A Genetic Algorithm-based Classifier Ensemble Optimization for Activity Recognition in Smart Homes

Iram Fatima, Muhammad Fahim, Young-Koo Lee and Sungyoung Lee
Department of Computer Engineering, Kyung Hee University, 446-701, South Korea
[e-mail: iram.fatima@oslab.khu.ac.kr, fahim@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr]
*Corresponding author: Young-Koo Lee

Abstract

Over the last few years, one of the most common purposes of smart homes is to provide human centric services in the domain of u-healthcare by analyzing inhabitants' daily living. Currently, the major challenges in activity recognition include the reliability of prediction of each classifier as they differ according to smart homes characteristics. Smart homes indicate variation in terms of performed activities, deployed sensors, environment settings, and inhabitants' characteristics. It is not possible that one classifier always performs better than all the other classifiers for every possible situation. This observation has motivated towards combining multiple classifiers to take advantage of their complementary performance for high accuracy. Therefore, in this paper, a method for activity recognition is proposed by optimizing the output of multiple classifiers with Genetic Algorithm (GA). Our proposed method combines the measurement level output of different classifiers for each activity class to make up the ensemble. For the evaluation of the proposed method, experiments are performed on three real datasets from CASAS smart home. The results show that our method systematically outperforms single classifier and traditional multiclass models. The significant improvement is achieved from 0.82 to 0.90 in the F-measures of recognized activities as compare to existing methods.

Keywords: Activity recognition, Classifier ensemble, Weisghted classification, Genetic algorithm, Smart Homes

1. Introduction

With the growing healthcare requirements of aging population, smart home technology has attracted a lot of attention. A smart home is an intelligent agent that perceives the state of resident and the physical environment using sensors. It is one of the best solutions that allow the provision of monitoring and health assistance for persons with special needs and the elderly to receive services in their own home environments [1]. In recent years, several smart homes have been developed, such as CASAS and MavHome [2] at Washington State University, Aware Home [3] at Georgia Tech University, Adaptive House [4] at University of Colorado, House_n [5] at Massachusetts Institute of Technology (MIT), and House A [6] at Intelligent Systems Laboratory. The advancement of sensor technology has proven itself to be robust, cost-effective, easy to install and less intrusive for inhabitants. This fact is supported by a large number of applications developed using activity recognition to provide solutions to a number of real-world problems, such as remote health monitoring, life style analysis, interaction monitoring, and behavior mining [7] [8].

Many researchers have designed a variety of models and methods to recognize the activities of daily living and have greatly contributed to improve the smart home technology [7-10]. Despite the great work and diversity in the existing classification methods, the most notable problem is that single classifier cannot always lead to good recognition results. Sometimes, a classifier can outperform other classifiers on a particular problem but in general, it is not always the case when one classifier overrides the others in all possible situations. The process of selecting an appropriate classifier is still a trial and error process that clearly depends on the relationship between the classifier and the smart home characteristics [11]. The predictable factors, such as the available amount of training data, the spatial variability of data samples, deployed sensors in smart homes, and the total activity occurrences in the dataset influence the performance of classifiers to a significant degree. Suppose there are four classifiers k_1, k_2, k_3 and k_4; and three classes c_1, c_2, and c_3. Let us assume that depending on the training set and the set of features used, classifier k_1 is more efficient to classify given points in class c_1. Similarly, depending on the configuration, classifiers k_2, k_3, and k_4 are more efficient in classifying points in classes c_2, c_3, and c_1, respectively. In most of the practical applications, the reliabilities of predictions vary among the various classes in any classifier.

In order to overcome the problems of single classifiers, classifier combination has been proven to be more accurate and robust than an excellent single classifier in many application domains [12-14]. A major factor in classifier combing methods is that the individual classifiers can be as diverse as possible. Therefore, in this study, a novel method to recognize daily life activities is proposed by optimizing the measurement level output in terms of weighted feature vectors of classifiers using GA. This solves the problem in which single classifier learners suffer from statistical, computational, and representational problems which may affect the accuracy. Statistical problems arise due to high dimensional variance in the sensors data that excessively increase the size of the search space. Computational problems occur when the training data is computationally complex and can get stuck in local optimum. Our proposed ensemble optimization method combines the complementary performance advantages of classifiers for more accurate results and does not require any technique to select the most suitable subset of classifiers from a set of base classifiers. In the proposed method, the weighted feature vector of a classifier is determined from its training performance for each class, which indicates the possibility that the input sensor values pertains to the class. As a result, the weighted feature vectors of all classifiers are ensemble together in GA to learn the

optimized activity rules for final decision about activity class label. We integrated the weighted feature vectors of probabilistic and statistical methods, such as the Artificial Neural network (ANN) [9], Hidden Markov Model (HMM) [7], Conditional Random Field (CRF) [8], and Support Vector Machines (SVM) [10] as base classifiers. To evaluate and validate the proposed approach, experiments are performed using three real datasets collected in the CASAS smart home [2], a research project at Washington State University (WSU). The recognition results show the significance of proposed approach in terms of high accuracy as compare to traditional methods.

The rest of the paper is organized as follows. We briefly describe related work and their limitations in Section 2. In Section 3, we introduce our proposed GA based classifier ensemble optimization method for activity recognition in smart homes. In Section 4, we analyse and evaluate our experimental results to validate the proposed approach. Finally, conclusion and future work is presented in Section5.

2. Related Work

In the last decade, a lot of research has been done in the area of classifier ensemble for designing high performance classification systems [12-15]. A classifier ensemble is used under different names, such as combing classifiers, committees of learners, mixtures of experts, classifier fusion, and multiple classifier systems [14-16]. The combine decision has been proven to be more accurate in long run than classification decision of the best single classifier. In this regard, Genetic algorithm (GA) is known as one of the best search algorithms for the optimization of classifiers output [17]. There exist several previous studies on the usage of GA as learning classifiers that may vary in the number of classifiers, using different combing methods for intended application domains.

Matthew *et al.*[18] proposed an extended version of learning classifier and utilized GA to produce generalizations over the space of all possible condition-class combinations. They optimized the learning speed for terabytes of data in their parallel data mining systems. Ekbal *et al*. [12] applied GA to recognize named entities for Bengali, Hindi, Telugu, and Oriya. To find more accurate results they quantify the amount of votes per classifier for each output class. They used Maximum Entropy (ME), Conditional Random Fields (CRF), and Support Vector Machine (SVM) as base classifiers for GA based ensemble. Kuncheva *et al*. [17] used a GA to design the classifier fusion system and determined that, as a learner component, GA outperforms other classifier models. They combined Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), and logistic classifier for better accuracy of results. Rongwu *et al*.[19] proposed classifier ensemble as a learning paradigm where many classifiers are jointly used to solve the prediction problem. They used seven wearable sensors including five accelerometers and two hydrophones. Their used classifiers are Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier, k-Nearest Neighbor (k-NN) and Classification and Regression Trees (CART). Also, GA has been successfully used as a learner to select optimal genes for analyzing DNA microarrays [20] where classification is treated as two class problem. A survey about evaluation of ensemble classifiers with imbalanced data problem in protein-protein interactions is found in [21]. All theoretical and empirical studies of GA learning show higher accuracy in real world applications, such as spam email filters, character recognition, text categorization, face recognition, computer-aided medical diagnosis, pattern recognition and gene expression analysis [18] .

The state-of-the-art and most popular activity recognition techniques are based on probabilistic and statistical models like Hidden Markov Models (HMM)[7], Conditional

Random Fields (CRF)[8], Artificial Neural Network (ANN)[9], Support Vector Machine[10], and some other classification methods[22-24]. A survey of all these works can be found in [25]. Kamrul *et al.*[26] resolved the problem that when number of sensor activation sequence or any combination will be huge and handling the combination is beyond the capability of traditional classifiers. They used multiclass Adaboost to get a lightweight classifier for activity recognition. Adaboost is suitable for situation when there is no key feature but the features together, may form a strong classifier. However, a number of difficulties and limitations remain with these approaches. The learning capability of these models depends on the observation of activity class distribution and the transitions between adjacent activities according to the characteristics of smart homes. In practice, no single classifier can achieve an acceptable level of accuracy on all datasets [11]. Every classifier operates well on different aspects of the training or test data. To overcome the limitations of existing work, we proposed an alternative state-of-the-art GA based classifier ensembles optimization for the recognition of daily life activities. The measurement level outputs of individual classifiers are fed into GA ensemble as input for combining the benefits of classification performance for each classifier. As a result, optimization of the output weights of multiple classifiers improves the accuracy of results when compared with existing method.

