StreamAR: Incremental and Active Learning with Evolving Sensory Data for Activity Recognition

Zahraa Said Abdallah*§ , Mohamed Medhat Gaber†, Bala Srinivasan*and Shonali Krishnaswamy‡
*Centre for Distributed Systems and Software Engineering
Monash University,Melbourne, Australia
Email: zahraa.said.abdallah@monash.edu
Email: srini@monash.edu
† School of Computing, University of Portsmouth
Portsmouth, Hampshire, England, PO1 3HE, UK
Email: mohamed.gaber@port.ac.uk
‡Institute for Infocomm Research (I2R), Singapore
Email: shonali.krishnaswamy@monash.edu
§Faculty of Computer and Information Sciences
Ain Shams University, Egypt

***Abstract*—Activity recognition focuses on inferring current user activities by leveraging sensory data available on today's sensor rich environment. Supervised learning has been applied pervasively for activity recognition. Typical activity recognition techniques process sensory data based on point-by-point approaches. In this paper, we propose a novel cluster-based classification for activity recognition Systems, termed *StreamAR*. The system incorporates incremental and active learning for mining user activities in data streams. The novel approach processes activities as clusters to build a robust classification framework. *StreamAR* integrates supervised, unsupervised and active learning and applies hybrid similarity measures technique for recognising activities. Extensive experimental results using real activity recognition datasets have evidenced that our new approach shows improved performance over other existing state-of-the-art learning methods.**

I. Introduction

Activity recognition has become one of the emerging applications in the area of ubiquitous computing. The availability of real time sensory information through sensors has led to emerging research into Activity Recognition (AR) . This focuses on inferring the current activities of users by leveraging the rich sensory data that is available from on-body sensors, environmental sensors, today's smartphone and rich information sources. Successfully recognising people's activities enables a wide range of pervasive computing applications in the fields of healthcare, social networks, environmental monitoring, surveillance, emergency response and mobile services.

The state of the art in mobile activity recognition research has focused on traditional classificatory learning techniques. First, data is collected and annotated by users. Then, labelled data is deployed to build and train the classifier learning model. When the model is ready, system is used to predict activities from the sensory data. A wide range of classification models has been used for activity recognition such as Decision Trees, Naive Bayes and Support Vector Machines. However, this approach has no notion of adaptation or refinement of the model that is already built. In realistic conditions, change of activities may emerge over time which includes modifying user activities patterns. Current approaches do not allow refinement of the deployed model. Moreover, personalisation of model to suit a specific user had a little focus in the research area. Typically, walking for one user may well be running for another, therefore tuning the general model to recognise a given user's personal activity is crucial for building a robust activity recognition system.

We propose an adaptive system for robust activity recognition with evolving sensory data streams. We coined our technique *StreamAR*. The novel system integrates supervised, unsupervised and active learning for activity recognition. *StreamAR* extends the state-of-the-art in *AR* by providing the following advantages.

- *Build an adaptable model with evolving sensory data streams:* One of the characteristics of *StreamAR* framework is the flexibility to be updated as the data evolves. Thus, the updated model is personalised and adapted to the most recent changes detected in the user's activities in streaming data.
- *Combination of modelling techniques:* The system combines supervised, unsupervised and active learning all in one data stream model. We initially build the learning model with supervised learning. When new data received, unsupervised learning is deployed to cluster activities. Active learning is also employed in the event of confusion on cluster labels.
- *Adaptability to the nature of activity recognition data:* People perform activities in a sequential manner (i.e., performing one activity after another). Therefore, activity recognition data stream typically composites of sequence of chunks that represents various activities. Different from other activity recognition systems, *StreamAR* is a cluster based classification that deals with activities as clusters

rather than processing each point. The novel approach is adapted for activity recognition data nature. Therefore, computation and processing time are conserved when dealing with the entire cluster instead of processing each point.

- *Robustness with hybrid similarity measure:* Learning model in *StreamAR* contains clusters that represent different activities. When new cluster is emerged, hybrid similarity measure is deployed to match up similarities of the new cluster/activity with the existing ones. These measures are namely distance, density, gravity and within cluster standard deviation (*WICSD*). Applying the aforementioned similarity measures for activity recognition shows superiority over the use of individual ones, and therefore enhances the system robustness across users.

To the best of our knowledge, no other existing activity recognition system addresses all aforementioned points in a single framework. The rest of the paper is organised as follows. Section 2 provides a discussion of the research context. Explanation of the proposed *StreamAR* framework and its details are presented in Section 3. Section 4 reports the experimental results and analysis. Finally, Section 6 concludes the paper with a summary.

