% overlap)	1Hz (0% overlap)
2 seconds	11335	5670
4 seconds	5670	2835
6 seconds	3780	1890

Table III shows the number of instances belonging to each window length and overlapping percentage parameter. These numbers decrease as we increase the window length. This is because we are expanding the time frame for which we want to find out the input statistical features and their corresponding output labels. We did not want to go further with the window length since it does not show significant improvement. Moreover, in the near future, if we deploy this model to a server, it will delay the output of class labeling.

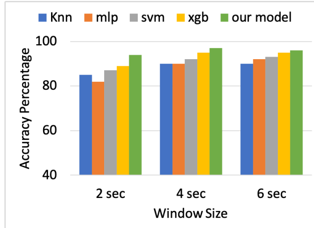
Candidate Algorithms: We have compared our Hierarchical Model with the Flat Model. When we say Flat model, we refer to the in-built classifiers, which are readily available in common machine learning libraries (e.g., sklearn in Python). The flat models attempt to learn a single decision boundary amongst all our classes of interest. For the flat models, we chose the following classifiers - a) KNN (K-nearest neighbor with k=5) b) MLP (multi-layer perceptron with 3 hidden layers having 13 neurons each) c) SVM (support vector machine with kernel='rbf') d) Logistic Regression e) Random Forest f) Gaussian Naive Bayes g) XGB (Extreme Gradient Boosting).

In the case of our hierarchical model, we tried different classifiers for each binary classifier mentioned in the tree (Figure 2) and chose the classifier which gives the best accuracy. In our implementation, we made this decision on the basis of the training accuracy as the test data is considered to be "hidden" by definition. In our experiments, we found out that in most of the cases, XGB was chosen at each level. We use python language for the implementation of our models.

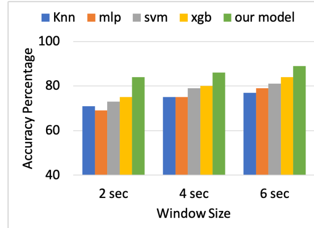
Training and Evaluation Metrics: We have used window lengths of 2sec, 4sec, and 6sec in our experiments. The window overlap parameter θ was varied across 0.5 and 0 (i.e.,



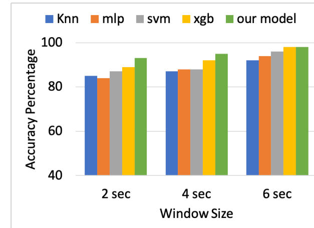
(a) All features were used. θ = 0.50,frequency = 1Hz



(b) All features were used. θ = 0.50,frequency = 5Hz



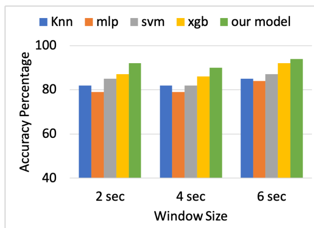
(c) All features were used. $\theta = 0$, frequency = 1Hz.



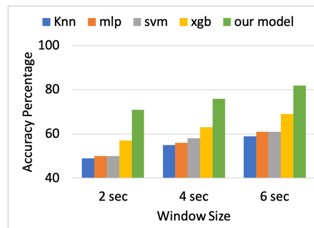
(d) All features were used. $\theta = 0$, frequency = 5Hz

Fig. 3: Effect of window length on overall accuracy when all features are used.

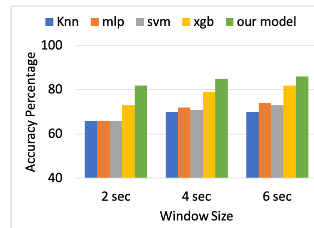
(a) Only Accelerometer were used. $\theta = 0.5$, frequency = 1Hz



(b) Only Accelerometer were used. $\theta = 0.5$, frequency = 5Hz



(c) Only Gyroscope were used. $\theta = 0.5$, frequency = 1Hz



(d) Only Gyroscope were used. $\theta = 0.5$, frequency = 5Hz

Fig. 4: Effect of window length on overall accuracy when only accelerometer or only gyroscope features are used.

