ERSP: An Energy-efficient Real-time Smartphone Pedometer

Thomas Olutoyin Oshin, Stefan Poslad
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, England
Email: {thomas.oshin, stefan}@eecs.qmul.ac.uk

***Abstract*—Smart devices such as Apple's iPod nano (5th generation), Nike+, and existing smartphone applications can provide the functions of a pedometer using the accelerometer. To achieve a high accuracy the devices must be worn on specific on-body locations such as on an armband or in footwear. Generally people carry smart devices such as smartphones in different positions, thus making it impractical to use these devices due to the reduced accuracy. Using the embedded smartphone accelerometer in a low-power mode we present an algorithm named Energy-efficient Real-time Smartphone Pedometer (ERSP), which accurately and energy-efficiently infers the real-time human step count within 2 seconds using the smartphone accelerometer. Our method involves extracting 5 features (4 novel and 1 derived) from the smartphone 3D accelerometer without the need for noise filtering or specific smartphone on-body placement and orientation; ERSP classification accuracy is approximately 94% when validated using data collected from 17 volunteers.**

Index Terms*—Pedometer, Accelerometer, Smartphone, Activity classification

I. Introduction

Smartphones provide sophisticated real-time sensor data for processing. Researchers have studied a large number of sensors such as accelerometer, gyroscope, rotation vector, and orientation sensors in human step count projects. Of these the accelerometer is the most valuable non-transceiver sensor used to provide the data for activity monitoring as it gives more information about movement forces [3]. Hence the core focus of this paper is on using solely the smartphone accelerometer for human step count. The accelerometer has three key advantages over transceiver based location signal sensors such as GPS. First, low energy consumption of 60 mW [2]. Second, there is no delay when starting the accelerometer, however receiving location updates in GPS depends on the start mode. In a hot start mode the Termed-Time-to-Subsequent-Fix (TTSF) is about 10 seconds and in a cold start mode the Time-To-First-Fix (TTFF) could take up to 15 minutes. Third, sensor readings are continuously available with the accelerometer as compared to GPS and Wi-Fi which could be obstructed from signals transmitted by GPS satellites and being out of range of Wi-Fi signals respectively.

Human activity classification using smartphones requires a mobility state recognition technique that can function regardless of the alignment of the smartphone because placing accelerometers on specific parts of the body makes it impractical for use in the real-world. Acceleration data differs for similar activities, thus making it more difficult to finely secernate between certain types of activity. Restrictions have been found in the range of mobility activities identified by use of a single sensor and; due to the complexity of human mobility and noise of sensor signals, mobility classification algorithms tend to be probabilistic [1]. They have instead designed a multimodal sensor board that simultaneously captures data from multiple sensors. A major challenge in the design of ubiquitous, context-aware smartphone applications is the development of algorithms that can detect the human activity using noisy and equivocal sensor data [10].

We present a method named Energy-efficient Real-time Smartphone Pedometer (ERSP), an Android based smartphone application to accurately count human steps. The novelties of this research as compared to existing systems are: 1) ERSP extracts five features (4 novel and 1 derived) from the accelerometer data. 2) This system employs an energy-efficient light-weight mathematical model to process in real-time the activity accelerometer data without need for noise filtering and works regardless of the smartphone on-body placement and orientation.

II. Related Work

Smartphone based context-aware sensing is a hot research topic. Several smartphone sensing based architectures exists to classify pedestrian step count. Architectures by [1]-[11] all use the smartphone as a major system component.

Shyi-Shiou et al. [5] present an Android based Pedometer system which uses the accelerometer and orientation sensors to detect the user's walking motion. The system provides three main action modes: time-based, distance-based, and count-based. The time-based mode notifies the user once a configurable walking time is reached, distance-based mode notifies the user once a configured walking distance is reached, and count-based mode notifies the user once a configured number of walking steps are achieved. This architecture requires the smartphone to be worn on the waist.