II. Related Work

Our technique is related to both data stream classification and activity recognition. An efficient approach based on data mining has been recently proposed in a number of research projects considering the activity recognition from the machine learning perspective. Methods commonly used for activity classification were reviewed in [1]. Supervised learning has been deployed pervasively for activity recognition. One example system is explained in [2]. In this system, three classification techniques from WEKA [3] to induce models for predicting the user activities are decision trees (J48), logistic regression and multilayer neural networks. Some other systems used fuzzy classifiers for activity recognition as in [4] and [5]. Parkka et al [6] implemented a real time classification method using a binary decision tree with only four nodes. The system used default parameters that have been trained to give optimum performance for average users. The user might be interested in personalising the activity-recognition algorithm to achieve better recognition accuracy.

Few studies considered unsupervised learning techniques for activity recognition and change detection. For example, Lee et al. [7] used unsupervised learning for abnormality detection. To detect whether a pattern is registered or not, a probability model based on the past activity pattern is created. The Expectation-Maximisation (EM) algorithm [8] is used with the feature vectors to decide whether the activity is abnormal behaviour or not. In [9], the feasibility of applying a specific type of unsupervised learning to high-dimensional, heterogeneous sensory input was analysed. The correspondence between clustering output and classification input was proposed as well. Typically there is only a small set of labelled training data available in addition to a substantial amount of unlabelled training data. Therefore, some studies considered labelling only profitable samples of data or continue learning while system is running. Longstaff et al. [10] investigated methods of further training classifiers after a user begins to use them using active and semi-supervised learning. *StreamAR* combines supervised, unsupervised and active learning for building a robust activity recognition system across users. Typical activity recognition stream is formed from a sequence of data chunks representing activities. Thus, *StreamAR* treats data input as a stream and uses clustering to avoid having to respond to each input data point. As the stream evolves, there is a need to assess old and new clusters and this is handled with a hybrid similarity measure. None of the above systems and as far as we know have dealt with the streaming nature of unlabelled sensory data for recognising different activities. Practically, unlabelled data streaming from sensors requires real time classification. Current recognition systems use a static training model which is built offline to recognise new data. Yet, there is no notion to expand the model after it is already deployed. Indeed, analysing sensory data for real time model adaptation and personalisation is crucial to reflect changes in activities with evolving data streams.

Data stream classification has been an interesting research topic for years, and many approaches are available. Data stream classification techniques maintain and incrementally update a classification model and effectively respond to concept-drift discussed in [11], [12] and [13]. Cluster based technique for data stream is represented in [14]. This technique detected novel concepts in an unsupervised incremental learning fashion and applied to intrusion detection in computer networks. However, this is also a ”single-class” technique, where authors assume that the learning model has only one labelled class. Thus, it not directly applicable to activity recognition environment with multi classes.

Masoud et al. [15] integrated classification with novel class detection in concept-drifting data streams. However, The system focused on detecting novel classes but not adaptation and refinement of the initial learning model. It processed each point and test similarities and differences with the normal data to detect novel classes.

Our approach is different from the above in several aspects. First, most of the existing novelty detection techniques assume that the initial model is static. Existing techniques focus on novelty or outlier detection rather than learning model refinement and adaptation. Second, existing techniques test data stream individually in a point based approach, whereas our technique deals with data collectively to recognise activities and update the model according to the most recent changes in data stream. Third, our framework initial model composites of multi- classes. Thus, it can recognise and refine different activities. On the other hand, most of the existing streaming techniques can only deal with one normal class. And therefore, considered as ”one-class”classifiers. Finally, Our technique integrates active learning with stream mining for activity recognition.

