UDetect: Unsupervised Concept Change Detection for Mobile Activity Recognition

Sulaimon A. Bashir
School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen, United Kingdom
s.a.bashir@rgu.ac.uk

Andrei Petrovski
School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen, United Kingdom
a.petrovski@rgu.ac.uk

Daniel Doolan
School of Computing and Digital Technology,
Birmingham City University,
Birmingham, United Kingdom
daniel.doolan@bcu.ac.uk

ABSTRACT

One of the major challenges in activity recognition task is the need to adapt a classification model during its operation. This is important because the underlying data distribution between those used for training and the new evolving stream of data may change during online recognition. The changes between the two sessions may occur because of differences in sensor placement, orientation and user characteristics such as age and gender. However, many of the existing approaches for model adaptation in activity recognition are blind methods because they continuously adapt the classification model without explicit detection of changes in the concepts being predicted. Therefore, we propose a concept change detection method for activity recognition under the assumption that a concept change in the model of an activity is followed by changes in the distribution of the input data attributes as well which is the realistic case for activity recognition. Our change detection method computes change detection statistic on stream of multi-dimensional unlabelled data that are classified into different concept windows. The values of the change indicators are then processed for detecting peak points that indicate concept change in the stream of activity data. Evaluation of the approach using real activity recognition dataset shows consistent detections that correlate with the error rate of the model.

Categories and Subject Descriptors

H.5.2 [**User/Machine Systems**]; I.5 [**Pattern Recognition**]: Metrics—*Percentage Accuracy*

Keywords

Activity Recognition, Concept change detection, Machine learning algorithms

MoMM '16, November 28-30, 2016, Singapore, Singapore

DOI: http://dx.doi.org/10.1145/3007120.3007144

1. INTRODUCTION

Smartphone for activity recognition employs inertial sensors such as tri-axial accelerometer or gyroscope to infer activities of users. The accelerometer measures the amount of acceleration forces experienced by the device along x, y and z axes. The patterns of acceleration forces experienced by the device correspond to the intensity of the activity being carried out by the phone possessor. The basic procedure for mobile activity recognition involves: i) collection of labelled data from the subjects that perform sample activities to be recognised ii) classification model generation by using collected data to train and test classification algorithms iii) a model deployment stage where the learnt model is transferred to the mobile device for identifying new unseen activities data. This traditional approach for activity recognition performs the model generation phase on remote systems and transferred the generated model to the phone to recognise new user activities. The drawback of this approach is that the model is static and does not reflect possible changes in the distribution of new evolving data. Another approach that aims to eliminate this, induced the model by using the user self-annotated data on the phone so that the model can be tuned to individual user.

The two approaches are still not immune from changes that may occur in the underlying distribution of the unseen incoming data. This usually results into decreasing performance in accuracy of the model. Thus, for example, a model that is trained to recognise walking activity given a specific data may take a new data from another slightly different distribution that correspond to say jogging for another user and classify it as walking. At this point the model has suffered from the phenomenon call concept change. The sources of change can be known or unknown. But for activity recognition problem, it has been shown to be caused by a number of factors such as dissimilarities between the user profiles used during training and the users using the model during recognition [11, 10]. It may also be caused by the displacement of the sensors and orientation effect on the sensor readings [4, 15].

Hence, various approaches have been developed for model adaptation during their online operation. Many of these approaches [17, 11, 1] are blind in the sense that they do not identify concept changes before they start the adaptation process. In the field of stream mining, a related problem exists and two approaches have been identified for handling concept drift. We have the informed and uninformed handling techniques [8]. The informed concept drift adaptation approach attempts to react to the occurrence of concept drift by ensuring that the drift point is detected before taking any action whereas, the uninformed handling of concept drift is proactive and does not have any explicit detection mechanism. Rather, it incrementally and continuously updates the model at each time step a sample or set of samples are integrated into the model. The drawback of uninformed adaptation is that they react slowly to concept drift and consumes system resources as it continuously adapts the old concept which may be required to be replaced out-rightly or maintained without adaptation [8]. Hence an informed adaptation scheme is better and required for better management of concept drift.

Furthermore, many of the approaches[7, 14] for the detection of concept drift require the presence of labels to detect the change. However, this assumption of label availability is not realistic in the domain of activity recognition because labels are not easy to come by during online recognition as the user will be required to provide label for each activity being performed. This is impractical and tedious to do. Hence our approach based on unsupervised detection eliminates the need for ground truth to detect changes that caused decrease in the accuracy of the recognition model.

The rest of this paper is organised as follows: Section 2 examines the existing methods for concept change detection; Section 3 presents our new method; Section 4 describes the experiment and the datasets used for evaluation while the results are presented in Section 5. Finally, the paper is concluded in Section 6.

2. CONCEPT DRIFT DEFINITION

Concept drift or change is a phenomenon in classification problem where a classifier built to recognize certain concepts from set of training data becomes inaccurate over time because the distribution of the data being classified has changed from the initial distribution known to the model [8]. The changes in the data can manifest either as changes in the class label or changes in the attributes of the new unobserved samples. Changes in the class labels can occur while the attributes themselves remain unchanged. That is, given a sample with a class label say '0', when changes occur the same sample now has label '1'. On the other hand, the attributes of the data may change while the class labels remain unchanged. The two parts of the data can also change simultaneously. In the first and third situations, the classifier will need to be updated with the new emerging distribution of the data while the second situation may or may not affect the decision boundary and hence may not require model update. Another possible but infrequent change is the change in the prior probabilities of classes termed concept evolution that result in emergence of new concepts or merging of existing concept [12].

