Pattern Recognition 56 (2016) 63–76

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Incremental granular relevance vector machine: A case study in multimodal biometrics

Hunny Mehrotra [a], Richa Singh [b], Mayank Vatsa [b,*], Banshidhar Majhi [a]

[a] National Institute of Technology Rourkela, India
[b] Indraprastha Institute of Information Technology (IIIT), Delhi, India

ARTICLE INFO

Article history:
Received 24 September 2013
Received in revised form
3 September 2015
Accepted 15 November 2015
Available online 15 December 2015

Keywords:
Incremental learning
Granular computing
Relevance vector machine
Biometrics

ABSTRACT

This paper focuses on extending the capabilities of relevance vector machine which is a probabilistic, sparse, and linearly parameterized classifier. It has been shown that both relevance vector machine and support vector machine have similar generalization performance but RVM requires significantly fewer relevance vectors. However, RVM has certain limitations which limits its applications in several pattern recognition problems including biometrics such as (1) slow training process, (2) difficult to train with large training samples, and (3) may not be suitable to handle large class imbalance. To address these limitations, we propose *i*GRVM which incorporates incremental and granular learning in RVM. The proposed classifier is evaluated in context to multimodal biometrics score classification using the NIST BSSR1, CASIA-Iris-Distance V4, and Biosecure DS2 databases. The experimental analysis illustrates that the proposed classifier can be a good alternative for biometric score classification with faster testing time.

1. Introduction

Classifiers are an integral component of a pattern classification system. In order to determine the class of any query, the data is processed, a representation is computed, and the classifier classifies it into one of the classes. Before testing, the classifier learns a model using the given training data. For instance, in a biometric verification problem, there are two classes, *genuine* and *imposter*. The task is to match the probe image with the corresponding gallery image and determine whether the probe is a genuine match or imposter. Existing biometric recognition algorithms have used different classifiers such as linear threshold, Bayesian classification, and Support Vector Machine (SVM) [1].

For training an accurate classification model, it is generally assumed that sufficient and representative training data is available during the training stage. However, in real world applications, there are several challenges in ensuring the availability of good quality training data:

- There exists the possibility that the entire training data is not available simultaneously. For example, in the case of India's Aadhaar project [2] or US-VISIT program [3], users are enrolled on a continuous basis. In such a scenario, training data is available only in an incremental manner. Training the classifiers in batch mode with every incremental update can be computationally expensive.
- Training databases can be highly unbalanced where data from one class is over populated compared to other class(es). In a biometric system that has n users in the database each having m samples ($n \gg m$), the number of genuine scores available for training is $nm(m-1)/2$ in comparison to $n(n-1)m^2/2$ impostor scores.
- Some classifiers are inherently computationally expensive, they perform well if the training size is small but on large training data they may require significant computational time or become intractable.

To address some of these challenges, researchers have proposed multiple solutions. The availability of sequential training data is addressed by incremental learning and online learning algorithms [4]. In incremental learning, classifiers are trained with new batches of data, as they arrive, while preserving the knowledge of previous learning. Some incremental learning approaches are incremental Principal Component Analysis (IPCA) [5], incremental learning of Bidirectional Principal Component Analysis [6], incremental Linear Discriminant Analysis (ILDA) [7], incremental Subclass Discriminant Analysis (ISDA) [8], and incremental and decremental SVM [9,10].

[*] Corresponding author. Tel.: +91 9654653404; fax: +91 11 26907405.
E-mail addresses: hunny@nitrkl.ac.in (H. Mehrotra), rsingh@iiitd.ac.in (R. Singh), mayank@iiitd.ac.in (M. Vatsa), bmajhi@nitrkl.ac.in (B. Majhi).

http://dx.doi.org/10.1016/j.patcog.2015.11.013
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

In the literature, several researchers have also explored the challenge of class imbalance [11,12]. Chawla et al. [13] have stated that class imbalance problem is handled either by assigning distinct cost to training data [14–16] or by resampling the entire database [17]. The resampling approaches work by either oversampling the minority class and under-sampling the majority class, or by combining the under-sampling and oversampling approaches [18,19]. To balance class distributions, random under-sampling may lead to information loss whereas random oversampling can increase the chances of overfitting. Tang et al. [20] have proposed an under-sampling approach using granular learning. Granular learning divides the data into granules represented as either classes, clusters or subsets and solves the problem in each information granule locally [21]. The challenge of large training database for learning computationally expensive classifiers has also been addressed by granular computing approaches [22].

Since the formulation of every classifier is different, the extension of an existing classifier that operates in batch mode to the corresponding incremental version is also different. In designing the incremental or granular variant of an existing classifier, it is important to ensure that the updated variants do not reduce the accuracy while reducing the training time or computational complexity. Therefore, researchers have proposed specific formulations for individual classifiers, such as SVM.

SVM has been shown to yield good results in several pattern classification problems including biometrics. It avoids overfitting and leads to good generalization by finding the separating hyperplane that maximizes the margin width. The subset of training data points used to represent the hyperplane are denoted as support vectors. Several formulations have been proposed for online training of SVM and addressing the class imbalance problem [10,20,22,23]. However, SVM suffers from the following limitations [24]:

1. The number of support vectors required for classification is relatively large,
2. In classical SVM, there is a need to fine tune the regularization parameter (C) during the training phase, and
3. The kernel function must satisfy the Mercer conditions [25].

Relevance vector machine (RVM) [24], on the other hand, is a probabilistic classifier which introduces a prior over each weight governed by the set of hyper-parameters. RVM is a sparse linearly parameterized model like SVM and it has been shown that the generalization performance of RVM is comparable to that of SVM with significantly fewer relevance vectors [24]. Another advantage of RVM is that it has very few parameters to be optimized while training. Along with these advantages, RVM has the following challenges owing to which it has not been well explored particularly in biometrics.

1. The native formulation of RVM requires expensive matrix inversion which makes it difficult to learn conventional RVM with very large training databases. Further, the amount of memory required to store the product of basis functions also limits its utilization for considerably large training databases.
2. RVM is trained in batch mode and if new batch of data arrives, the classifier has to be re-trained with new as well as old data. This is not feasible for many real-time applications such as biometrics where it may be required to continuously update the classifier to adjust the changes (in data and template) that happen over time.
3. RVM may not be suitable to handle large class imbalance in the training data and may get biased towards the class with more number of training samples.

To address these challenges, in this paper, we propose an incremental granular RVM that can be trained with large unbalanced training data to perform efficient classification. As shown in Fig. 1, the learning process starts by considering batches of training data which are divided into granules. An RVM is trained on each granule independently and the results are amalgamated to obtain a robust boundary for classification. For online learning, the knowledge from the previous training is carried forward to learn the next batch of training database. The major contributions of this research are:

1. *Incremental RVM* (*i*RVM) is proposed which is scalable with new enrollments and also reduces the training time.
2. *Granular RVM* (GRVM) handles the class imbalance problem by training the classifier locally for each granule.
3. *Incremental Granular RVM* (*i*GRVM) combines the advantages of both incremental and granular learning into RVM.

The proposed variant provides a good alternative to existing classifiers and overcomes the limitations of native RVM classifier. The performance of incremental granular RVM is evaluated using a case study in multimodal biometrics with two classes (genuine and imposter). The match scores obtained from different modalities, units and algorithms are normalized followed by incremental granular RVM classification. Experiments performed on three match score databases show that the proposed classifier is comparable to existing approaches in terms of classification performance and provides significant reduction in computational time.

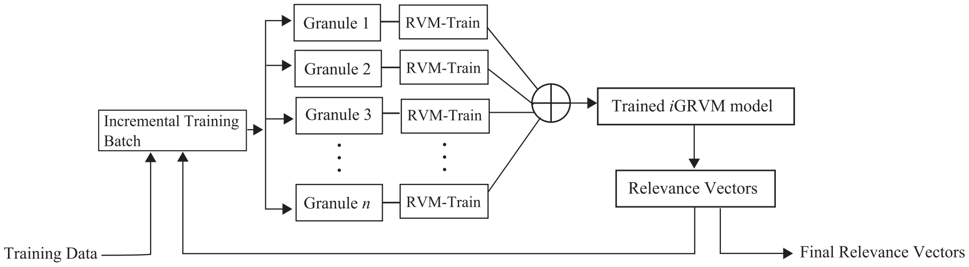

Fig. 1. Block diagram of incremental granular relevance vector machine (*i*GRVM).

2. Incremental granular relevance vector machine

This section describes the formulation of *i*GRVM for data classification. The proposed classifier is designed to incrementally update the learnt model and decision boundary for new batches of training data. Training RVM using data divided into granules may further boost the performance. Our hypothesis is that this unique combination of granulation and incremental learning when applied to RVM can improve the performance. The proposed variant of RVM is more focussed towards developing an adaptive and unbiased learning framework.

