On the Parallelization and Optimization of the Genetic-Based ANN Classifier for the Diagnosis of Students with Learning Disabilities

T-K Wu, S-C Huang, Y-L Lin, H. Chang
National Changhua University of Education
Changhua City, Taiwan
tkwu@im.ncue.edu.tw

Y-R Meng
National HsinChu University of Education
HsinChu City, Taiwan
myr321@mail.nhcue.edu.tw

***Abstract*—Diagnosis of students with learning disabilities has long been a difficult issue as it requires extensive man power and takes a long time. Through genetic algorithm based feature selection method and genetic based parameters optimization, artificial neural network (ANN) classifier has proven to be a good predictor to the diagnosis of students with learning disabilities. In this study, we keep focusing on the ANN model and compare three strategies of parallelizing the ANN parameter optimization procedure with OpenMP and MPI APIs. Not surprisingly, the outcomes show that all three parameter optimization procedures indeed converged or executed more quickly with the aid of parallel processing. In particular, the genetic-based method tends to derive the best accuracy and require less execution time. Most important of all, potentially due to a more diverse search space provided by the distributed parallel processing environment, the accuracy of the genetic-based ANN classifier may also be improved in general. In addition, with appropriate combinations of features and parameters setting, the accuracy in LD identification model has exceeded the 90% mark (using 5-fold cross validation), which is the best we have achieved so far. The result suggests that genetic-based (or perhaps similar) optimization methods may be benefited, both in reducing execution time and achieving better outcome, from current grid-based computing technologies.**

***Keywords*—Learning Disabilities, Genetic Algorithm, Evolutionary Computation, Grid Computing, Neural Network**

I. Introduction

The term "learning disabilities" (LD) was first used in 1963 [1]. However, experts in this field have not yet completely reach an agreement on the definition of LDs and its exact meaning [2]. According to definition given by the United States National Center for Learning Disabilities [3], a learning disability is:

"a neurological disorder that affects the brain's ability to receive, process, store, and respond to information. The term learning disability is used to describe the seeming unexplained difficulty a person of at least average intelligence has in acquiring basic academic skills. These skills are essential for success at school and work, and for coping with life in general. LD is not a single disorder. It is a term that refers to a group of disorders."

As a result, a person can be of average or above average intelligence, without having any major sensory problems (like blindness or hearing impairment), and yet struggle to keep up with people of the same age in learning and regular functioning.

Due to the implicit characteristics of learning disabilities as stated above, the identification of students with LDs has long been a difficult and time-consuming process. In the United States, the so called "Discrepancy Model" [4], which states that a severe discrepancy between intellectual ability and academic achievement has to exist in one or more of these academic areas: (1) oral expression, (2) listening comprehension (3) written expression (4) basic reading skills (5) reading comprehension (6) mathematics calculation, is one of the commonly adopted criteria to evaluate whether a student is eligible for special education services.

In Taiwan, the diagnosis procedure pretty much follows the "Discrepancy Model" and is roughly separated into 4 stages: (1) application for screening of potential students with LDs by parents, general education teachers and/or junior-level evaluation personnel, (2) identification of potential students with LDs by junior-level evaluation personnel, (3) diagnosis of possible students with LDs by senior-level evaluation personnel, and (4) final confirmation by special education specialists (usually college or university professors with LD major) [5]. Note, junior-level or senior-level evaluation personnel is selected special education teachers with days' (junior level) or weeks' (senior level) training on LD diagnosis related procedure.

The sources of input parameters required in such prolonged process include information from parents, general education teachers, students' academic performance and a number of standard achievement and IQ tests. To guarantee collection of required information regarding to students suspected with LD, usually checklists of some kind are developed to assist parents and regular education teachers. The Learning Characteristics Checklists (LCC), a Taiwan locally developed LD screening checklist [6], is commonly used in most counties of Taiwan. Among the standard tests, the Wechsler Intelligence Scale for Children, Third or Fourth Edition (WISC III or IV) plays the most important role in the third and fourth stages of the current LD diagnosis model. The WISC-III is composed of 13 sub tests

This work was supported in part by the National Science Council of Taiwan under Grant NSC 98-2511-S-018-014-MY2.

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

[7]. The scores of the sub-tests are then used to derive 3 IQs, which include Full scale IQ (FIQ), Verbal IQ (VIQ), Performance IQ (PIQ), and 4 indexes, which include Verbal Comprehension Index (VCI), Perceptual Organization Index (POI), Freedom from Distractibility Index (FDI), Processing Speed Index (PSI). There are also a number of locally developed standard achievement tests (AT), which typical consists of reading, math, and fields that related to students' academic achievement.