Hongman et al. [6] present an Android based Pedometer system which uses the accelerometer and orientation sensors. The architecture studies the top (peak) and bottom (trough) of the acceleration wave. The paper compares a single acceleration sensor vs. a multi-sensor pedometer. For the single acceleration sensor, a configured threshold is used to filter the accelerometer noise. Two methods to determine the threshold

978-1-4799-0652-9/13/$31.00

are: fixed (using test data) and dynamic (real-time user data). There results show that a multi sensor (accelerometer and orientation) is more accurate and presents more differences in terms of acceleration in the gravity direction. The human steps are calculated using the wave crest (trough). The sensor sampling frequency used is 20Hz. The architecture isn't real-time as it expects consecutive step count time intervals to be $0.2 \sim 2$ seconds.

Sugimori et al. [7] presents an automatic human walking authentication system using the embedded smartphone accelerometer. They study forward steps using the left and right leg with the smartphone worn on the waist. Smartphone on-body placements also considered are front and rear pockets using 5 male subjects. The features extracted for walking recognition are the highest and lowest value of the 3-axis compound acceleration and the spectrum processed by FFT. The experiment uses three classifiers. The classifiers are C4.5 decision tree, naive bayes and, support vector machine. Results show naive bayes was the best performing classifier with classification accuracy of approximately 90%.

Inoue et al. [8] propose a two-tier approach involving multi-level segmentation and activity recognition using microphone sound and accelerometers. The data features extracted include mean, frequency-domain energy, and frequency-domain entropy of each axis. Also the correlation of the combined axis was extracted. The architecture was validated using data gathered from nurses working at a hospital with the smart device placed in their breast pockets with a fixed direction. The accelerometer sensor sampling frequency used was 20Hz. Using 216 IPod Touches as smartphones by several users, sensor data was gathered 35310 times for several human activities during a 14 month period. The paper authors state the activity recognition results were poor due to the following four reasons. First, the algorithm required the smartphone to be fixed to the body, but this wasnt the case. Second, similar activities classes, e.g., "eat.sit"-"sit" and "sit"-"train.sit" are often misclassified. Third, disparity in data gathered for similar activities by different participants. Fourth, misinterpretation and low data quality from users.

Henpraserttae et al. [9] investigates two major issues in using the embedded smartphone accelerometer for continuous activity monitoring. The issues identified are the smartphone orientation and on-body placement. They propose a two-step signal transformation method to generate uniform signals from different placements and orientations. The first step involves pre-processing input signals by normalization with the mean and standard deviation. The second step applies eigen-decomposition to the covariance matrix of the projected data. The data gathering was done by 10 participants for 6 daily activities using 16 different smartphone orientations and 3 smartphone on-body placements (front shirt pocket, front trouser pocket, and front waist). The accelerometer sampling rate used is 50Hz. The features extracted are mean, standard deviation, and variance magnitude. The results using a single device orientation show that classification accuracy with the proposed signal transformation method was approximately 42% to 51% better than without signal transformation. The on-body placement results show that attaching the smartphone to the waist had highest recognition accuracy followed by front shirt pocket then front trouser pocket.

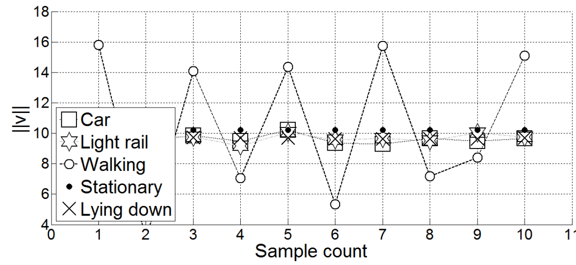


Fig. 1: $||v||$ plot for selected urban activities.

The surveyed systems can classify human activities such as calculate the pedestrian step count over a given time period. None of these architectures can in real-time energy-efficiently calculate the human step without need for sensor noise filtering and specify smartphone on-body placement and orientation. For e.g., most of the surveyed systems sample the accelerometer at a rate $\geq$ 20Hz which is energy-inefficient.