2.1. Relevance vector machine for classification

RVM is a probabilistic model which is similar to SVM in its functional form. In RVM, learning is performed using Bayesian approach where weights are iteratively estimated from the training data and governed by a set of hyper-parameters [24,26]. Unlike SVM, training vectors corresponding to non-zero weights, also known as relevance vectors, are not necessarily close to decision boundary but are representative examples which help in classification. As discussed earlier, the generalization performance of RVM is comparable to SVM with relatively fewer relevance vectors. In this section, we briefly discuss the formulation of RVM [24]. Let $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$ be the pair of input labeled data, with $\mathbf{x}_n$ being the data point and t_n being the scalar valued target label. RVM by design follows a Bayesian probabilistic model for learning and the predictions are based on the function

$$y = \mathbf{w}^T \Phi \tag{1}$$

where $\mathbf{w} = (w_0, w_1, w_2, \ldots, w_N)$ is the weight matrix, $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)]$ is the set of basis functions, $\phi(\mathbf{x}_n) = [1, K(\mathbf{x}_n, \mathbf{x}_1), K(\mathbf{x}_n, \mathbf{x}_2), \ldots, K(\mathbf{x}_n, \mathbf{x}_N)]$, and the output of RVM (y) is a linear combination of weighted basis functions. The weights ($\mathbf{w}$) are computed during training and the training samples corresponding to non-zero weights are called *relevance vectors* (RVs).

The objective function of RVM is to predict the posterior probability of class membership for the given input $\mathbf{x}$. Since, in this research, RVM is applied for two-class classification, it is desirable to model the posterior probability of input data using Bernoulli distribution. Further, the linear model in equation (1) is generalized by applying the logistic sigmoid function $\sigma(y) = 1/(1+e^{-y})$ and adopting the Bernoulli distribution to define the likelihood as,

$$P(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} \sigma\{y(\mathbf{x}_n, \mathbf{w})\}^{t_n}[1-\sigma\{y(\mathbf{x}_n, \mathbf{w})\}]^{1-t_n} \tag{2}$$

where $\mathbf{t} = (t_1, t_2, \ldots, t_N)$. To obtain the marginal likelihood analytically, Mackay's iterative procedure [27] is used which is based on the Laplace's method. Let α be the vector of hyper-parameters and each individual α value is associated with every weight value. For the fixed values of α, most probable weights $\mathbf{w}$ are found, giving location of the mode of posterior distribution [24]. Since

$$p(\mathbf{w}|\mathbf{t}, \alpha) \propto P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha), \tag{3}$$

this is equivalent to finding the maximum of

$$\log\{P(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)\} = \sum_{n=1}^{N}[t_n \log y_n + (1-t_n)\log(1-y_n)] - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} \tag{4}$$

over $\mathbf{w}$, where $y_n = \sigma\{y(\mathbf{x}_n, \mathbf{w})\}$. Equation (4) is differentiated twice to obtain

$$\nabla_{\mathbf{w}}\nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{t}, \alpha)|_{\mathbf{w}} = -(\Phi^T\mathbf{B}\Phi + \mathbf{A}) \tag{5}$$

where $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$ and $\mathbf{B} = \mathrm{diag}(\beta_1, \beta_2, \ldots, \beta_N)$ with $\beta_n = \sigma\{y(\mathbf{x}_n)\}[1-\sigma\{y(\mathbf{x}_n)\}]$. The covariance matrix Σ and the posterior over weights centered at ($\mathbf{w_{MP}}$) are defined as,

$$\Sigma = (\Phi^T\mathbf{B}\Phi + \mathbf{A})^{-1} \tag{6}$$

$$\mathbf{w_{MP}} = \Sigma\Phi^T\mathbf{B}\mathbf{t} \tag{7}$$

Using Σ and $\mathbf{w}$, the hyper-parameters are updated as,

$$\alpha_i = \frac{\gamma_i}{\mathbf{w}_i^2} \tag{8}$$

where $\mathbf{w}_i$ is the i^{th} posterior weight computed using equation (7), $\gamma_i \equiv 1 - \alpha_i\Sigma_{ii}$ and Σ_{ii} is the i^{th} diagonal element of Σ. The convergence criteria for the above mentioned iterative procedure is defined as

$$\delta = \sum\nolimits_{i=1} \alpha_i^{n+1} - \alpha_i^n \tag{10}$$

Re-estimation stops when $\delta < \delta_\tau$, where δ_τ is the threshold value for change of α between iterations.

The training samples corresponding to $\mathbf{w} \neq 0$ are termed as relevance vectors ($\mathbf{R}$). The weights and relevance vectors obtained from Algorithm 1 are used to find an estimate of the target value pertaining to the new input x'.

$$y' = \mathbf{w}^T\phi(x') \tag{11}$$

where y' is the probabilistic class value predicted by the trained model. This classification algorithm is provided in Algorithm 2.

2.2. Incremental relevance vector machine (iRVM)

Traditionally, RVM is trained in offline (batch) mode, where the entire training data is used to learn the classifier. Since training RVM requires matrix inverse operation (Eq. (6)) that performs $O(N^3)$ computations [24], where N is the size of training data, RVM is not suitable for learning if the size of N is very large. Tipping proposed to reduce the training time of traditional RVM using fast marginal likelihood maximization [28]. This learning approach helps to achieve speed-up by considering an empty model and sequentially adding basis vectors to increase marginal likelihood. Tzikas et al. proposed a modification to fast marginal likelihood maximization approach which also learns the location and scale parameters of the kernel during training [29]. These approaches [28,29] focus on dynamically adding or deleting the relevance vectors from the complete batch of training data. However, in several real-time applications, the requirement is not only to train with large data but also to update the learnt model as training data arrives in batches. For instance, very large scale biometric applications such as UIDAI and US VISIT programs continuously enroll new subjects and new training data is obtained incrementally. Therefore, it is important to introduce incremental learning strategies in the classifier [4]. In this research, we extend the original formulation of RVM to design incremental RVM (iRVM) which is scalable to new enrollments. This facilitates updating the classifier with the availability of new enrollment data. Incremental learning also helps to keep memory and time computations (Hessian inverse) simple.

Algorithm 1. RVM-Train.

```
Input: $\mathbf{x}$: Input matrix of $N$ scores with dimension $d$, $\mathbf{t}$: corresponding target values
Output: $\mathbf{R}$: model of relevance vectors, $\mathbf{w}$: weight matrix
Generate $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)]$
Initialize $\delta_\tau$    // Threshold for convergence
Initialize $\alpha, \beta$    // Initialization of hyper-parameters
repeat
    $\mathbf{A} = \text{diag}(\alpha),\ \mathbf{B} = \text{diag}(\beta)$;
    $\Sigma = (\Phi^T \mathbf{B} \Phi + \mathbf{A})^{-1}$;
    $\mathbf{w} = \Sigma \Phi^T \mathbf{Bt}$;
    $\gamma_i \equiv 1 - \alpha_i \Sigma_{ii}$;
    $\alpha_i = \frac{\gamma_i}{\mathbf{w}_i^2}$
    $\delta = \sum_{i=1} \alpha_i^{n+1} - \alpha_i^n$;
untill convergence
$\mathbf{R} = \mathbf{x}(\mathbf{w}_{\text{index}})$    // Training samples corresponding to non-zero weights
```

Algorithm 2. RVM-Classify.

```
Input: $x'$: Test data for classification, $\mathbf{R}$: Relevance vectors, $\mathbf{w}$: weight matrix
Output: $y'$: Predicted class membership
Generate $\phi(x')$ using $\mathbf{R}$
$y' = \mathbf{w}^T \phi(x')$
```

The proposed *i*RVM considers only relevance vectors from the previous training batch with the new batch of data for re-training. This approach is based on the assumption that only RVs are essential for making predictions and all other training samples pertaining to the previous batches can be safely removed. This concept is similar to Tipping's constructive training approach [24] which reduces the total number of basis functions and the computational complexity. As shown in Fig. 2, the steps involved in *i*RVM learning are:

- Initial training data ($\mathbf{T}_1$) is provided as input to the RVM and the relevance vectors ($\mathbf{R}_1$) are obtained.
- RVM is re-trained using the new batch of training data ($\mathbf{T}_i$) along with the relevance vectors from the previous iteration ($\mathbf{R}_{i-1}$). RVM-Train ($\mathbf{T}_i \oplus \mathbf{R}_{i-1}$) computes the new relevance vectors $\mathbf{R}_i$ along with the weights ($\mathbf{w}_i$) which represents the updated relevance vector for the aggregated training data.
- The previous step is repeated on encountering new training samples in $\mathbf{T}_i$ for $i = 2, 3, 4, \ldots, m$, where m is the number of incremental training batches.

The learning strategy of *i*RVM is summarized in Algorithm 3. The relevance vectors from the last iteration ($\mathbf{R}_m$) are finally considered for making predictions until the new batch of training data arrives. The formulation of *i*RVM is given by

$$y = \mathbf{w}^T \Phi(\mathbf{T}_m \oplus \mathbf{R}_{m-1}) \tag{12}$$

where $\oplus$ denotes the concatenation operation. The classification algorithm for *i*RVM is same as RVM-Classify (Algorithm 2). It is interesting to note that on encountering every new batch of data, the objective of the proposed *i*RVM is to adapt to new enrollments via incremental learning.