Diagnosis of students with LDs then involves mainly interpreting the standard test scores and comparing them to the norms that are derived from statistical method. As an example, in case the difference between VIQ and PIQ is greater than 15, representing significant discrepancy between a student's cultural knowledge, verbal ability, etc, and his/her ability in recognizing familiar items, interpreting action as depicted by pictures, etc, is a strong indicator in differentiating between students with or without LD [7]. A number of similar indicators together with the students' academic records and descriptive data (if there is any) are then used as the basis for the final decision (by senior evaluation personnel and special education specialists). Confirmed possible LD students are then evaluated for one year before admitting to special education. However, it deserves to note that a previous study in Taiwan reveals that the certainty in predicting whether a student is having a LD using each one of the currently available predictors is in fact less than 50% [8].

As we can see, the above procedure involves extensive manpower (mainly the overloaded special education teachers) and resources. In addition, the diagnosis process requires that the special education teachers having a strong background in both psychology and statistics. Unfortunately, those were not commonly included in their training at the college level. Furthermore, a lack of nationally regulated standard for the LD diagnosis procedure and criteria result in possible variations on the outcomes of diagnosis. In most cases, the difference can be quite significant [5]. Accordingly, the quality of interpretation varied and the pressure is primarily on the special education specialists at the final stage.

With the advance in artificial intelligence (AI) and its successful applications to various classification problems, it is interesting to investigate how these AI-based techniques perform in identifying students with LDs. In our previous study, we made attempts in adopting two well-known artificial intelligence techniques, artificial neural network (ANN) and support vector machine (SVM), together with various feature selection algorithms and evolutionary computation, to the LD diagnosis problem [5, 9]. The results show that ANN classifier does well in positively identifying students with LDs. In subsequent studies, we combined various feature selection techniques and genetic-based parameters optimization with the ANN classifier, which further improves the overall identification accuracy [10, 11]. However, although ANN-based classifier performs well in LD diagnosis problem, the procedure is computation-intensive and may take a while to proceed. The situation is even more difficult when ANN is wrapped with genetic-based algorithm for parameter optimization. Accordingly, grid-based parallel computing has been used to speedup the ANN training and validation procedure [12].

In this paper, we still focus on the ANN classification model and try to further explore the potential limit that ANN classifier can achieve through the use of parallelized genetic algorithm based feature selection procedure and parameters optimization algorithm. Our codes are modified so that they can take full advantage of current multi-core processor technology and grid-based distributed parallel computing environment. The outcomes show that parallel processing not only reduce the ANN training time substantially, but also improve the correct identification rate in LD identification.

This rest of the paper is organized as follows. Section 2 briefly describes history of AI techniques on the special education applications. Section 3 and 4 presents the experiment settings, design and corresponding results. Finally, Section 5 gives a summary of the paper and lists some issues that deserve further investigation.

II. RELATED WORK

Artificial intelligence techniques have long been applied to special education. However, most attempts occurred in more than one or two decades ago and mainly focused on using the expert systems to assist special education in various ways [5].

There were also numerous classification techniques other than neural network that have been developed and widely used in various applications [13]. For a classification problem, it is necessary to first try to estimate a function $f: R^N \rightarrow \{\pm 1\}$ using training data, which are l N-dimensional patterns $\mathbf{x}_i$ and class labels y_i, where

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \in R^N \times \{\pm 1\} \tag{1}$$

such that f will classify new samples ($\mathbf{x}, y$) correctly.

Among all the classification techniques, artificial neural network (ANN) has received lots of attentions due to their demonstrated performance and has gained widely acceptance beginning from the 1990s [14]. An artificial neural network is a mathematical representation that is inspired by the way the brain process information. Many types of ANN models have been suggested in the literatures, with the most popular one for classification being the multilayer perceptron (MLP) with back propagation. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to predict the outcome when the desired output is unknown. MLP with back propagation is typically composed of an input layer, one or more hidden layers and an output layer, each consisting of several neurons. Each neuron processes its inputs and generates one output value that is transmitted to the neurons in the subsequent layer.

