International Journal of Information Technology and Language Studies (IJITLS)

High performance GA-LDA feature selection model for Brain Computer Interface data

Sahar Alwadei [1], **Mohamed Dahab** [2], and **Mahmoud Kamel** [3]
saalwadei@nu.edu.sa; mdahab@kau.edu.sa; miali@kau.edu.sa

[1] Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran, Saudi Arabia
[2] Department of Computer Science, Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia
[3] Department of Information Systems, Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia

Abstract. High-Performance Computing (HPC) proved notable performance enhancements in many different applications. The research is considering the Brain-Computer Interface (BCI) data, precisely the P300 based system. BCI is a system that provides a direct communication control channel between the brain and the external world, but its data processing is exceedingly time-consuming. That system consists of many components where Feature Selection is a primary key to its performance. There is a need for search algorithms and classifiers to build a feature selection model using the wrapper approach. Hence, they are the concern of this research where the Genetic Algorithm (GA) implemented as a search algorithm and Linear Discriminant Analysis (LDA) as a classifier. Thus, the model for features selection formed as GA-LDA. The Evolutionary Algorithm (EA) used, GA, estimates an optimal solution and saves an enormous amount of time more than other algorithms such as brute force search. Furthermore, HPC techniques had implemented since the computational power was one of the main obstacles besides the problem's size causing an extensive processing time. This model saves 98.1% of the original time consumed while using common computing facilities. It also maintains an accuracy rate of 72.5% selecting 46.2% of the original features only.

Keywords: High performance computing, Feature selection, Dimension reduction, Genetic algorithm, Linear discernment analysis, Islands model, Message passing interface.

1. Introduction

For the last few decades, the applications of High-Performance Computing (HPC) have been a field of interest in many disciplines. It proved an exceptional performance and attained acceptable results. One of the most critical fields attempts to get the most of HPC is machine learning where data plays a key role. A set of features would describe any data but having a massive number of them could impede the process of learning from that data. This problem is known as the curse of dimensionality where the features referred to as the dimensions of the problem defined by that data. Therefore, reducing the number of dimensions has been there, and its techniques fall into categories: feature selection and feature extraction. These techniques result in less number or higher quality of features, and that would have a positive impact while processing the data.

Feature selection method preferred since the original features preferred to be known for further implementations. The process of finding the smallest number of vital features following that technique consumes a massive amount of time using conventional computing resources. Thus, HPC gets involved as it offers distinctive potentials being able to reduce the needed time to solve a problem providing high-quality solutions. In this paper, a model that implements both dimension reduction and HPC techniques has proposed.

Brain-Computer Interface (BCI) is one of the fields that can make significant progress by implementing GA-LDA model. Finding out the minimum number of features to be considered while

proceeding its data is one of the issues affecting the researches in this field. Therefore, implementing HPC techniques along with the dimension reduction method, feature selection, delivers promising improvements in the BCI system efficiency and quality as well as its time-saving. This course would take the BCI system to the next level and boost its performance where lots of opportunities are there to proceed with many related applications.

The following section would offer the needful background to go through the paper. Then the methodology used to design the model proposed described. Later, the model has to a BCI dataset, and the final sections show the results and discuss them.

2. Background

2.1. Dimension Reduction

In the latest years, analyzing data has received considerable attention due to its substantial effect in many fields such as biology, engineering, business, economics, astronomy and so on. The problem can be described by the use of a massive number of features to describe some perceived objects. Not all of these features are important to learn about those objects or to mark the aspects of related interests. This leads to high dimensional datasets; dimensions represent the features, that need specific computationally costly methods for analyzing (Fodor, 2002).

In statics, such a problem is known as ***"Big p Small n"*** where explanatory variables ***p***, features, tops the number of samples ***n***. Thus, the number of objects in a dataset is insignificant compared to the number of features defining them (Cunningham, 2008). Thus, dimension reduction techniques endorsed to be employed while analyzing the features describing the considered objects. Thus, that would enhance the computational efficiency and the results accuracy of data analysis. These techniques classified based on the learning manner they apply as supervised or unsupervised. The supervised learning techniques get used in the proposed model. They learn first from a defined subset of data, examples, as a resource. In a different perspective, dimension reduction methods fall into two categories: feature selection or feature extraction (Cunningham, 2008).

Feature selection is a concept that refers to the algorithms select the optimistically best subset of the original set of features. On the other hand, the methods create new features based on some transformations or combinations of the original feature set known as feature extraction. Given a set of features: $F = \{f_1, \ldots, f_i, \ldots, f_n\}$, a feature selection method proposed to find a subset F' out of F where feature extraction maps F to get another feature set F'', (Kamel & Anas, 2014). A detailed study of fast feature selection algorithms described in (Alharbi & Dahab, 2018).