More formally, concept drift arises as a result of differences in the relationship between input variable x any target variable y between two points in time t_0 and t_1 i.e. $P(x,y)_{t_0} \neq P(x,y)_{t_1}$ where $x \in R^n$ are the input attributes and $y \in \{y_i : i = 1...c \ \ number\ of\ classes\}$. The changes in this relationship can manifest in the form of changes in the class conditional probability $P(x/y)$ where the attributes values changes for given y_i but the class label y remains unaffected or it may result in posterior probability $p(y/x)$ changes which means the attributes remain unchanged but the class labels changed or there could be a simultaneous changes in posterior and class conditional probability. It is also possible to have prior probability changes leading to emergence of new concepts.

Table 1: Categories of Drift

Types of Drift	**Notation**	**Comment**
Real Drift	$p(y/x)_{t_0} \neq p(y/x))_{t_1}$	This drift affect the decision boundary
Virtual Drift	$p(x/y))_{t_0} \neq p(x/y))_{t_1}$	Does not affect the decision boundary
Virtual Drift with Decision Boundary Change	$p(x/y))_{t_0} \neq p(x/y))_{t_1}$ and $p(y/x)_{t_0} \neq p(y/x))_{t_1}$	Simultaneous drift in class conditional probability and posterior probability which affects the decision boundary.
Concept Evolution	$p(y)_{t_0} \neq p(y)_{t_1}$	Concept evolution results in emergence of new classes other than the known classes.

Change that arises from $p(y/x)$ is regarded as real concept drift, while that of $p(x/y)$ is referred to as virtual drift. The virtual drift can also occurred when both $p(x/y)$ and $p(y/x)$ changes simultaneously. The changes in $p(y)$ is referred to as concept evolution.

The following are some of the existing change detection methods in the literature: Online Cumulative Sum Test [13]: It is a sequential test that can be applied on stream of numerical data to detect change point. The test monitors the cumulative sum of the attribute of the data stream such as the mean or the error rate of a classification model and alert a change when the value exceeds a pre-set threshold γ. Specifically, the test begins by initializing the sum of the target value to 0. i.e. $S_0 = 0$, then computes the cumulative sum after receiving each observation x_i as $S_{i+1} = max(0, S_i + x_i - \xi)$ where ξ allowed magnitudes of change. The efficacy of this approach depends on the choice of the parameters of the test.

Page Hinkely Test: This is a sequential test for change point detection originally devised by Page in 1954 for change detection in signal processing [13]. The approach is similar to CUSUM but rather than computing cumulative sum, it computes two test statistics, the cumulative difference between the observed values and their mean up till the moment of the test defined as $c_T = \sum_{t=1}^{T}(xt - \overline{x} - \theta)$, where $\overline{x} = \frac{1}{T}\sum_{1}^{T} x_t$ and θ is the accepted magnitude of tolerable changes and the minimum of c_t defined as $Ct = min(ct, t = 1...T)$. The two parameters are compared as $PHtest = c_t - C_T$ and if the result is greater than a threshold ζ a change is signalled.

Methods based on Statistical Process Control: These methods unlike other sequential approach consider the system being monitored as a process and try to monitors the normal operation of the process from unwanted variations. Once the variation of the process is beyond the acceptable threshold a drift alarm is signalled. Notable methods of process controls include P-charts, X-chart, R-chart, CUSUM chart and a host of other process control charts. An often cited work that considers learning as a process and applied the principles of process control to concept change detection include DDM [7, 9] . The methods monitors the performance evolution of a classifier and relies on the availability of ground truth to determine when the classifier gives a correct or incorrect prediction. The method incrementally computes the proportion of errors produced by the current model with $p_i = p_{i-1} + (x - p_{i-1})/n$ with x =1 if the prediction is incorrect and x=0 if the prediction is correct. The average error is thus computed incrementally. The standard deviation of the error rate s_i at each time step of the learning process is also computed. Two register p_{min} and s_{min} are maintained and they are updated with pi and s_i respectively whenever $p_i + s_i < p_{min} + s_{min}$. DDM has two thresholds to take decision on the drift: if $p_i + s_i \geq p_{min} + 2 * s_{min}$ it implies a warning level. Subsequent examples after this point are stored in anticipation of a possible change of concept. If $p_i + s_i \geq p_{min} + 3 * s_{min}$ a drift level is signalled after a series of warning state concept drift is declared, the model induced by the learning method is reset and a new model is learnt using the examples stored since the warning level triggered. The values for p_{min} and s_{min} are reset to 0. The intuition behind this method is that, in the absence of concept drift the error rate should decrease indicating a stationary distribution. However, if the error rate increases significantly it means the classifier is no more in tandem with the distribution of the data. Thus a concept drift has occurred and the model has to be rebuilt. Authors in [3] extends DDM to account for the distance between error points while [6] used DDM as a component for their adaptation algorithm to make them informed.

3. UDETECT METHODS FOR CHANGE DETECTION IN MULTI-CLASS MODEL IN ACTIVITY RECOGNITION

This section presents our new proposed method for change detection in activity recognition. The method does not assume the presence of ground truth with each arrival of new unseen sample to be classified. Hence, it reflects a realistic scenario for detecting concept change in activity recognition where the stream of activity data being classified does not usually comes with ground truth label. The method transform a multivariate data stream to univariate detection stream.