Figure 1 provides an example of an MLP with one hidden layer and one output neuron. The output of i-th hidden neuron is then computed by processing the weighted inputs and its bias term b_i as follows:

$$h_i = f^h\left(b_i + \sum_{j=1}^{n} w_{ij} x_j\right) \quad (2)$$

where w_{ij} denotes the weight connecting input x_j to hidden unit h_i.

Similarly, the output of the output layer is computed as follows:

$$y = f^{output}\left(b + \sum_{j=1}^{n} w_j x_j\right) \quad (3)$$

with n being the number of hidden neurons and w_j represents the weight connecting hidden unit j to the output neuron. A threshold function is then applied to map the network output y to a classification label. The transfer functions f^h and f^{output} allow the network to model non-linear relationships in the data. Also note that the number of hidden layer nodes does not need to be the same as the number of input nodes.

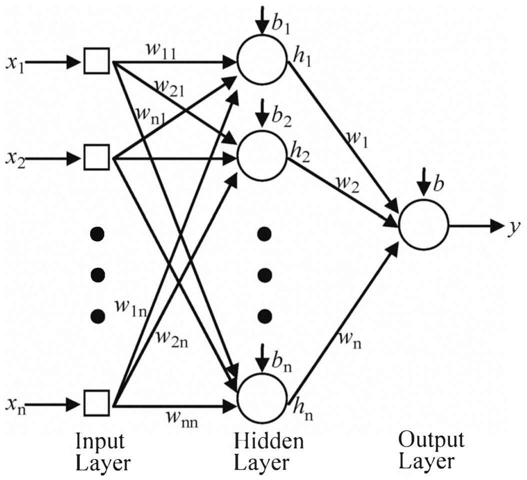

Figure 1. MLP with one hidden layer.

The applications of ANN range from signal processing in communications to pattern recognition in business, engineering and medicine [15]. In the field of special education, ANN has been used as an arithmetic training tool for the children with learning disabilities [16] and for prediction of successful or unsuccessful completion of special education programming for students diagnosed with SED (Serious Emotional Disturbance) with up to 64% of accuracy [17]. In addition, ANN has also been applied to the diagnosis of autism with very high correct identification rate [18]. However, the autism diagnosis is not as complicated as the LD diagnosis problem.

There are also a number of researches indicate that application of data dimensionality reduction pre-processing step prior to the classification procedure does improve the overall classification performance [19, 20]. Furthermore, feature selection can also provide a better understanding of the underlying process that generated the data [21]. Feature selection approaches can be categorized into filter and wrapper based. For wrapper approach, feature selection is wrapped around the classification algorithm and the result of classification is used as the evaluation criterion to guide the search of optimal features set. Although computational expensive for larger data set, wrapper approach may perform better in finding useful subsets of relevant variables [21].

In addition, genetic algorithms have been applied to a number of optimization problems [22]. For example, combining genetic algorithm with the above mentioned feature selection procedure has been shown to be quite effective in searching for the optimal feature subsets [20]. Genetic algorithm has also been used to train the ANN classification model and construct the structure of networks [23].

However, the ANN model training and genetic algorithms based procedures may require extensive computation and take quite a long time to process [11]. Fortunately, researches have shown parallel processing may provide affordable computational power to speedup the time-consuming process [12, 24]. For network connected cluster or grid environment, message passing interface (MPI) is usually used to coordinate computing nodes for completing a common task. On the other hand, to take full advantage of the currently available multi-core processor technology, OpenMP may be used explicitly to direct multi-threaded, shared memory parallelism.

III. Data Sets and Experimental Settings

Our objective in this study is to explore how the distributed (and multi-core) parallel processing environment, like grid-computing, may improve performance of the genetic-based parameter optimization in terms of processing speed and accuracy in constructing a ANN-based LD identification model. The data sets, as listed in Table I, we use are collected from three counties in the northern and southern Taiwan, all have been tested in previous studies [5, 9-12]. All data samples in Table I are pre-processed with a normalization procedure as depicted in equation (2).

$$\frac{L - L_{\min}}{L_{\max} - L_{\min}} \quad (2)$$

where L is the value of the original data sample to be normalized and $(L_{\min}, L_{\max})$ are the minimum and maximum values of samples in that particular feature.