3. Classifier Ensemble Optimization with Genetic Algorithm

Here, we propose a GA-based classifier ensemble optimization method that integrates the measurement level classification results generated by multiple classifiers in terms of weighted feature vector. Let $\Omega = \{S_1, \dots, S_n\}$ be a set of n embedded sensors, e.g., {stove-sensor, refrigerator-sensor, microwave-sensor, door-sensor etc.} Let daily life activities be divided into a set of m classes $C = \{c_1, \dots, c_m\}$. Consider activity recognition problem where set of n sensors are assigned to one of the m possible class. Let's assume that we have k classifiers and $W_k = [w_1, w_2, \dots, w_m]$ is the weighted feature vector representing the relative significance of kth classifier for all classes. The weight w_i is the degree of importance of kth classifier for class i. This implies the estimation of how important k^{th} classifier is in the classification of the class i compared to the other classifiers. The search space for classifier ensemble is defined as $SP = (c_i, [W_{1k}, \dots, W_{mk}])$ and the activity rule space is defined as $R = \propto (c_i, [OW_{1k}, \dots, OW_{mk}])$, where OW_{ik} can be value of optimized weighted feature vector of kth classifier on a given threshold $\propto$ with "don't care term". A GA based Classifier Ensemble Learner (CEL) for m classes is a mapping from search space to optimized activity rule space and is defined as:

$$CEL_1^m: \ SP \rightarrow R \tag{1}$$

The architecture of the proposed method is shown in **Fig. 1**, it consists of four major components: (1) Data preprocessing: to represent the sensory data as an observation vector for classifier input, (2) base classifier for Activity Recognition (AR): to provide details about applied classifiers with preferred parameters settings, (3) a GA based classifier ensemble learner: to optimize the weighted feature vectors of multiple classifiers, and (4) recognition phase: to recognize the performed activities. The details of each component are described in the following sections.

3.1 Data Pre-processing

Data pre-processing is an important step towards accurate training of machine learning

techniques [27]. Data collected from ubiquitous sensors based on subject interactions are stored in sensor logs "*SL*" and annotation files "*AF*" with attributes start time, end time, sensor id, sensors value and activity label. In order to recognize the performed activities, recorded dataset is pre-processed into the form of $PD = \{(x_1, y_1), \dots, (x_n, y_m)\}$. The "$x_i$" is the vectors whose components are the values of embedded sensors $\{S_1, \dots, S_n\}$ such as stove-sensor, refrigerator-sensor, and door-sensor. The values of "y" are drawn from a discrete set of classes $\{c_1, \dots, c_m\}$ such as a "Leave Home", "Read", and "Sleep". Furthermore, excessive information such as multiple header lines is also removed from the sensor logs and annotation files. The pseudo code for the data pre-processing is given in **Algorithm 1**.
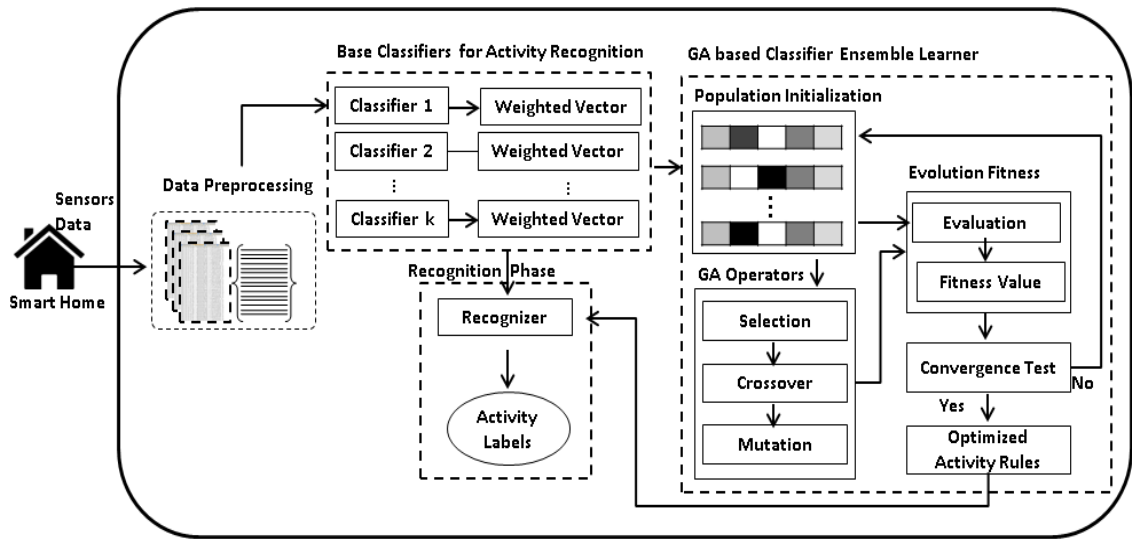
Fig 1. The proposed architecture for classifier ensemble optimization

Algorithm 1: Data Preprocessing

Input:	*SL* – Sensor Log
	AF – Annotation File
Output:	*PD* – Processed Data

Begin
```
1  for i=1: length(AF)
2    index=1
3    for j=1: length (SL)
4      if (AF(startTime(i) >= SL(startTime(j)) && AF(endTime(i) <= SL(endTime(j)))
5        firedEvents(index) = SL.value(j)
6        index = index+1
7      end
8    end
9    PD(i) = [firedEvents, AF(c(i))]
10 end
```
End

Algorithm 1. Data Processing

3.2 Base Classifiers for Activity Recognition (AR)

In this section, we introduce the base classifiers[1] (i.e., ANN, HMM, CRF, and SVM) used for ensemble optimization in the proposed method with preferred settings of our experiments. For this purpose, an input is the set of sensor values and output is its corresponding measurement (weight) for each activity class. Brief description of each classifier is given as

3.2.1 Artificial Neural Networks (ANN) for AR

It is an information processing network of artificial neurons connected with each other through weighted links. In activity recognition, the structure of the network, number of hidden layers, number of neuron in each layer with number of deployed sensors and total occurrences of activities in smart homes affects the learning process of different activities. The activation of the neurons in the network depends on the activation function [9]. In the proposed method, multilayer neural network with back propagation learning algorithm is utilized to recognize the human activities [9] and the weights are updated by the following equation:

$$\Delta w_{ki} = -c \left[-2 \sum_{j} \{(y_{j(desired)} - y_{j(actual)})f'(act_j)w_{ij}\} f'(act)_i x_k \right] \tag{2}$$

where Δw is the weights adjustment of the network links. In our network, we use one hidden layer with twenty neurons, tangent sigmod function as an activation function given below:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \tag{3}$$

Learning of the network is limited to maximum 1000 epochs. We use Neural Network Toolbox [28], the multi-layer neural network can be seen as an intuitive representation of a multi-layer activity recognition system

3.2.2 Hidden Markov Model (HMM) for AR

It is a generative probabilistic graph model that is based on the Markov chains process [7]. The training model is based on the number of states (activity class labels) and their transition weight parameters. Parameters are learned through observation (sensors value) and following parameters are required to train the model:

$$\lambda = \{A, B, \pi\} \tag{4}$$

Where λ is graphical model for activity recognition, A is a transition probability matrix, B represents the output symbol probability matrix, and π is the initial state probability [7]. We use Baum-Welch algorithm [29] to determine the states and transition probabilities during training of HMM. The ith classification weight of an activity is given as:

$$\lambda_i = \{A_i, B_i, \pi_i\}, \qquad i = 1, \dots, N \tag{5}$$

---

[1] The scope of variables used to describe the functionality of base classifiers is limited to this section.