The multidimensional training data which, are taken as the reference data that represent the normal activity of the users are sorted into separate classes. The data in each class are converted into uni-dimensional stream of data by segmenting them into windows of equal sizes and compute the change statistics (x_d) from each window. The change statistic from each window is the average distance to the centre of each window computed according to Equation 1. These values are then subsequently used to compute the parameters of the Shewart Control Charts [16] a statistical process control method.

The chart is made up of two charts; the individual observation and the moving range of the observation charts. The moving range chart is constructed by computing the moving range of two successive change statistics as $mR_i = |x_{d(i)} - x_{d(i-1)}|$. The mean $\overline{mR}$ of the ranges are also computed. This value is then used to compute the with upper control limit as $3.27\overline{mR}$. This upper limit indicates the acceptable level of drift which, the new observations should not exceed. Similarly, control chart for the individual change statistics x_d, is constructed using average range value $\overline{mR}$ to compute upper control limit as $\overline{x_d} + 2.66\overline{mR}$ and the lower control as $\overline{x_d} - 2.66\overline{mR}$. Afterwards, these parameters were used to monitor the statistic computed from the new batches of unseen data in order to detect concept change points.

Once the parameters of the charts have been computed using the reference training data, a nearest neighbour based classifier [5] that was trained on the data was used to recognize activity from new set of test data. Figure 1 shows the architecture of our detection method. As the pre-trained model classifies data samples, they are kept in a buffer . Then, the samples from the buffer are transmitted to the detection windows dedicated to each class from which the change statistic is computed on equal size amount of window data. The values are then charted on the existing charts to detect out of control points that indicate the model misclassification rate is increasing beyond control.

Intuitively, if the classifier is not misclassifying samples to a class, the distribution of the attributes in the samples classified to the same class should be stable. It should be noted that changes in concept in activity recognition does not only manifest in the change in class label but also simultaneous changes in the distribution of the attributes and the label. A change in the distribution of the samples classified to the same class is evidenced if the value of computed change statistics of the new samples deviate significantly from that of the referenced training data set. Our method relies on this assumption and monitors the parameters computed from the batches of data that are classified to the same class. If this parameter is within a threshold no change is detected but if this parameter exceeds a threshold a change is signalled in the class of data. Hence the method can precisely localize the part of the model that need to be updated rather than updating the entire classifier.

$$x_d = \frac{\sum_{1=1}^{n} Euclidean(X^{(i)}, WindowCenter)}{n} \quad (1)$$

where $X^{(i)} \in R^n$ are the instances in the window and WindowCenter is defined as the centroid of the data in the window computed as:

$$WindowCenter = \frac{\sum_{1=1}^{n} X^{(i)}}{n} \quad (2)$$

n being the size of the window and Euclidean(E) between two vectors a and $b \in R^n$ is computed as:

$$E(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2} \quad (3)$$

In another words, the distribution of correctly classified samples of a class should correlate with the distribution of the training data for that class. The changes in the distribution of test examples indicated by out of control limits on the chart is an indication that the model is misclassifying the samples and therefore the model needs to be reviewed. The revision carried out on the model depends the underlying assumption of the recognition process. Two scenarios can be envisaged. The first is to assume that new data that signal the change detection are outliers that should be discarded and indicate that the model need to be fine-tuned to continue in its original settings. All that is required is to identify the causes of the variation from the normal s be-

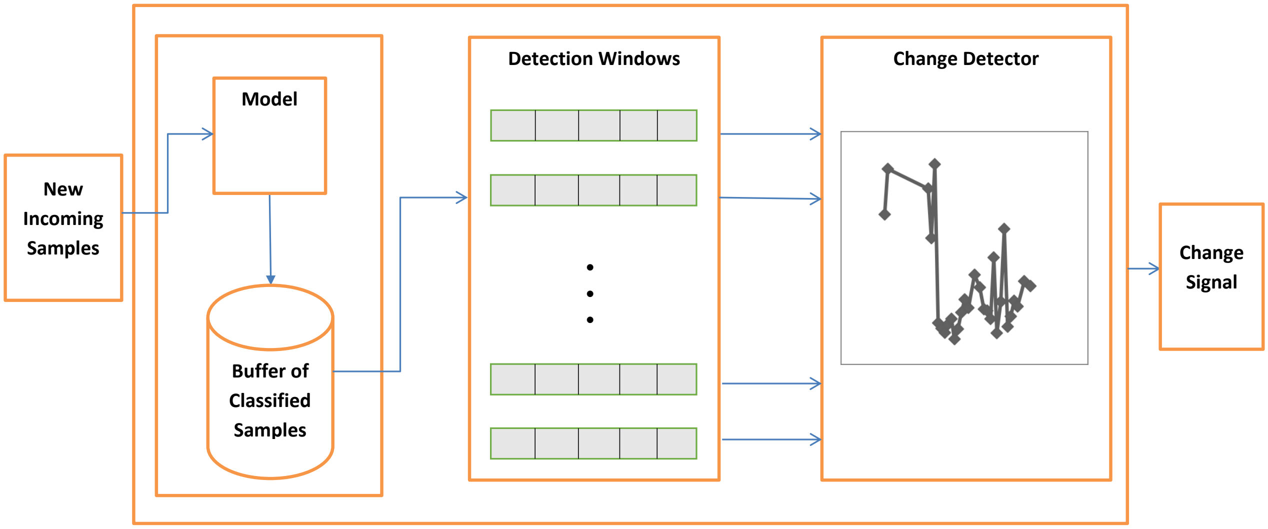


Figure 1: Change Detection Architecture for Multi-Class Activity Recognition

haviour. The variation in such situation can be caused by sensor misplacement or fault. In this case, the cause of the variation should be eliminated i.e. by replacing the faulty sensor. The second possible scenario is to assume that the data at the change points reflect the current situation of the user that is using the model and in this case the model should be reviewed to suit the new distribution. Thus, the model will need to be adapted by making use of model adaptation technique. Many adaptation techniques are available in the literature. It is out of scope of this paper to deal with adaptation of the model.