In addition, a wrapper-based genetic feature selection procedure combining SVM learner (with RBF kernel and C-SVC SVM type, as used in [5]) is applied to the three data sets. Note that SVM learner is chosen because it is known to be able to minimize the structured risk while separating the two classes [25], which may result in models that have better generalization capability. For the genetic-based feature selection procedure, we use binary chromosome encoding with one-point crossover and tournament selection. Probabilities of crossover and mutation are 0.8 and 0.1, respectively. The size of population and generation are set at thirty and fifty, respectively. In addition, accuracy in classification is used to evaluate the fitness of populations. The above feature selection procedure is performed fifty times and then output the feature set that results in the best accuracy in LD identification. The selected features for the three data sets are listed in Table II.

TABLE I. DATA SETS AND THEIR FEATURES USED IN THIS STUDY

	sample size	percentage of students with LDs	feature×size*
data set 1	652	22.7%	WISC-III×7 、WISC-III×13
data set 2	125	19.5%	WISC-III×7 、LCC×6 、AT×3
data set 3	159	47.8%	WISC-III×7 、LCC×6 、AT×3

*WISC-III×7 includes three IQ scores and 4 indexes, while WISC-III×13 includes the 13 WISC subtests. Please refer to [7] for further details. LCC represents learning characteristics checklist, which include LCC overall index and indexes A~E. Please refer to [6] regarding details of LCC. AT represents achievement test, for dataset 2 it includes Word Recognition (WR), Reading and Math sub-tests, while for dataset 3 it includes Chinese, English and Math sub-tests.

TABLE II. SELECTED FEATURES FOR SEBSEQUENT EXPERIMENTS

data set	features
1	VIQ, PIQ, FIQ, VCI, POI, FDI, PSI (WISC-III×7)
2	Math, VIQ, PIQ, FIQ, VCI, FDI, PSI
3	Chinese, Math, LCC-B, LCC-C, LCC-E, LCC-T, PIQ, POI, FDI, PSI

Three strategies for constructing a better ANN classification model are experimented and compared, which include parallel exhaustive search, parallel genetic-based search, and one combining the both. In all three approaches, the number of hidden layer, number of input and output neurons are set to one, number of features, and one, respectively.

Table III lists pseudo code (in sequential form) for the exhaustive search strategy. We iterate the number of hidden nodes (between 1 and 26), momentum (M=0.0→1.0, step 0.1), and learning rate (L=0.0→1.0, step 0.1) at the first stage. In subsequent stages, we narrow down the range of momentum and learning rate (controlled by the parameter *depth*) of the search process. For example, if *depth*=1 and L=0.5 achieves the best accuracy in stage one, L would then be varied in between 0.45 and 0.55 (L =0.45→0.55, step 0.01) in stage 2.

With genetic-based search approach, the three parameters (number of hidden nodes, learning rate and momentum) are real-value encoded into the chromosome. Note, real-value encoding is chosen as it performs slightly better in terms of speed and accuracy as compared to binary encoding in our cases. In addition, random number seeds are also encoded in the chromosome since it may affect the initial weights and bias of neural network. Unless otherwise specify, the population (number of chromosomes) and generation are set to twenty and fifty. For combination of the above two approaches, we replace stage two of Table III with genetic-based search approach and bound the search space of the number of hidden nodes, learning rate and momentum to [1, 26], [L_0-0.05, L_0+0.05] and [M_0-0.05, M_0+0.05], respectively (L_0 and M_0 are learning rate and momentum that achieve the best accuracy in stage one). In each stage of all the three strategies, the number of epoch in ANN model training is fixed at 500.

TABLE III. EXHAUSTIVE PARAMETERS SEARCH PROCEDURE

For *data-set* = {data set 1, data set 2, data set 3}
Stage 1:
  For L = 0 to 1.0 step 0.1
    For M = 0 to 1.0 step 0.1
      For H = 1 to 26 step 1
        With *data-set*, Perform 5-fold cross validation
      Performance evaluation & Store the result;
  Output best CIR, and L_0 / M_0 that achieve the best CIR
Stage 2:
  For (D = 1 to depth)
    For H = 1 to 26 step 1
      For $L = (L_0 - \frac{1}{2\times10^D})$ to $(L_0 + \frac{1}{2\times10^D})$ step $(\frac{1}{10^{D+1}})$
        For $M = (M_0 - \frac{1}{2\times10^D})$ to $(M_0 + \frac{1}{2\times10^D})$ step $(\frac{1}{10^{D+1}})$
          With *data-set*, perform 5-fold cross-validation & store the result;
      Output best CIR, *and* L_D / M_D that achieve the best CIR

A mini-grid environment containing 11 nodes, include 1 server node and 10 computing nodes (as shown in Table IV) running Linux operating system (Fedora 10) connected via local area network (see Figure 2), is established with the aid of Globus toolkit version 4.2.1 [26] for the experiments.

