

A Novel Partially Connected Cooperative Parallel PSO-SVM Algorithm: Study Based on Sleep Apnea Detection

Yashar Maali

Faculty of Engineering and IT
University of Technology, Sydney
Sydney, Australia
Yashar.Maali@student.uts.edu.au

Adel Al-Jumaily

Faculty of Engineering and IT
University of Technology, Sydney
Sydney, Australia
Adel.Al-Jumaily@uts.edu.au

Abstract— Sleep disorders are common in a general population. It effect one in 5 adults and has several short term and long term bad side effects on health. Sleep apnea (SA) is the most important and common component of sleep disorders. This paper presents an automatic approach for detecting apnea events by using few bio-singles that are related to breathe defect. This work uses only air flow, thoracic and abdominal respiratory movement as input. The proposed algorithm consists of three main parts which are signal segmentation, feature generation and classification. A new proposed segmentation method intelligently segments the input signals for further classification, then features are generated for each segment by wavelet packet coefficients and also original signals. In classification phase a unique parallel PSO-SVM algorithm is investigated. PSO used to tune SVM parameters, and also data reduction. Proposed parallel structure used to help PSO to search space more efficiently, also avoiding fast convergence and local optimal results that are common problem in similar parallel algorithms. Obtained results demonstrate that the proposed method is effective and robust in sleep apnea detection and statistical tests on the results shown superiority of it versus previous methods even with more input signals, and also versus single PSO-SVM. Using fewer signals means more comfortable to subject and also, reduction of cost during recording the data.

Keywords; Sleep apnea; PSO; parallel programming; SVM

I. INTRODUCTION

Sleep disorders are important because they are common in a general population; a survey in 1987 [1] reported that at least one symptom of disturbed sleep was present in 41% of all subjects; and still sleep disorder is common now [2], for instance, Young reported that one daytime sleepiness in 5 adults in 2004 [3].

The sleep disorders have several short term and long term bad side effects[4]. Short-term effects lead to impaired attention and concentration, lowered life quality, increased rates of absenteeism with less productivity, and greater possibility of accidents at work, home or on the road. Long-term consequences of sleep deprivation include increased morbidity and mortality from more automobile accidents, coronary artery disease, heart failure, high blood pressure, obesity, type 2 diabetes mellitus, stroke and memory impairment as well as depression. Long-term consequences, however, are still open [5] for further academic research..

Sleep apnea (SA) is one of the most common and important component of sleep disorders. Unfortunately, because of

person's unawareness, sleep apnea may go undiagnosed for years [6, 7]. Usually the SA case is often observed via patient's spouse, a roommate, or a family member who has witnessed the apnea periods alternating with arousals and accompanied by loud snoring [8, 9]. The patients - who have symptoms of SA - should be checked through an overnight sleep study, in sleep center. The diagnosing of SA is usually achieved through analysis of recorded signals by polysomnography, an integrated device comprising EEG, EMG, EOG, ECG and oxygen saturation [10]. The polysomnography also recording the airflow through the mouth and nose, thoracic and abdominal respiration measurement units [11], thoracic breathing movements and the position of the body during sleep. From overnight sleep studies apnea/hyponea index (AHI) can be found, the AHI will holds the sum of apneas, hypopneas, and respiratory arousals per hour during sleep after been standardized. AHI is used to assess the severity of apnea according to the Chicago criteria: AHI<5, normal; AHI =5- 15, mild; AHI=15-30, moderate; and AHI >30, severe [12].

The polysomnography recording need to be reviewed in order to detect events of SA, the manual review by experts is high cost and time-consuming. Therefore, many efforts have been done to develop systems that analyse recording singles automatically [13-15]. For this reason several Artificial Intelligent algorithms are used in this area, like as fuzzy rule based system or genetic-SVM algorithms that proposed by authors previously[16].

In this paper, sleep apnea events detected by a PSO-SVM algorithm that implemented in a new parallel structure. Details of the detection process are explained in future sections. This paper is organized as follow, firstly, some preliminaries about algorithms and a basic concept about parallel PSO implementation has been introduced. In section III, details of the proposed approach comprised of signal segmentation, feature generation and classification by PSO-SVM is explained. Finally, in section IV, experimental results on real data are reviewed and the paper is conclude in section V

II. PRELIMINARY TECHNIQUES INFORMATION

A. Support vector machines

The Support vector machines (SVM) pioneered by Vapnik[17] is known as an excellent tool for classification and regression problems. We merely outline its main ideas here.