The number of extracted features could or could not be less than the original number of features. In contrast, feature selection techniques executed to select a smaller subset of useful features out of the original ones. Not only the number of features does matter, but the resulting features themselves also do. For instance, inferring the output of feature extraction algorithms can often prove to be problematic. The reconstructed features may have no physical meaning to the domain expert as they will not map to the original ones. On the other hand, the dimensions held by a feature selection procedure can be directly interpreted (Cunningham, 2008).

Feature selection preferred to be the method applied in this research where the most relevant features being selected rather than being extracted. It would lead to a better data acquisition later considering the significant ones only. Moreover, there is an extended benefit of reducing the number of features processed over having the same number of features. It would enhance data processing performance by consuming less time.

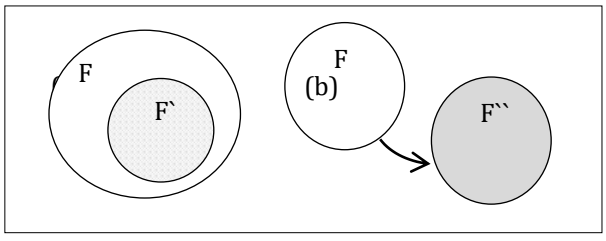

Figure 1. Dimension reduction techniques (a) Feature Selection (b) Feature Extraction

2.2. Evolutionary Algorithms (EAs)

Evolutionary algorithms (EAs) try to estimate the optimal solutions to save the time consumed while searching for all the possible solutions of a problem, brute force search. They are being used to approximate a solution that provides an acceptable rate of accuracy and tries to avoid any misuse of resources. They implement a mechanism simulates biological evolution, such as mutation, recombination, selection, and reproduction. Nominee solutions to a problem play the role of individuals in a population, and a fitness function, object, determines the quality of each solution. That said, any EA, in general, would have three phases. Initializing the individuals of the initial population is randomly based on the solution description. Each of these individuals considered as a solution and a fitness function evaluates it in the second phase. The fitness function used to rank the solutions in the population for selection and to calculate the average fitness value of the population. The third phase would have a new population by neglecting the disturbance in the existing population of solutions (Xin-She, 2010). The population usually disturbed by individuals having fewer fitness values.

Figure 2 shows the flow of applying the three phases. A newer population repeatedly generated if stopping criteria determined earlier are not met. This scenario repeated until at least one stopping criterion satisfied. On the one hand, **fitness value** could be calculated mathematically as it also could be found by applying other methods. It defined after analyzing the considered problem.

On the other hand, **stopping criteria** can be static or dynamic. If it is static, the algorithm allowed to run for a fixed number of iterations while it is a dynamic one, the algorithm allowed to repeat until a specified percent of the found solution is the best considering some percentage. Furthermore, a combination of stopping criteria could perform as well.

In case that an evolutionary algorithm used, especially for optimization problems and dimension reduction in specific, the solution representation must be determined based on the characteristics of the evolutionary algorithm. The population generated within the algorithm might have infeasible solutions. Therefore, designing the solution representation is very critical; thus, the algorithm is more likely to produce feasible solutions. The representation of the solution can be direct or indirect, but the latter must be valid to decoded into a probable solution. A decoding procedure often applied along with any indirect representation in complex problems to transform it into a valid solution. Afterward, the fitness function evaluated once that solution decoded. Defining which features to reduce, each member of an EA population is representing a subset of these features. For example, a population member formed by a random sequence of 0's and 1's where their number is equal to the primary number of features. These zeroes and ones would act as on and off switchers assigning the features to be considered or not within the resulting subset of features. The feature corresponding to "0" is going to be neglected in that subset but it is "1" then the feature will be included.

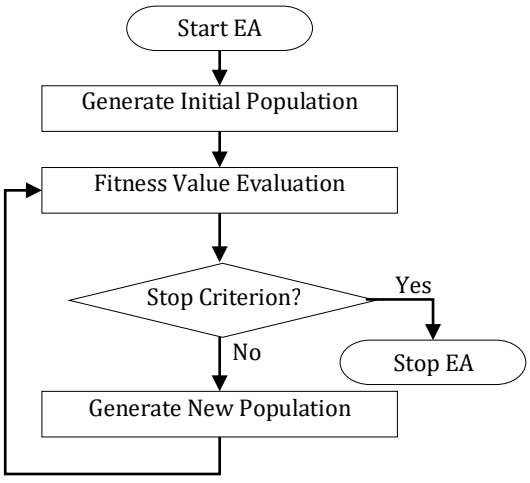

Figure 2. Evolutionary algorithms' phases

Moreover, two critical parameters rather than the solution representation must be determined primarily. Those parameters are the maximum iteration number the size of the population. Both have a significant influence on the quality and accuracy of the solution and the time it would take to find. These

values are always determined empirically through pilot runs in practice even though there are many values suggested and tested as well.