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.8	0.9	0.86	0.83	0.81	0.9	0.99	87%		0.81	0.83	0.86	0.78	0.68	0.68	0.96	80%
Only Accelo	0.76	0.84	0.82	0.75	0.77	0.84	0.98	82%		0.75	0.68	0.75	0.69	0.53	0.58	0.97	71%
Only Gyro	0.34	0.78	0.8	0.57	0.73	0.71	0.96	71%		0.37	0.57	0.65	0.55	0.46	0.44	0.93	57%

(a) $\theta = 0.50$, frequency = 1Hz, window = 2seconds

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.9	0.96	0.93	0.89	0.91	0.93	0.99	93%		0.84	0.88	0.86	0.85	0.75	0.79	0.98	85%
Only Accelo	0.82	0.88	0.84	0.82	0.82	0.89	0.98	87%		0.81	0.83	0.82	0.81	0.65	0.7	0.96	80%
Only Gyro	0.45	0.89	0.85	0.59	0.74	0.74	0.99	76%		0.46	0.65	0.71	0.58	0.53	0.51	0.95	63%

(b) $\theta = 0.50$, frequency = 1Hz, window = 4seconds

Fig. 5: F1-scores of Our Hierarchical Model and Flat Classifier XGBoost (XGB) for $\theta = 0.50$ and frequency = 1 Hz.

no overlap amongst temporally consecutive windows). For a given set of windows $\mathcal{W}$ with its corresponding θ values, we divide it into a train and a test data set in the ratio of $4:1$, i.e., 80% of the data is allocated for training, and the rest 20% is allocated for testing. To get reliable results, we divided our given dataset into training and test portions 10 times (randomly). Following this, we trained (and tested the learned model) on each of the previously mentioned 10 splits. Finally, we report the average of F-scores and Accuracies obtained across those 10 test datasets.

Consequences of varying window length on the final accuracy: Figure 3 and Figure 4 illustrate the results of this experiment. The test accuracy of our model is shown in the form of bar graphs along with the rest of the flat models. In this experiment, we tried window lengths of 2sec, 4sec, and 6secs. Overlap parameter θ was taken as 0 and 0.50. Figure 3 shows the results corresponding to the case where all features were used in training. Whereas, Figure 4 displays the results corresponding to cases where only accelerometer used (Figure 4a and Figure 4b), or only gyroscope was used (Figure 4c and Figure 4d). It is important to note that, the performance of all the models decreased when only gyroscope features were used. Overall, we observed that our proposed model outperformed the alternative approaches consistently. Moreover, our experimental results also indicate

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.75	0.84	0.83	0.71	0.73	0.88	0.92	83%		0.78	0.79	0.82	0.65	0.68	0.63	0.91	76%
Only Accelo	0.71	0.63	0.71	0.68	0.7	0.83	0.94	75%		0.72	0.58	0.74	0.73	0.48	0.55	0.89	71%
Only Gyro	0.41	0.68	0.78	0.51	0.68	0.76	0.93	69%		0.34	0.52	0.61	0.57	0.38	0.39	0.87	53%

(a) $\theta = 0$, frequency = 1Hz, window = 2seconds

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.8	0.89	0.89	0.76	0.78	0.92	0.99	86%		0.81	0.82	0.87	0.7	0.7	0.68	0.96	80%
Only Accelo	0.74	0.69	0.74	0.71	0.74	0.87	0.98	78%		0.76	0.61	0.78	0.75	0.52	0.58	0.92	73%
Only Gyro	0.44	0.73	0.82	0.55	0.72	0.79	0.98	72%		0.36	0.56	0.64	0.6	0.4	0.4	0.9	56%

(b) $\theta = 0$, frequency = 1Hz, window = 4seconds

Fig. 6: F1-scores of Our Hierarchical Model and Flat Classifier XGBoost (XGB) for $\theta = 0$ and frequency = 1Hz.

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.89	0.96	0.93	0.91	0.91	0.96	1	94%		0.92	0.95	0.94	0.93	0.86	0.89	0.98	93%
Only Accelo	0.89	0.93	0.9	0.92	0.89	0.94	1	92%		0.88	0.88	0.89	0.9	0.77	0.83	0.98	88%
Only Gyro	0.58	0.87	0.89	0.71	0.82	0.81	0.99	82%		0.57	0.77	0.79	0.74	0.69	0.64	0.95	74%