Given a training set of data-label pairs (x_i, y_i) , $i = 1, 2, \dots, m$ where $x_i \in R^n$ and $y_i \in \{+1, -1\}$, the generalized

linear SVM finds an optimal separating hyperplane $f(x) = \langle w, x \rangle + b$, by solving the following optimization problem:

$$\begin{aligned} & \min \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \\ & \text{s. t. } y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \quad \xi_i \geq 0, i = 1, \dots, l. \end{aligned}$$

Where C is a penalty parameter on the training error, and ξ_i is the non-negative slack variable. SVM finds the hyperplane that provides the minimum number of training errors (i.e., to keep the constraint violation as small as possible). This optimization model can be solved by introducing the Lagrange multipliers α_i for its dual optimization model. After the optimal solution α_i^* is obtained, the optimal hyperplane parameters w^* and b^* can be determined, and the indicator function (classifier) can be written as:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i \alpha_i^* (x_i \cdot x_j) + b^* \right) \text{ or } \text{sgn} ((w^* \cdot x) + b^*).$$

The nonlinear SVM maps the training samples from the input space into a higher dimensional feature space via a mapping function ϕ . By performing such a mapping, the training samples could be linearly separated by applying the linear SVM formulation. The scalar product $\langle \phi(x_i), \phi(x_j) \rangle$ is calculated directly by computing the kernel function $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ for given training data in an input space.

There are 3 common kernel functions in SVM:

Polynomial kernel: $K(x_i x_j) = (x_i x_j + 1)^q$

RBF kernel: $K(x_i x_j) = e^{-\gamma |x_i - x_j|^2}$

Sigmoid kernel: $K(x_i x_j) = \tanh(\gamma x_i^T x_j + c)$

Here q, γ, c are kernel parameters.

By introducing the kernel function, the nonlinear SVM classifier has the following forms:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i \alpha_i^* k(x_i, x_j) + b^* \right).$$

B. Particle swarm optimisation (PSO)

In 1995, Kennedy and Eberhart introduced particle swarm optimization (PSO)[18]. PSO is a population-based stochastic optimization technique based on the movement of swarms and inspired by social behavior of birds or fishes. In comparison with other metaheuristic algorithms like genetic algorithms (GAs), PSO have fewer complicated operations, fewer defining parameters, can be coded in just a few lines and it is highly dependent on stochastic processes. Because of simplicity and good performance of the PSO, this algorithm has received increasing attention in recent years [19, 20].

For the D -dimensional feature space including M particles, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ and $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ are the position and velocity of the i th particle respectively, $i = 1, 2, \dots, M$. The performance of x_i is evaluated by its fitness function value. In each iteration, the particle updates own position and velocity by tracking its best solution and the global best solution discovered by all particles in the swarm.

Let p_{best} denote the best previous position encountered by the i th particle, and g_{best} denote the global best position so far. Having find p_{best} and g_{best} , the i th particle updates its position and velocity according to formula (1) and (2).

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1^k (p_{best}^k - x_{id}^k) + c_2 r_2^k (g_{best}^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

Where r_1^k and r_2^k are random numbers in $[0, 1]$, k is the iteration counter and positive constant c_1 and c_2 are personal and social learning factors. x_{id}^k and v_{id}^k are the current position and velocity of the d th ($d = 1, 2, \dots, D$) dimension in the k th iteration of the i th particle, respectively.

Later, Shi and Eberhart [21] introduced an inertia weight w , which control the impact of the previous velocity on the current velocity, by modifying Eq.(1)to

$$v_{id}^{k+1} = w v_{id}^k + c_1 r_1^k (p_{best}^k - x_{id}^k) + c_2 r_2^k (g_{best}^k - x_{id}^k) \quad (3)$$

Suitable inertia weight w provides a balance between global and local exploration and exploitation, and on average results in less iterations required to find a sufficiently optimal solution. Previous studies [22] have shown that a time-dependent weight factor often outperforms a fixed factor. The most common functional form for this weight factor is linear, and changes with time step as follows:

$$w(k) = w_{max} - \frac{w_{max} - w_{min}}{N_{iter}} k \quad (4),$$

where N_{iter} is the maximum number of iterations and w_{max} and w_{min} are often selected to be 0.9 and 0.4, respectively[22].

The velocity v_{id} is restricted to the $[v_{min}, v_{max}]$ range. *This range* determines the resolution of the search regions between the present and target position. If v_{max} is too high, the particle may fly over the good solutions. If v_{max} is too small, the particle may not explore sufficiently beyond local solutions and get trapped in a local optimum.

The constants c_1 and c_2 represent the weight of the stochastic acceleration terms that pull each particle toward p_{best} and g_{best} . Low values allow particles to roam far from target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, the target regions. Hence, the acceleration constants c_1 and c_2 are often set to be 2.0[23].

C. Parallel PSO

Recently, the availability of cheap and fast parallel hardware has encouraged researcher to possibility of implementing of parallel type of metaheuristic algorithms for

large scale problems. Specially population based algorithms, like as genetic algorithms, are good candidate for parallelization[24]. PSO is also selected to implemented parallel in many studies recently [25, 26].