4. EXPERIMENT

The aim of this experiment is to examine the efficacy of our change detection approach in detecting changes between the distribution of training data used to train a model and the new examples being classified by the models.

To carry out this experiment, we have employed an activity recognition dataset that is based on accelerometer and gyroscope sensors of mobile phones. The dataset has characteristics that enable us to assess the applicability of our approach. The dataset is obtained from different subjects and marked appropriately to distinguish one subject data from another. The Human Activity Recognition Using Smartphone Dataset [2] is collected from a set of 30 volunteers who are within an age bracket of 19-48 years. Each subject performed six designated activities of walking (class-0), walking-upstairs (class-1), walking-downstairs (class-2), sitting (class-3), standing (class-4), and laying (class-5) while wearing a smartphone attached to their waists. The data were obtained from gyroscope and accelerometer sensors of the smartphone. Each data sample in the dataset is represented by 561 features containing both time domain and frequency domain features and a corresponding activity label. The features were obtained from 128 fixed-width sliding windows of 2.56sec with 50% overlap.

The first part of the experiment focused on identifying the level of differences among the individual subject data. We performed a cross-user model evaluation. This is done by using one user data for training and another user data for testing. This is repeated for each of the user data in the dataset one after the other.

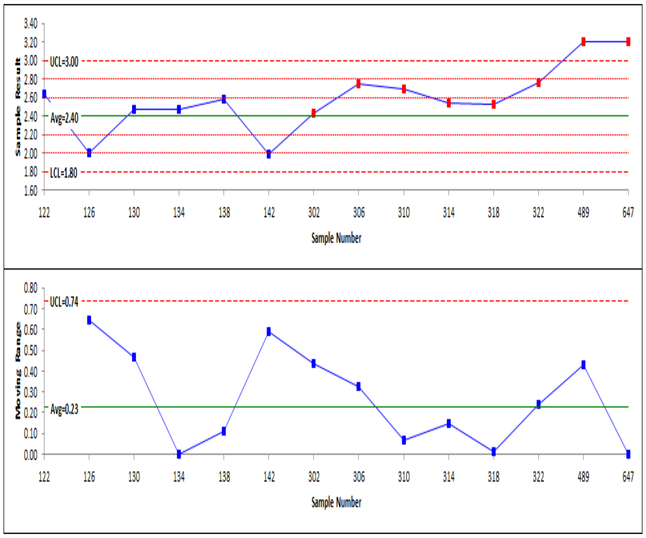
In the second stage of the experiment, we detected changes between training data and new unseen data during activity recognition. To do this, we employed the data of one user for training and another subject data for evaluating the change detection. We used a known amount of one user data as training data to create a bespoke up-to date model. The training and the test data are then combined and passed to the model so that if there are differences between the distributions of the activity data between the users which, cause model misclassification, our method should be able to identify the change points after the first user data. Hence, the first set of data to test are from the original user while the rest are from another user.

5. RESULTS

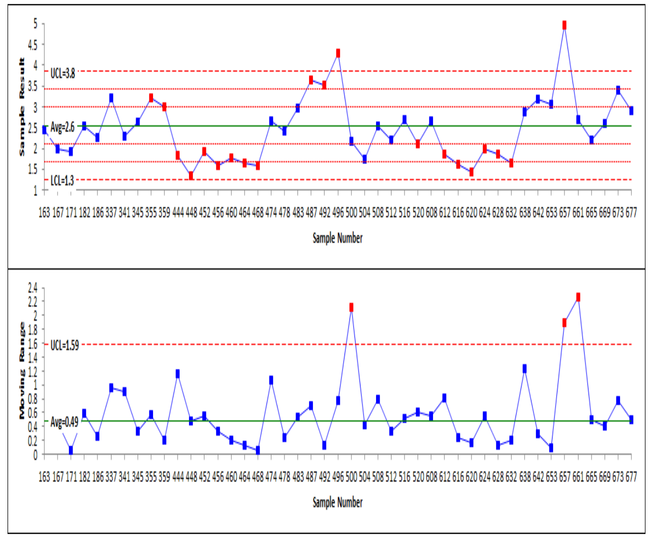
We present the result of the experiment performed using the datasets mentioned above. The accuracies of using one user as training are listed against other individual user's data as testing in the columns of Table 2. In an ideal situation, where the pattern of performing similar activity is the same between any two users, the accuracy should be very high. However, if the pattern of performing similar activity between users is not the same the accuracy will be low.

Table 2 shows the results for the first set of 5 users. We can observe from the result that the accuracy of within user model where the same user is used for training and testing the model is the highest across all users. The levels of accuracy between a user and other set of users vary from user to user and this indicate that there is a difference in pattern in how users perform similar activity. User with dissimilar activity profiles have lower accuracies while those with similar profiles have higher accuracies.