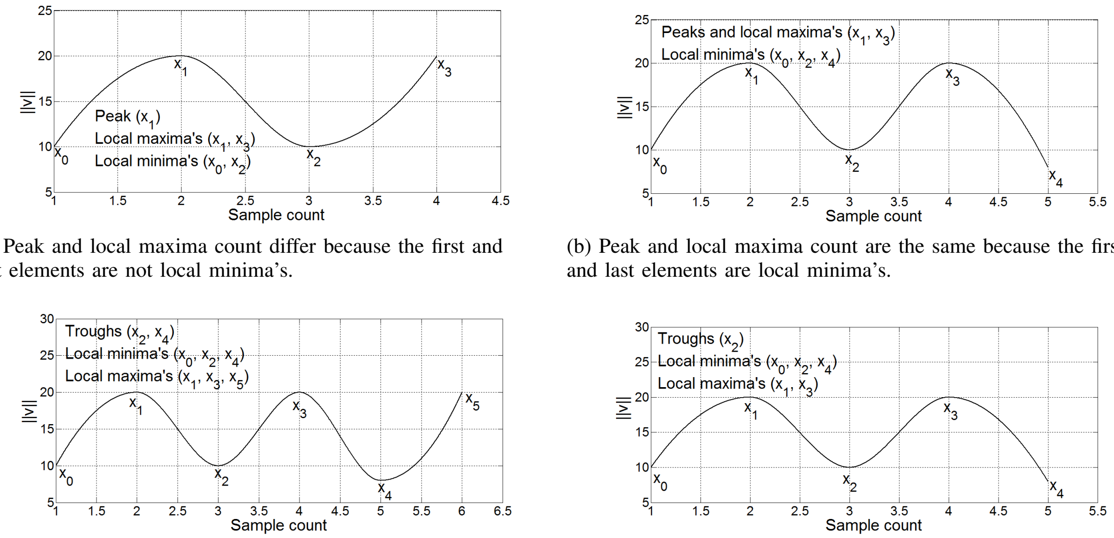
III. Methodology

In the time-domain, activities generate accelerometer data readings that are patterns of varying peaks and troughs (waves). Several state of the art existing methods study these patterns. Our method also studies the accelerometer magnitude waves because our experiments found that some time-domain feature characteristics are unique across activities. Fig. 1 shows the $||v||$ plot for selected urban human activities. As shown the activities all present different accelerator oscillations.

Using the Sun SPOT[1] accelerometer we were able to detect patterns based on the vertical (y) axis. The vertical axis presented the largest deviation because of the fixed orientation of the device during the experiments. The case is the contrary for accelerometer data gathered for similar activities with different smartphone on-body placements. In such cases, more than one axis had to be taken into consideration for a pattern match to be demodulated. To amalgamate (x, y, z) readings regardless of the smartphone orientation we make use of the magnitude of the accelerometer signal vector ($||v||$). Given the accelerometer readings (x, y, z), the $||v||$ is calculated using the formula $||v|| = \sqrt{x^2 + y^2 + z^2}$. It should be noted that once 8 accelerometer samples (2 seconds) are gathered, the processing time to calculate human step count complete in $\leq$ 15 milliseconds allowing ERSP to perform a real-time human step count classification. In the rest of this sub-section, we detail the steps undertaken to detect the total human step count within 2 seconds using the smartphone 3D accelerometer readings. The following light-weight computational features are extracted as classifiers from the accelerometer readings:

1) Peak (P): - this is the count of peaks every 2 seconds (8 accelerometer samples). The Peak is the local maxima if the first and last elements are local minima's. This is detailed in Figures 2a and b. The acceleration peak is calculated as

[1] Sun SPOT http://www.sunspotworld.com/



(a) Peak and local maxima count differ because the first and last elements are not local minima's.

(b) Peak and local maxima count are the same because the first and last elements are local minima's.

(c) Trough and local minima count differ because the first element isn't a local maxima.

(d) Trough and local minima count differ because the first and last elements are local minima's.