3.2.3 Conditional Random Fields (CRF) for AR

It is a discriminative probabilistic graph model for labelling the sequences. The structure of the CRF is similar to HMM but learning mechanism is different due to absence of the hidden states [8]. In CRF model, the conditional probabilities of activity labels with respect to sensor observations are calculated as follows:

$$p(y_{1:T}|x_{1:T}) = \frac{1}{Z(x_{1:T}, w)} exp \left\{ \sum_{j=1}^{N_f} w_j F_j(x_{1:T}, Y_{1:T}) \right\}$$ (6)

In equation 6, Z denotes normalized factor and $F_j(x_{1:T}, Y_{1:T})$ is a feature function. To make the inference in the model, we compute the most likely activity sequence weights as follows:

$$y_{1:T}^* = argmax_{y'_{1:T}} p(y'_{1:T}|x_{1:T}, w)$$ (7)

We use the crfChain [30] with UGM[31], a simple, customizable, and open source implementation of CRF for segmenting or labelling sequential sensor events.

3.2.4 Support Vector Machine (SVM) for AR

SVM is statistical learning method to classify the data through determination of a set of support vectors and minimization of the average error [10]. It can provide a good generalization performance due to rich theoretical bases and transferring the problem to a high dimensional feature space. For a given training set of sensors value and activity pairs, the binary linear classification problem require the following maximum optimization model using the Lagrangrian multiplier techniques and Kernel functions as:

$$\text{Maximize (w.r.t } \alpha) \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=0}^{n} \sum_{j=1}^{n} \alpha_i y_i \alpha_j y_j K(x_i, x_j)$$ (8)

$$\text{Subject to: } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$$ (9)

Where K is the kernel function that satisfies $K(x_i, x_j) = \Phi^T(x_i)\Phi(x_j)$. In our case, we use radial basis function (RBF) for recognizing the activities.

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{(2\sigma^2)}\right)$$ (10)

We use the SVM [32] classifier, by simply substituting every dot product of activities weights in dual form with RBF kernel function, SVM can handle its non-linear nature. The activity recognition is multi-class problem so "one-versus-one" method is adopted to classify the weights of different activities.

3.3. A GA-based Classifier Ensemble Learner

In this section, a GA-based classifier ensemble learner is designed from the output of base classifiers in terms of weighted feature vectors that optimizes the measurement level

classification results into a final decision about activity class label. The weighted features vectors are fed into GA through chromosomes that reflect the relative importance of each classifier for a particular activity class. The pseudo code of ensemble learner is presented in **Algorithm 2**. We model the population initialization, evolution fitness, and stochastic operators of the GA as follows:

Algorithm 2: GA-based Classifier Ensemble Learner

Input: λ - crossover rate
         n - number of generations
         μ - mutation rate
         s - population size
         t - training classifier data
         $\propto$ - fitness threshold
Output: R - optimized activity rules
Begin
1 $pop = randInitial(s)$
2 **while** $(!max(n)||convg(n))$
3 $\quad| \;\; fitness = compEvoultionFitness(pop, t, \propto)$
4 $\quad|$ **if** $!(fitness)$
5 $\quad||\;\; pop = fGAOperation(pop, \lambda, \mu)$
6 $\quad|$ **end**
7 **end**
8 $R = ranked(pop)$
End

Algorithm 2. GA-based Classifier Ensemble Learner

3.4. Population Initialization

In GA population, weights of classifiers are combined into a string of real values as chromosome. Here, the problem to be solved is the optimization of weighted feature vectors in order to combine different measurement values given by k classifiers and determine the final decision. The well-known Michigan approach [33] is used to maintain a population consisting of candidate weighted feature vectors as a set of genes in chromosome that represents a single activity rule. Each activity rule of length β consists of two portions, the antecedent portion (1 to β-1) is the logical combination of weights of k classifiers and the subsequent portion (last value i.e., 4) represents the activity class c_i . The size of the activity rule $\beta = m \times k$ is fixed depending on the number of k classifiers and m classes in a smart home. The representation of weights is divide into k parts for k classifier system: $W_1,...,W_k$. W denotes the weight vector for k^{th} classifier, w_{ik} is the ith weight of W_k, and represent the relative importance of the kth classifier for class i. It is a positive number between 0.0 and 1.0. For example, when m=3 and k=4, a possible chromosome encoding is shown in **Fig. 2**.

Sample Values	0.59	0.12	0.29	0.33	0.25	0.42	0.35	0.49	0.16	0.15	0.24	0.61	4
Classifier weights	w_{11}	w_{12}	w_{13}	w_{21}	w_{22}	w_{23}	w_{31}	w_{32}	w_{33}	w_{41}	w_{42}	w_{43}	c_i
		k_1			k_2			k_3			k_4		Activity Class

Fig. 2. Activity Rule Encoding

The initial population is generated by setting the weights in each chromosome randomly. Once initial population is generated, the GA stochastic operations iteratively update the population according to the evolution fitness to assign the activity class.

3.5. Evolution Fitness

In each iteration, an objective function called evolution fitness is used to qualify each activity rule and score it according to its performance in the classification optimization process. The evolution fitness function "EF" evaluates the candidate weighted vectors wv of classifiers measurement level output in search space against optimized activity rules R for a m classes on the following basis.

$$EF = \frac{\sum_{i=1}^{pop} F_i(wv, R_i)}{pop} \tag{11}$$

Where "pop" is the size of GA population and

$$F_i(\text{wv}, R_i) = \begin{cases} 1 & \sqrt{\sum_{j=1}^{mk}(wv_j - R_{ij})^2} \leq \propto \\ 0 & otherewise. \end{cases}$$

here, $\sqrt{\sum_{j=1}^{mk}(wv_j - R_{ij})^2}$ is the measure to calculate the difference between search space and activity rule space based on activity class differences where $\propto$ is the constant to control the influence of success on overall learning process. Chromosomes are ranked according to these scores called fitness values.

3.6. GA Stochastic Operations

In the each iteration of GA, a new population is generated by probabilistically selecting the fittest chromosomes from the previous population. Some of the chromosomes are transferred intact into the next generation. The others are used as a basis for creating new offspring by applying genetic operators, such as selection, crossover, and mutation. The pseudo code for GA stochastic operations is illustrated in **Algorithm 3**.

```
Algorithm 3: fGAOperation - GA stochastic operations
Input:    ipop(1...m) – initial population
          λ – crossover rate
          μ – mutation rate
Output:  fpop(1...n)  - final population
Begin
1  for i=1: (⌊ipop(λ)⌋)
2  |  cp₁ = rand (c)
3  |  cp₂ = rand (c)
4  |  par₁ = rand (upper (ipop/2))
5  |  par₂ = rand (lower (ipop/2))
6  |  ofsp₁ = concat (par₁ (1: cp₁), par₂ (cp₁: cp₂), par₁ (cp₂: end))
7  |  fpop (1...n) = rand (⌊ofsp₁(μ)⌋)
8  |  ofsp₂ = concat (par₂ (1: cp₁), par₁ (cp₁: cp₂), par₂ (cp₂: end))
9  |  fpop (1...n) = rand (⌊ofsp₂(μ)⌋)
10 end
End
```

Algorithm 3. GA Stochastic Operations

3.6.1 Selection

In the proposed solution, selection of the fittest chromosome is based on its ranking according to evolution fitness. The whole population is sorted in descending order of fitness values, and a pair of parent selection for crossover operator incorporated the low fitness chromosome with the best fit chromosome. After ranking of population, in order to guarantee exploration of whole search space, one parent is randomly selected from top 50% of highly ranked population, while the other is selected from the other half of the population.

3.6.2 Crossover

Crossover is performed on the selected parents to create the new offspring. A dynamic two point crossover is applied as a reproduction operator. Two uniform cut points are selected at random from the integer range $[1, \beta-1]$ and two new state strings *Ofsp1* and *Ofsp2* are created by swapping the values between cut points. For example, if the value of cut points $cp_1=3$, and $cp_2=9$ are selected randomly as crossover points, we exchange the values around that point as shown in **Fig. 3**. In the each iteration of GA, the fittest replacement mechanism is applied so the entire generation is replaced with the new generation by keeping the best fit from last generation.

					cp_1					cp_2		
Par_1	0.23	0.56	0.21	0.32	0.15	0.53	0.68	0.21	0.11	0.74	0.15	0.11
Par_2	0.26	0.59	0.15	0.31	.052	0.35	0.65	0.07	0.28	0.32	0.23	0.45
$Ofsp_1$	0.23	0.56	0.21	0.31	.052	0.35	0.65	0.07	0.28	0.74	0.15	0.11
$Ofsp_2$	0.26	0.59	0.15	0.32	0.15	0.53	0.68	0.21	0.11	0.32	0.23	0.45

Fig. 3. An example of two-point crossover

3.6.3 Mutation

In the proposed approach mutation inaugurates the diversity in current population to increase the fitness of chromosomes. The mutation operator flips a random value on randomly selected genes of a chromosome according to the mutation rate. This operation confirms the diversity in the weighted ensemble of classifiers and avoids the stagnation of search space during the optimization process.