Generally, parallel processing can be classified to two major classes as pipeline processing and data parallelism. Pipeline processing separates the problem into a cascade of tasks where each of the tasks is executed by an individual processor. Data are transmitted through each processor which executes a different part of the process on each of the data elements. Since the program is distributed over the processors in the pipeline and the data moves from one processor to the next, no processor can proceed until the previous processor has finished its task. Data parallelism is an alternative approach which involves distributing the data to be processed amongst all processors which then executes the same procedure on each subset of the data. [27].

Data parallelism used widely for implementing metaheuristic algorithms like PSO. The parallel PSO can be classified into three categories:

(a) Global or Master–slave PSO: This model uses a single global population and the fitness evaluation is done on different processors. The nature of PSO is not changed because an algorithm still works with the whole population. In this model, the gain is only the speedup in the optimization process[28] as shown in figure 1.

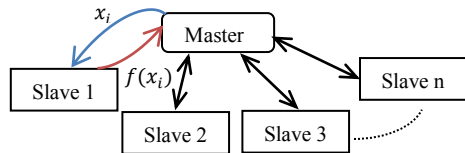


Figure1. Master- Slave parallel structure.

(b) Migration PSO or island model (Coarse-grained): The population is divided into a few large subpopulations. Each of these subpopulations is maintained by different processors. Then, according to some “migration-strategy”(commonly a given number of iterations) individuals are exchanged[28]. Island PSO models also referred in literatures as multi PSO [29] as figure 2 shown.

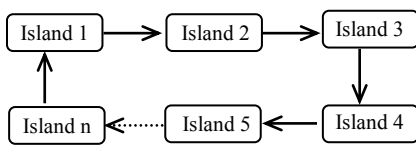


Figure2. Island parallel structure.

(c) Fine-grained PSO or cellular PSO: The population is separated into a large number of very small subpopulations, which are maintained by different processors. The subpopulation may be only a particle. This model is suitable for massively parallel architectures – machines consisting of a huge number of basic processors and connected with a specific high speed topology. The computer structure limits an

interaction between individuals. As illustrated in Figure 3, each individual I_k has 4 neighbors. Such neighborhood restriction delays the information exchange between non-neighbor processes, increasing the diversity in the search[28]. The use of local selection and reproduction rules leads to a continuous diffusion of individuals over the population. Therefore this model is also called diffusion mode[30].

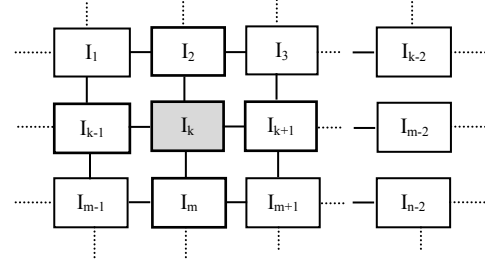


Figure 3. Cellular parallel structure.

III. APPROACH AND METHOD

This paper introduces new technique for Sleep apnea detection through using new segmentation method and novel parallel PSO-SVM algorithm. It is based on using just three signals, airflow, abdominal and thoracic movement signals as inputs. The methodology comprised of three main stages.

- 1- Signal segmentation: in this stage, data are segmented to create a Reasoning Units (RUs) that may be containing one or more SA case for further analysis.
- 2- Feature generation: this stage contains generating of several statistic features for each RU from wavelet packet coefficients and original signals.
- 3- Parallel PSO-SVM: in this stage, PSO algorithm used for selecting best training data and features subset interactively by SVM, as fitness evaluator, for final classification of RUs to sleep apnea or normal ones, this PSO-SVM applied in parallel by a new architecture to achieve better performance and avoiding of local optimal solutions.

Details of new algorithms proposed in each part, are as follows:

A. Signal Segmentation

In the first step, the input signals segmented to Reasoning Units (RUs)[31]. Each RU represents an interval which has more chance to contain sleep apnea event. The length of each segment selected as 30 seconds based on consideration of state of art in sleep disorders and especially sleep apnea literatures. The proposed signal segmentation algorithm in this study is based on [32] with some modification. The original algorithm [32] used only airflow for detecting sleep apnea events but our proposed modified version consider three signals as input to generate RUs. Pseudo code of the modified algorithm is as follow.

- Pre processing: Each signal normalized by attention to its mean and variances.

- Amplitude calculator: Input signals are 10 Hz, so for each signal, difference between local maximum and local minimum of each 10 consecutive data, from each second, is considered as amplitude of that second.
- Amplitude classifier: for classifying each amplitude vector value (v_i). First of all, the amplitude reference value (AV_{REF}) is computed, that is the last maximum amplitude value in a past two minutes interval, previous to (v_i). Then, if the amplitude value (v_i) is greater than $0.25*AV_{REF}$, it's classified as normal. Otherwise it's classified as feasible event. The factor 0.25 is based on [32].
- Amplitude reviewer: Amplitude reviewer takes the classified amplitude vectors as input and reviews the feasible events, this task accomplishes as an event fusion (a and b steps) as follows:

(a) As a complete respiratory cycle lasts at least 3 seconds, amplitude values located between feasible event values and separated no more than 3 seconds, are classified as a feasible event.