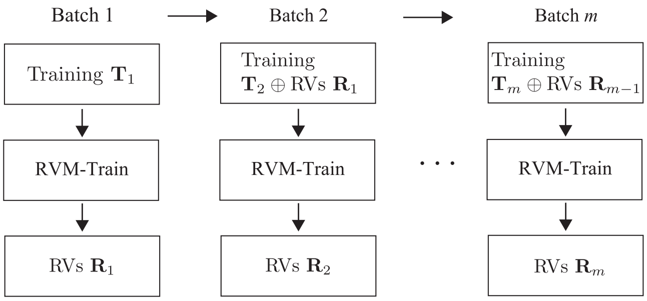

Fig. 2. Illustrating the steps involved in training the proposed incremental relevance vector machine. The relevance vectors from the previous/last iteration ($\mathbf{R}_{i-1}$) are used to perform training with new batch of data ($\mathbf{T}_i$).

Algorithm 3. *i*RVM-Train.

```
Input: $\mathbf{T}_m$: Training data, $\mathbf{t}_m$: equivalent target values.
Output: $\mathbf{R}_m$: relevance vectors, $\mathbf{w}_m$: most probable weight matrix
foreach Batch m of training do
    $\mathbf{T}_m \leftarrow \mathbf{T}_m \oplus \mathbf{R}_{m-1}$   // combine training data with RVs from last iteration
    $[\mathbf{R}_m, \mathbf{w}_m] = \text{RVM} - \text{Train}(\mathbf{T}_m, \mathbf{t}_m)$
end
```

2.3. Granular relevance vector machine

In several applications, training is performed with highly imbalanced data. For instance, in biometrics applications, positive/genuine class contains lesser samples and negative/impostor class contains larger number of samples. Training the classifier with such data leads to a biased model which classifies in favor of the majority class. This imbalance problem can be handled by considering various strategies such as oversampling the genuine scores (minority class) and under-sampling the imposter scores (majority class). However, random data sampling may lead to information loss, for instance some informative samples may be lost. An alternative approach is to split the training data into smaller segments and train the classifier on each segment locally; i.e., in granulation manner. Granular computing [22] solves the problem in each data granule locally while not compromising the global generalization.

In general, there are two approaches to granular computing: the first approach performs granule split using divide and conquer strategy whereas the second approach is based on granule shrink. Granule split approach divides a large problem into a sequence of smaller problems, solves them individually and then combines the results. Granule shrink removes redundant and unnecessary information to reduce the data size so that it can be solved more efficiently. Inspired from these two approaches, this research proposes two variants of granular RVM using the *split* and *shrink* approaches.

2.3.1. Granular RVM using repetitive under-sampling

As previously stated, only relevance/support vectors are required for classification. This motivates us to use only RVs to sample the data without loss of information. The process of learning granular RVM based on repetitive under-sampling (henceforth named as GRVM-*ru*) is shown in Fig. 3. The steps involved in this under-sampling approach are described below.

- Since the genuine class has lesser number of training samples, all the samples of genuine class ($\mathbf{T}_1^+$) from the initial training data ($\mathbf{T}_1$), are considered informative and they form the positive information granule.
- Since the number of samples in the imposter class is very large, only the relevance vectors from the imposter class ($\mathbf{R}_1^-$) that are obtained by training RVM with ($\mathbf{T}_1$) form the first negative information granule.
- To generate the next negative granule, the relevance vectors from imposter class ($\mathbf{R}_1^-$) selected in the first iteration are removed from the initial training data, ($\mathbf{T}_1$). The training data after this iteration reduces to $\mathbf{T}_2 = \mathbf{T}_1 \ominus \mathbf{R}_1^-$ by *granule shrink*, where $\ominus$ denotes removal of elements from the set. The second negative granule thus consists of relevance vectors obtained after training RVM on $\mathbf{T}_2$. These relevance vectors ($\mathbf{R}_2^-$) are the ones which were left undetected at the very first training of RVM but considered important. The process of granulation terminates on meeting the convergence criteria.
- Negative granules are combined with the positive granule using the aggregation operation ($\mathbf{T}_1^+ \oplus \mathbf{R}_1^- \oplus \mathbf{R}_2^- \oplus \ldots \oplus \mathbf{R}_n^-$). RVM is trained on this aggregated set, and if the classification performance improves, the next negative granule is extracted otherwise the under-sampling process stops.

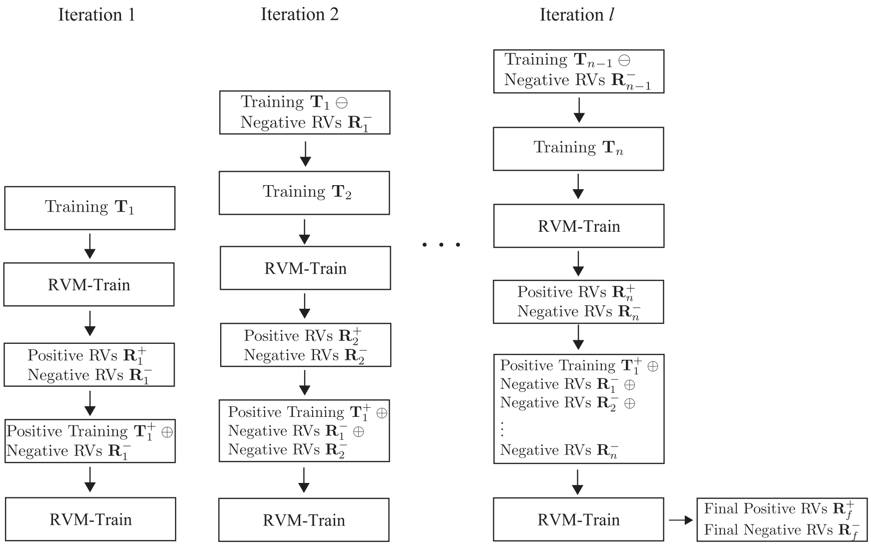
Iteration 1
Iteration 2
Iteration l
Training $\mathbf{T}_1$
RVM-Train
Positive RVs $\mathbf{R}_1^+$ Negative RVs $\mathbf{R}_1^-$
Positive Training $\mathbf{T}_1^+ \oplus$ Negative RVs $\mathbf{R}_1^-$
RVM-Train
Training $\mathbf{T}_1 \ominus$ Negative RVs $\mathbf{R}_1^-$
Training $\mathbf{T}_2$
RVM-Train
Positive RVs $\mathbf{R}_2^+$ Negative RVs $\mathbf{R}_2^-$
Positive Training $\mathbf{T}_1^+ \oplus$ Negative RVs $\mathbf{R}_1^- \oplus$ Negative RVs $\mathbf{R}_2^-$
RVM-Train
. . .
Training $\mathbf{T}_{n-1} \ominus$ Negative RVs $\mathbf{R}_{n-1}^-$
Training $\mathbf{T}_n$
RVM-Train
Positive RVs $\mathbf{R}_n^+$ Negative RVs $\mathbf{R}_n^-$
Positive Training $\mathbf{T}_1^+ \oplus$ Negative RVs $\mathbf{R}_1^- \oplus$ Negative RVs $\mathbf{R}_2^- \oplus$ ⋮ Negative RVs $\mathbf{R}_n^-$
RVM-Train
Final Positive RVs $\mathbf{R}_f^+$ Final Negative RVs $\mathbf{R}_f^-$

Fig. 3. Training of granular relevance vector machine using repetitive under-sampling (GRVM-*ru*). Here $\oplus$ denotes the concatenation and $\ominus$ denotes removing particular data from the set.