Fig. 2: Peak and trough vs. local maxima and minima.

follows:

$$Q_i = \begin{cases} 1, & \text{if } (x_{i+1} > x_i) \text{ and } (x_{i+2} < x_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$
$$P = \sum_{i=0}^{n-2} (Q_i) \tag{1}$$

2) Trough (T)*:* - this is the count of troughs every 2 seconds (8 accelerometer samples). The trough is the local minima if the first and last elements are local maxima's. This is detailed in Figures 2c and d. The acceleration trough is calculated as follows:

$$Q_i = \begin{cases} 1, & \text{if } (x_{i+1} < x_i) \text{ and } (x_{i+2} > x_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$
$$T = \sum_{i=0}^{n-2} (Q_i) \tag{2}$$

x_i is the $||v||$ of each accelerometer data point.
n is the total number of data points.
P is the total numbers of peaks.
T is the total numbers of troughs.

3) T_{PT}*:* - the sum of the total peak (P) and trough (T) acceleration values.

$$T_{PT} = P + T \tag{3}$$

4) mm*:* - the difference between the maximum peak and minimum trough every 2 seconds (8 accelerometer samples). The following is the mm equation:

$$mm = max_{\forall i(0<i\leq m)}(max_{\forall j(0<j\leq n)}(G_i^P - G_j^T)) \tag{4}$$

where i and j are integers.

G^P is the group of peak values, which has m elements.
G^T is the group of trough values, which has n elements.

5) P_{mm}*:* The difference between the maximum and minimum peak values given the T_{PT} range for the activity. Algorithm 2 details the pseudocode to generate the static range threshold per user activity.

$$P_{mm} = max_{\forall i(0<i\leq m)}(max_{\forall j(0<j\leq m)}(G_i^P - G_j^P)) \tag{5}$$

where i and j are integers.

G^P is the group of peak values, which has m elements.

A. Features threshold

The $T_{PT}, mm,$ and P_{mm} range thresholds are required to accurately align the algorithm to the user's step pattern. The algorithm must be able to adapt to the various variations while a user is performing a step activity. Human steps could occur based on walking, jogging, marching etc.

To personalize the application based on a specific activity, the user performs the activity for a one-off time of 14 seconds (56 accelerometer samples). 14 seconds was chosen because a minimum of 56 accelerometer samples are required to cover the T_{PT} range from 0 to 6. We selected the optimal value of 8 accelerometer samples which occurs every 2 seconds after iterations involving 1 second (4 samples) to 62.5 seconds (250 samples), because it presented the largest differences of $T_{PT}, mm,$ and P_{mm} within the shortest computation time. It should be noted that given 8 accelerometer samples the T_{PT} range is between 0 and 6.

1) T_{PT} *range estimation:*

Estimate the Gaussian distribution for T_{PT}. Calculate the $||v||$ for each (x, y, z) sample. At intervals of 8 samples extract the peaks and troughs for 7 iterations. Sum the count of peaks and troughs for each iteration and aggregate the T_{PT} value based on the percentage of occurrences within 0 to 6.

Given the Gaussian distribution, if the sum of the distribution for 2 or 3 consecutive T_{PT} values is $\geq 90\%$ then the T_{PT} range is between the corresponding minimum and maximum T_{PT} values. The 90% threshold was chosen based on the analysis of accelerometer data gathered from 17 volunteers. Our analysis show for walking, the distribution percentage sum for T_{PT} values 5 and 6 is $\geq 98\%$ and for jogging with T_{PT} values 4 and 5 is $\geq 96\%$. The T_{PT} range for both activities is $(5,6)$ and $(4,5)$ respectively. The pseudocode to calculate the T_{PT} range is shown in algorithm 1.

Algorithm 1 T_{PT} range estimation pseudocode

Require: $A = \{x_i \ldots x_n\}$ // Array with T_{PT} Gaussian // distribution.
Require: $S_A = size(A)$ // array size of A.
Ensure: $E = \emptyset$;
Ensure: $i = 0; k = 0$
  for all v in $\{A_0, A_1, \ldots A_{(S_A-1)}\}$ **do**
    $E_k = \sum_{i=k}^{k+1} v_i$ // Insert the sum of 2 consecutive v // elements in E
    $k = i$
  end for
  $M_E = max(E)$ // Maximum element in E
  if $M_E \geq 90$ **then**
    return $index(M_E, M_{E+1})$ // return index of the max // E and the next element.
  else
    reset(E) // reset to an empty set.
    for all v in $\{A_0, A_1, \ldots A_{(S_A-2)}\}$ **do**
      $E_k = \sum_{i=k}^{k+2} v_i$ // Insert the sum of 3 consecutive v // elements in E
      $k = i$
    end for
    $M_E = max(E)$
    if $M_E \geq 90$ **then**
      return $index(M_E, M_{E+1}, M_{E+2})$
    end if
  end if