   The stopping criterion for classifiers ensemble learner is either a fixed number of generations or all training instances passed correctly. Later in the results and evaluation section, we discuss the optimal values for the number of generations, the size of the population, the crossover rate, and the mutation rate.

```
Algorithm 4: Recognizing the activities
Input:    R(1 ...m) − optimized activity rules
          wv − base classifiers weighted features
          ∝ - fitness threshold
Output:  Al− Activity label
Begin
1 for j = 1: size(R)
2 |  if (cLabel (wv, R(1 ...m)) < ∝)
3 |  |  voteList[] = vote+1
4 |  end
5 |  Al = max (voteList)
6 end
End
```

Algorithm 4. Recognizing the Activities

3.7. Recognition Phase

This phase recognizes the activity class label based on the classifier weighted features and optimized activity rules. For a particular set of classifiers weighted features, optimized activity rules are fired after considering the fitness threshold to recognize activity class labels. If more than one activity rules are fired then conflicting class labels are resolved by majority voting. The pseudo code for the recognition phase is given in **Algorithm 4**.

4. Results and evaluation

In this section we present the results to evaluate and validate the feasibility of classifier ensemble optimization in the activity recognition domain.

4.1. Datasets Description

The experiments are performed on three datasets from CASAS [2] smart home. The *Tulum2009* dataset was recorded in an apartment by deploying 18 motion and 2 temperature sensors. Two volunteers performed 10 daily life activities for 83 days. In case of *Milan2009*, 31 motion sensors, 2 door sensors and 2 temperature sensors were deployed on everyday objects. One volunteer performed 15 daily life activities for 62 days. For *TwoSummer2009*, 51 motion sensor, 4 item sensor, 15 door sensors, 5 temperature sensors, 1 electricity usage sensors, and 10 light sensors were deployed on '*Kyoto*' test bed. Two volunteers performed 11 daily life activities for 55 days. The details description of the datasets and annotation method can be found in 2]. In **Table 1**, the characteristic of *Tulum2009*, *Milan2009*, and *TwoSummer2009* dataset are shown. The 'Num.' column shows activities count, 'Time' column shows the time in seconds and 'Sensor' column shows generated sensor events.

Table 1. Characteristics of the annotated activities of CASAS smart home datasets

Tulum2009				Milan2009			
Activities	**Num.**	**Time**	**Sensor**	**Activities**	**Num.**	**Time**	**Sensor**
Idle	-	102986.4	203408	Idle	-	911.233	5760
Wash Dishes	71	1204.84	24869	Bed to Toilet	89	379.37	1255
Watch TV	528	4955.43	52222	Sleeping	96	37217.9	22172
Enter Home	73	119.42	604	Leave Home	214	4229.47	4946
Leave Home	75	101.58	1854	Watch TV	114	5919.72	23688
Cook Breakfast	80	1440.31	33435	Chores	23	684.82	7587
Cook Lunch	71	972.74	24527	Desk Activity	54	743.74	7628
Group Meeting	11	1847.04	31084	Dining Rm	22	330.37	4295
R1 Eat Breakfast	66	932.87	20077	Evening Medicines	19	10.56	250
R1 Snack	491	4461.85	81183	Guest Bathroom	554	7526.81	128942
R2 Eat Breakfast	47	497.06	13649	Kitchen Activity	330	952.31	10601
TwoSummer2009				Master Bathroom	306	1946.33	15071
Idle	-	18550.79	267222	Master Bedroom	117	2168.97	27337
Meal Preparation	199	1788.32	41744	Meditate	17	109.94	1315
R1 Sleeping	52	18202.24	43914	Morning Medicines	41	45.97	1023
R2 Sleeping	26	9723.08	29762	Read	314	10942.7	50281
Cleaning	28	143.31	3375	-	-	-	-
R1 Work	443	22262.50	284570	-	-	-	-
R2 Work	352	7245.24	84965	-	-	-	-
Grooming	47	351.14	7172	-	-	-	-

R1 Shower	41	556.52	4125	-		-	-	-
R1 Wakeup	48	15.84	289	-		-	-	-
R2 Shower	28	321.52	4950	-		-	-	-
R2 Wakeup	25	36.11	393	-		-	-	-

4.2. Performance Measures

In order to evaluate our proposed method, the four standard metrics of precision, recall, F-measure and accuracy are used as performance measures. They are calculated using the values of the confusion matrix [34] and computed as:

$$\text{Precision} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{TP_i}{NI_i} \tag{12}$$

$$\text{Recall} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{TP_i}{NG_i} \tag{13}$$

$$\text{F} - \text{Measure} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{14}$$

$$Accuracy = \frac{\sum_{i=1}^{Q} TP_i}{Total} \tag{15}$$

Where Q is the number of performed activities, TP is the number of true positives, NI is the total number of inferred labels and NG is the total number of ground truth labels. 'Total' is the total number of a particular activity performed in the dataset.

4.3. Experiments and Discussion

The proposed method has been implemented in MATLAB 7.6. The configuration of the computer is an Intel Pentium(R) Dual-Core 2.5 GHz with 3 GB of memory and Microsoft Window 7. We split the dataset using the 'leave one day out' approach; therefore, the sensor readings of one day are used for testing and the remaining days for training. In **Fig. 4**, the optimal values for the number of generations, the size of the population, the crossover rate and the mutation rate are evaluated. For population size, we analysed its range from 25 to 60 and found an optimal point at 45 as shown in **Fig. 4(a)**. Similarly, different generation sizes are estimated for the convergence of the proposed method and a stable point is observed after 400 generations, there is no significant improvements are found after this point as shown in **Fig. 4(b)**. We also analysed the optimal point for crossover and mutation rate with different values and discovered optimal points at 0.6 and 0.04 as shown in **Fig. 4 (c) and (d)** respectively. Therefore in the experiments, the population size is set to 45, set of individuals is evolved to 400 generations, crossover, and mutation rate are set to 0.6 and 0.04 respectively. Furthermore, we set the value of $\propto= 0.25$ to control the influence of success in the evolution process. It ensures the survival of the fittest from the old population with stochastic operators that helps to form new individuals with higher fitness. The results of our experiments are shown in Tables 2, 3, and 4 in $m \times m$ confusion matrices. The ith row, and the jth column represents the number of times an activity a_i, is recognized as activity a_j.
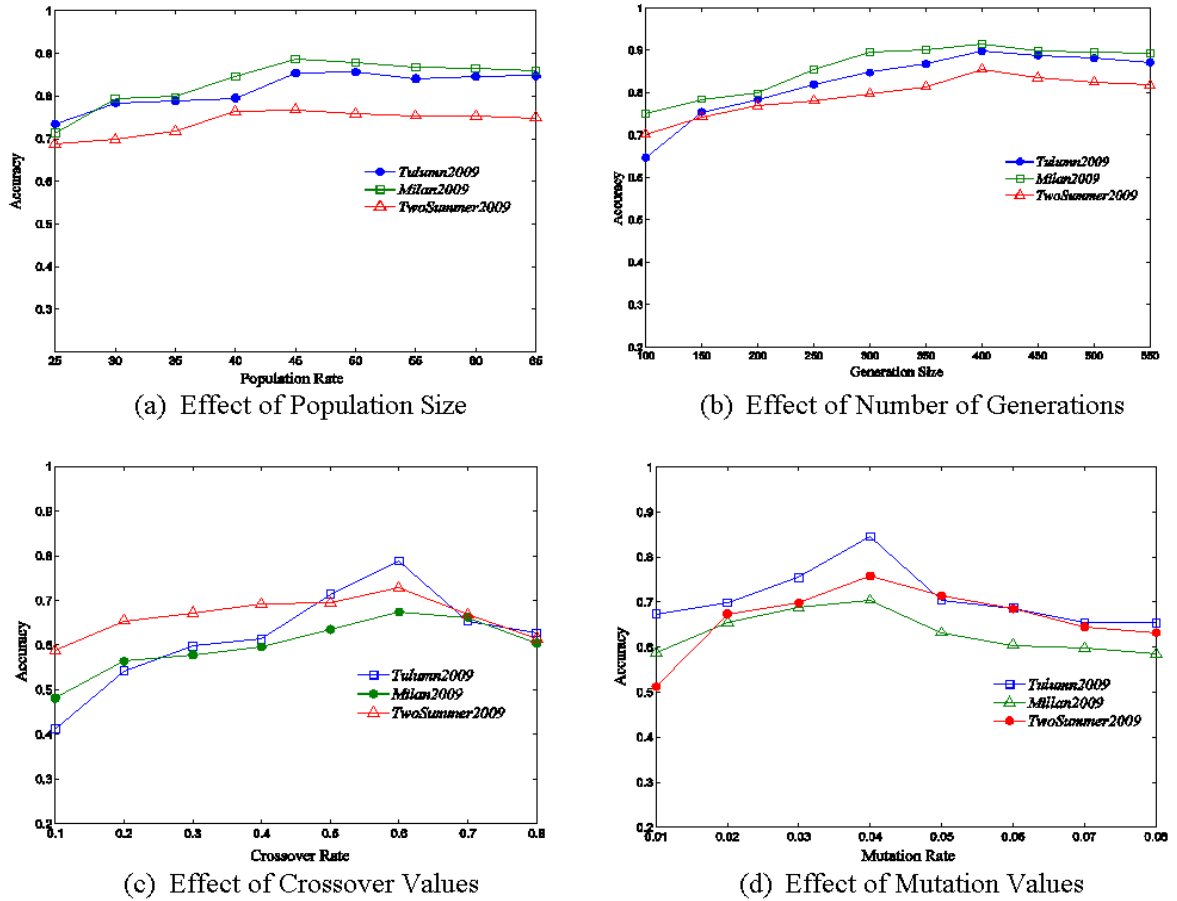
(a) Effect of Population Size