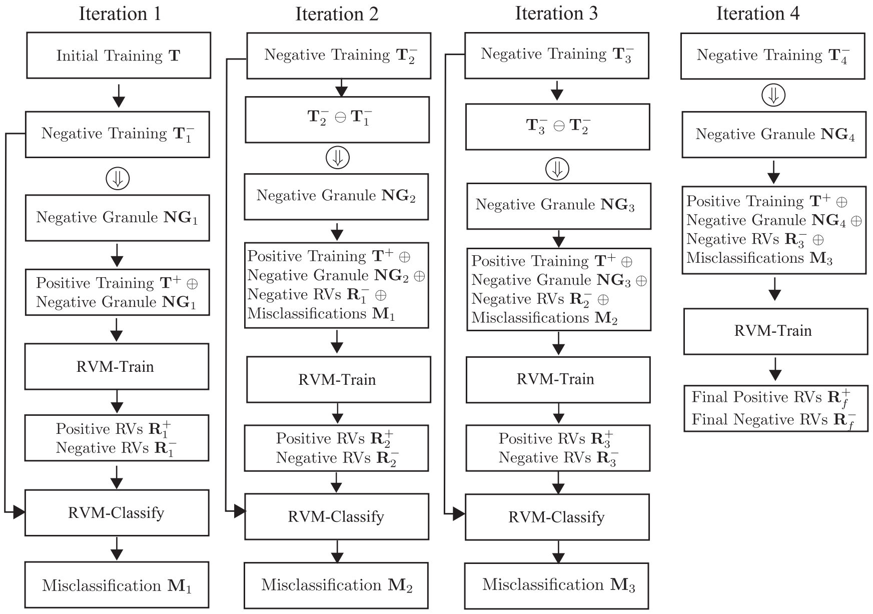
Iteration 1
Initial Training T
Negative Training T_1^-
Negative Granule NG_1
Positive Training $T^+ \oplus$ Negative Granule NG_1
RVM-Train
Positive RVs R_1^+ Negative RVs R_1^-
RVM-Classify
Misclassification M_1
Iteration 2
Negative Training T_2^-
$T_2^- \ominus T_1^-$
Negative Granule NG_2
Positive Training $T^+ \oplus$ Negative Granule $NG_2 \oplus$ Negative RVs $R_1^- \oplus$ Misclassifications M_1
RVM-Train
Positive RVs R_2^+ Negative RVs R_2^-
RVM-Classify
Misclassification M_2
Iteration 3
Negative Training T_3^-
$T_3^- \ominus T_2^-$
Negative Granule NG_3
Positive Training $T^+ \oplus$ Negative Granule $NG_3 \oplus$ Negative RVs $R_2^- \oplus$ Misclassifications M_2
RVM-Train
Positive RVs R_3^+ Negative RVs R_3^-
RVM-Classify
Misclassification M_3
Iteration 4
Negative Training T_4^-
Negative Granule NG_4
Positive Training $T^+ \oplus$ Negative Granule $NG_4 \oplus$ Negative RVs $R_3^- \oplus$ Misclassifications M_3
RVM-Train
Final Positive RVs R_f^+ Final Negative RVs R_f^-

Fig. 4. The steps involved in training granular relevance vector machine using 3σ rule (GRVM-3σ). Here $\Downarrow$ denotes the sub-sampling operation.

The relevance vectors ($\mathbf{R}_f^+, \mathbf{R}_f^-$) obtained from the final aggregation set are used to predict the class label of the probe sample. The formulation of GRVM-*ru* is given by

$$y = \mathbf{w}^T \Phi(\mathbf{T}_1^+ \oplus \mathbf{R}_1^- \oplus \mathbf{R}_2^- \oplus \ldots \oplus \mathbf{R}_l^-) \tag{13}$$

where l is the number of iterations used until GRVM-*ru* converges. The proposed formulation helps address the problem of unbalanced training data when the training data is small. However, the initial learning of GRVM-*ru* still requires the complete training data. With very large training data, the Hessian matrix becomes ill-conditioned for large values of N [30]. Therefore, it is unable to handle the problem of training with large highly unbalanced data.

2.3.2. Granular RVM using 3σ-rule (GRVM-3σ)

In the second approach, referred to as GRVM-3σ, the data from positive (under-represented) class is small and hence considered without sampling. However, the negative class has to be divided into granules using the *granule split* before applying RVM. It has been observed that imposter scores in biometrics generally follow a Gaussian distribution. According to the 3σ rule of Gaussian distribution, approximately 99.73% values lie within the data range covered by $\mu \pm 3\sigma$. Let $\mathbf{x}$ be the samples from the population that follow a Gaussian distribution with mean μ and standard deviation σ. The percentage of data covered can be given by

$$\begin{aligned} &p(\mu - \sigma \le \mathbf{x} \le \mu + \sigma) \approx 0.6827 \\ &p(\mu - 2\sigma \le \mathbf{x} \le \mu + 2\sigma) \approx 0.9545 \\ &p(\mu - 3\sigma \le \mathbf{x} \le \mu + 3\sigma) \approx 0.9973 \end{aligned} \tag{14}$$

This statistical property of Gaussian distribution is adopted to subsample the data from the majority class. Fig. 4 illustrates the steps involved in the proposed granular approach using 3σ subsampling which is explained below.

- Assuming that the positive class has lesser number of samples, all of them are considered to be informative and form the positive granule ($\mathbf{T}^+$).
- All the data from the negative class is represented by $\mathbf{T}^-$ with mean μ and standard deviation σ.
- The first negative granule ($\mathbf{NG}_1$) consists of only those samples from the negative (majority) class that are covered by

  $$\mathbf{NG}_1 = \left\{\mathbf{T}^- \mid \mathbf{T}^- \in \mu \pm \sigma\right\} \tag{15}$$

  This covers approximately 68% of the values. For large class imbalance there may be further difficulty in training RVM with 68% of the negative samples. This issue is further resolved by randomly under-sampling the data from each negative granule. The sampling is performed in the ratio of 1 : 2 : 3 : 4 for $(\mu \pm \sigma) : (\mu \pm 2\sigma) : (\mu \pm 3\sigma) : \ (> \mu \pm 3\sigma)$ respectively. This choice of ratios is made to handle the class imbalance problem by selecting equal amount of data from each negative granule. This also ensures that the relevance vectors are not biased towards any one part of the negative class.
- RVM (RVM$_1$) is then trained using the positive granule and the first negative granule ($\mathbf{NG}_1$). To find the training accuracy with this model, RVM$_1$ is tested on all the negative data prior to under-sampling that is covered within $\mu \pm \sigma$. The misclassifications ($\mathbf{M}_1$) that arise out of RVM$_1$ training are taken further to improve the performance of the classifier.
- The second negative granule ($\mathbf{NG}_2$) is formed by considering those samples that fall outside $\mu \pm \sigma$ but within $\mu \pm 2\sigma$. This can be given by the values of $\mathbf{T}^-$ that fall in the range of

  $$\mathbf{NG}_2 = \left\{\mathbf{T}^- \mid \mathbf{T}^- \in (\mu \pm 2\sigma) \;\&\; \mathbf{T}^- \notin (\mu \pm \sigma)\right\} \tag{16}$$

$\mathbf{NG}_2$ is further combined with negative relevance vectors ($\mathbf{R}_1^-$) from RVM$_1$ and the misclassifications from the last iteration ($\mathbf{M}_1$). The combined $\mathbf{NG}_2$ with positive granule is used to train RVM$_2$.

- The process is repeated for $\mathbf{NG}_3$ that contains negative samples as shown in Eq. (17). Subsampled $\mathbf{NG}_3$ along with the misclassifications of RVM$_2$ ($\mathbf{M}_2$) and $\mathbf{R}_2^-$ are used to train RVM$_3$.

$$\mathbf{NG}_3 = \{\mathbf{T}^- \,|\, \mathbf{T}^- \in (\mu \pm 3\sigma)\ \&\ \mathbf{T}^- \notin (\mu \pm 2\sigma)\}. \tag{17}$$

- The final negative granule ($\mathbf{NG}_4$) contains the remaining 0.3% of the samples that lie outside the range of $\mu \pm 3\sigma$. Thus, a final RVM (RVM$_4$) is trained on positive granule with $\mathbf{NG}_4$, misclassifications ($\mathbf{M}_3$) and negative relevance vectors ($\mathbf{R}_3^-$) of RVM$_3$.

Algorithm 4 outlines the steps involved in training granular RVM using 3σ rule (GRVM-3σ). The model trained with this data is finally used to perform classification. The formulation of GRVM-3σ is given by

$$y = \mathbf{w}^T \Phi(\mathbf{T}^+ \oplus \mathbf{NG}_4 \oplus \mathbf{R}_3^- \oplus \mathbf{M}_3) \tag{18}$$

The proposed granular RVM (GRVM-3σ) is capable of training highly imbalanced data. Note that the misclassifications that arise from the previous iteration are carried forward to re-train RVM with improved decision boundary. At the same time, use of RVs helps to avoid over-fitting. Unlike repetitive under-sampling, 3σ approach is relatively faster as RVM is trained using a smaller size of data.

Algorithm 4. GRVM-3σ-Train.

```
Input: T: Training data, t: Equivalent target values.
Output: R: Relevance vectors, w: Most probable weight matrix
$\mathbf{T}^+ = \{\mathbf{T} | \mathbf{T} \in \text{positive class}\}$
$\mathbf{T}^- = \{\mathbf{T} | \mathbf{T} \in \text{negative class}\}$
$n \leftarrow |\mathbf{T}^+|$
$PC \leftarrow 0$    // Percentage of data covered
$\mu \leftarrow \mathbf{mean}(\mathbf{T}^-)$
$\sigma \leftarrow \mathbf{sigma}(\mathbf{T}^-)$
$i \leftarrow 1$
while $PC < 1$ do
    $s \leftarrow n \times \frac{i}{10}$
    $l_i \leftarrow \mathbf{random}(s)$
    $\mathbf{T}_i^- = \{\mathbf{T}^- | \mathbf{T}^- \in \mu \pm i\sigma\}$
    If $i = = 1$ then
    |   $\mathbf{NG}_i \leftarrow \mathbf{T}_i^-(l_i)$        //     Undersamplingnegativegranule
    end
    |   $\mathbf{NG}_i \leftarrow [\mathbf{R}_{i-1}^- \oplus \mathbf{M}_{i-1} \oplus (\mathbf{T}_i^- \ominus \mathbf{T}_{i-1}^-)(l_i)]$
    end
    $\mathbf{T}_i \leftarrow \mathbf{T}^+ \oplus \mathbf{NG}_i$
    $[\mathbf{R}_i, \mathbf{w}_i] = \text{RVM} - \text{Train}(\mathbf{T}_i, \mathbf{t}_i)$
    $\mathbf{T}' \leftarrow \mathbf{T}^+ \oplus \mathbf{T}_i^-$
    $y = \text{RVM} - \text{Classify}(\mathbf{T}', \mathbf{R}_i, \mathbf{w}_i)$
    $\mathbf{M}_i = y \neq t'$
    $\mathbf{R}_i^- = \{\mathbf{R}_i | \mathbf{R}_i \in \text{negative class}\}$
    $PC \leftarrow |\mathbf{T}_i^-| / |\mathbf{T}^-|$
    $i \leftarrow i + 1$
end
```

2.4. Incremental granular relevance vector machine

The previous two sections propose algorithms to address the problem of large data and class imbalance individually. However, they cannot solve both the problems together. Therefore, this section proposes the Incremental Granular Relevance Vector Machine (*i*GRVM-3σ) to address both the problems using a single classifier.