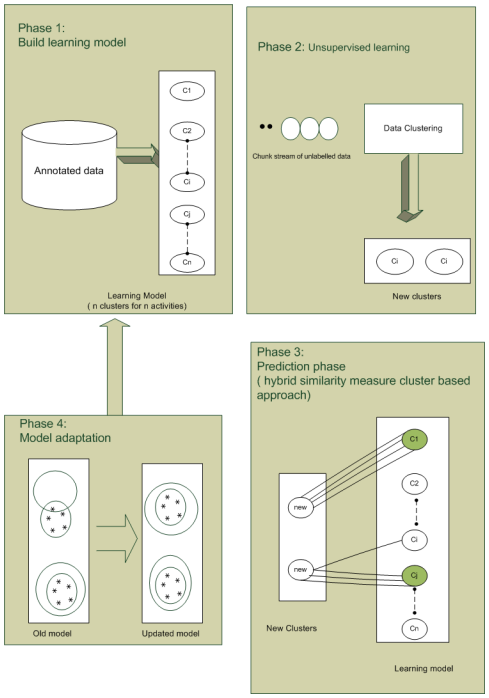


Fig. 1. Top Level Algorithm

III. StreamAR: Stream Learning technique for Activity Recognition

In this section we present *StreamAR*, our novel approach for incremental and active learning for activity recognition in sensory data streams. We start with describing the top level algorithm. Then we introduce the four phases applied for implementing the novel framework .

A. Top Level Algorithm

An overview is shown in Figure 1. In terms of the learning paradigm, it may be divided in four phases. A supervised learning phase, where a learning model is built from a set of examples that describe the data domain. An unsupervised learning phase, where chunks of unlabeled examples arriving from a data stream form new clusters. Recognition of new formed clusters is assessed by a hybrid similarity measure approach in phase 3. Based on the recognition, the learning model is refined and updated in real time to reflect recent changes as in phase 4.

B. StreamAR Phases

1) Phase 1: Build Learning Model : Initially, supervised learning is applied on labelled data to train and generate the learning model. The generated model consists of set of clusters. Each cluster represents one of the labelled activities that exist in data domain. After creating K clusters using the supervised algorithm, we extract and save summary of the statistics of the data points in each cluster as a micro-cluster and discard the raw data points. We will refer to the K micro-clusters built from training data as a learning model. We use these micro-clusters of the learning model to classify unlabelled received data.

Learning Model Purification: Each cluster lin the learning model has a label of the majority label among cluster instances. Training examples typically contain outliers and noisy data that might affect the quality of the model directly. Therefore, we add a filtration step that aims to purify clusters and therefore build more accurate learning model.

While creating new cluster from training data, cluster is purified by considering only true-labelled instances inside the cluster and ignoring other mis-clustered examples (instances with different labels). Building on the purified clusters, characteristics are extracted for each cluster and all raw points are then dismissed.

Considering computational, time and space complexity, *StreamAR* extracts features from each cluster and dismisses all raw data at the end of this phase. Micro clusters contain the basic information describes the learning model. Statistics about cluster include basically the cluster centroid, density, within cluster standard deviation and boundaries.

2) Phase 2: Unsupervised Learning for New Data: This step aims to create clusters of various activities exist in data received. When unlabelled data emerged, we apply clustering on data to generate clusters of the performed activities . Various clustering techniques such as k-means, Expectation Maximisation and DBScan [16] have used and compared to reach the best performance. Clusters Characteristics are extracted similar to the learning model building procedure. The output of this phase is the set of clusters'/activities' characteristics ready for the recognition phase.

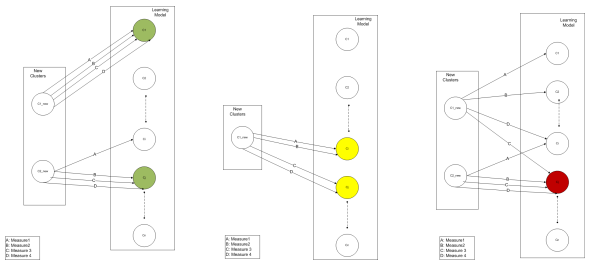
3) Phase 3: Prediction Phase : As the stream evolves, we assess new and learning model clusters to predict new clusters' labels. This is handled with a hybrid similarity measure approach. As raw data has been dismissed, characteristics of the new cluster are compared to the existing learning model ones. The predicted label is based on the characteristics of the most similar cluster in the learning model.

Clusters are similar if they match based on the hybrid similarity measure approach. For each new formed cluster, the algorithm checks how similar it is to other clusters already exist in the learning model. We apply various measures to test similarities among clusters. Each measure votes for its own " candidate" cluster from the measure respective. The predicted label is the candidate cluster with the majority of votes among all measures, while the true label of a cluster is the majority label among cluster instances. There are three cases expected from the voting procedure as follows.