TABLE IV. NODE DETAILS OF THE MINI-GRID IN OUR STUDY

node ID	CPU	memory
Server	Intel i7 960 Quad Core CPU 2.67GHz	8GB
node 1~4	AMD Athlon 64X2 Dual Core Processor 5600+	2GB
node 5~8	Intel Core 2 Duo CPU E8400 3.00GHz	2GB
node 9	AMD Athlon 64X2 Dual Core Processor 5000+	1GB
node 10	Intel Core 2 Dual CPU 6300 1.86GHz	2GB

Grid-enabled Message Passing Interface (MPICH-G2) [27] is adopted to coordinate computation among the computing resources. OpenMP APIs [28] are also used to explore the current multi-core processor technology within each individual node. For exhaustive parameter search strategy, the computation loads are statically and evenly distributed among the computing nodes and processing cores with *depth* set to 1 (please refer to Table III). For genetic-based parameter search, we adopt the asynchronous Parallel Distributed Genetic Algorithm (PDGA) proposed in [29] as it has been proven to be effective and easy to implement.

Accordingly, the parallelized genetic-based ANN parameter search procedure will be named PDGA-ANN hereafter.

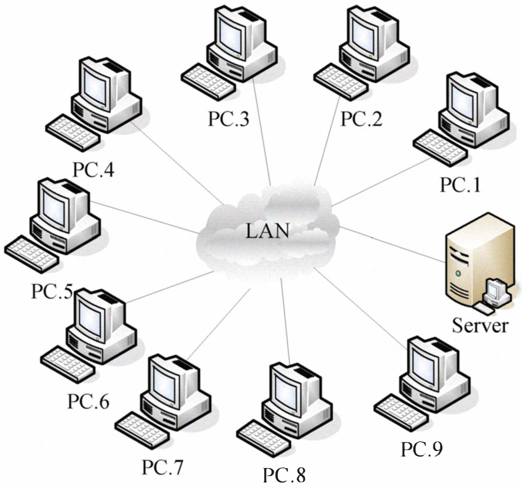

Figure 2. LAN-based mini-grid environment.

IV. Experiment Design and Results

We designed and conducted four experiments as listed follow. To evaluate the experiment outcomes, a performance index: correct identification rate (CIR) is defined as follows.

$$CIR = \frac{(\text{number of correct LD and non-LD identification})}{(\text{total number of cases})} \quad (3)$$

In the first experiment, we compare the three parallelized ANN classification parameter search strategies (exhaustive, genetic-based, and combination of the above two) described earlier in Section III in terms of performance in execution time and CIR on each of the three available datasets. The outcomes of experiment 1 are shown in Table V. Note, all the numbers shown represent the average of twenty consecutive runs on a 5-node (node 1~4 and node 9) mini-grid computing environment.

TABLE V. Performance comparison on three parameters search strategies (All time in seconds)

data set	1		2		3	
strategy	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*
exhaustive	87.7% (88.1%)	3378	87.2% (88.8%)	1563	87.3% (88.7%)	2163
PDGA	87.5% (88.1%)	**2315**	**87.5% (89.6%)**	**703**	87.0% **(88.8%)**	**1390**
exhaustive + PDGA	**87.8% (88.4%)**	4931	**87.5% (89.6%)**	1623	**87.5%** (88.7%)	2521

According to the results, PDGA approach performs much faster than the other two. For comparison, in an extreme scenario, the serial version of the two-stage exhaustive strategy (with *depth* equals 1) takes slightly more than 11,000 seconds on node 1. On the other hand, it is the combination of the exhaustive and PDGA approach that results in the best average CIR in all cases. However, for best CIR, it appears that PDGA won in most cases. The exhaustive + PDGA strategy, unlike its serial implementation that performs quite well in term of CIR [11], only achieves slightly better result in data set 1. However, it may take more than twice the execution time as compared to the PDGA approach. It appears that parallel processing may contribute to more randomized or diverse search space (since there are more distributed nodes involved) and thus has better chance in converging early and finding parameters that result in decent CIRs.