(b) Effect of Number of Generations

(c) Effect of Crossover Values

(d) Effect of Mutation Values

Fig. 4. Optimal Values for GA Stochastic Operations

For *Tulum2009,* the results are presented in **Table 2**, the activities 'Group Meeting', 'Watch TV', and 'Leave Home' are recognized with high accuracy. The most confusion takes place during the 'Cook Breakfast', 'Cook Lunch' and 'R1 Snack' activities. These were recognized correctly most of the times but mixed with five to six other cooking and eating activities.

Table 2. The Confusion Matrix of Recognized Activities in the Tulum2009

Activities	Cook Breakfast	Cook Lunch	Enter Home	Group Meeting	Leave Home	R1 Eat Breakfast	R1 Snack	R2 Eat Breakfast	Wash Dishes	Watch TV
Cook Breakfast	**74**	1	-	2	-	1	-	1	1	-
Cook Lunch	3	**48**	-	1	-	-	10	5	2	2
Enter Home	-	-	**69**	2	2	-	-	-	-	-
Group Meeting	1	-	-	**10**	-	-	-	-	-	-
Leave Home	-	-	3	-	**72**	-	-	-	-	-
R1 Eat Breakfast	2	-	-	-	-	**56**	8	-	-	-
R1 snack	1	1	-	-	-	1	**484**	1	1	2
R2 Eat Breakfast	2	2	-	-	-	-	-	**42**	1	-
Wash Dishes	1	-	-	-	-	-	7	2	**59**	2

Watch TV	-	-	-	-	-	-	3	-	-	**525**

In the case of *Milan2009*, 'Desk Activity', 'Guest Bathroom', 'Sleep' and 'Watch TV' are recognized with the highest accuracy and confused with only one activity, while the confusion rate of 'Read' and 'Leave Home' activities is high compared to other activities. The 'Bed to Toilet' activity is recognized with the lowest accuracy, which is correctly classified 15 times and confused with 3 other activities 74 times, as shown in **Table 3**.

Table 3. The Confusion Matrix of Recognized Activities in the Milan2009

Activities	Bed to Toilet	Chores	Desk Activity	Dining Rm	Eve Meds	Guest Bathroom	Kitchen Activity	Leave Home	Master Bathroom	Master Bedroom	Meditate	Morning Meds	Read	Sleep	Watch TV
Bed to Toilet	**15**	-	-	-	-	6	-	-	60	-	-	-	-	8	-
Chores	-	**15**	4	-	-	-	-	1	-	2	-	-	1	-	-
Desk Activity	-	-	**53**	-	-	-	-	-	-	-	-	-	-	-	1
Dining Rm	-	-	-	**20**	-	-	1	1	-	-	-	-	-	-	-
Eve Meds	-	2	-	-	**8**	-	1	-	-	-	2	6	-	-	-
Guest Bathroom	-	-	-	-	-	**329**	1	-	-	-	-	-	-	-	-
Kitchen Activity	-	-	-	1	-	-	**550**	-	-	-	-	1	2	-	-
Leave Home	-	5	-	-	-	-	5	**185**	-	10	-	-	6	-	3
Master Bathroom	3	-	-	-	-	4	-	-	**290**	6	-	-	-	3	-
Master Bedroom	-	-	-	-	-	-	-	-	1	**115**	-	-	-	1	-
Meditate	-	1	-	1	-	-	-	-	-	-	**14**	-	1	-	-
Morning Meds	-	1	-	1	6	-	2	-	-	-	-	**31**	-	-	-
Read	-	-	10	4	-	-	10	-	-	-	-	-	**290**	-	-
Sleep	-	-	-	-	-	-	-	-	-	1	-	-	-	**95**	-
Watch TV	-	-	1	-	-	-	-	-	-	-	-	-	-	-	**113**

The confusion matrix for *TwoSummer2009* shows that 'Grooming' and 'R1 Shower' are recognized with the highest accuracy as they only confused 4 to 5 times with other activities. While, all other activities are recognized with the acceptable accuracy, confusion among activities does not exceed more than 13 times that is insignificant in comparison to the total occurrences of the activities. The most confusion takes place in the recognition of 'R1 Work' and 'R2 Work' activities as shown in **Table 4**. The above confusion matrices show that our proposed method not only recognize dissociated activities, such as 'Sleeping and 'Watch TV' with significant accuracy but its outperforms all other techniques for the recognition of activities that are highly correlated, such as 'Cook Breakfast', 'Cook Lunch' and 'Snack Activity'. This validates the significance of the proposed method for recognizing the daily life activities.

Table 4. The Confusion Matrix of Recognized Activities in the TwoSummer2009

Activities	Cleaning	Meal Preparation	Grooming	R1 Shower	R1 Sleeping	R1 Wakeup	R1 Work	R2 Shower	R2 Sleeping	R2 Wakeup	R2 Work
Cleaning	**22**	2	-	1	1	-	1	-	-	1	-
Meal Preparation	2	**190**	-	-	-	2	3	-	-	2	-
Grooming	-	-	**43**	2	-	-	2	-	-	-	-
R1 Shower	-	-	-	**36**	1	1	2	-	1	-	-
R1 Sleeping	-	-	-	4	**41**	-	2	2	-	3	-

R1 Wakeup	-	2	-	-	4	**39**	-	-	2	1	-
R1 Work	-	3	6	2	7	-	**405**	3	7	-	10
R2 Shower	-	2	-	3	-	-	1	**20**	-	-	2
R2 Sleeping	-	-	-	3	3	-	-	1	**18**	-	1
R2 Wakeup	-	3	-	-	-	2	1	-	-	**19**	-
R2 Work	-	13	4	1	2	-	11	-	1	-	**320**