1) ***Correct prediction:*** This case occurs when the majority of votes are for the micro cluster in the learning model with the true label. As showed in Figure 2(a), the majority of measures have chosen the candidate micro cluster with the true label - label of C_1 .
2) ***Active learning:*** In case of equal votes are assigned to a specific two micro clusters, user input is required to label the new cluster. Active learning is explained in Figure 2(b). In this case, exactly two measures vote for

C_i and other two measures vote for C_j. The algorithm inquires about the correct label as an input from user in an active learning mode with either the label of C_i or C_j.

3) ***Incorrect prediction:*** The cluster is incorrectly classified when a total confusion with lowest confidence among all measures for clusters or voted for an incorrect cluster. In this case, the algorithm has an incorrect prediction as explained in Figure 2(c).



(a) Correct Prediction (b) Active Learning (c) Incorrect Prediction

Fig. 2. Illustration of Prediction Cases

Hybrid similarity measure technique implements four measures namely distance, density, gravitational force and within cluster standard deviation. We concern in these measures about the distance among the cluster centroids, clusters density, how strong is the gravitational force among clusters and the cohesion inside the cluster. Learning model ***LM*** consists of ***n*** micro clusters/ activities.

$$LM = \{C_1, C_2, C_3, ...C_n\} \tag{III.1}$$

Micro Clusters contain statistics summary of the data points belong to the clusters. The summary includes the following statistics: the total number of points, the centroid of the cluster, boundary and density of the cluster. When new cluster C_{new} arrives, the algorithm deploys the various similarity measures to choose the best candidate micro cluster C_i from ***n*** clusters exist in LM. The four measures are:

- ***Distance:*** Micro cluster centroid is an n dimensional array of mean *n*- dimensional instances inside the cluster. C_i is the candidate micro cluster from distance perspective if distance between C_{new} and C_i centroids is the shortest among n clusters in LM.
- ***Density:*** Each cluster has its own density that distinguishes it from other clusters. We first define the Cluster Radii.

  Def: Cluster Radii: This measure gives a better understanding of the boundaries of cluster and how various clusters are close or far away from each other. It is considered as the maximum distance between any point insides the cluster and the cluster centroid as described in Equation III.2:

$$ClusRadii = \max(EDistance(P_i, \textit{ClusCntr})) \; \forall P_i \in C; \tag{III.2}$$

Where (C) is the cluster tested, $(EDistance)$ is the Euclidean Distance, (*P*) is a n- dimensional data point inside the cluster and *ClusCntr* is the cluster (*C*) centroid. Cluster density reflects the distribution of data points inside the cluster. It is described by the Formula III.3:

$$ClusDens = \frac{ClusMass}{ClusVolume} \tag{III.3}$$

$$ClusVolume = \frac{4}{3}\pi ClusRadii^3; \tag{III.4}$$

Where (*Clus Mass*) is the number of points in the cluster, (*Clus Volume*) is the volume of the cluster as a sphere. (*ClusRadii*) the maximum distance between any point insides the cluster and the cluster centroid.

When new cluster merged with an existing one, density of the new merged cluster is recalculated. There are two possible options may occur in case of merging. First, the density of the new cluster is bigger than the original one. Thus, the new emerged cluster increases the density of existing cluster. Second, density decrease when new cluster is merged with the learning model cluster.

In order to select the best candidate micro cluster in the *LM* from density perspective, we check the density gain/ loss if the new cluster merged with each of the existing *LM* micro clusters. Density gain is the difference between the new density of micro cluster C_i when merged with the new cluster C_{new} and old density before merging. C_i with the highest gain/ lowest loss among n clusters in LM is chosen as the candidate cluster by density measure.

- ***Gravitational force:*** Gravitational force has been previously applied in machine learning such as in [17] , [18], and [19] . There exists a natural attraction force between any two objects in the universe and this force is called gravitation force. According to Newton universal law of gravity, the strength of gravitation between two objects is in direct ratio to the product of the masses of the two objects, but in inverse ratio to the square of distance between them. The law is described in Equation III.5:

$$F_g = G\frac{m_1 m_2}{r^2} \tag{III.5}$$

Where F_g is the gravitation between two objects (clusters); *G* is the constant of universal gravitation; m_1 is the mass of object 1 (size of cluster 1); m_2 is the mass of object 2 (size of cluster 2); *r* the distance between the two objects (Euclidean distance between clusters' centroids) .

According to Equation III.5, each cluster generates its own gravitational force created from its weight. The bigger the weight of the candidate the stronger the gravitational force produced around it. Therefore, the probability it could attract more data object would be increased. When the gravitational force between $Cnew$ and C_i is bigger than with other micro clusters existing in LM, then C_i is the candidate micro cluster from gravitational force perspective.