2) mm range estimation:

The range between the minimum peak and trough values min (P,T) and the corresponding maximum values max (P,T) over the 14 seconds (56 accelerometer samples) personalization phase.

3) P_{mm} range calculation:

This is the range between the minimum and maximum peak values given the T_{PT} range for the activity. Algorithm 3 details the pseudocode to generate P_{mm} range given the $||v||$ data for the user activity.

Algorithm 3 shows the pseudocode to calculate the human step count given the values of P (peak), m^{mod}_{min} (min (P,T)), m^{mod}_{max} (max (P,T)), k^{mod}_{min} (min T_{PT}), k^{mod}_{max} (max T_{PT}), and p^{mod}_{min} (min p_{mm}), p^{mod}_{max} (max p_{mm}).
k^{mod}_{min} and k^{mod}_{max} are derived from algorithm 1 and; p^{mod}_{min} and p^{mod}_{max} are derived from algorithm 2.

IV. Results

Several mobility classification systems require sensors such as accelerometers to be placed on specific parts of the body [4].

Algorithm 2 P_{mm} range pseudocode

Require: $A = \{x_i \ldots x_n\}$ // Peak values. i = 0, n = 6.
Require: $S_A = size(A)$ // array size of A.
Ensure: $E = \emptyset; i = 0; k = 0$
  for all v in $\{A_0, A_1, \ldots A_{(S_A-1)}\}$ **do**
    $E_k = \sum_{i=k}^{k+1} v_i$ // Sum of 2 consecutive elements in E.
    $k = i$
  end for
  $M_E = max(E)$
  if $M_E \geq 90$ **then**
    $min_p = min(M_E, M_{E+1})$
    $max_p = max(M_E, M_{E+1})$
    return (min_p, max_p) // p_{mm}
  else
    reset(E) // reset to an empty set.
    for all v in $\{A_0, A_1, \ldots A_{(S_A-2)}\}$ **do**
      $E_k = \sum_{i=k}^{k+2} v_i$ // Sum of 3 consecutive elements in E.
      $k = i$
    end for
    $M_E = max(E)$
    if $M_E \geq 90$ **then**
      $min_p = min(M_E, M_{E+1}, M_{E+2})$
      $max_p = max(M_E, M_{E+1}, M_{E+2})$
      return (min_p, max_p) // p_{mm}
    end if
  end if

Algorithm 3 Pseudocode to calculate the human step count.

  while $(mod \neq null)$ **do**
    if $((mm \geq m^{mod}_{min} \wedge mm < m^{mod}_{max}) \wedge (T_{pt} \geq k^{mod}_{min} \wedge T_{pt} \leq k^{mod}_{max}) \wedge (p_{mm} \geq p^{mod}_{min} \wedge p_{mm} < p^{mod}_{max}))$ **then**
      $steps = p$; // human step count = number of peaks
    end if
    $\forall$ $mod \in$ {human activies e.g., walking, etc.}
  end while

Our method is relatively insensitive to the smartphone on-body placement and orientation. ERSP was validated against the smartphone accelerometer data gathered from 17 able-bodied volunteers for a minimum of 20 steps in 4 different smartphone on-body positions. The on-body smartphone positions are: palm, front trouser pocket, backpack, and top jacket pocket. Users 1 to 13 were permitted to carry the smartphone regardless of the on-body placement and smartphone orientation. Users 14 and 17 had to place the smartphone in the 4 previously identified on-body positions. This allowed us to study the differences in accelerometer readings based upon different on-body placements. Fig. 3 shows a graph of the $||v||$ for 10 real-world human steps. The experiments involved real world data gathered using Android based smartphones. Fig. 4 shows a screenshot of the ERSP Android application.