Incremental RVM (*i*RVM) and granular RVM (GRVM-3σ) are fused hierarchically to generate a novel classifier – *i*GRVM-3σ. This classifier is robust to updates and handles large class imbalance problem as well. *i*GRVM-3σ functions hierarchically where for each new batch of training data, granular RVM is used to find the relevance vectors. Here, the experiments are performed using both GRVM variants but GRVM-3σ is preferred due to its ability to handle large unbalanced training data. The steps involved in *i*GRVM-3σ are:

- The process starts with incremental learning of the initial batch of training data ($\mathbf{T}_1$). Under Gaussian assumption on the negative training samples, the negative granule ($\mathbf{NG}_{11}$) is generated by retaining samples from $\mu \pm \sigma$.
- RVM is trained with the positive granule ($\mathbf{T}_1^+$) and the sub-sampled negative granule ($\mathbf{NG}_{11}$). The trained model is used to check for misclassifications ($\mathbf{M}_{11}$) that arise after learning.

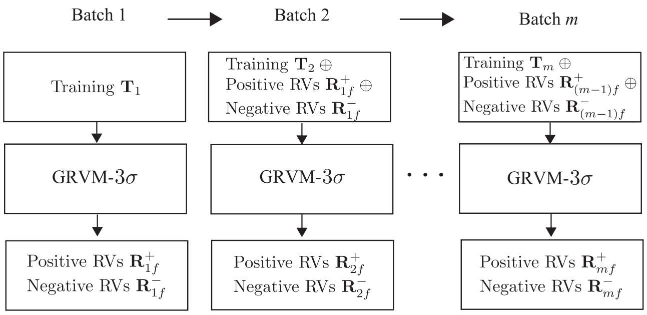
Batch 1
Batch 2
Batch m
Training $\mathbf{T}_1$
Training $\mathbf{T}_2 \oplus$ Positive RVs $\mathbf{R}_{1f}^{+} \oplus$ Negative RVs $\mathbf{R}_{1f}^{-}$
Training $\mathbf{T}_m \oplus$ Positive RVs $\mathbf{R}_{(m-1)f}^{+} \oplus$ Negative RVs $\mathbf{R}_{(m-1)f}^{-}$
GRVM-3σ
GRVM-3σ
GRVM-3σ
Positive RVs $\mathbf{R}_{1f}^{+}$ Negative RVs $\mathbf{R}_{1f}^{-}$
Positive RVs $\mathbf{R}_{2f}^{+}$ Negative RVs $\mathbf{R}_{2f}^{-}$
Positive RVs $\mathbf{R}_{mf}^{+}$ Negative RVs $\mathbf{R}_{mf}^{-}$

Fig. 5. Steps involved in training of *i*GRVM-3σ.

- The next negative granule ($\mathbf{NG}_{12}$) covers data in the range of $\mu \pm 2\sigma$ exclusive of values from $\mu \pm \sigma$. The RVM is trained with $\mathbf{T}_1^{+}$ along with $\mathbf{R}_{11}^{-}$ and $\mathbf{M}_{11}$. This process is iterated until the complete impostor training data is covered.
- The relevance vectors ($\mathbf{R}_{1f}^{+}, \mathbf{R}_{1f}^{-}$) after final training from $\mathbf{T}_1$ along with the next batch of training data ($\mathbf{T}_2$) are used to incrementally update the learnt model. The new training batch is thus represented as $\mathbf{T}_2 = \mathbf{T}_2 \oplus \mathbf{R}_{1f}^{+} \oplus \mathbf{R}_{1f}^{-}$.
- RVM is re-trained with this new unified training set ($\mathbf{T}_2$). The learning process of *i*GRVM-3σ is iterated for i=2,…,m until the new training batch arrives.
- The RVs from the latest enrolled batch of *i*GRVM-3σ ($\mathbf{R}_{mf}^{+}, \mathbf{R}_{mf}^{-}$) are finally used to make predictions using Eq. (11).

*i*GRVM-3σ provides a stable set of relevance vectors with significantly reduced training time. The steps involved in training the proposed *i*GRVM-3σ approach are summarized in Fig. 5 and Algorithm 5. The formulation of *i*GRVM-3σ can also be shown using Eq. (19)[1].

$$y = \mathbf{w}^T \Phi\left(\mathbf{T}_m \oplus \mathbf{R}_{(m-1)f}^{+} \oplus \mathbf{R}_{(m-1)f}^{-}\right) \tag{19}$$

The proposed *i*GRVM-3σ has the following attributes:

1. *Sparse*: It holds the sparsity property of RVM [24], compared to SVM, lesser number of relevance vectors are used to define the boundary.
2. *Scalable*: *i*GRVM-3σ performs re-training on encountering a new batch of training data. Hence the classifier is adaptive to environmental dynamics, which makes it scalable to new enrollments as well.
3. *Faster*: RVM requires matrix inverse operation for computing the covariance matrix (Eq. (6)) which is a costly operation for very large training databases. *i*GRVM-3σ recurrently trains on sub-sampled granules, thus the learning is faster compared to conventional RVM.
4. *Unbiased*: *i*GRVM-3σ handles class imbalance problem by the sub-sampling the data from the overpopulated class using granular computing. The model trained on the positive and sub-sampled negative data is unbiased in nature.

3. Case study: multimodal biometric match score classification

The formulations of the proposed algorithms are particularly helpful when the size of databases is large and they are unbalanced in terms of samples per class. Biometrics projects have both these characteristics. For instance, projects such as Aadhaar and US VISIT have millions of enrollments. The recognition pipeline of such projects involves four primary steps: (a) preprocessing, (b) segmentation, (c) feature representation, and (d) matching. *i*GRVM can be used for matching the extracted features or perform classification using match scores. For a pair of input biometric features, the match scores (either a distance score or a similarity score) generated by matching these features belong to two classes: *genuine* and *imposter*. Out of these, the genuine class is significantly less populated compared to the impostor class. Further, with increase in the number of individuals, there is concept drift of data distribution and this requires retraining of the learnt classification model. However, re-training the classifier in offline/batch mode every time on encountering new data can be computationally expensive. An adaptive classifier that incrementally adjusts the decision boundary has greater significance in biometrics. Therefore, in this research we select biometrics as the case study. Biometric scores are used in a *classification* based approach which accepts scores from multiple modalities and assigns a label to the integrated score vector using a classifier.

Algorithm 5. *i*GRVM-3σ-Train.

```
Input: $\mathbf{T}_m$: Training data, $\mathbf{t}_m$: equivalent target values.
Output: $\mathbf{R}_m$: relevance vectors, $\mathbf{w}_m$: most probable weight matrix
foreach Batch m of training do
    $\mathbf{T}_m \leftarrow \mathbf{T}_m \oplus \mathbf{R}_{m-1}$
    $[\mathbf{R}_m, \mathbf{w}_m] = \mathrm{GRVM}-3\sigma-\mathrm{Train}(\mathbf{T}_m, \mathbf{t}_m)$
end
```

[1] As the training of each RVM variant is different so the weight matrix ($\mathbf{w}$) is also different.

Table 1
Databases used in this case study.