- WICSD(Within Cluster Standard Deviation): This measure considers the cohesion inside each cluster. Standard deviation of n dimensional points inside the cluster is calculated as the equation III.6.

$$WICSD = \sqrt{\frac{\sum_{i=1}^{m} EDistance(P_i, ClusCntr)^2}{m}} \quad \text{(III.6)}$$

Where $EDistance$ is the Euclidean Distance, (*m*) is the number of points into the cluster, (P) is a n- dimensional data point inside the cluster and (*ClusCntr*) is the cluster centroid.

Clusters that have similar standard deviation are more likely to present the same activity/label. C_i is the candidate cluster form *WICSD* perspective if it has the smallest difference in standard deviation measure with C_{new} among n micro clusters in LM.

4) Phase 4: Model adaptation: StreamAR periodically updates the learning model to ensure that it represents the recent changes of users' activities. In case of incorrect and active learning of activity, user labels new clusters and feed it into the system for real time model adaptation.

Learning model micro clusters are updated to reflect changes in data stream in four aspects.

- Micro Cluster Centroid: As stream evolves, it is crucial to update the cluster centroids for maintaining high system accuracy. A numerically stable algorithm is given below in Algorithm 1. It computes the mean due to Knuth et al [20] that has been thoroughly analysed in [21].

Algorithm 1 $UpdateCentroid(data, Centroid)$

for x in $data$
n = n+ 1
$delta$ = x - $Centroid$
$Centroid$ = $Centroid$ + $delta/n$

Where n is the micro cluster size before update. The new cluster has the majority of a single true label. The system inquires about the entire cluster label instead of labelling each point. Cluster Centroid is updated inclemently by adding each point to micro cluster.

- *WICSD*: Centroid update algorithm is extended for updating variance and standard deviation [20]. Algorithm 2 illustrates the incremental update for *WICSD*.

Algorithm 2 $UpdateWICSD(data, Centroid, WICSD)$

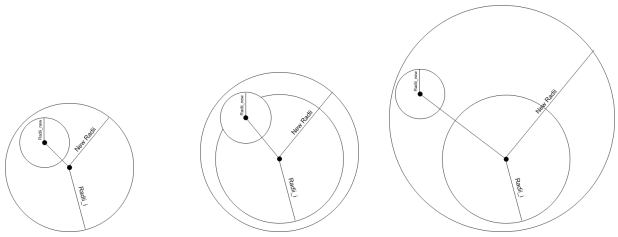
$M = (WICSD * n)^2$
for x in $data$
n = n+ 1
$deltaa$ = x - $Centroid$
$Centroid$ = $Centroid$ + $delta/n$
M = M + $delta$*(x - $Centroid$)
End for
$variance$ = M/n
$WICSD$= SQRT($variance$)

Where n is the micro cluster size before update.

- Cluster Gravity: Cluster gravity is automatically updated according to Equation III.5 with the updated sizes and centroids.
- Micro Cluster Density: Equation III.3 explained the calculation of density in creating the learning model. When adding new points into cluster C_i, we recalculate the density based on the new parameters. The algorithm updates micro cluster's mass by adding up the C_new size to the old size (prior to update). The updated cluster Radii relies on position of the new cluster, C_new. As illustrated in Figure 3, three cases occur when updating the Radii of C_i. Radii remains the same if the new cluster is fully contained inside the micro cluster C_i, as shown in Figure 3 (a). The other two cases are illustrated in Figure 3 (b)(c). These cases occur when C_new intersects with or fully separated from C_i. The radii is adapted as Equation III.7

$$NewRadii = \frac{EDis(Cen_i, Cen_{new}) + Rad_i + Rad_{new}}{2} \quad \text{(III.7)}$$

Where $EDis$ is the Euclidean Distance,Cen_i is the learning model cluster centroid, Cen_{new} is a centroid of the new cluster, Rad_i is the radius of C_i, and Rad_{new} is the new emerged cluster radii.



(a) Contained (b) Intersected (c) Separated

Fig. 3. Radii Update Cases

IV. Empirical Results

In this section we discuss the data sets used in the experiments, the system setup, the results, and analysis.

A. Experimental setup

We conducted our experiments on two real activity recognition data sets.