(b) The amplitude values that primary classified as feasible event will be classified as normal if its duration is less than 10 seconds.

- RU generation: First of all computed amplitude of the airflow signal is reviewed and for each amplitude value that labeled as feasible event, one RU with length equal to 30 seconds is created. After generating RUs related to airflow's feasible events of abdominal and thoracic movement signals are reviewed. In this step, a new RU will be created between previous RUs if a feasible event found in abdominal or thoracic amplitudes and also if there is room for that RU between previous made RUs.

By completing this algorithm some not overlapped RUs will be generated. Each generated RU has chance to contain at least one SA event. In future steps these RUs will be classified and filtered to SA or normal by more comprehensive methods. It must be noticed that segmenting the signals to the not overlapped windows, without any smart process or actually by a blind segmentation can reduce the final accuracy. As the figure 4 describes, in 4.a whole sleep apnea event correctly located in one window but in 4.b the length of sleep apnea event in each window is less than 10 seconds so these windows can be wrongly classified as normal in future steps. This problem has been dealt with it in our propose signal segmentation method.

B. Features Extraction

After determining RUs, some statistical measures corresponding to signals relevant to each RU are generated as features. Features extraction plays an important role in recognition systems, in this paper features generated from wavelet coefficients and also from original signals.

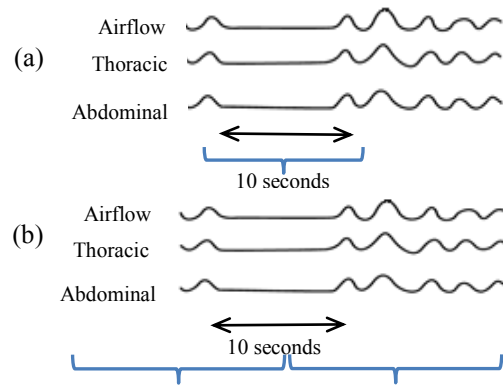


Figure 4. (a) Suitable RU for apnea detection; (b) these two RUs cannot classify as apnea events currently because length of apnea event in each of them is less than 10 seconds.

To find these features, 3 levels "Haar" wavelet packet applied on input signals. Then several statistical measures are computed by attention to the coefficients related to each RUs. These features represent the inputs of proposed PSO-SVM algorithm in the next step. Full list of proposed features are included in Table I.

$\log(\text{mean}(x^2))$	$\text{kurtosis}(x^2)$	$\text{geomean}(\text{abs}(x))$
$\text{std}(x^2)$	$\text{var}(x^2)$	$\text{mad}(x)$
$\text{skewness}(x^2)$	$\text{mean}(\text{abs}(x))$	$\text{mean}(x^2)$
$\text{skewness}(x)$	$\text{kurtosis}(x)$	$\text{var}(x)$
$\text{geomean}(x^2)$	$\text{mad}(x^2)$	$\text{std}(x)$

Table I: List of statistical features, x is coefficients of wavelet.

Exact mathematical definitions of these features are available in Appendix A.

Also for each signal, difference of local maximum and minimum of each 10 seconds is computed and then same features as Table 1 are computed for these differences during each RU.

C. Paralle PSO-SVM algorithm

After generating RUs and features related to each RU, more attractive classification method needed to separate the RUs to sleep apnea or normal ones. For this reason, a partially connected parallel PSO-SVM algorithm with a new architecture is proposed. This algorithm is used for features selection and also tuning the parameters of the SVM.

First of all, RUs of each subject (patient) separated randomly to three equal sets as: train, test and validation. We want to use learning method by using training and test data and then validate it by using validate set.

The learning algorithm, PSO-SVM, used for feature subset selection, but also during this work founded that selecting best

training data, from fixed set of training data, have important impact on overall performance as well as the selecting the best subset of features. This can be result of interferences between training data that might lead to low classification accuracy, so we actually filter training data. Training data reduction also, applied in previous studies [33].

It must be noted that the PSO-SVM tries to find a best training data and features subset for classification of the fixed test set during learning phase, and then final performance of the method obtained by classification of an unseen validation set by the selected training and features data. Besides, if the whole training data results a better classification, this algorithm automatically will use all the training data. Detail of the PSO-SVM algorithm is as follows.

Particle representation

As mentioned before, PSO algorithm is used in this paper for selecting best features subset and also best training data from the total training set. Also, size of selected features and selected training samples are determined by the proposed algorithm itself. Finally, parameters of SVM like as Type of SVM (C- SVM, nu-SVM and nu-SVM), kernel type, degree, gamma and cost are determined by the PSO.