We compared our proposed method Classifier Ensemble (CE) with the results of base classifiers (ANN, HMM, CRF, and SVM). Similarly, we performed experiments in comparison to Majority Voting (MV)[35] and Adaboost[26], in order to evaluate our proposed method with other well-known combing methods for activity recognition. MV is a well-known method for combining the decisions of several classifiers in order to arrive at improved recognition results. The MV method goes with the decision when there is a consensus for it or at least more than half of the classifiers agree on it. Adaboost used multiclass to combine features together in order to form the strong classification strategy. The activity features in form of weak classifiers are fed into Adaboost. That's why it is appropriate for state when there is no key feature but the features together form a strong classifier. We keep all the data settings unchanged and report the results in **Figs. 5**, **6**, and **7**. A remarkable improvement in terms of accuracy has been achieved compared to the previous work. As can be seen from **Fig. 5**, our proposed model achieves significant improvement in all recognized activities in *Tulum2009* except 'Wash Dishes', in comparison to ANN and 'R1 Snack' and 'Cook Breakfast' in comparison to CRF. The most noticeable improvements are achieved for the recognition of 'Cook Lunch', 'Enter Home, 'Leave Home' and 'R2 Eat Breakfast' activities.
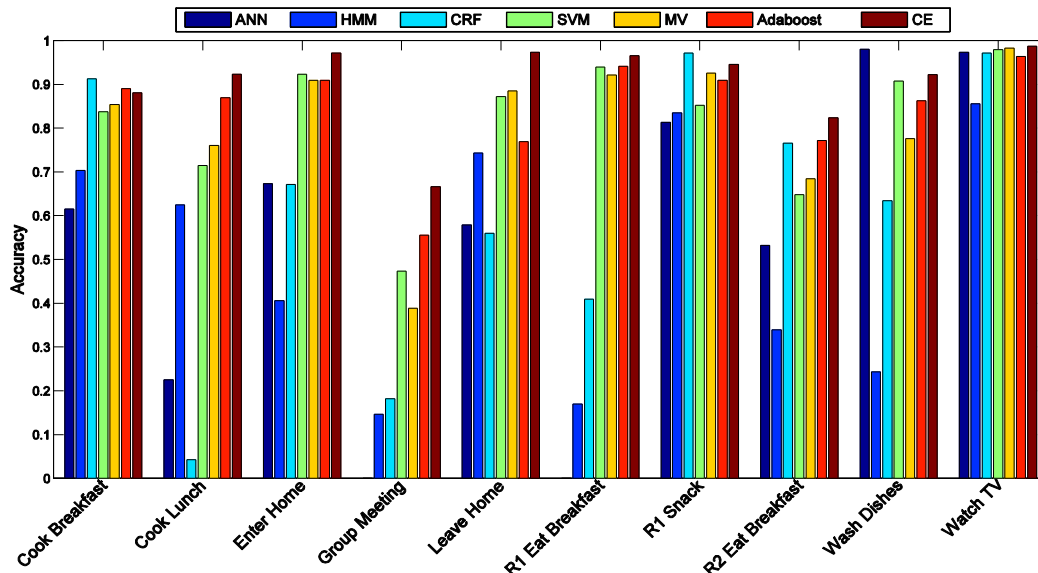

Fig. 5. The Tulum2009 Activity Recognition Results

In the case of *Milan2009*, we achieved significant improvement in all recognized activities against all classification techniques except 'Chores' in comparison of MV and 'Desk Activity' in comparison of CRF. The most noticeable improvements are in case of 'Master Bathroom', 'Master Bedroom' and 'Morning Meds' as shown in **Fig. 6**.

For *TwoSummer2009*, significant improvements in accuracy are achieved except 'R2 Sleeping in Bed', 'Grooming', 'R1 Shower' and 'R2 Wakeup' in comparison of ANN. The most noticeable improvements are in case of 'Cleaning, and 'R2 Shower' as shown in **Fig. 7**. Class level comparison of accuracies shows the variations in performance of all classifiers and validates the better performance of the proposed approach in most of the cases.
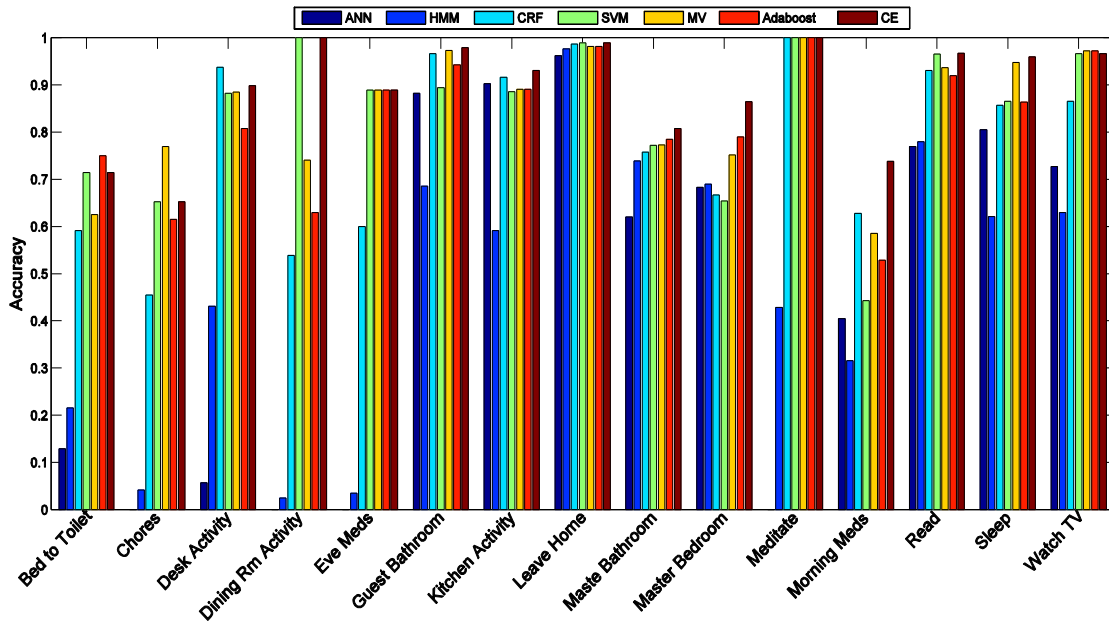

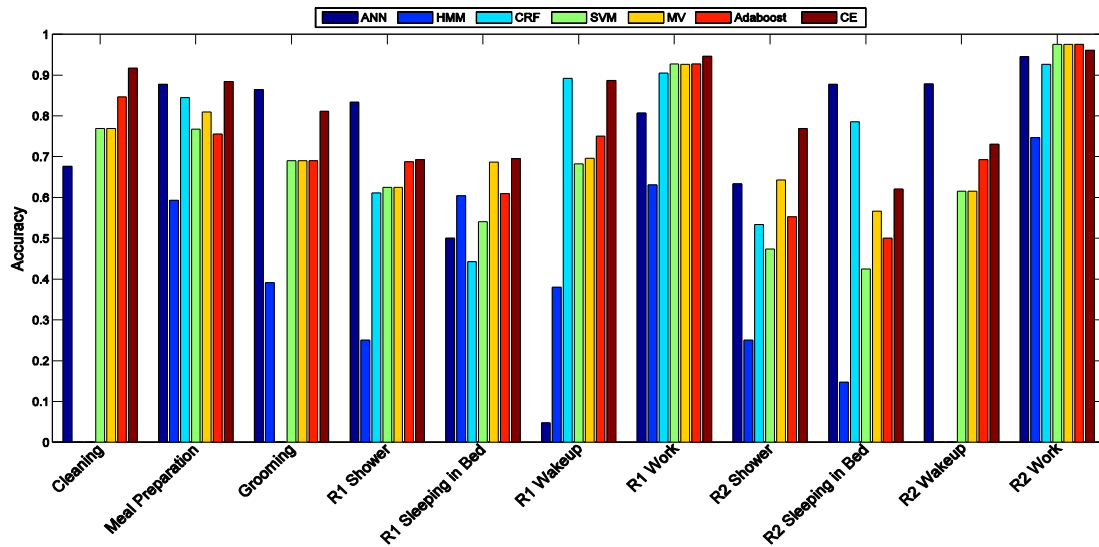**Fig. 6.** The Milan2009 Activity Recognition Results