- OPPORTUNITY dataset [22]
  In the recently started European research project, the OPPORTUNITY dataset has been recorded to recognise complex activities from accelerometer sensors in addition to highly rich environmental sensors. They designed the activity recognition environment and scenario to generate many activity primitives, yet in a realistic manner. Thus, the dataset has labels for user activities like (sitting, walking and running) streaming form accelerometer sensors attached to the user's body. It consists of an annotated complex, multidimensional and naturalistic

activities, with a particularly large number of atomic activities (around 30000 for each segment), collected in a very rich sensor environment. The OPPORTUNITY dataset contains annotated four activities for five subjects across five different segments.

- COSAR dataset [23]
  This dataset is for 10 different activities performed both indoor and outdoor by volunteers having different attitude to physical activities. Each activity was performed by 4 different volunteers. While performing activities, volunteers wore one sensor on their left pocket and one sensor on their right wrist to collect accelerometer data, plus a GPS receiver to track their current physical location. the dataset is composed of 5 h of activity data. The dataset is composed of 18,000 activity instances. For each activity instance, accelerometer readings were merged to build a feature vector composed of 148 features, including means, variances, correlations, kurtosis, and other statistical measures.

To measure our system performance, we define some terms as performance meters. Let F_c = total existing class instances correctly classified, F_a = total existing class instances trigger active learning, F_i = total existing class instances misclassified, S = total instances the dataset. We use the following performance metrics to evaluate our technique. M_c: % of class instances correctly classified = $\frac{F_c * 100}{S}$, M_a: % of class instances requires active learning = $\frac{F_a * 100}{S}$, and ERR: % of misclassification class instances = $\frac{F_i * 100}{S}$.

StreamAR is tested on the OPPORTUNITY and COSAR datasets. Part of the data is used to build the learning model; other new date is applied for testing. Testing data is a new data that has not used for training the model. Learning and testing data could be for the same user but different segments or for different users. The default chunk size = 50 unless otherwise stated. For each chunk of incoming data stream we apply k-means clustering with k=2.

B. Learning model purification

Model purification is an essential step for pruning and refining the learning model. It has a superior advantage of building a robust model that filters outliers and wrong-labelled examples. Therefore, purified model represents activity of majority label with high confidence. We built the learning model with Subject 2 data - $S2$, while applying data for $S2$, $S3$ and $S4$ in testing the System. As shown in Table I, the system shows better performance with purified learning model.

Purification of the learning model has different effect on the various measures deployed. Purification reduces confusion among measures and therefore reduces active learning occurrence, M_a. Eliminating outliers and wrong-labelled instances affects the position of clusters' centroids. Therefore, purification boosts measures that rely mainly on the coordinates of centroids (i.e. distance). On the other hand, purification might eliminate far away instances or instances belong to low dense clusters that seemed to be outliers. Therefore, density

TABLE I
LM PURIFICATION AND *StreamAR* PERFORMANCE

LM	*Dataset*	M_c	M_a	ERR
$S2$ Pur	$S1$	69.28%	14.16%	16.56%
$S2$	$S1$	41.56%	45.73%	12.71%
$S2$ Pur	$S3$	61.59%	17.87%	20.54%
$S2$	$S3$	50.58%	29.47%	19.94%
$S2$ Pur	$S4$	57.59%	20.93%	21.48%
$S2$	$S4$	50.51%	29.72%	19.77%

and gravity might be affected negatively with purification. Although, purification has not enhanced the performance of all the measures, it has an overall positive effect with increasing M_c and reducing M_a when combining all measures.

C. Model adaptation

We conducted our experiments on both static and adaptable model. Dealing with streams that evolves over the time, model adaptation is essential to cope with recent changes. Figure 4 shows the StreamAR performance with both static and adaptable learning model. All runs are deployed on a purified learning model.

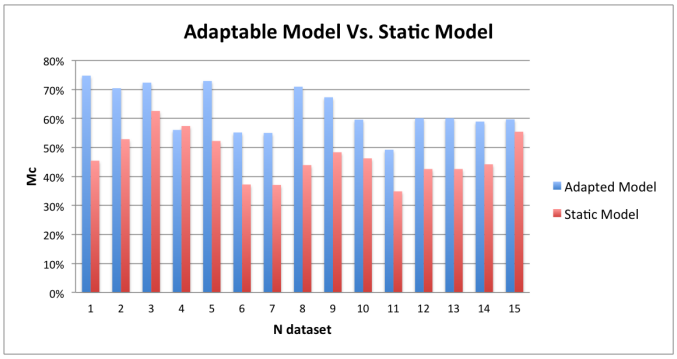


Fig. 4. *StreamAR* Performance with Adaptation

Adaptable technique out performed the static one for all runs. Indeed, feeding the system with true labelled points that have previously actively or mis-classified enhances the system accuracy. Figure 5 shows the error reduction due to adaptation for the same datasets.