In subsequent two experiments, our focus is on the genetic-based ANN parameter search approaches. However, in the former (experiment 2), we port the code onto a single workstation equipped with a four-core CPU (the server in Table IV). OpenMP APIs are used to multi-thread the most time-consuming fitness function computation, which would be a five-fold cross validation involving numerous ANN model constructions and verifications in our case. A simple static scheduling that evenly assigns population to the available threads (cores) is adopted. In experiment 3, we further parallelize the multi-threaded genetic-based parameter search procedure with MPI and execute it on the mini-grid nodes (the ten computational nodes in Table IV). The objectives of the two experiments is to assess how the CPU-level and grid-level parallelism affect the performance, both in speeding up the process and improving the CIR, of the ANN-based LD classifier. The results of experiment 2 and 3 are shown in Table VI and VII respectively. All numbers in Table VI and VII are average of results from twenty consecutive runs on the parallelized codes. Note, "1(100)" in Table VII represents the genetic-based ANN procedure with population size equals 100 is executed in 1 node, while "5(20)" represents that similar computation load is distributed among 5 nodes with each node being responsible for only twenty chromosomes. In other words, 5(20) is the parallelized counterpart of 1(100) with the same number of population and should be compared directly. The same applies to 1(200) and 10(20).

TABLE VI. Performance comparison under multi-core environment (All time in seconds)

data set	1		2		3	
cores (threads)	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*
1	**87.2% (87.9%)**	1966	83.8% (86.4%)	430	86.4% (87.5%)	823
2	87.0% (87.5%)	822	**84.2%** (86.4%)	211	86.3% (88.1%)	367
3	86.9% (87.6%)	733	84.1% **(88.0%)**	159	86.4% **(88.7%)**	272
4	86.8% (87.2%)	522	**84.2%** (87.2%)	124	**86.8%** (88.1%)	192

Based upon the results shown in Table VI, the genetic-based ANN LD classifier does improve in term of execution time through the use of current multi-core processor technology. However, it is not the case for CIR in LD identification. Either the best or average CIR does not seem to relate to the number of threads (cores) used. The outcome appears to be reasonable since our code just distributes the overall populations onto available threads (cores) without affecting the random number seed at the initialization phase. Accordingly, it may worth trying to develop schemes so that each core executes the search procedure in quite different direction by modifying the initialization step.

TABLE VII. PERFORMANCE COMPARISON BETWEEN SINGLE NODE AND GRID-BASED PARALLEL PROCESSING ENVIRONMENT (ALL TIME IN SECONDS)

Data Set	**1**		**2**		**3**	
# of nodes (population)	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*	*average (best) CIR*	*average exec. time*
1 (100)	87.3% (87.7%)	7194	84.6% (87.2%)	1984	86.8% (88.6%)	3434
5 (20)	87.5% (87.8%)	1958	85.4% (86.4%)	844	**87.5% (89.3%)**	1119
1 (200)	87.5% (88.0%)	14172	84.9% (86.4%)	3629	87.0% (88.7%)	6913
10 (20)	**87.6% (88.2%)**	1967	**86.6% (88.8%)**	701	**87.5%** (88.8%)	1228

On the other hand, results in Table VII show that both execution time and (best/average) CIR are improved under the grid-based parallel processing environment. It is not surprising that the parallelized version performs better in term of execution time. However, it is interesting that genetic-like optimization algorithms may seem to potentially benefit from parallelism (and diverseness in search space) provided by the distributed environment. To further verify this point, a cross campus grid environment with more computing nodes is being built as a test-bed. Note that the performance in term of speedup of the gird-based parallel implementation does not seem to be good enough. The issue is related to the inconsistent computational power of the nodes and scheduling schemes for the computation load, which should be addressed in the future.

Finally, in experiment 4 we parallelize the feature selection procedure described earlier using the asynchronous Parallel Distributed Genetic Algorithm (PDGA) (and thus named PDGA-SVM). Each selected feature set will then be validated multiple times (thirty times in our experiment) using PDGA-ANN. Note, five-fold cross validation procedure with linear sampling is adopted in the PDGA-ANN phase (as opposed to a simple validation procedure, which depends merely on a single split of data, used in [5]). The above procedure is repeated twenty times. The objective of this experiment is to explore the potentially "best" CIR (without concerning the execution time) that may be achieved for LD identification. Table VIII presents the procedure in program-like format. The outcomes are shown in Table IX.

Note that the best feature combination of data set 1 remains the 7 WISC-III IQs and indexes. Also note that the CIRs derived in Table IX are the best that we have got so far with any given data set (using 5-fold cross validation), which shows that the combined genetic-based distributed parallel computing procedure is indeed effective in finding the "optimal" parameters settings, which then achieves the better than ever results. The comparison bewteen this study and previous ones are depicted in figure 3.