Database	Modalities	Subjects	Scores	
			Genuine	Imposter
BSSR1 (Set 1)	Face, Finger	517	517	266,772
BSSR1 (Set 2)	Fingerprint	6000	6000	35,994,000
BSSR1 (Set 3)	Face	3000	6000	17,994,000
CASIA-Iris-Distance	Iris	142	37,314	5,259,588
BioSecure DS2	Face, Finger, Iris	51	Sessions 1 and 2	

Gutschoven and Verlinde [31] applied Support Vector Machines for multimodal biometric fusion. In some applications, the classification based approaches [31,32] have shown to outperform score fusion approaches. Further, online SVM has been applied in biometrics with a case study in near infrared face verification [10]. Singh et al. [33] fused visible and long wave infrared face images using 2ν-granular SVM. In this paper, the proposed incremental granular RVM learning algorithm is applied for biometric score classification and the results show that it can be used as an effective alternative that can optimize computational time without compromising the accuracy. The steps in d-dimensional biometric score vector classification using iGRVM-3σ are:

- *Training* iGRVM-3σ: The training set $(\mathbf{x}, \mathbf{t})$ consists of N match scores $\{x_{i1}, x_{i2}, \ldots, x_{id}, t_i\}_{i=1}^{N}$ obtained from d different sources with the corresponding class labels in t_i. The objective is to apply a function to $\mathbf{x}$ that provides a clear separation between the genuine and impostor classes. The proposed iGRVM-3σ-Train$(\mathbf{x}, \mathbf{t})$ is called for training score vectors to generate the weights $\mathbf{w}_m$ along with relevance vectors ($\mathbf{R}m$).
- *Classification*: The trained iGRVM-3σ model is used to predict the class of d-dimensional multimodal score vector of the probe sample given by x' using Eq. (11). The output from iGRVM-3σ is a probabilistic value which is used to take final decision.

3.1. Databases and algorithm used for evaluation

In this case study, three different databases are used to analyze the performance of iGRVM-3σ under different scenarios like combining multiple modalities, units, and algorithms. Table 1 summarizes the details of the databases used.

- *NIST Biometrics Score Set – Release 1 (BSSR1)* [34] contains similarity scores from two face and one fingerprint matchers. This database helps to perform a diversified study on multi-modal, multi-unit, and multi-classifier biometric systems. BSSR1 database is divided into three sets. *Set 1* consists of face and fingerprint scores from 517 individuals. This can be represented as Multimodal $=[\text{Face}_1, \text{Face}_2, \text{Finger}_{LI}, \text{Finger}_{RI}]$. This set can be used to study multi-modal fusion. *Set 2* consists of fingerprint scores from one matcher for 6000 individuals. For each individual, the score vector contains match scores from left and right index fingers respectively. This helps to study multi-unit classification of score vectors represented as Finger $=[\text{Finger}_{LI}\ \text{Finger}_{RI}]$. The scores from *set 3* are computed from two different classifiers for 3000 individuals. The score vector is represented as Face $=[\text{Face}_1\ \text{Face}_2]$ to highlight multi-classifier fusion. Since BSSR1 is a match score database, no feature extraction is required. The match scores are normalized using min–max approach and used for classification.
- *CASIA-Iris-Distance Version 4* [35] database contains iris images captured using a long-range multi-modal biometric image acquisition and recognition system. The images of CASIA-Iris-Distance are captured by a high resolution camera so both dual-eye iris and face patterns are included. The database consists of 142 individuals with both left and right eye images, and the score vector can be represented as Iris $=[\text{Left}_i\ \text{Right}_i]$. Both the irises are localized and the scores for both left and right units of iris are obtained individually using the VeriEye commercial matcher[2].
- *Biosecure DS2* [36] database contains face, fingerprint and iris images collected in two sessions with one month interval. For evaluation, cross device quality dependent protocol is used on the development set. The task becomes more challenging as the probe and gallery sets are taken from different sensors with an emphasis that gallery is always acquired from a high quality sensor. The development set consists of scores and quality measures corresponding to eight different modalities, distributed to the participants for algorithm development. The first session of the development set is used for training the classifier and the scores from session 2 are used for classification. In this paper, we perform a study on fusion of multiple modalities independent of quality so only the match scores corresponding to eight different modalities are considered for classification. Since this database also contains raw match scores only, feature extraction is not required.

The proposed iGRVM-3σ is compared with some existing classification approaches that have already been applied in biometrics.

- Likelihood Ratio (LR) [37],
- Support Vector Machine [1][3]and
- Relevance Vector Machine [24][4].

[2] http://www.neurotechnology.com/verieye.html

[3] LIBSVM – A Library for Support Vector Machines http://www.csie.ntu.edu.tw/cjlin/libsvm. The Matlab code uses mex files to provide an interface between C and Matlab.

[4] Sparse Bayesian Models. http://www.miketipping.com/sparsebayes.htm

Table 2
Protocols for experiments 1 and 2.

Database	Experiment 1				Experiment 2			
	Training		Testing		Training		Testing	
	Genuine	Imposter	Genuine	Imposter	Genuine	Imposter	Genuine	Imposter
BSSR1 (Set 1)	52	2668	465	264,104	52	26,677	465	240,095
BSSR1 (Set 2)	600	3599	5400	35,990,401	600	3,599,400	5400	32,394,600
BSSR1 (Set 3)	600	3599	5400	17,990,401	600	1,799,400	5400	16,194,600
CASIA-Iris-Distance	746	3156	36,568	5,256,432	–	–	–	–
Biosecure DS2	–	–	–	–	51	21,012	102	21,012

3.2. Experimental protocol

Two sets of experiments are performed with three times random cross validation to study the performance of *i*GRVM-3σ under different scenarios.

- Experiment 1 is performed to study the performance of *i*GRVM-3σ over other classifiers such as LR, SVM and RVM. In this experiment, all three sets of BSSR1 database and CASIA distance database are used. Table 2 shows the number of training samples used from each class for learning the classifiers. From the complete batch of training data, 10% genuine and imposter scores are used for incrementally learning the classifier in batches.
- Experiment 2 is performed to study the performance of *i*GRVM-3σ for highly unbalanced training datasets. Table 2 shows the experimental protocol followed for this study. The databases used in this case study are BSSR1 and Biosecure DS2. Biosecure DS2 by design is highly unbalanced, whereas for BSSR1, we have partitioned the datasets to showcase the unbalanced nature and classification performance.

3.3. Experiment 1: iGRVM-3σ performance

This experiment is performed to measure the performance of the proposed classifier as well as LR, SVM and RVM classification. The classifiers are trained using the experimental protocol mentioned in Table 2. The receiver operating characteristic (ROC) curve for Experiment 1 is shown in Fig. 6. Table 3 illustrates genuine acceptance rate (GAR) at 0.01% false acceptance rate (FAR). The key results and observations are as follows:

- For BSSR1 (all three sets) and CASIA distance databases, the proposed GRVM-3σ and *i*GRVM-3σ yield comparable results as other classifiers (LR and SVM) in terms of accuracy (Table 3). Since the imposter scores from biometric database are assumed to follow Gaussian distribution, GRVM-3σ divides the database into granules following the statistical property of Gaussian distribution. Further, GRVM-*ru* requires the complete database for initial training, so the problems inherent to large training persist whereas GRVM-3σ performs training on such large data using the divide and conquer approach. The amalgamation operation is performed at each stage of granulation so the training errors are considered simultaneously which further improves the performance in comparison to GRVM-*ru*.
- Table 4 shows that RVM training is costly while training SVM requires minimum time for all the databases[5]. It is to be noted that SVM is a *C* implementation whereas RVM is written using MATLAB, so a comparison of time across platforms is not reasonable. In practical applications, there is always an effort to minimize the testing time as training is an offline process. Testing time is directly proportional to the number of non-zero vectors used for making predictions. Table 4 shows the number of SVs/RVs provided by individual classifiers. RVM generates significantly reduced number of relevance vectors and hence is preferred over SVM. However, *i*GRVM-3σ performs iterative training to minimize the training error. RVs generated using GRVM-3σ are more compared to RVM but significantly less compared to SVM. As shown in Table 4, testing time on the entire testing set using the proposed variants of RVM are significantly less compared to SVM. While one can argue that the performance should also be compared with online SVM. The objective of this research is not to showcase the superiority of the proposed algorithm over SVM but to show that the modification over RVM makes it applicable to unbalanced large databases such as the ones observed in biometrics related problems.
- From the results shown in Table 3, it can be observed that the proposed *i*GRVM-3σ performs comparable to GRVM-3σ. In fact, all the incremental variants of RVM perform equivalent to their offline/batch mode including RVM. The incremental learning approaches facilitate learning with very large training datasets in batches which cannot otherwise be enrolled in offline mode. We also experimentally observed that considering only RVs for learning does not lead to reduction in performance compared to batch-based RVM learning. Therefore, incremental variants provide added advantage that can be useful in biometrics.

3.4. Experiment 2: unbalanced databases

This experiment is performed to understand the nature of these classifiers on highly unbalanced databases. The experimental protocol followed for highly unbalanced database is shown in Table 2. As mentioned previously, for incremental learning 10% of the training data is used.The ROC plots for unbalanced database is shown in the second column of Fig. 6. The key observations from this study are:

- Table 5 shows the accuracy of different classifiers on highly imbalanced data. LR outperforms other approaches in this experiment. In most of the cases, LR is designed to handle imbalance in the training database [37]. The performance of GRVM-3σ and *i*GRVM-3σ is

[5] The testing time reported in this paper is for complete testing samples as given in the experimental protocol (Table 2).