There was no accelerometer data noise filtering or data simulation. Noise filtering isn't practical because the algorithm only considers the extractable features T_{PT} and mm over a 2 seconds window. We found noise reduction using kalman filtering stymied the computational features.

The validation process involved using 10 different models of Android based smart devices. The smart devices include

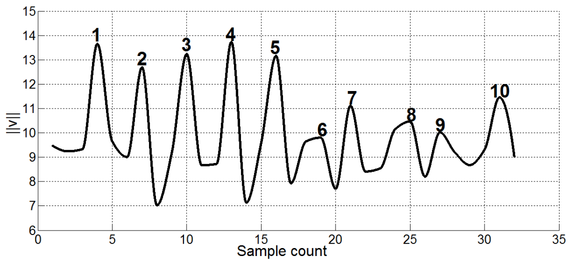


Fig. 3: Real-world smartphone accelerometer data showing 10 human steps.

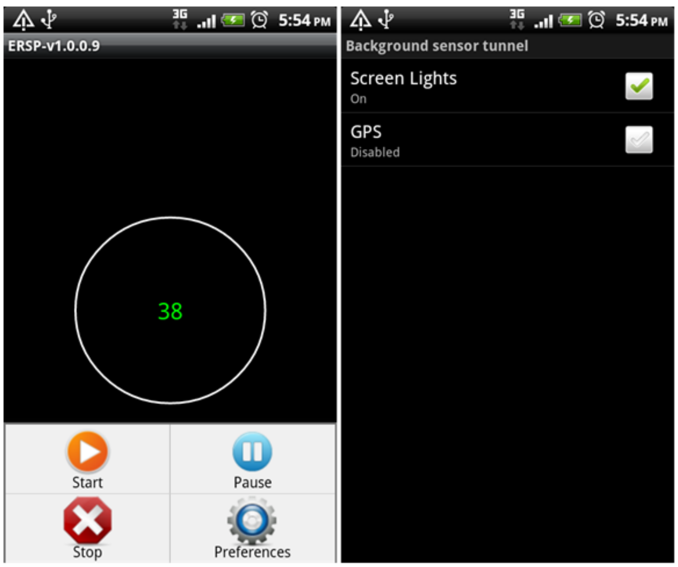


Fig. 4: Screenshot of ERSP Android application.

HTC Desire HD running Android version 2.3.5, Samsung Galaxy S smartphone running Android version 2.1-update1, Samsung S II running Android version 4.0.3, Samsung Galaxy Note I running Android version 4.0.4, Hauwe 300C running Android version 2.3, Lynk 3D II running Android version 2.3.3, Samsung Galaxy II running Android version 4.0.4, Samsung Galaxy Tab GT-P5110 running Android version 4.0.4, Sony Xperia U ST25i running Android version 4.0.4, and Samsung Galaxy Tab GT-N8010 running Android version 4.1.1. The devices have a dynamically user-selectable full scale acceleration range of $\pm 2g/\pm 4g$ $/\pm 8g/\pm 16g$. We found no discrepancies in the classification accuracy of the results which implies ERSP can be applied generally across Android based smartphones since the (x, y, z) accelerometer readings are similar and the Micro-electro-mechanical systems (MEMS) specifications are similar across the devices.

A. *Smartphone orientation and on-body placement*

We studied the impact of different smartphone orientations and on-body placements on ERSP with two commercial human step count Android applications. The applications are

Application	Palm	Front trouser pocket	Backpack	Top jacket pocket
Runtastic	(3,5)	(13,12)	(15,14)	(9,11)
Accupedo	(11,12)	(18,21)	(26,22)	(14,15)
ERSP	(10,10)	(12,10)	(11,11)	(10,11)

TABLE I: Results from 10 human steps over 2 iterations using the step count applications.