TABLE VIII. PARALLELIZED EXHAUSTIVE PARAMETERS SEARCH PROCEDURE

For *data-set* = {data set 1, data set 2, data set 3}
  For *i*=1 to 20
    Apply PDGA-SVM Feature Selection Algorithm on *data-set* and output the *feature-set* with best CIR
    For *j*= 1 to 30
      Perform PDGA-ANN on *data-set* with *feature-set* and record the best CIR in each run;
  Output the best CIR

TABLE IX. BEST CIRS IN THE TRHEE DATA SETS AND THE CORRESPONDING FEATURE SETS.

data set	*best CIR*	*features*
1	88.63%	VIQ, PIQ, FIQ, VCI, POI, FDI, PSI
2	90.40%	WR, Reading, Math, LCC-B, LCC-D, VIQ, PIQ, FSIQ, POI, PSI
3	90.63%	Math, LCC-A, VIQ, PIQ, VCI, POI, FDI, PSI

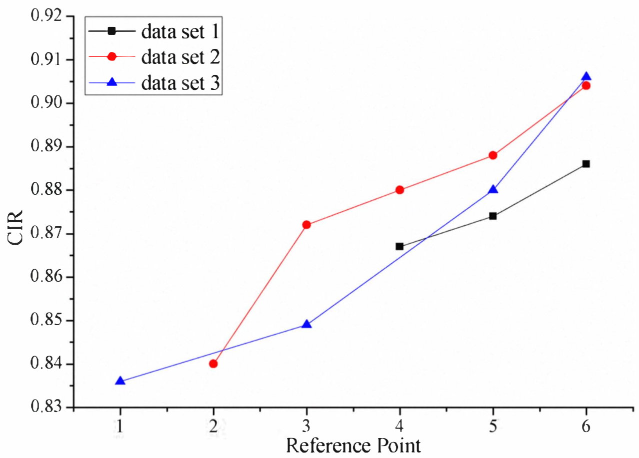

Figure 3. Comparison of results derived in this (reference point 6) and earlier studies (results in reference points 1~5 are published in [9], [10], [5], [11], and [12], respectively). All CIRs are results of five-fold cross validation with linear sampling.

SUMMARY AND FUTURE WORK

In this study, we keep working on improving the ANN classification model for diagnosing students with learning disabilities. We parallelize the genetic based feature selection procedures and ANN parameters optimization algorithms so that they can take full advantage of current multi-core processor technology and grid-based distributed computing environment. The outcomes show that parallel processing not only reduce the ANN training time substantially, but may also improve the correct identification rate for the LD identification. As a matter of fact, the best CIRs (with 5-fold cross validation) we got on data set 2 and 3 in this study have exceeded 90%, which are the highest ever so far.

In addition, it also appears that the number of computing nodes of a distributed parallel processing environment may have positive effect on the genetic-based optimization process. More experiments may be required to further verify this observation, which will be one of our future research topics.

Finally, our grid-based parallelized codes have not yet taken into consideration the load balancing issue. As a result, the overall execution time may depend on some less capable machine or some slow converging process. Accordingly, addressing this load balancing issue should be our other concern in the future.

REFERENCES

[1] S. A. Kirk, "Behavioral Diagnosis and Remediation of Learning Disabilities," Proceedings of the Conference on the Exploration into the Problems of the Perceptually Handicapped Child, 1963.

[2] J. M. Fletcher, W. A. Coulter, D. J. Reschly, and S. Vaughn, "Alternative Approach to the Definition and Identification of Learning Disabilities: Some Questions and Answers," Annals of Dyslexia, vol. 54, No. 2, pp. 304-331, 2004.

[3] National Center for Learning Disabilities, http://ncld.org/ld-basics/ ld-explained/basic-facts/what-are-learning-disabilities, extracted on Nov. 16th, 2009.

[4] J. Schrag, "Discrepancy Approaches for Identifying Learning Disabilities," Alexandria, VA: National Association of State Directors of Special Education, 2000.

[5] T.-K. Wu, S.-C. Hwang and Y.-R. Meng, "Evaluation of ANN and SVM Classifiers as Predictors to the Diagnosis of Students with Learning Disabilities," Expert Systems with Applications, vol. 34, no. 3, pp. 1846-1856, April 2008.