Fig. 7. The TwoSummer2009 Activity Recognition Results

It can be seen from **Fig. 8**, our proposed model shows stable results in comparison of avg., max., and min. accuracy of all techniques. In case of min. accuracy some of the base classifiers show zero results that means they fail to classify even a single occurrence of at least one activity class label. The maximum accuracy is either high or similar, while the minimum and average accuracy is always high in all classes. The above results and statistics clearly show that dataset characteristics highly affect the classifiers' individual class level assignments and thus their overall performances. For example, in case of *Tulum2009* and *TwoSummer2009* SVM outperforms other base classifiers however, for *Milan2009* CRF outperforms others. Similarly, in combing method MV outperforms Adaboost in case of *Milan2009* and *TwoSummer2009*. However Adaboost is better than MV for *Tulum2009*. Our proposed CE method shows overall better performance in case of all three datasets.
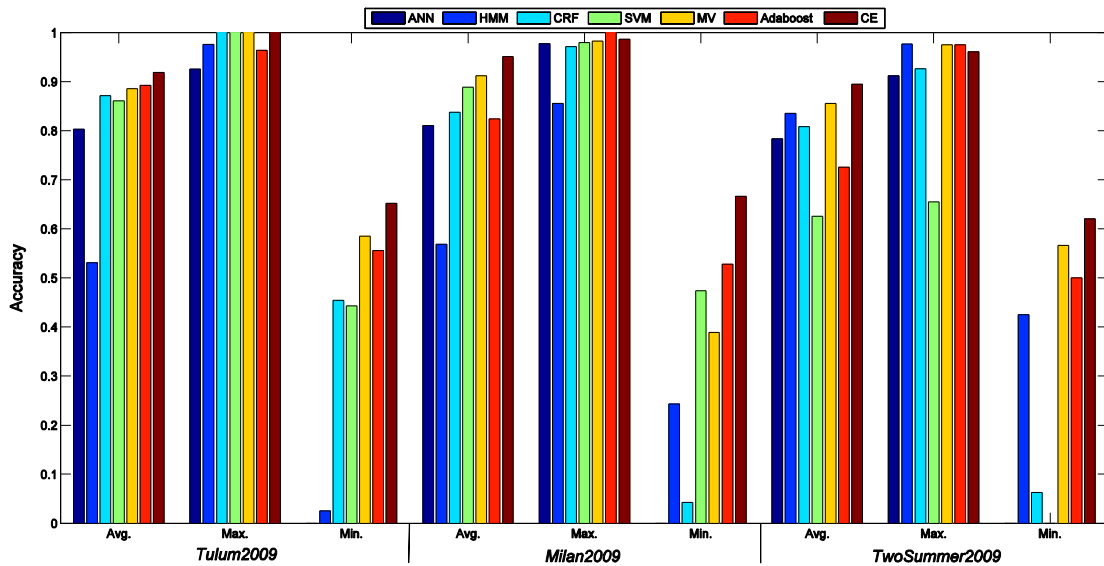

Fig. 8. Accuracy Comparisons (Avg., Max., and Min.)

We computed the precision, recall, F-measure and accuracy, as shown in **Table 5**. For all the datasets, the proposed method performed better, the highest increase of 45.31% in F-measure is achieved in case of base classifier (HMM) for *TwoSummer2009* and 10.29% increase in F-measure is achieved compared to majority voting for *Tulum2009*. Similarly, in comparison to maximum accuracy of base classifier (SVM) we achieved 6.94% more accurate results for *TwoSummer2009* and 4.02% increase in accuracy is achieved in comparison to Adaboost for *Tulum2009*. Furthermore, we performed the statistical significance test and our CE achieves significant improvement (p-value < 0.05) regarding to the classification accuracy.

Table 5. Precision, Recall, F-Measure and Accuracy

Dataset	Model	Precision	Recall	F-Measure	Accuracy
Tulum2009	NN	0.5284	0.5511	0.5395	0.8109
	HMM	0.6161	0.5065	0.5559	0.5684
	CRF	0.7478	0.6119	0.6731	0.8374
	SVM	0.7605	0.8146	0.7866	0.8889
	MV	0.7996	0.7971	0.7984	0.9068

	Adaboost	0.8442	0.7983	0.8206	0.9128
	CE	**0.8968**	**0.9058**	**0.9013**	**0.9510**
Milan2009	NN	0.4934	0.4649	0.4787	0.8029
	HMM	0.4701	0.4801	0.4751	0.5306
	CRF	0.7363	0.7796	0.7574	0.8718
	SVM	0.7855	0.8380	0.8109	0.8610
	MV	0.8006	0.8478	0.8235	0.8858
	Adaboost	0.8242	0.8767	0.8050	0.8759
	CE	**0.8267**	**0.8903**	**0.8573**	**0.9190**
TwoSummer2009	NN	0.73561	0.7261	0.7308	0.8083
	HMM	0.3704	0.3628	0.3666	0.5252
	CRF	0.5451	0.5179	0.5311	0.7832
	SVM	0.73648	0.6809	0.7076	0.8250
	MV	0.75197	0.7274	0.7394	0.8556
	Adaboost	0.7359	0.7174	0.7266	0.8379
	CE	**0.82948**	**0.8102**	**0.8197**	**0.8944**

5. Conclusion and Future Work

In this paper, we proposed a novel technique of classifier ensemble optimization for activity recognition in smart homes. Our main idea is based on the fact that usually classifiers perform complementary to each other for the recognition of activities to be identified. The advantage of the proposed approach is to combine the measurement level decisions of base classifiers by considering their relative competence in the context of assigned weights to each activity class. We have used ANN, HMM, CRF, and SVM as base classifiers for activity recognition. The proposed design of GA, optimized the weighted feature factors for different output classes in each classifier for final activity class label. In the proposed method the weights are encoded in a chromosome as a string of real values. The optimal parameters for in-depth investigation are determined to accelerate the convergence of GA. Hence, a GA based ensemble of multiple classifiers leads to a significant accurate results for recognizing the daily life activities. For evaluation, experiments are performed on three publically available smart home datasets and results show the effectiveness of our proposed approach with the promising results in comparison to the state-of-the-art single classifiers and multi model techniques.

In this study, daily life activities are recognized independently with high accuracy but in reality the activities performed by human users are highly complex and interdependent on each other. An individual can perform a set of activities at the same time in parallel. This limits the applicability of this model at present; however, the generic nature of training and implementation will lead to the success of proposed method for conceivable complex situations. In our future research, we will extend our proposed method to recognize the interleaved, consecutive and parallel activities with comparable accurate results.

References

[1]   R.C. Parisa, J.H. Diane, B. Lawrence and S. Maureen "Discovering Activities to Recognize Track in a Smart Environment". *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp.527-539, 2011. Article (CrossRef Link)

[2]   D. Cook, CASAS Smart Home project. [Online]. http://www.ailab.wsu.edu/casas/ (accessed Sept. 10, 2012).

[3]   B. Jones Aware Home. [Online]. http://awarehome.imtc.gatech.edu/ (accessed Sept. 10, 2012).