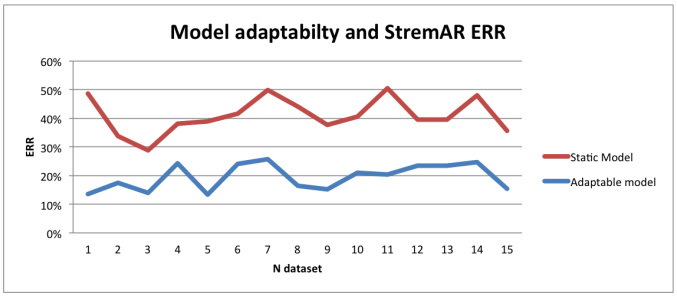
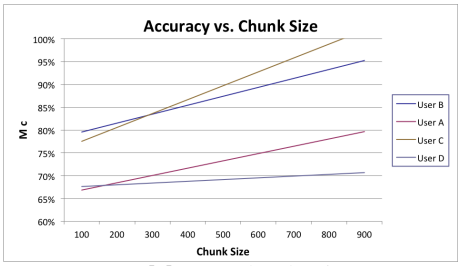


Fig. 5. *StreamAR* Performance with Adaptation

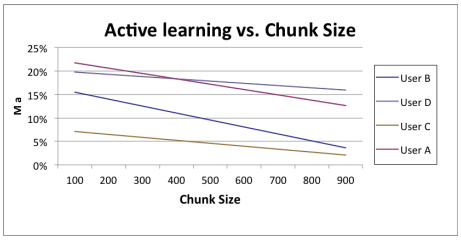
D. Chunk size

StreamAR allows user to specify chunk/window size for recognition. Bigger chunk size gives more exposure to data

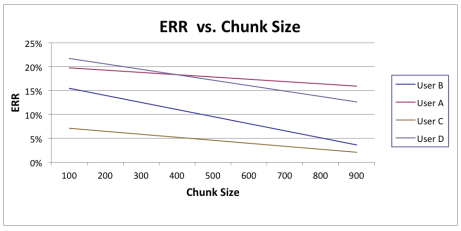
and allows better recognition. However, processing large data chunk has other drawbacks especially with time and complexity. Figure 6 illustrates the performance trend with various chunk sizes with COSAR dataset. Big chunks allow better understanding of data, therefore M_c showed better accuracy. Moreover, M_a and ERR gradually decreased as the data chunk enlarges.



(a) M_c vs. Chunk Size



(b)M_a vs.Chunk Size



(b)ERR vs. Chunk Size

Fig. 6. Chunk Size and *StreamAR* performance

E. StreamAR and other classification techniques

Table II illustrates the performance of *StreamAR* across various learning and testing datasets in comparison to other iconic classifications techniques. Across-users test deployed; We built the learning model with data from one user and test the system with different users. Learning model is adaptable and purified for all runs. Chunk size is set to 500 instances. The static classification methods applied are Weka built in methods namely: Support Vector Machine, Naive Bayed, Decision Trees and Random Forest Tree. To the best of our knowledge, there is no approach that applied stream mining for activity recognition. Therefore, we compare our system with the static classification methods applied pervasively in *AR*. However, all of these techniques are static and cannot adapt with evolving data and concept drift streams.

TABLE II
StreamAR RECOGNITION PERFORMANCE

Test	StreamAR		SVM	N.Bayes	DTree	RFTree
	M_c	M_a				
$D1$	**74.7%**	**11.61%**	77.8%	78.1%	63.3%	55.3%
$D2$	70.4%	12.1%	76.9%	**90.2%**	75.5%	37.9%
$D3$	72.3%	13.6%	75.6%	**90.4%**	73.2%	42.25%
$D4$	**55.1%**	**20.7%**	48.0%	72.8%	52.5%	58.6%
$D5$	**69.4%**	**11.1%**	52.5%	76.9%	53.9%	53.9%
$D6$	**49.2%**	**30.3%**	67.6%	40.9%	24.7%	34.72%
$D7$	**60.0%**	16.54%	41.1%	44.9%	29.6%	22.3%
$D8$	**60.0%**	16.5%	52.6%	46.2%	22.2%	30.3%
$D9$	**59.6%**	24.8%	46.5%	52.4%	22.0%	34.4%
$D10$	**67.2%**	**17.4%**	52.0%	73.8%	51.9%	61.0%

V. CONCLUSION

We address a realistic problem of stream mining with activity recognition. The novel technique combines active and incremental learning approach for recognising various activities. We integrate supervised, unsupervised and active learning to build a robust and efficient recognition system. Previous approaches for stream classification did not address this vital problem. We tested our technique on real datasets and discussed the system performance compared to other classification techniques.