[6] Y.-R. Meng and L.-R. Chen, "On Discussing the Differences about the Learning Characteristics of LD," Bulletin of Special Education, vol. 23, pp. 75-93, 2002 (in Chinese).

[7] C. L. Nicholson and C. L. Alcorn, "Interpretation of the WISC-III and Its Subtests," Proc. 25th Annual Meeting of the National Association of School Psychologists, 1993.

[8] T.-S. Huang, "A Study on the Characteristics of WISC-III for Students With Learning Disabilities," Master thesis, Graduate Institute of Special Education, National HsinChu University of Education, Hsinchu, Taiwan (in Chinese).

[9] T.-K. Wu, Y.-R. Meng, and S.-C. Hwang, "Identification of Students with Learning Disabilities using Artificial Neural Network," Proceedings of the 2006 International Conference on Artificial Intelligence, 2006.

[10] T.-K. Wu, S.-C. Hwang, and Y.-R. Meng, "Effects of Feature Selection on the Identification of Students with Learning Disabilities using ANN," Lecture Notes in Computer Science, Springer Berlin/Heidelberg, vol. 4221, pp. 565 – 574, 2006.

[11] T.-K. Wu, S.-C. Hwang, and Y.-R. Meng, "Improving ANN Classification Accuracy for the Identification of Students with LDs through Evolutionary Computation," Proceedings of the 2007 IEEE Congress on Evolutionary Computation, 2007.

[12] W.-H. Chang, T.-K. Wu, and Y.-R. Meng, "Improving ANN-Based Classification Accuracy and Performance for the Diagnosis of Students with Learning Disabilities through Grid Computing," Proceedings of the 20th International Conference on Information Management, 2009 (in Chinese).

[13] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking State-of-the-Art Classification Algorithms for credit scoring," Journal of the Operational Research Society, vol. 54, 627–635, 2003.

[14] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.

[15] M. A. Razi & K. Athappilly, A Comparative Analysis of Neural Network (NNs), Nonlinear Regression and Classification and Regression Tree (CART) Models, Expert Systems with Application, vol. 29, no. 1, pp. 65-74, 2005.

[16] W. C. Mead, "Development of Neural Network Tools and Techniques for Arithmetic Training of Children with Learning Disabilities: Phase I - Theory and Simulations," AERA '97, 1997.

[17] K. R. Linstrom, and A. J. Boye, "Neural Network Prediction Model for a Psychiatric Application," Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005.

[18] I. L. Cohen, D. Landon-Jimenez, and M. Keogh, "A Neural Network Approach to the Classification of Autism," Journal of Autism and Developmental Disorders, vol. 23, no. 3, 1993.

[19] X. J. Fu, and L. P. Wang, "Data Dimensionality Reduction with Application to Simplifying RBF Network Structure and Improving Classification Performance," IEEE Trans. System, Man, Cybernetics, Part B, 33, pp. 399-409, 2003.

[20] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality Reduction Using Genetic Algorithms," IEEE Trans. Evolutionary Computation, vol. 4, pp. 164 – 171, 2000.

[21] I. Guyon, and A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[22] T. Bäck, and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," Evolutionary Computation, vol. 1, issue 1, pp. 1 – 23, Spring 1993.

[23] E. Cantú-Paz, and C. Kamath, "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 35, no. 5, pp. 915-927, 2005.

[24] N. Sakamoto, K. Ozawa, and T. Niimura, "Grid computing solutions for artificial neural network-based electricity market forecasts," Proceedings of the 2006 International Joint Conference on Neural Networks, 2006.

[25] N. Cristianini, and J. Shawe-Taylor, "An Introduction to Support Vector Machine and Other Kernel-based Learning Methods," Cambridge: Cambridge University Press, 2000.

[26] I. Foster, "IFIP International Conference on Network and Parallel Computing," Lecture Notes in Computer Science, Springer-Verlag, Berlin / Heidelberg, vol. 3779, pp 2-13, 2005.

[27] N. Karonis, B. Toonen, and I. Foster, "MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface," Journal of Parallel and Distributed Computing, Vol. 63, No. 5, pp. 551-563, May 2003.

[28] The OpenMP API specification for parallel programming, http://openmp.org/wp/about-openmp/, extracted on Feb. 28, 2010.

[29] K. Kazunori, M. Hiroshi, and I. Masaaki, "Asynchronous Parallel Distributed GA using Elite Server," Proceedings of the 2003 congress on evolutionary computation, 2003.