[4]   M.C. Mozer, The Adaptive House. [Online].
      http://www.cs.colorado.edu/~mozer/index.php?dir=/Research/Projects/Adaptive (accessed Sept.
      10, 2012).
[5]   K. Larson House_n. [Online]. http://architecture.mit.edu/house_n/ (accessed Sept. 10, 2012).
[6]   T. Kasteren, A .Noulas, G. Englebienn and K. Ben "Accurate Activity Recognition in a Home
      Setting," in *Proc. of the Tenth International Conference on Ubiquitous Computing*, Korea, pp. 1-9,
      2008. Article (CrossRef Link)
[7]   T.L.M. Kasteren, G. Englebienne and B.J.A. Krose "An Activity Monitoring System for Elderly
      Care using Generative and Discriminative Models", *Personal and Ubiquitous Computing*, vol. 14,
      no. 6, pp. 489-498. 2010. Article (CrossRef Link)
[8] B.J.A. Krose, T.L.M. Kasteren, C.H.S. Gibson and T. Dool, "CARE: Context Awareness in
      Residences for Elderly," in *Proc. of 6th International Conference of the International Society for
      Gerontechnology*, pp. 101-105, 2008.   Article (CrossRef Link)
[9]   S. Helal, E. Kim and S. Hossain. "Scalable Approaches to Activity Recognition Research," in
      *Proc. of the workshop of How to do good activity recognition research? Experimental
      methodologies, evaluation metrics, and reproducibility issues*, pp. 450-453, 2010.
      Article (CrossRef Link)
[10] A. Fleury, M. Vacher and N. Noury "SVM-Based Multimodal Classification of Activities of Daily
      Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results". *IEEE
      Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp.274-83, 2010.
      Article (CrossRef Link)
[11] J.K. Aggarwal and M.S. Ryoo, "Human activity analysis: A review", *ACM Comput. Survey*, vol.
      43, no. 3, pp. 1–43, 2011. Article (CrossRef Link)
[12] A. Ekbal and S. Saha, "Weighted Vote-Based Classifier Ensemble for Named Entity Recognition:
      A Genetic Algorithm-Based Approach". *ACM Trans. Asian Lang. Inf. Process*. vol. 10, no. 2, pp.
      1-37., 2011. Article (CrossRef Link)
[13] M. Behrouz, K. Gerd and F. William, "Optimizing classification ensembles via a genetic
      algorithm for a web-based educational system," in *Proc. of the International Workshop on
      Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, pp. 397-406,
      2004. Article (CrossRef Link)
[14] M. Kim, S. Min and I. Han. "An evolutionary approach to the combination of multiple classifiers
      to predict a stock price index". *Expert Systems with Applications*. vol. 31, no. 2 pp. 241-247 , 2006.
      Article (CrossRef Link)
[15] M. Bahrepour, N. Meratnia, Z. Taghikhaki and P.J.M. Havinga. "Sensor Fusion-based Activity
      Recognition for Parkinson Patients". *Sensor Fusion - Foundation and Applications. InTech*, pp.
      171-190. ISBN: 978-953-307-446-7 Article (CrossRef Link)
[16]   A.B. Jon , K. Josef and R. Fabio, "Multiple Classifier Systems", *Lecture Notes in Computer
      Science*, 5519, Iceland, Springer, 540, 2009 Article (CrossRef Link)
[17] L.I. Kuncheva and L.C. Jain. "Designing classifier fusion systems by genetic algorithms". *IEEE
      Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 327-336, 2000.
      Article (CrossRef Link)
[18]   L. Bull, M. Studley, A. Bagnall and I. Whittley, "Learning Classifier System Ensembles with
      Rule-Sharing". *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 496-502,
      2007. Article (CrossRef Link)
[19] X. Rongwu and H. Lin. "GACEM: Genetic Algorithm Based Classifier Ensemble in a
      Multi-sensor System", *Sensors* ,vol. 8, no.10, pp. 6203-622, 2008. Article (CrossRef Link)
[20] K. Kyung-Joong and C. Sung-Bae, "An Evolutionary Algorithm Approach to Optimal Ensemble
      Classifiers for DNA Microarray Data Analysis". *IEEE Transactions on Evolutionary
   Computation*, vol. 12, no. 3, pp. 377-388, 2008. Article (CrossRef Link)

[21] M. A. Seenay and I. Sminu. "A Survey: Evaluation of Ensemble Classifiers and Data Level Methods to Deal with Imbalanced Data Problem in Protein Protein Interactions", *Review of Bioinformatics and Biometrics (RBB)*, vol 2 no. 1, pp. 2013. Article (CrossRef Link)

[22] L. Chen, C.D. Nugent and H. Wang. "A Knowledge-Driven Approach to Activity Recognition in Smart Homes". *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961-974, 2011. Article (CrossRef Link)

[23] L. Chen and C.D. Nugent, "Ontology-based activity recognition in intelligent pervasive environments:. *Int. J. Web Inf. Syst*., vol. 5, no. 4, pp. 410–430, 2009. Article (CrossRef Link)

[24] G. Okeyo, L. Chen, and H. Wang, "Ontology-based learning framework for activity assistance in an adaptive smart home, in Activity Recognition in Pervasive Intelligent Environments", *Atlantis Ambient and Pervasive Intelligenc*e, vol 4, pp. 237–262, 2011. Article (CrossRef Link)

[25] C. Liming, H. Jesse, D.N. Chris, J.C. Diane and Y. Zhiwen, "Sensor-Based Activity Recognition", *IEEE Transactions on Systems, Man, and Cybernetics*,vol. 99, pp. 1-19, 2012. Article (CrossRef Link)

[26] M. H. Kamrul Hasan and L. Sungyoung, L. Young-Koo Lee, "Using Sensor Sequences for Activity Recognition by Mining and Multi-Class Adaboost", *The 2010 International Conference on Artificial Intelligence*, Las Vegas, USA, pp. 735-740., 2010. Article (CrossRef Link)

[27] T. Mitchell, Machine Learning. McGraw Hill, Columbus, 1997.

[28] Neural Network Toolbox [Online]. http://www.mathworks.co.kr/products/neural-network/index.html (accessed Sept. 10, 2012).

[29] M. Piccardi, "Hidden Markov Models with Kernel Density Estimation of Emission Probabilities and their Use in Activity Recognition". *In proceeding of IEEE conference of Computer Vision and Pattern Recognition*, pp. 1 – 8, 2007. Article (CrossRef Link)

[30] crfChain [Online]. http://www.di.ens.fr/~mschmidt/Software/crfChain.html (accessed Sept. 10, 2012).

[31] UGM [Online]. http://www.di.ens.fr/~mschmidt/Software/UGM.html (accessed Sept. 10, 2012).

[32] LIBSVM [Online]. http://www.csie.ntu.edu.tw/~cjlin/libsvm/(accessed Sept. 10, 2012).

[33] S. Guan and F. Zhu, "An incremental approach to genetic-algorithms-based classification:. *IEEE Transactions on Systems, Man, and Cybernetics,* vol.35, no 2, pp. 227-239, 2005. Article (CrossRef Link)

[34] T.L.M Kasteren, H. Alemdar and C. Ersoy, "Effective Performance Metrics for Evaluating activity Recognition Methods". *In proceedings of 2nd Workshop on Context-Systems Design", Evaluation and Optimization*. Italy , (2011). Article (CrossRef Link)

[35] D. Ruta and B. Gabrys, "Classifier Selection for Majority Voting". *Information fusion*, vol. 6, no.1, pp. 63–81, 2005. Article (CrossRef Link)

Iram Fatima received her BS from the Department of Computer Science, Government College University (GC University) in 2004. She received his MS in Information Technology from the School of Electrical Engineering and Computer Science, National University of Sciences and Technology in 2009. Since March 2010, she has been working on his PhD in the Department of Computer Engineering at Kyung Hee University, Korea. Her research interests include healthcare, social networks, clinical decision support system and activity recognition.
Email: iram.fatima@oslab.khu.ac.kr

Muhammad Fahim received his B.S. with distinction from Gomal University, Pakistan in 2007. He got M.S. from National University of Computer and Emerging Sciences (NUCES), Pakistan in 2009. Since March 2010, he has been working on his PhD degree at the Department of Computer Engineering at Kyung Hee University, Korea. His research interests include activity recognition in smart homes and smartphone, digital signal processing, Machine Learning and artificial intelligence.
E-mail:fahim@oslab.khu.ac.kr
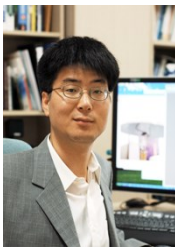
Young-Koo Lee received his B.S.,M.S., and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Korea in 1988, 1994 and 2002, respectively. Since 2004, he has been an assistant professor at the Dept. of Computer Engineering, College of Electronics and Information, Kyung Hee University, Korea. From 2002 to 2004, he was a Post Doctoral Fellow Advanced Information Technology Research Center (AITrc), KAIST, Korea, and a Postdoctoral Research Associate at Dept. of Computer Science, University of Illinois at UrbanaChampaign, USA. His research interests are Ubiquitous Data Management, Data Mining, Activity Recognition, Bioinformatics, On-line Analytical Processing, Data Warehousing, Database Systems, Spatial Databases, and Access Methods.
E-mail:yklee@oslab.khu.ac.kr

Sungyoung Lee received his Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1991. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 1993. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing, Cloud Computing, Intelligent Computing, and eHealth. E-mail:sylee@oslab.khu.ac.kr