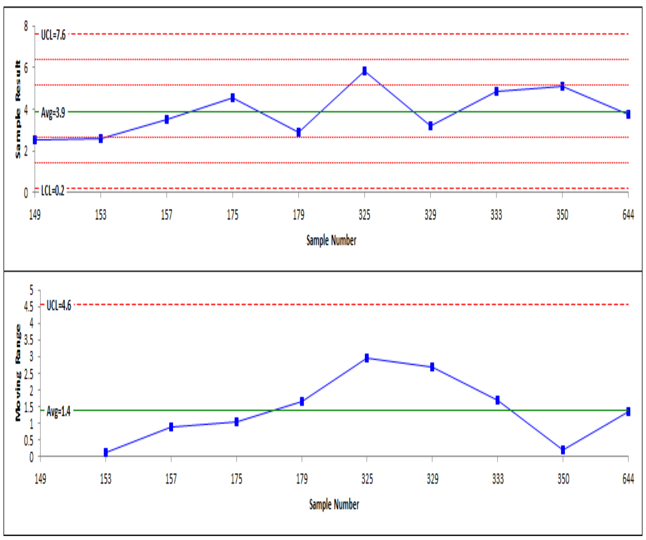
The second phase of the experiment evaluates our change detection method in detecting the variations between one



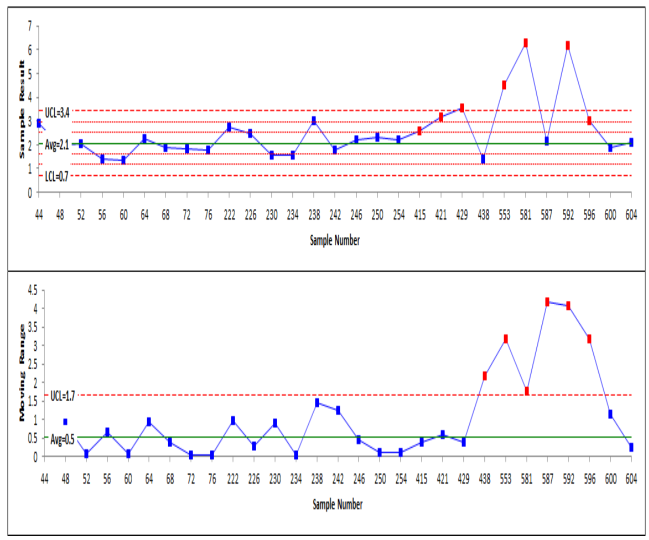
(a) Class 0 Chart



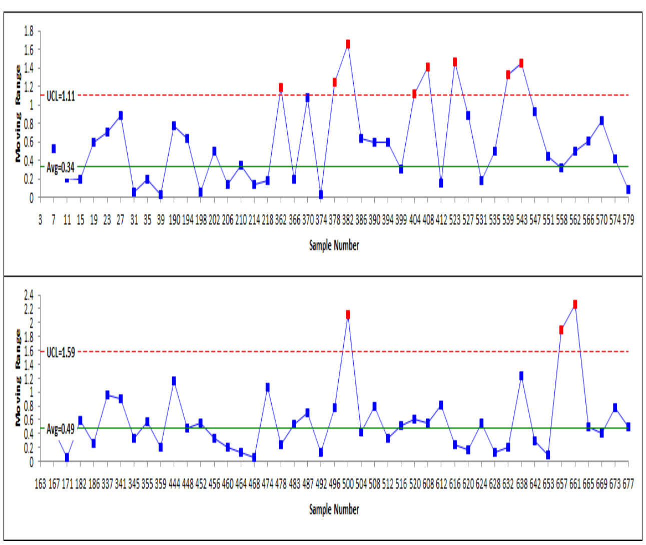
(b) Class 1 Chart



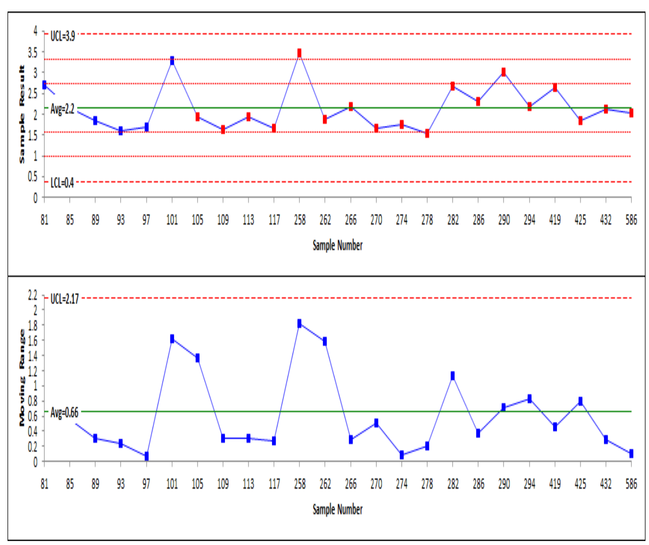