REFERENCES

[1] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors: a review of classification techniques," *Physiological Measurement*, vol. 30, no. 4, pp. R1–R33, 2009.

[2] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, pp. 74–82, March 2011.

[3] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[4] M. Helmi and S. M. T. Almodarresi, "Human activity recognition using a fuzzy inference system," in *IEEE International Conference on Fuzzy Systems*, 2009, pp. 1897–1902.

[5] J.-Y. Yang, Y.-P. Chen, G.-Y. Lee, S.-N. Liou, and J.-S. Wang, "Activity recognition using one triaxial accelerometer: A neuro-fuzzy classifier with feature reduction," in *Entertainment Computing - ICEC 2007, 6th International Conference*, 2007, pp. 395–400.

[6] J. Pärkkä, L. Cluitmans, and M. Ermes, "Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree," *Trans. Info. Tech. Biomed.*, vol. 14, pp. 1211–1215, September 2010.

[7] M. hee Lee, J. Kim, K. Kim, I. Lee, S. H. Jee, and S. K. Yoo, "Physical activity recognition using a single tri-axis accelerometer," in *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I, WCECS '09, October 20 - 22, 2009, San Francisco, USA*, ser. Lecture Notes in Engineering and Computer Science, S. I. Ao, C. Douglas, W. S. Grundfest, and J. Burgstone, Eds., International Association of Engineers. Newswood Limited, 2009, pp. 14–17.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, B*, vol. 39, pp. 1–38, 1977.

[9] F. Li and S. Dustdar, "Incorporating unsupervised learning in activity recognition," in *AAAI Workshops; Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[10] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth) 2010*, march 2010, pp. 1 –7.

[11] S. Chen, H. Wang, S. Zhou, and P. S. Yu, "Stop chasing trends: Discovering high order models in evolving data," pp. 923–932, 2008.

[12] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *IN PROC. OF THE 2001 ACM SIGKDD INTL. CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING*. ACM SIGKDD, 2001, pp. 97–106.

[13] Y. Yang, "Abstract combining proactive and reactive predictions for data streams," pp. 710–715, 2005.

[14] E. J. Spinosa, A. P. de Leon F. de Carvalho, and J. a. Gama, "Olindda: a cluster-based approach for detecting novelty and concept drift in data streams," in *Proceedings of the 2007 ACM symposium on Applied computing*, ser. SAC '07. New York, NY, USA: ACM, 2007, pp. 448–452. [Online]. Available: http://doi.acm.org/10.1145/1244002.1244107

[15] M. M. Masud, L. Khan, and B. Thuraisingham, "Integrating novel class detection with classification for concept-drifting data streams," in *In ECML PKDD*, 2009, pp. 79–94.

[16] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[17] L. Peng, B. Yang, Y. Chen, and A. Abraham, "Data gravitation based classification," *Inf. Sci.*, vol. 179, no. 6, pp. 809–819, 2009.

[18] Z. S. Abdallah and M. M. Gaber, "Ddg-clustering : A novel technique for highly accurate results," in *Proceedings of the IADIS European Conference on Data Mining*, 2009.

[19] ——, "Kb-cb-n classification: Towards unsupervised approach for supervised learning," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence*, 2011, pp. 283–290.

[20] D. E. Knuth, *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.

[21] T. F. Chan, G. Golub, and R. J. LeVeque, "Algorithms for computing the sample variance: Analysis and recommendations," *The American Statistician*, vol. 37, pp. 242–247, 1983.

[22] D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Tröster, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, and J. del R. Millán, "Opportunity: Towards opportunistic activity and context recognition systems," in *Proc. 3rd IEEE WoWMoM Workshop on Autononomic and Opportunistic Communications*, 2009.

[23] D. Riboni and C. Bettini, "COSAR: hybrid reasoning for context-aware activity recognition," *Personal and Ubiquitous Computing*, vol. 15, no. 3, pp. 271–289, 2011.