(a) $\theta = 0.50$, frequency = 5Hz, window = 2seconds

	F1-Scores of Our Model									F1-Scores of XGB Model							
	cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy		cook	Sweep	Mop	Walk	CU	CD	non-NEAT	Overall Accuracy
All Features	0.96	0.98	0.97	0.95	0.95	0.97	0.99	97%		0.93	0.94	0.95	0.94	0.9	0.91	0.98	93%
Only Accelo	0.88	0.83	0.82	0.92	0.9	0.96	0.98	90%		0.87	0.82	0.81	0.91	0.82	0.86	0.97	86%
Only Gyro	0.61	0.92	0.89	0.79	0.86	0.86	0.99	85%		0.68	0.83	0.81	0.77	0.77	0.78	0.96	80%

(b) $\theta = 0.50$, frequency = 5Hz, window = 4seconds

Fig. 7: F1-scores of Our Hierarchical Model and Flat Classifier XGBoost (XGB) for $\theta = 0.50$ and frequency = 5Hz.

that all models perform better when windows overlap (i.e., $\theta = 0.50$). A similar increase in performance with an increase in overlap has also been reported in other works [35]. This is possibly due to the fact that in the case of overlapping windows, the effect of outliers is reduced as "good data" sort of "spawns" more "good data" (inadvertently) through the process of overlapping the windows.

F1-scores of individual classes: The illustration of f1-scores for each of the classes can be seen in Figure 5, Figure 6 and Figure 7. The left side represents our model, and the right side represents the flat model. We have shown only XGBoost (XGB) since it was best among all flat models. We have demonstrated f1-scores for three different types of feature sets. In the first set, we have used all statistical features (i.e., accelerometer x, y, z-axis, the magnitude of accelerometer, gyroscope x, y, z-axis, and the magnitude of a gyroscope). In the second set, we have used only accelerometer features (i.e., individual axis and its magnitude), and in the third set, we have considered only gyroscope features (same as an accelerometer). Our experiments indicate the following: (a) Our proposed hierarchical model outperforms XGBoost (and other flat models) for all the parameter values of θ, window lengths, and data sampling frequency explored in the experiments. (b) Best accuracy (and F1 scores) is obtained when we use both accelerometer and gyroscope features together. (c) As expected, both our model and XGBoost perform better when high-frequency data (5Hz) is used. (d) Both models perform better with $\theta = 0.50$ and window length 4secs.

VI. Conclusion

Distinguishing the typical home activities in a home environment from smartwatch sensor data (low-frequency sampling data) is not a trivial problem. Presently, the work closely related to our application deals with high frequency or multiple sensors. Moreover, they do not include basic home activities like sweeping, mopping, or cooking on smartwatch. In contrast, our proposed model can distinguish these seven activities from each other using data sampled at low frequency (1Hz). Our experiments show that the proposed approach gives better overall accuracy compared to all flat models. Our work demonstrates the potential of a wrist smartwatch for identifying the typical NEAT activities done in a home setting. In the future, we would like to work on converting the time spent in NEAT activities into calories burnt.

Acknowledgement: Authors would like to thank IIT Ropar and OPERA grant from BITS Pilani Goa for their support in this work.

References

[1] G. Milette and A. Stroud, *Professional Android Sensor Programming*, 1st ed. Birmingham, UK, UK: Wrox Press Ltd., 2012.

[2] J. A. Levine, "Non-exercise activity thermogenesis (neat)," *Best Practice and Research Clinical Endocrinology and Metabolism*, vol. 16, no. 4, pp. 679–702, 2002.

[3] P. A. Villablanca, J. R. Alegria, F. Mookadam, D. R. Holmes, R. S. Wright, and J. A. Levine, "Nonexercise activity thermogenesis in obesity management," *Mayo Clinic Proceedings*, vol. 90, no. 4, pp. 509–519, 2015.

[4] J. Levine, M. Vander Weg, J. Hill, and R. Klesges, "Non-exercise activity thermogenesis: The crouching tiger hidden dragon of societal weight gain," *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 26, no. 4, pp. 729–736, 2006.

[5] H. Hamasaki, O. Ezaki, and H. Yanai, "Nonexercise activity thermogenesis is significantly lower in type 2 diabetic patients with mental disorders than in those without mental disorders," *Medicine*, no. 2, January 2016.

[6] N. Owen, A. Bauman, and W. Brown, "Too much sitting: a novel and important predictor of chronic disease risk?" *British Journal of Sports Medicine*, vol. 43, no. 2, pp. 81–83, 2009.

[7] L. Zheng and et al., "A novel energy-efficient approach for human activity recognition," *Sensors*, vol. 17, no. 9, 2017.