(c) Class 2 Chart



(d) Class 3 Chart

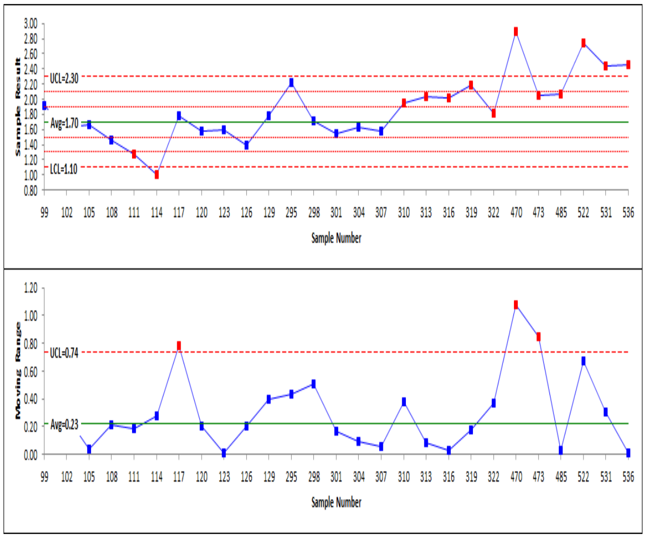


(e) Class 4 Chart

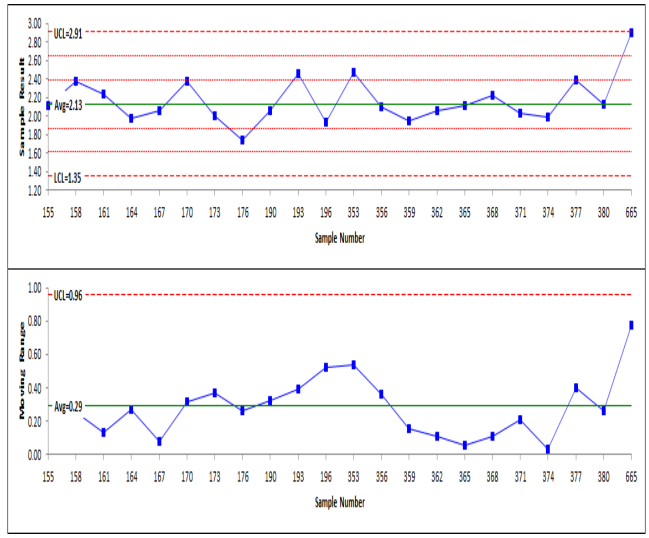


(f) Class 5 Chart

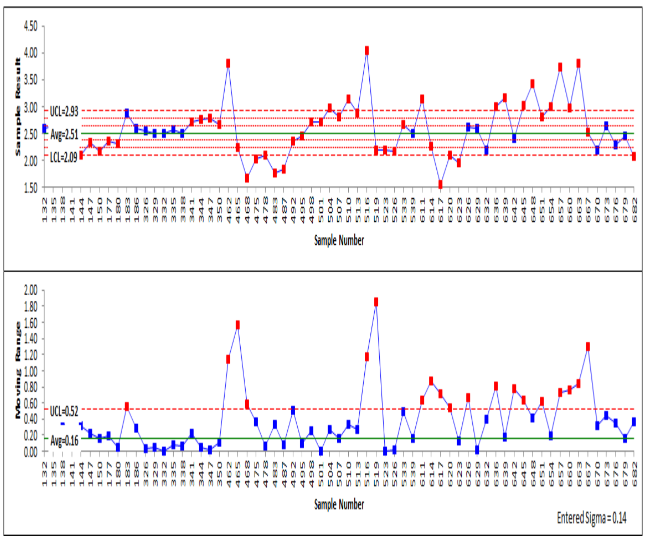
Figure 2: Chart of Window Parameters for Each Class of Activity User 19 Against 14