Runtastic[2] and Accupedo[3]. We calculated the accuracy for 10 human steps while walking with the smartphone placed in the four previously identified on-body positions. Table I shows details of the comparison. The results show the human step count accuracy of ERSP was slightly inflated when the smartphone was placed in the backpack. The additional steps were mainly recorded during the transfer of the smartphone to and from the backpack. This was a similar case for the front trouser pocket. The human step count accuracy for carrying the smartphone in the palm and top-jacket pocket positions was unaffected regardless of the smartphone orientation and on-body placement. This overcomes a lack of flexibility in requiring the smartphone position to be fixed.

B. *Energy-efficiency*

An Android application named AppResource was developed to study the energy-efficiency. AppResource calculates the average consumed resources in terms of CPU and RAM (Mb) usage of active and idle applications over a configurable time period. Over a 60 second window our results show ERSP consumed on average less than 1% of the CPU and 3Mb RAM in active execution as compared to Accupedo (13% CPU, 17.6Mb RAM) and Runtastic (12.7% CPU and 29.3Mb RAM). The energy consumption was also relatively low as compared to standard applications such as playing music (2% CPU, 19.1Mb RAM) and internet browsing (45.4% CPU, 60.5Mb RAM). In the Android OS, sampling the accelerometer in normal mode doesn't consume any additional energy as long as the screen is lit or the CPU is running. ERSP is based on the embedded smartphone accelerometer running in normal sensing mode. The accelerometer sampling rate is approximately 4 HZ by invoking the Android SensorManager module.

V. Analysis

Meta-level classifiers such as bagging have a higher classification accuracy for activity recognition from a single accelerometer as compared to base-level classifiers such as decision tables [11]. We evaluated ERSP with known existing classifiers. The classifiers are J48, Bagging, Decision Table (DT), and Naive Bayes (NB). Fig's. 5 and 6 show the precision and recall comparison of ERSP vs. the classifiers. We trained the classifiers using a data set comprising of pre-classified accelerometer data on the following urban activities: sitting in a moving light rail train, sitting in a moving car, stationary with no movements, walking, and stationary with slight movements e.g., lying down. These 5 user activities were selected because

[2]Runtastic Pedometer http://www.runtastic.com/

[3]Accupedo Pedometer http://www.accupedo.com/

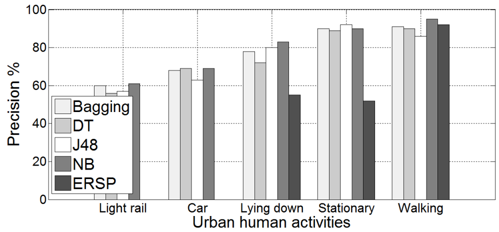


Fig. 5: Chart showing the precision comparison of ERSP vs. existing classifiers.

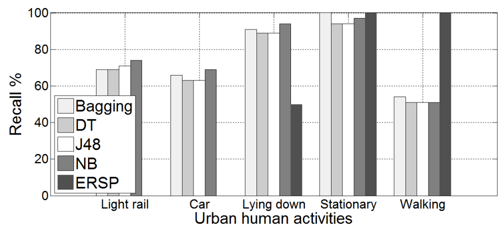


Fig. 6: A comparison of recall for ERSP vs. existing classifiers.

step counts are expected only from walking. Also the user activities were amongst the most popular types of modality and offered a wide range in normal urban commuting. To obtain a model, the classifiers were trained using 175 accelerometer magnitude samples for each selected activity and with a 10 fold cross-validation. Once a model was obtained for each classifier we used 100 instances for predictions of unknown samples.

With respect to the confusion matrices, for motorized movement by car, ERSP falsely classified 1 car data sample as walking. None of the light rail, lying down, stationary, and walking user data activities were misclassified by ERSP.

Naive Bayes had the highest precision and recall for walking in comparison to the existing classifiers. ERSP had the highest average classification accuracy which was calculated from the predictions of unknown samples. We define accuracy as the sum of correct classifications over the total number of input instances. Fig. 7 shows the classification accuracy of ERSP vs. existing classifiers. ERSP outperformed existing classifiers with a weighted average accuracy of 93.8%.