Therefore, the particle is comprised of five parts, Size of the features subset, Size of the selected training data (integer-values), the features and training weights, which are real numbers between 0 and 1, and finally, five SVM parameters. Figure 5 shows the representation of first four parts of a sample particle with dimension of “ $nf + nt + 6$ ” where nf is total number of features and nt is total number of training data

Figure 5, represented a particle corresponding to the sample problem with 3 features and 4 training data, except the SVM parameters part. By attention to the first two cells, start from the left side; two features and three training data must be selected from original feature and training data sets. For selecting this features and training data, by attention to the importance of features, which represented in cells numbers three till five. Features 2 and 3 are selected due to their importance, and similarly, training data with number 1,2, and 4 must be selected.

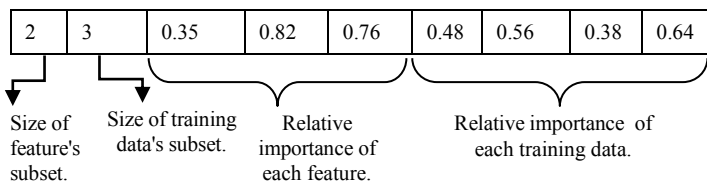


Figure 5. representation of one sample particle.

It must be noted that, integer PSO is not used for the first two cells. Actually, regular PSO used in this work and then obtained result rounded to the closest integer number for the integer parts.

Initialization

C_1 and c_2 of PSO are set to 0.2. Position and velocity of each particle generated randomly, by normal distribution. For example, number of selected training data is a random numbers between, size of training data/2, and size of training data. And

number of selected features is a random number between 3 and 20.

For the parallel structure, different settings are tested, like as 5 masters and 1 slave, 4 masters and 2 slaves and so on. Frequency of sending the best local particle sets to 5.

Particle evaluation

After determining features subset and training data subset corresponding to each particle, SVM used to classify testing data set by selected training data and features. Accuracy of this classification is used as the fitness of corresponding particle.

Support vector machines used in this study because of the primary advantage that is ability to minimize both structural and empirical risk leading to better generalization for new data classification[33]. This property of SVM is so important especially when we using optimization approach with a classifier that can be led to over fitting problem.

Parallel structure

In this work with consideration to the big size of search space, single PSO hasn't a good performance for this problem and lead to local optimum with low accuracy for classification. Therefore Parallel structure selected to perform better exploration in search space.

In this study a master-slave (multi swarms) structure with new cooperative strategy among swarms is introduced. In traditional master-slave model, swarms classified to one master and several slaves swarms[34]. But, in the proposed strategy, swarms classified to several masters and one (or more) slave. Master swarms have access to the best particle of others swarms but slaves ones have no access to others information's. Sending the best local partition information among masters and from slave(s) to masters can be perform in each iteration or after specified number of iterations. Pseudo code of the proposed multi swarm PSO is as follows.

Begin

Select the number of master and slave swarms and size of particles for each sub-warm. Select one of the master swarms as center.

Initialize the position and velocity of each particle

Do in parallel until the maximum number of iterations has reached {

Evaluate the fitness value of each particle

Find out the local best position in each sub-swarm

If meet sending condition

Sending the local best particle from each swarm to the center swarm.

End If

Updating global best particle in center swarm.

Sending the global best swarm to all of the master swarms.

Calculate the new velocity of each particle in each sub-swarm

Update the position of each particle in each swarm

End Do}

Return the best solution (the global best particle) of the algorithm

End

Figure 6 shows a sample of proposed parallel structure with 5 masters and 1 slave swarms. In this implementation, master 1 is selected as the center swarm, so all of the swarms send their local best particles to this swarm. After computing the global best swarm, it is send to all of the master swarm. So, in this structure slave swarm, provide information for others swarms, but not get profit from others information.

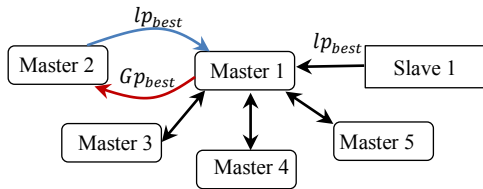


Figure 6. Proposed parallel structure with 5 masters and one slave.

Indeed fast convergence is one of the disadvantages of PSO which heightened in parallel structure. The proposed model tries to expand exploration and exploitation ability of PSO, by integrating isolated swarms with linked swarms together. Slave swarm(s) in this model help to prevent premature convergence and local optimums.

IV. RESULTS AND DISCUSSION

Experimental data containing two sets of data; the first one is 6 overnight sleep recordings were provided by concord hospital in Sydney, and the second set consist of 12 samples from [31] were both two sets events annotated by experts.

In the first step RUs generated for each sample, Table 2 and 3 tabulated total number of sleep apnea events, number of generated RUs and percentage of coverage for each sample. It must be noted that we consider an event as coverage with a RU if middle point of that event is in the RU and also at least 10 seconds of the event happened in that RU.