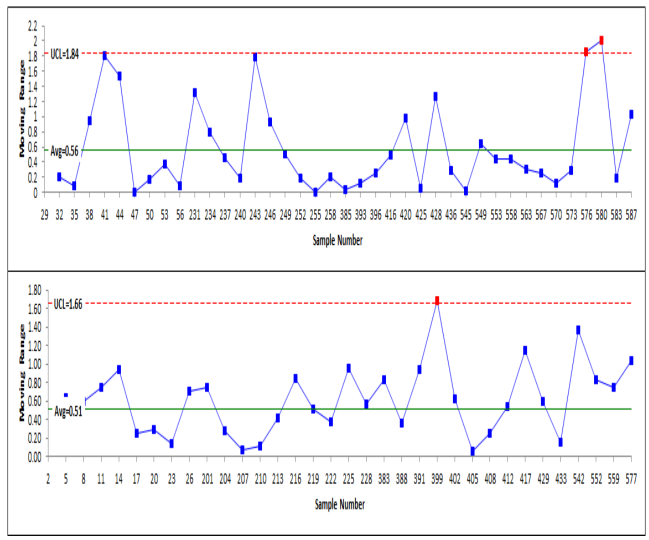
(a) Class 0 Chart



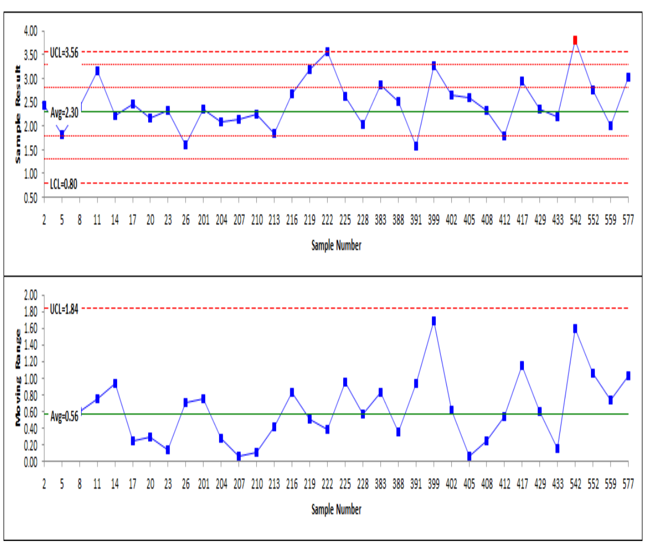
(b) Class 1 Chart



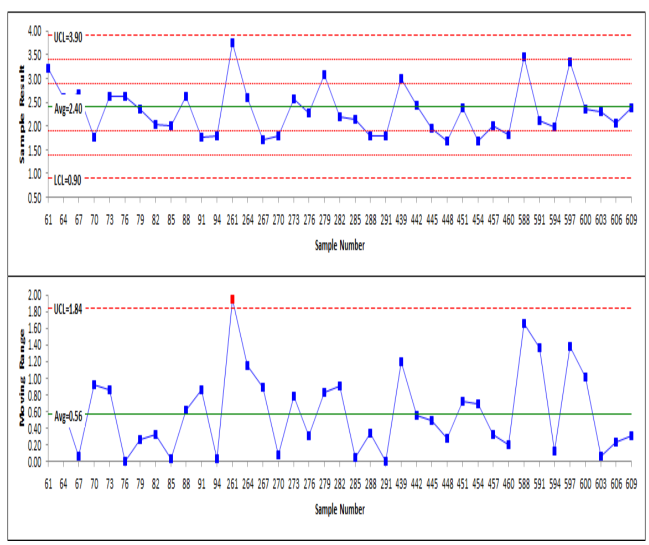
(c) Class 2 Chart



(d) Class 3 Chart



(e) Class 4 Chart



(f) Class 5 Chart

Figure 3: Chart of Window Parameters for Each Class of Activity User 30 Against 2

Table 2: Cross User Data Classification Accuracy

ID	1	ID	2	ID	3	ID	4	ID	5
16	64.48%	**10**	68.03%	**10**	63.27%	**14**	62.85%	**14**	57.28%
14	67.80%	**19**	70.00%	**19**	66.94%	**30**	64.75%	**9**	58.68%
28	69.11%	**4**	71.92%	**8**	69.04%	**9**	68.06%	**21**	62.50%
18	70.88%	**7**	72.40%	**9**	69.79%	**10**	68.71%	**16**	62.57%
20	71.75%	**18**	72.53%	**14**	72.14%	**16**	69.95%	**24**	64.83%
6	72.00%	**14**	72.76%	**18**	72.80%	**20**	73.16%	**30**	65.27%
5	75.17%	**30**	74.15%	**16**	73.50%	**22**	73.52%	**28**	66.49%
10	75.17%	**9**	75.00%	**30**	76.24%	**28**	74.87%	**20**	67.23%
19	75.28%	**3**	75.37%	**20**	76.27%	**8**	76.16%	**18**	67.58%
9	75.35%	**8**	75.80%	**15**	76.52%	**19**	76.67%	**8**	67.97%
2	75.50%	**23**	77.69%	**28**	76.70%	**6**	76.92%	**13**	70.03%
24	75.59%	**28**	78.27%	**29**	78.49%	**17**	77.17%	**7**	70.78%
21	76.72%	**20**	78.53%	**6**	80.31%	**29**	77.33%	**10**	71.43%
22	77.57%	**12**	79.38%	**4**	81.07%	**7**	78.57%	**17**	72.55%
17	77.99%	**29**	79.94%	**5**	81.79%	**11**	79.11%	**3**	73.31%
30	78.59%	**6**	80.00%	**17**	82.88%	**18**	79.12%	**29**	74.42%
8	79.00%	**16**	80.60%	**22**	83.49%	**23**	80.38%	**12**	74.69%
12	79.06%	**17**	82.34%	**11**	83.54%	**15**	81.10%	**23**	75.81%
25	79.08%	**5**	82.78%	**25**	84.18%	**1**	83.00%	**1**	76.66%
26	79.08%	**11**	82.91%	**26**	84.18%	**12**	83.13%	**22**	78.82%
4	80.44%	**15**	82.93%	**23**	85.22%	**21**	83.33%	**11**	80.70%
11	81.33%	**1**	83.86%	**2**	86.75%	**2**	86.09%	**2**	80.79%
23	81.72%	**13**	84.10%	**13**	86.85%	**13**	87.16%	**15**	81.10%
7	83.44%	**25**	84.95%	**1**	88.76%	**24**	87.40%	**4**	83.60%
3	86.51%	**26**	84.95%	**12**	89.38%	**25**	88.01%	**19**	84.17%
15	88.72%	**24**	86.09%	**24**	89.50%	**26**	88.01%	**25**	84.95%
27	89.63%	**22**	87.23%	**21**	90.20%	**5**	89.40%	**26**	84.95%
13	90.21%	**21**	87.25%	**27**	90.43%	**3**	89.74%	**6**	85.54%
29	93.60%	**27**	89.36%	**7**	90.91%	**27**	92.02%	**27**	88.03%
1	100.00%	**2**	100.00%	**3**	100.00%	**4**	100.00%	**5**	100.00%

user and another user data. The results obtained show correlations between the chart of the change statistics and the level of model accuracy in recognizing the various activity types of new users. Figure 2 -3 show the samples of the change detection charts.

The data plotted are the change statistics computed from the batches of data samples coming from the moving windows designated for each class of activity being recognised. We utilized a batch size of 3 to compute the sample statistics for all the experiments. Each type of activity has its own dedicated window for detecting the variability in the data classified to the class. This variability indicates the rate of misclassification in the window. The charts show this by out of control points indicating non-uniformity in the data classified to the same window. The charts also indicate the change points by means of out of control points. A change point indicates the point where different classes of data other than the original class begin with a threshold of delay.