Based on the accelerometer patterns, one of the outstanding differences between walking, motorized movement, and stationary activities is the unique acceleration peak and trough frequency caused by walking within a given time period. As we focused more on improving the detection and extraction of T_{PT}, mm, and P_{mm} features in our method, ERSP can classify walking activity more accurately than known classifiers.

VI. Conclusion

Numerous step count techniques exist, but they tend to require a fixed device orientation. In this paper, we have detailed a probabilistic and feature extraction method on the accelerometer data to accurately determine the count of human steps. Our method can be applied to a wide-range of application areas and is ideally suited for pervasive health, mobility profiling, energy-efficient location sensing applications. The

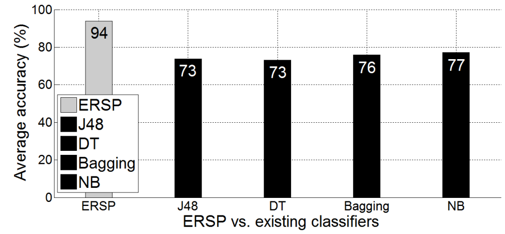


Fig. 7: Chart showing the weighted average accuracy of ERSP vs. existing classifiers.

benefits of ERSP over related architectures are: 1) ERSP functions regardless of the smartphone orientation and on-body placement. 2) Real-time human step count calculation. 3) No requirement for accelerometer sensor noise filtering. 4) Energy-efficient due to the light-weight accelerometer data feature extraction and smartphone accelerometer sensing mode at 4 samples per second.

Acknowledgment

This work has been carried out as part of the ASSET (Adaptive Security for Smart Internet of Things in eHealth) research project funded by The Research Council of Norway VERDIKT program (Grant No: 213131/O70)

References

[1] T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, et al., *The Mobile Sensing Platform: An Embedded Activity Recognition System* IEEE Pervasive Computing, vol. 7, no. 2, pp. 32-41, 2008.

[2] Y. Wang, J. Lin, M. Annavaram, et. al., *A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition* Proc. of the 7th Int'l Conf. on Mobile systems, applications, and services (MobiSys), pp. 179-192, 2009.

[3] N. Gyorbíró, Á. Fábián, and G. Hományi, *An activity recognition system for mobile phones* Springer Mobile Networks and Applications, vol. 14, no. 1, pp 82-91, 2009.

[4] G. Hache, E.D. Lemaire, and N. Baddour, *Mobility Change-of-State Detection Using a Smartphone-based Approach* IEEE Int'l Workshop on Medical Measurements and Applications Proc. (MeMeA), pp. 43-46, 2010.

[5] W. Shyi-Shiou and W. Hsin-Yi, *The Design of an Intelligent Pedometer Using Android* 2nd Int'l Conf. on Innovations in Bio-inspired Computing and Applications (IBICA), pp. 313-315, 2011.

[6] W. Hongman, Z. Xiaocheng, and C. Jiangbo, *Acceleration and Orientation Multisensor Pedometer Application Design and Implementation on the Android Platform* 1st Int'l Conf. on Instrumentation, Measurement, Computer, Communication and Control, pp. 249-253, 2011.

[7] D. Sugimori, T. Iwamoto, and M. Matsumoto, *A Study about Identification of Pedestrian by Using 3-Axis Accelerometer* 17th Int'l Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA), pp. 134-137, 2011.

[8] S. Inoue and Y. Hattori, *Toward High-Level Activity Recognition from Accelerometers on Mobile Phones* Proc. of the 4th Int'l Conf. on Cyber, Physical and Social Computing, pp. 225-231, 2011.

[9] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, *Accurate Activity Recognition Using a Mobile Phone Regardless of Device Orientation and Location* Int'l Conf. on Body Sensor Networks (BSN), pp. 41-46, 2011.

[10] L. Bao, and S.S. Intille, *Activity Recognition from User-Annotated Acceleration Data* Proc. of Springer, Heidelberg, pp. 1-17, 2004.

[11] N. Ravi, N. Dandekar, P. Mysore, et al., *Activity Recognition from Accelerometer Data* 17th Int'l Conf. on Innovative Applications of Artificial Intelligence, pp. 1541-1546, 2005.