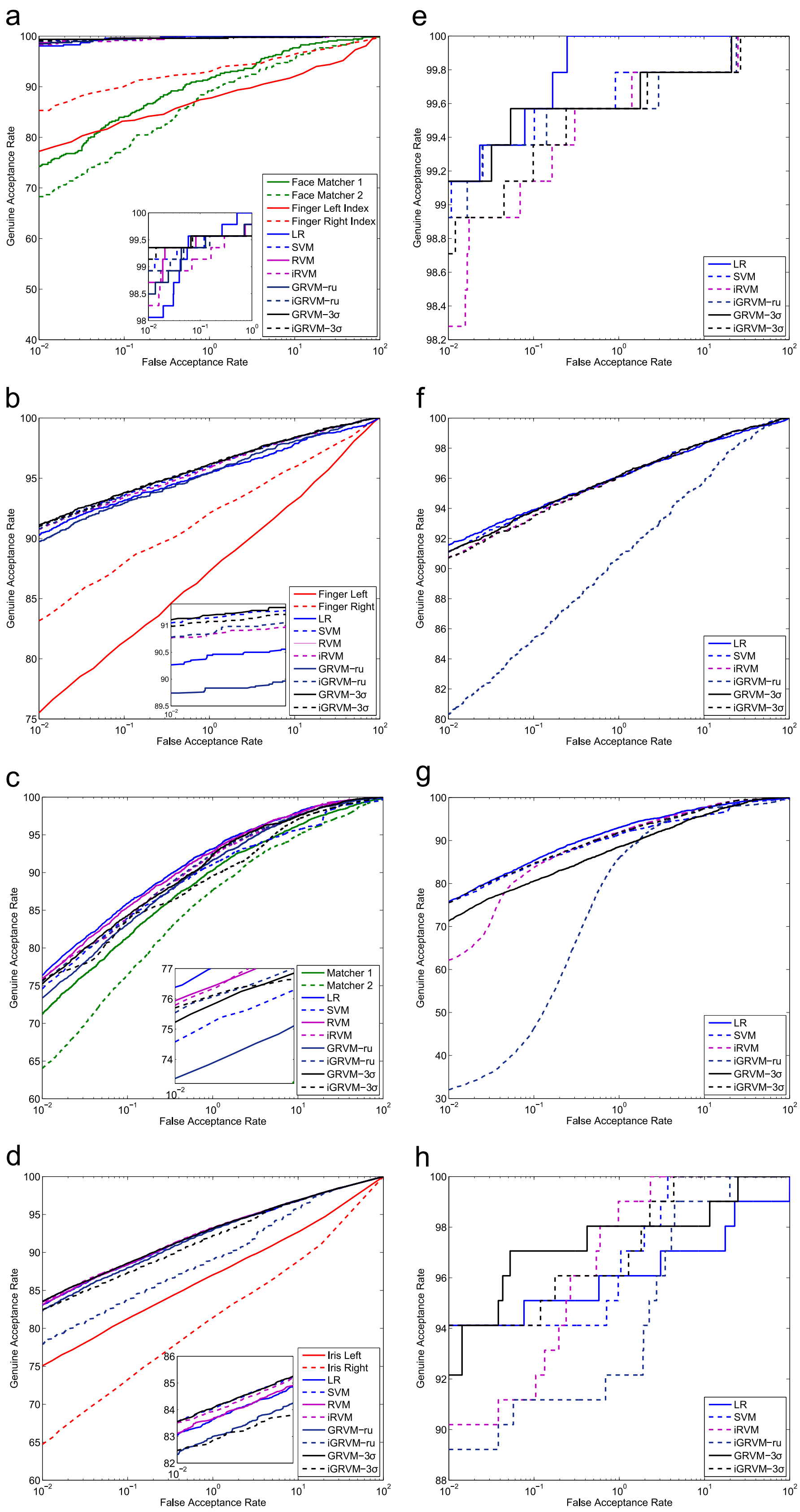

Fig. 6. ROC curves demonstrating the performance of the classifiers on the two experimental protocols. The first column shows the results for Experiment 1 on the (a) NIST Multimodal, (b) NIST Fingerprint, (c) NIST Face, and (d) CASIA-Iris-Distance databases. The second column shows the results on highly skewed training data (Experiment 2 protocol) for the (e) NIST Multimodal, (f) NIST Fingerprint, (g) NIST Face, and (h) BioSecure DS2 databases.

Table 3
Genuine acceptance rate (%) at 0.01% false acceptance rate for the NIST BSSR1 and CASIA databases.

Algorithm	**NIST BSSR1**			**CASIA**
	Set 1	Set 2	Set 3	
Face Matcher 1	74.27	–	71.22	–
Face Matcher 2	68.28	–	64.06	–
Finger Left	77.18	75.52	–	–
Finger Right	85.30	83.13	–	–
Iris Left	–	–	–	75.07
Iris Right	–	–	–	64.70
LR	98.06	90.26	**76.39**	83.04
SVM	98.92	91.05	74.58	83.55
RVM	98.70	89.74	75.85	83.11
*i*RVM	98.27	90.77	75.89	83.51
GRVM-*ru*	98.49	89.74	73.38	82.32
*i*GRVM-*ru*	98.92	90.78	75.61	77.98
GRVM-3σ	**99.35**	**91.10**	75.21	**83.55**
*i*GRVM-3σ	99.14	90.98	75.70	82.48

Table 4
Time taken (in seconds) by different classifiers on the complete training and testing databases.

Approach	**BSSR1**									**CASIA-Iris-Distance**		
	Set 1			Set 2			Set 3					
	Training	Testing	# SV/RV	Training	Testing	# SV/RV	Training	Testing	# SV/RV	Training	Testing	# SV/RV
LR	0.81	0.72	G:3, I:3	0.45	69.78	G:3, I:3	1.48	50.42	G:2, I:5	0.56	1.05	G:3, I:3
SVM	0.02	0.23	54	0.27	1068.77	1200	0.94	242.44	576	0.05	10.89	408
RVM	84.46	0.09	4	425.19	471.43	4	361.54	138.47	10	3443.93	3.55	281
*i*RVM	2.92	0.05	4	149.76	8.74	4	22.75	147.77	20	1029.85	0.10	5
GRVM-*ru*	350.16	0.06	3	405.22	358.17	4	1101.70	118.24	4	2421.92	0.13	8
*i*GRVM-*ru*	9.82	0.13	4	253.72	441.54	14	113.13	77.52	4	150.94	0.32	28
GRVM-3σ	6.90	0.20	22	554.21	673.41	48	7.27	121.31	9	7.02	0.28	21
*i*GRVM-3σ	2.48	0.34	19	31.07	374.08	10	11.23	126.14	4	61.44	1.07	3

Table 5
Genuine acceptance rate (%) at 0.01% false acceptance rate for highly skewed databases.

Algorithm	**Databases**			
	Set 1	Set 2	Set 3	BioSecure DS2
LR	99.14	91.54	75.76	94.12
SVM	98.92	91.10	75.50	94.12
*i*RVM	98.28	90.74	62.15	90.20
*i*GRVM-*ru*	98.92	80.33	32.07	89.22
GRVM-3σ	99.14	91.11	71.30	92.36
*i*GRVM-3σ	98.71	90.71	75.59	94.12

Table 6
Training and testing time taken (in seconds) by various learning based classification approaches for highly unbalanced data.

Approach	**BSSR1**									**BioSecure DS2**		
	Set 1			Set 2			Set 3					
	Training	Testing	#SV/RV	Training	Testing	#SV/RV	Training	Testing	#SV/RV	Training	Testing	#SV/RV
LR	3.40	0.75	G:3, I:10	227.96	24.84	G:3, I:3	122.97	10.88	G:3, I:5	4.50	0.03	G:2, I:2
SVM	0.06	0.41	58	391.75	1552.39	1200	1113.71	546.30	1200	0.06	0.03	46
*i*RVM	51.25	0.12	4	16,395.50	411.44	7	27.25	100.84	5	235.19	0.01	8
*i*GRVM-*ru*	167.49	0.09	3	72,409.20	250.48	4	6650.31	71.49	10	2568.83	0.03	20
GRVM-3σ	11.31	0.13	4	251.70	426.01	378	23.86	151.40	39	2.09	0.003	5
*i*GRVM-3σ	18.66	0.06	3	298.51	203.36	11	24.39	82.47	55	6.49	0.006	3

comparable to LR. Classification accuracy improves for more data but the performance of some classifiers such as *i*RVM and *i*GRVM-*ru* reduces due to biased distribution of data.

- The training database is very large so some offline classifiers such as RVM and GRVM-*ru* cannot be trained. As explained earlier, these approaches require complete data for initial training. If training database is large enough as given in Table 2, RVM and GRVM-*ru* generate memory error and classifier fails to learn.
- In almost all the experiments of unbalanced nature it is observed that the proposed GRVM-3σ performs comparable to LR and SVM. This approach samples the database following the statistical property of Gaussian and is preferred over random data sampling. The information loss is minimal and hence the classifier performs well for highly unbalanced biometric databases.
- Testing time required by *i*GRVM-3σ is less compared to SVM (Table 6). Thus, *i*GRVM-3σ is comparable to other existing classifiers in terms of time and performance. This approach can be taken as an alternative to existing learning based approaches.
- BioSecure DS2 is the most challenging database in terms of class imbalance. This database is utilized to study the performance of *i*GRVM-3σ under such large variations. Table 5 shows that *i*GRVM-3σ performs as good as LR and SVM with an accuracy of 94.12%. However, it requires minimum testing time on this database due to significantly reduced number of relevance vectors.