	SA events	Number of RUs	Coverage
#1	281	826	97.86
#2	63	831	100%
#3	241	796	98.76
#4	249	815	100%
#5	73	729	100%
#6	702	792	98.43

Table 2. RUs generated for the first set of data with 6 samples.

After generating of RUs, features corresponding to each RU are computed, by applying a 3 levels "Haar" wavelet packet, 295 features are generate as described before. These features used as input for the parallel PSO-SVM algorithm.

The parallel structure consists of 5 masters and one slave that implemented by MATLAB parallel toolbox. An SVM package, OSU-SVM used in this work.

	SA events	Number of RUs	Coverage
#1	238	528	94.53
#2	660	1110	97.27
#3	286	1072	100%
#4	4	683	100%
#5	664	920	91.86
#6	406	1075	98.52
#7	542	1239	97.78
#8	289	1091	100%
#9	451	1241	95.78
#10	553	981	95.29
#11	61	838	100%
#12	664	915	97.28

Table 3. RUs generated for the second set of data with 12 samples.

After training the system by train set sets and test sets, the whole system evaluated by the validate sets. Tables 4 and 5 summarize the performance of the system for two data sets. All automatically detected apnea events are comparing with manual annotations by medical expert and coded as a True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN), respectively. In this work after training the system on each sample the validation set is evaluated, in this way, Sensitivity and Specificity and Accuracy Values are calculated.

	TP	TN	FP	FN	Sens.	Spec.	Accu.	Elapsed Time (seconds)
#1	59	164	28	24	71.1	85.4	81	337
#2	11	255	7	4	73.3	97.3	96	117
#3	63	165	32	5	92.6	83.8	86	235
#4	60	153	42	16	79	78.5	78.6	779
#5	14	205	22	2	87.5	90.3	90.1	461
#6	196	35	23	10	95.1	60.3	87.5	540
Total	403	977	154	61	86.9	86.4	86.5	2469

Table 4. Results of the validation for apnea event detection for the first set.

	TP	TN	FP	FN	Sens.	Spec.	Accu.	Elapsed Time (seconds)
#1	78	73	14	11	87.6	83.9	85.8	177
#2	212	108	17	33	86.5	86.4	86.5	299
#3	117	124	18	8	93.6	87.3	90.3	247
#4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
#5	143	134	16	13	91.6	89.3	90.5	458
#6	126	186	23	23	84.5	89	87.1	419
#7	156	191	34	32	83	84.9	84	519
#8	75	256	26	16	82.4	90.8	88.7	365
#9	112	256	26	19	85.5	90.8	87.9	486
#10	161	138	16	12	93.1	89.6	91.4	286
#11	13	252	12	2	86.6	95.5	95	327
#12	154	112	12	27	85.1	90.3	87.2	369
Total	1342	1830	214	201	87.3	89.5	88.4	3952

Table 5. Results of the validation for apnea event detection for the second set.

The automatic detector system shown a good performance, obtaining a Sensitivity and accuracy of 86% for first set of data that shows the proposed system can detect efficiently both apnea and normal RUs. This high performance also achieved for second set of data. Also result of table 5 can be compared with [31], which used same data but by using one more input as oxygen saturation and achieved sensitivity of 87% and specificity of 89%, in total. To determine if there is significant

differences between the obtained results and result of [31] we performed two sample t -tests with the significance level chosen as $\alpha = 0.05$. T -test shows higher mean in all of the indexes; sensitivity, specificity and accuracy, even by using one less input signal, oxygen saturation, than [31]. As Table 5 shows performance indexes for sample 4 is not provided, and it is because this sample only contain 4 sleep apnic events, so after dividing this sample to three sets randomly, there is no event in training data, so this algorithm fails to predict them. But, this is not important because, generally, this algorithm and algorithms like this applied to samples that have sleep apnea and contain enough apnic events.

The optimization in training phase does no lead to over fitting in this work. Actually, if considering all the 12 and 6 samples together this algorithm achieved sensitivity, specificity and accuracy as 87.19, 88.40 and 87.93 for validation sets, respectively. These indexes are as 91.62, 92.56 and 92.20 for training phase, respectively, that are not so far from validation results.

The total processing time of this algorithm in parallel by the described structure is as 6421 seconds with accuracy of 87.93. We can consider the isolated slave swarm, as a sample of series implementation, which has process time equal to 6151 seconds and accuracy of 83.66; it can be shows that the proposed parallel structure helps us to improve the performance by acceptable extra time. For designing the parallel structure three different strategy as; S1: 1 slave and 5 Masters, S2: 2 slaves and 4 Masters and finally S3: 3 slaves and 3 masters are tested to find the best combination between masters and slaves swarms. Figure 6 shows accuracies for these different strategies for the first database with 6 samples. By attention to the accuracy for all of these six samples the first strategy selected which, also applied on second database.