Table 3: Proportion of Error Per Class

	Class0	Class1	Class2	Class3	Class4	Class5
Exp. 1	0.15	0.49	0.00	0.32	0.26	0.00
Exp. 2	0.10	0.00	0.45	0.19	0.11	0.00

Sample results from the experiment are presented in Figure 2 and 3. The charts in Figure 2 is obtained by setting the data of user with ID 19 as training set while the combination of user 19 and 14 are set as test data. The first 360 samples of the testing data belong to user 19 while the remaining 323 data points of the total 683 belongs to user 14. We noted this demarcation points to be able to identify the peak points in the sequences of plotted values for detecting the changes between the two data.

Figure 2a, show the chart of sequence of the individual change statistics obtained from the samples classified into window of class 0 in the upper part while the lower part of the figure indicates the moving range of the values. A change is detected at the time step 489 on the individual chart. This is the point where the computer sample statistic exceeds the upper control limit. This indicates the point where the distribution of the data changes from the initial data distribution. It also indicates that the samples around these time steps are misclassified which makes their computed statistic goes out of control limit. The more the out of control points the more the proportions of the misclassified samples that are classified into this window. Table 3 shows the proportions of misclassification for each class. We can see that there is classification error of 0.15 in this class. This error rate is computed looking at the amount of incorrectly classified samples to the class.

Similarly, changes are detected in the activity types 1, 3 and 4 shown in Figures 2b, 2d, and 2e respectively. The change points are indicated by the out of control limit points in the individual chart and moving range chart. The changes detected show the variation in the activity of the initial user and the test user data. The proportions of misclassified samples in these classes as shown in Table 3 corroborate the non-homogeneity of the data that are classified to the window dedicated for each class. Thus the approach is able to detect changes in the distribution of the initial user data that belongs to the original activity and those that comes from another user. It should be noted that points after the change points that are within control limits indicate instances from test data have the same and correct class as the initial training data. There is no detection of change in the activity class 2 and 5. This is evident by the absence of out of control points in the two charts for the two classes. This is because there is no variability in the training data of the user and the test data from another user and hence the proportions of their misclassified samples are 0 for each of the two classes as shown in Table 3.

The second experiment result charts in Figure 3 are obtained by setting the user 30 data as training set and combination user 30 and 2 as test data. The first 383 samples of the testing data belong to user 30 while the remaining 302 data points of the total 685 belongs to user 2. The same observations are recorded as in the sample experiment 1. The changes detected correlate with the proportion of error in the misclassified samples in each of the window dedicated for each class of activity. This implies that the distribution of the test data has changed for some part of the test data.

Changes are detected in activity types 0, 2, 3 and 4 while no change is detected in activity types 1 and 5. These are shown in Figure 3a-f. The detected changes in classes 0,2,3 and 4 are corroborated by the misclassification levels of $0.1, 0.45, 0.19$ *and* 0.11 obtained for each of the activity type respectively. The non-change classes 1 and 5 have misclassification rate of 0. This implies that the more the misclassification levels observed from the windows dedicated to each activity, the more the changes that are detected.

It should be noted that the individual chart is the main chart that indicates changes. The moving range chart is meant to corroborate the detections. Once the individual chart detects the change there is no need to look at the range chart. But if the range chart detects the change we have to confirm from the individual chart before taken a decision on the admissibility of the change detected.

6. CONCLUSION

This paper presented a novel concept change detection method for activity recognition. The method is based on processing chunks of data classified to the same class and extract the change statistic value that characterised each chunk. The average distance to centre parameter computed from each batch are monitored by using Shewart Control Charts as the change point detector to identify outlier peaks that represent the drift point. Such points indicate that the model is misclassifying the samples to the wrong class and thus need to be diagnosed to react to drift. The main benefits of this method compared to the traditional drift detection approach in data stream domain is that it does not rely on the ground truth to detect drift in the data and thus is more realistic approach for activity recognition.

The method is evaluated using real activity recognition dataset obtained from smartphone of diverse subjects who perform the designated activity. The result indicates that the method can identify the precise drift point in the data.

7. REFERENCES

[1] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, 150:304–317, 2015.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, 2013.

[3] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86, 2006.

[4] O. Banos, M. A. Toth, M. Damas, H. Pomares, and I. Rojas. Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023, 2014.

[5] S. Bashir, D. Doolan, and A. Petrovski. Clusternn: A hybrid classification approach to mobile activity recognition. In *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, MoMM 2015, pages 263–267. ACM New York, NY, USA, 2015.

[6] A. Bouchachia. Fuzzy classification in dynamic environments. *Soft Computing*, 15(5):1009–1022, 2011.

[7] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *Advances in artificial intelligence–SBIA 2004*, pages 286–295. Springer, 2004.

[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[9] R. Klinkenberg and I. Renz. Adaptive information filtering: Learning drifting concepts. In *Proc. of AAAI-98/ICML-98 workshop Learning for Text Categorization*, pages 33–40. Citeseer, 1998.

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[11] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 355–364. ACM, 2011.

[12] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *Knowledge and Data Engineering, IEEE Transactions on*, 23(6):859–874, 2011.

[13] E. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[14] P. Sobhani and H. Beigy. *New drift detection method for data streams*. Springer, 2011.

[15] Y. E. Ustev, O. Durmaz Incel, and C. Ersoy. User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1427–1436. ACM, 2013.

[16] D. J. Wheeler. *Understanding Variation: The Key to Managing Chaos*. SPC Press, second edition edition, 1993.

[17] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. Cross-people mobile-phone based activity recognition. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence- Volume Three*, pages 2545–2550. AAAI Press, 2011.