4. Conclusions

The main contribution of this research is to propose incremental and granular learning in RVM and develop *i*GRVM classifier. The proposed classifier not only preserves the *sparse* property of original RVM classifier, but it is also scalable, faster and can be trained with unbalanced large training samples. The case study on multibiometric score classification is performed using the NIST BSSR1, CASIA-Iris-Distance V4, and Biosecure DS2 databases. Experimental results suggest that the proposed variant is better than original RVM classifier and comparable to existing classifiers such as LR and SVM. The key advantage of the proposed classifier is in terms of testing time which is significantly better than SVM. It is our assertion that *i*GRVM can be considered as an attractive alternative to approaches such as LR and SVM. As future research direction, we plan to extend the proposed *i*-GRVM to semi-supervised version and apply it for different applications including context switching [38].

Conflict of interest

None declared.

References

[1] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag New York, Inc., 1995.
[2] Unique Identification Authority of India. http://uidai.gov.in.
[3] US-VISIT. http://www.dhs.gov/us-visit-office.
[4] N. A. Syed, S. Huan, L. Kah, K. Sung, Incremental Learning with Support Vector Machines, in: Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence, 1999.
[5] H. Zhao, P. Yuen, J. Kwok, A. Novel, Incremental principal component analysis and its application for face recognition, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 36 (4) (2006) 873–886.
[6] C.-X. Ren, D.-Q. Dai, Incremental learning of bidirectional principal components for face recognition, Pattern Recognit. 43 (1) (2010) 318–330.
[7] H. Zhao, P. Yuen, Incremental linear discriminant analysis for face recognition, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 38 (1) (2008) 210–221.
[8] H. Lamba, T. Dhamecha, M. Vatsa, R. Singh, Incremental subclass discriminant analysis: a case study in face recognition, in: 19th IEEE International Conference on Image Processing, 2012, pp. 593–596.
[9] G. Cauwenberghs, T. Poggio, Incremental and decremental support vector machine learning, Adv. Neural Inf. Process. Syst. 13 (2001) 409–415.
[10] R. Singh, M. Vatsa, A. Ross, A. Noore, Biometric classifier update using online learning: a case study in near infrared face verification, Image Vis. Comput. 28 (7) (2010) 1098–1105.
[11] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[12] R. Barandela, J. Snchez, V. Garca, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognit. 36 (3) (2003) 849–851.
[13] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
[14] P. Domingos, MetaCost: a general method for making classifiers cost-sensitive, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 155–164.
[15] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.
[16] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.
[17] M.A. Tahir, J. Kittler, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recognit. 45 (10) (2012) 3738–3750.
[18] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: 14th International Conference on Machine Learning, 1997, pp. 179–186.
[19] R. Barandela, R. Valdovinos, J. Sanchez, F. Ferri, The imbalanced training sample problem: under or over sampling?, Struct. Syntactic Statistical Pattern Recognit. (2004) 806–814.
[20] Y. Tang, Y.-Q. Zhang, N. Chawla, S. Krasser, SVM modeling for highly imbalanced classification, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 39 (1) (2009) 281–288.
[21] T. Lin, Data mining and machine oriented modeling: a granular computing approach, Appl. Intell. 13 (2) (2000) 113–124.
[22] Y. Tang, Granular support vector machines based on granular computing, soft computing and statistical learning, (Ph.D. thesis). Georgia State University, 2006.
[23] S. Nikitidis, N. Nikolaidis, I. Pitas, Multiplicative update rules for incremental training of multiclass support vector machines, Pattern Recognit. 45 (5) (2012) 1838–1852.
[24] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (5) (2001) 211–244.
[25] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press 2001.
[26] M.E. Tipping, The relevance vector machine, in: S. Solla, T. Leen, K. Müller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, 2000, pp. 652–658.
[27] J.C. MacKay, The evidence framework applied to classification networks, Neural Comput. 4 (5) (1992) 720–736.
[28] M. Tipping, A. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, in: Ninth International Workshop on Artificial Intelligence and Statistics, 36, 2003.
[29] D. Tzikas, A. Likas, N. Galatsanos, Incremental relevance vector machine with kernel learning, in: 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications, 2008, pp. 301–312.
[30] X. Xiang-min, M. Yun-feng, X. Jia-ni, Z. Feng-le, Classification Performance Comparison between RVM and SVM, in: IEEE International Workshop on Anti-counterfeiting, Security, Identification, 2007, pp. 208–211.

[31] B. Gutschoven, P. Verlinde, Multi-modal Identity Verification using Support Vector Machines (SVM), in: International Conference on Information Fusion, vol. 2, 2000, pp. 3–8.

[32] H. Mehrotra, M. Vatsa, R. Singh, B. Majhi, Biometric match score fusion using RVM: a case study in multi-unit Iris recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 65–70.

[33] R. Singh, M. Vatsa, A. Noore, Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition, Pattern Recognit. 41 (3) (2008) 880–893.

[34] NIST BSSR1. http://www.nist.gov/itl/iad/ig/biometricscores.cfm.

[35] CASIA Iris Image Database Version 4.0, http://biometrics.idealtest.org.

[36] N. Poh, T. Bourlai, J. Kittler, A Multimodal biometric test bed for quality-dependent, cost-sensitive and client-specific score-level fusion algorithms, Pattern Recognit. 43 (3) (2010) 1094–1105.

[37] K. Nandakumar, Y. Chen, S. Dass, A. Jain, Likelihood ratio-based biometric score fusion, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 342–347.

[38] S. Bharadwaj, H.S. Bhatt, R. Singh, M. Vatsa, A. Noore, Qfuse: online learning framework for adaptive biometric system, Pattern Recognit. 48 (11) (2015) 3428–3439.

Hunny Mehrotra received M.Tech and Ph.D. degrees in Computer Science in 2010 and 2014 respectively from National Institute of Technology Rourkela, India. Her area of research includes biometrics, image processing, and computer vision. She has been conferred with various prestigious awards such as Google India Women in Engineering Award in 2010, Innovative Student Project Award in 2010 by INAE, and fellowship in 2012 from Department of Science and Technology under Women Scientist Scheme, Government of India. She is currently Lead Engineer at Samsung Research Institute, India.

Richa Singh received the M.S. and Ph.D. degrees in computer science from West Virginia University, Morgantown, USA, in 2005 and 2008, respectively. She is currently an Associate Professor and the Kusum and TV Mohandas Pai Faculty Research Fellow with the Indraprastha Institute of Information Technology Delhi, India. Her research has been funded by the UIDAI and DeitY, India. Her areas of interest are biometrics, pattern recognition, and machine learning. She is a recipient of the FAST Award by DST, India. She is also an Editorial Board Member of Information Fusion (Elsevier), IEEE Access, and the EURASIP Journal of Image and Vision Processing (Springer). She is a member of the Computer Society and the Association for Computing Machinery. She has co-authored over 150 research papers and received several best paper and best poster awards in international conferences. She is a recipient of the NVIDIA Innovation Award 2015 and the Best Reviewer Award at the IAPR International Conference on Biometrics 2013. She serves as the General Co-Chair of the IEEE International Conference on Identity, Security and Behavior Analysis 2017 and PC Co-Chair of the IEEE International Conference on Biometrics: Theory, Applications, and Systems 2016.

Mayank Vatsa received the M.S. and Ph.D. degrees in computer science from West Virginia University, Morgan- town, USA, in 2005 and 2008, respectively. He is currently an Associate Professor and AR Krishnaswamy Faculty Research Fellow with the Indraprastha Institute of Information Technology Delhi, India. His research has been funded by the UIDAI, DST, and DeitY. He has authored over 150 research papers and received several best paper and best poster awards and the NVIDIA Innovation Award 2015. His areas of interest are biometrics, image processing, machine learning, and information fusion. He is a recipient of the FAST Award by DST, India. He is a member of the Computer Society and the Association for Computing Machinery. He is the Vice President (Publications) of the IEEE Biometrics Council, an Area Editor of the IEEE Biometric Compendium, and an Associate Editor of the Information Fusion Journal (Elsevier) and the IEEE ACCESS. He was the Program Committee Co-Chair of the IAPR International Conference on Biometrics 2013 and the IEEE/IAPR International Joint Conference on Biometrics 2014. He also serves as the PC Co-Chair of IEEE International Conference on Identity, Security and Behavior Analysis 2017.

Banshidhar Majhi is Professor in Department of Computer Science and Engineering, National Institute of Technology Rourkela, India since 2006. He has 20 years of teaching and research experience. He has published several articles in refereed journals and international conferences. He has worked on several government funded projects. His area of interest includes data structures, image processing, cryptography, biometrics, parallel processing and soft computing.