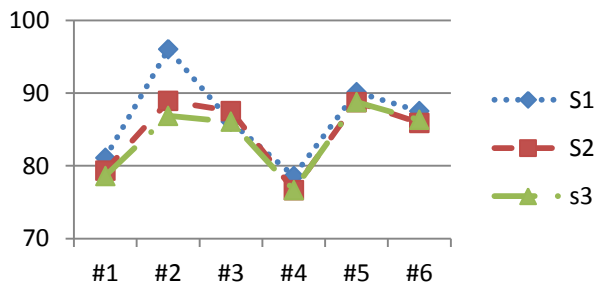


Figure 6. Accuracies for different combinations of slaves and masters swarms

V. Conclusion

In this paper, a novel Parallel PSO-SVM algorithm is proposed to detect sleep apnea. It works based three input signals (airflow, abdominal and throat movements) to the proposed algorithm. This algorithm consists of three main parts; signal segmentation, feature generation, and classification. In each part new algorithm is developed by attention to the nature of sleep apnea events.

In the signal segmentation phase, signals are segmented to non overlapped time windows named RUs. Each RU may contain at least one Sleep apnea event. This segmentation by smart

processing the signals helps to achieve better result in future generation phase versus blind segmentation. After segmentation, features are generated by wavelet packet coefficients and also original signals. Finally, a parallel PSO-SVM classified RUs as sleep apnea or normal. The unique structure of the parallel algorithms helps PSO to have both exploration and exploitation together, by use of connected and isolated swarms.

Experimental results on real data shows high accuracy of this algorithm. Also, by comparing with results of [31] the proposed approach performs advantageously. It must be considered that [31] used one more signal as input, Oxygen saturation, than this work. Using less one signal means more comfortable to subject during recording the data in sleeping time.

Appendix A.

Mean, variance (VAR) and standard deviation (STD) are common and well known statistical tools, so other statistical features, in this study, are reviewed here.

Kurtosis: The kurtosis of a distribution is a measure of how outlier-prone a distribution is, and defined as follow

$$k = \frac{E(x - \mu)^4}{\sigma^4}$$

Geomean: Geomean is the geometric mean and computed as follow:

$$m = \left[\prod_{i=1}^n x_i \right]^{1/n}$$

Skewness: Skewness is a measure of the asymmetry of the data around the sample mean, and defined as follow

$$s = \frac{E(x - \mu)^3}{\sigma^3}$$

Mad: mad is mean absolute deviation of the sample as, $mean(|(x - mean(x))|)$.

REFERENCES

- [1] Klink, M. and S.F. Quan, Prevalence of reported sleep disturbances in a general adult-population and their relationship to obstructive airways diseases. *Chest*, 1987. **91**(4): p. 540-546.
- [2] Chokroverty, S., et al., Approach to the Patient with Sleep Complaints, in *Sleep Disorders Medicine (Third Edition)*2009, W.B. Saunders: Philadelphia. p. 255-274.
- [3] Young, T.B., Epidemiology of daytime sleepiness: Definitions, symptomatology, and prevalence. *Journal of Clinical Psychiatry*, 2004. **65**: p. 12-16.
- [4] Chokroverty, S., et al., *Sleep deprivation and sleepiness*, in *Sleep Disorders Medicine (Third Edition)*2009, W.B. Saunders: Philadelphia. p. 22-28.
- [5] Chokroverty, S., *Overview of sleep & sleep disorders*. *Indian Journal of Medical Research*, 2010. **131**(2): p. 126-140.
- [6] Ball, E.M., et al., Diagnosis and treatment of sleep apnea within the community - The Walla Walla project. *Archives of Internal Medicine*, 1997. **157**(4): p. 419-424.
- [7] Kryger, M.H., et al., Utilization of health care services in patients with severe obstructive sleep apnea. *Sleep*, 1996. **19**(9): p. S111-S116.
- [8] Stradling, J.R. and J.H. Crosby, Relation between systemic hypertension and sleep hypoxemia or snoring- analysis in 748 men drawn from general-practice. *British Medical Journal*, 1990. **300**(6717): p. 75-78.