[8] A. Stisen and et al., "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in *Proc. of the Conference on Embedded Networked Sensor Systems*, ser. SenSys. ACM, 2015, pp. 127–140.

[9] X. Jiang and et al., "Air: Recognizing activity through ir-based distance sensing on feet," in *Proc. of the 2016 ACM Intl. Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 97–100.

[10] A. Mannini, "Activity recognition using a single accelerometer placed at wrist or ankle (vol 45, pg 2193, 2013)," *MEDICINE AND SCIENCE IN SPORTS AND EXERCISE*, vol. 47, no. 2, pp. 448–449, 2015.

[11] M. Zhang and A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. of the 2012 ACM Conf. on Ubiquitous Computing*, 2012, pp. 1036–1043.

[12] K. Murao and T. Terada, "A recognition method for combined activities with accelerometers," in *Proc. of the Intl. Joint Conf. on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp '14 Adjunct, 2014, pp. 787–796.

[13] H. Guo and et al., "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *Proc. of Intl. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1112–1123.

[14] S. Inoue, N. Ueda, Y. Nohara, and N. Nakashima, "Mobile activity recognition for a whole day: Recognizing real nursing activities with big dataset," in *Proc. of Intl. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 1269–1280.

[15] H. Ponce and et al., "A novel wearable sensor-based human activity recognition approach using artificial hydrocarbon networks," *Sensors*, vol. 16, no. 7, p. 1033, 2016.

[16] L. Sun and et al., "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous Intelligence and Computing*. Springer, 2010, pp. 548–562.

[17] S. Thiemjarus, A. Henpraserttae, and S. Marukatat, "A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone," in *2013 IEEE International Conference on Body Sensor Networks*. IEEE, 2013, pp. 1–6.

[18] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235 – 244, 2016.

[19] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450 – 457, 2014, the 9th Intl. Conf. on Future Networks and Communications (FNC'14)/The 11th Intl. Conf. on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops.

[20] X. Sun, L. Qiu, Y. Wu, and G. Cao, "Actdetector: Detecting daily activities using smartwatches," in *2017 IEEE 14th Intl. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 1–9.

[21] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.

[22] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.

[23] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, pp. 138 – 154, 2014.

[24] K. Sankaran and et al., "Using mobile phone barometer for low-power transportation context detection," in *Proc. of Conf. on Embedded Network Sensor Systems*, ser. SenSys '14, 2014, pp. 191–205.

[25] S. Zhang, P. McCullagh, C. Nugent, and H. Zheng, "Activity monitoring using a smart phone's accelerometer with hierarchical classification," in *2010 Sixth International Conference on Intelligent Environments*, 2010, pp. 158–163.

[26] E. M. Tapia and et al., "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *11th IEEE Intl. Symp. on Wearable Computers*, 2007, pp. 37–40.

[27] L. Stenneth and et al., "Transportation mode detection using mobile phones and gis information," in *Proceedings of the ACM SIGSPATIAL*. ACM, 2011, pp. 54–63.

[28] R. C. Shah and et al., "Classifying the mode of transportation on mobile phones using gis information," in *Proc. of Intl. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 225–229.

[29] S. Hemminki and et al., "Accelerometer-based transportation mode detection on smartphones," in *Proc. of Conf. on Embedded Networked Sensor Systems*, ser. SenSys '13. ACM, 2013, pp. 13:1–13:14.

[30] H. Xia and et al., "Using smart phone sensors to detect transportation modes," *Sensors*, vol. 14, no. 11, pp. 20 843–20 865, 2014.

[31] X. Liang and G. Wang, "A convolutional neural network for transportation mode detection based on smartphone platform," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 338–342.

[32] M.-C. Yu and et al., "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1429–1440, Aug. 2014.

[33] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *Proc. of the 24th ACM Intl. Conf. on Multimedia*, ser. MM '16. ACM, 2016, pp. 392–396.

[34] S. Fang, Y. Fei, Z. Xu, and Y. Tsao, "Learning transportation modes from smartphone sensors based on deep neural network," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, 2017.

[35] A. Dewan and et al., "A hierarchical classifier for detecting metro-journey activities in data sampled at low frequency," in *MoMM 2019: The 17th Intl. Conf. on Advances in Mobile Computing & Multimedia*. ACM, 2019, pp. 46–55.