- [9] Hoffstein, V., *Snoring*, in *Principles and practice of sleep medicine*, M.H. Kryger, T. Roth, and D.W. C., Editors. 2000, Saunders: Philadelphia, PA: W.B. p. 813-826.
- [10] Penzel, T., et al., Systematic comparison of different algorithms for apnoea detection based on electrocardiogram recordings. *Medical & Biological Engineering & Computing*, 2002. **40**(4): p. 402-407.
- [11] Kryger, M.H., *management of obstructive sleep-apnea*. *Clinics in Chest Medicine*, 1992. **13**(3): p. 481-492.
- [12] Flemons, W.W., et al., Sleep-related breathing disorders in adults: Recommendations for syndrome definition and measurement techniques in clinical research. *Sleep*, 1999. **22**(5): p. 667-689.
- [13] Cabrero-Canosa, M., E. Hernandez-Pereira, and V. Moret-Bonillo, *Intelligent diagnosis of sleep apnea syndrome*. *Ieee Engineering in Medicine and Biology Magazine*, 2004. **23**(2): p. 72-81.
- [14] Cabrero-Canosa, M., et al., *An intelligent system for the detection and interpretation of sleep apneas*. *Expert Systems with Applications*, 2003. **24**(4): p. 335-349.
- [15] de Chazal, P., et al., Automated processing of the single-lead electrocardiogram for the detection of obstructive sleep apnoea. *Ieee Transactions on Biomedical Engineering*, 2003. **50**(6): p. 686-696.
- [16] Maali, Y. and A. Al-Jumaily, Genetic Fuzzy Approach for detecting Sleep Apnea/Hypopnea Syndrome. 2011 3rd International Conference on Machine Learning and Computing (ICMLC 2011), 2011.
- [17] Vapnik, V., *Statistical Learning* 1998: Wiley.
- [18] Kennedy, J. and R. Eberhart, *Particle swarm optimization*. In *Proceedings of the IEEE International Conference on Neural Networks*, ed. I.P.o.t.I.I.C.o.N. Networks. Vol. 4. 1995, Perth.
- [19] Rana, S., S. Jasola, and R. Kumar, *A review on particle swarm optimization algorithms and their applications to data clustering*. *Artificial Intelligence Review*, 2011. **35**(3): p. 211-222.
- [20] Thangaraj, R., et al., *Particle swarm optimization: Hybridization perspectives and experimental illustrations*. *Applied Mathematics and Computation*, 2011. **217**(12): p. 5208-5226.
- [21] Shi, Y.H., R. Eberhart, and Ieee, *A modified particle swarm optimizer*. 1998 Ieee International Conference on Evolutionary Computation - Proceedings 1998. 69-73.
- [22] Shi, Y. and R.C. Eberhart, Parameter Selection in Particle Swarm Optimization, in *Proceedings of the 7th International Conference on Evolutionary Programming VIII 1998*, Springer-Verlag.
- [23] Cui, S.M. and D.S. Weile, *Application of a parallel particle swarm optimization scheme to the design of electromagnetic absorbers*. *Ieee Transactions on Antennas and Propagation*, 2005. **53**(11): p. 3616-3624.
- [24] Herrera, F., M. Lozano, and C. Moraga, *Hierarchical distributed genetic algorithms*. *International Journal of Intelligent Systems*, 1999. **14**(11): p. 1099-1121.
- [25] Kalivarapu, V., J.L. Foo, and E. Winer, *Synchronous parallelization of Particle Swarm Optimization with digital pheromones*. *Advances in Engineering Software*, 2009. **40**(10): p. 975-985.
- [26] Wu, D.F., et al., Prediction of Parkinson's disease tremor onset using a radial basis function neural network based on particle swarm optimization. *International Journal of Neural Systems*, 2010. **20**(2): p. 109-116.
- [27] Chang, J.F., et al., *A parallel particle swarm optimization algorithm with communication strategies*. *Journal of Information Science and Engineering*, 2005. **21**(4): p. 809-818.
- [28] Waintraub, M., R. Schirru, and C. Pereira, Multiprocessor modeling of parallel Particle Swarm Optimization applied to nuclear engineering problems. *Progress in Nuclear Energy*, 2009. **51**(6-7): p. 680-688.
- [29] Zhang, J.Z. and X.M. Ding, *A Multi-Swarm Self-Adaptive and Cooperative Particle Swarm Optimization*. *Engineering Applications of Artificial Intelligence*, 2011. **24**(6): p. 958-967.
- [30] Madhuri, M. and K. Deep, *A State-of-the-art Review of Population-based Parallel Meta-heuristics*. 2009 World Congress on Nature & Biologically Inspired Computing, ed. A. Abraham, et al. 2009. 1603-1606.
- [31] Alvarez-Estevéz, D. and V. Moret-Bonillo, *Fuzzy reasoning used to detect apneic events in the sleep apnea-hypopnea syndrome*. *Expert Systems with Applications*, 2009. **36**(4): p. 7778-7785.
- [32] Hernandez-Pereira, E., et al., *An amplitude signal based technique for hypopneas detection*. *Proceedings of the 11th Iasted International Conference on Artificial Intelligence and Soft Computing*, ed. A.P. DelPobil 2007. 127-130.
- [33] Khandoker, A.H., M. Palaniswami, and C.K. Karmakar, *Support Vector Machines for Automated Recognition of Obstructive Sleep Apnea Syndrome From ECG Recordings*. *Ieee Transactions on Information Technology in Biomedicine*, 2009. **13**(1): p. 37-48.
- [34] Fan, S.K.S. and J.M. Chang, Dynamic multi-swarm particle swarm optimizer using parallel PC cluster systems for global optimization of large-scale multimodal functions. *Engineering Optimization*, 2010. **42**(5): p. 431-451.