In the literature (K Y & El-Sharkawi, 2008; Rody, 2010; Kachitvichyanukul, 2012), it is stated clearly that for real-world problems, we could get optimal global solutions using evolutionary algorithms. The most dominant EAs are Differential Evolution (DE), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). GA is the chosen algorithm to be applied within the proposed model. In GA, solutions decoded into binary numbers, then each of these solutions, called chromosomes also, regenerated using defined lower and upper limits into real values if needed. Later, each chromosome gains a fitness value using a specified fitness function. GA begins with a randomly generated population. This population evolves where an optimal solution would occur. GA uses three operators to process its populations from the current generation to the next one: selection, crossover, and mutation. The selection comes first, and it selects the right chromosomes according to their fitness values and forms the crossover population. The second operator, the crossover, transmits the best features of the current population to the next population. For instance, a new chromosome could compose out of two of the best previous population ones by filling half of its features from some of the first one's features and filling the second half by the other chromosome features. The developed chromosome does not guarantee a higher fitness value. The goodness of a chromosome is a result of a specific combination of features, and this combination can suffer during this process. The crossover rate is typically quite huge between 70% and 95% of the total number of `members in a population where the rest of would be kept unchanged. The last operator, mutation, supports the diversity in the features of a population and it is used to restrict the algorithm from getting trapped in a local minimum. These three steps repeated until some criteria met.

2.3. Classification

Foretelling the value of a categorical variable, target or class, is a data mining task. By constructing a model based on one or more numerical, predictors, categorical variables or attributes. Using the classification concept with different subsets of features would appoint the best to produce the most likely accurate results. The classifier considered in the proposed model is a supervised machine learning one, and these results used to train classifiers. Such a classifier called supervised classifier and classifiers of this type categorized in general into linear and non-linear methods. The applied classifier is linear, and it is using linear functions to distinguish classes, it is called Linear Discriminant Analysis (LDA). There are many categories of linear classifiers, and it goes under a covariance matrix category which is one of four distinct ones such as frequency table, similarity functions, and others.

LDA is a classification method initially developed in 1936 by R. A. Fisher. It is simple, reliable and often produces good accuracies as those results from more complex methods. It relies on the concept of searching for a linear combination of variables (features or predictors) that separates two classes (target and non-target for example). To declare the idea of classes separated, Fisher defined the following score function:

$$Z = \beta_1 + \beta_2 x_2 + \ldots + \beta_d x_d$$

$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta} \text{ score function}$$

$$S(\beta) = \frac{\bar{Z}_1 - \bar{Z}_2}{Variance\ of\ Z\ within\ groups}$$

The primary matter is to estimate the linear coefficients that maximize the score function, and these equations can solve that:

$$\beta = C^{-1}(\mu_1 - \mu_2) \text{ Model coefficients}$$

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 - n_2 C_2) \text{ Pooled covariance matrix}$$

where

β: *Linear model coefficients*

C_1, C_2: *Covariance matrices*

μ_1, μ_2: *Mean vectors*

The discrimination effectiveness stated by calculating the Mahalanobis distance between the two groups. If it is greater than three, then it is in two averages differ by more than three standard deviations, and the overlap is quite small.

$$\Delta^2 = \beta^T(\mu_1 - \mu_2)$$

Δ: Mahalanobis distance between two groups

Then, a new point classified by projecting it on the direction that maximally separating the classes and classifying it as c_1 if:

$$\beta^T\left(x - \left(\frac{\mu_1 - \mu_2}{2}\right)\right) > \log\frac{p(c_1)}{p(c_2)}$$

LDA implemented as a fitness function of GA. Each proposed solution evaluated by the LDA classifier that gets trained then tested later. While testing, classification errors computed then divided over the size of test data. The result makes a classification accuracy returned to GA to assign as a fitness value to that solution.

2.4. High-Performance Computing (HPC)

High-performance computing (HPC) is the use of parallelism for running programs efficiently, reliably and quickly. The term applies to systems that function above a teraflop or 1012 floating-point operations per second. HPC occasionally used as a synonym for supercomputing. Some supercomputers perform more than a petaflop or 1015 floating-point operations per second. Through the literature given earlier, there were different critical points have been declared, and those have significant consequences on data handling. Both EAs and classifiers require a substantial amount of time and high computational power to process big datasets. That is the main problem to run such technique on ordinary computing facilities, and that leads to low-performance levels. Assuming m population and g generations in GA, w x w grid in parameter tuning and t seconds for LDA plus 10-fold cross-validation, then the whole runtime will be mgw2t seconds. For instance, as a sample data, let m =50, g = 20, and w = 10 then we need 11.57 days. Therefore, the use HPC get recommended where these days could be hours only. This case is the base on which the model' performance evaluated later.

One of the most popular libraries built for parallel computing is Message Passing Interface (MPI). MPI has many implementations as CUDA, **Open MPI**, MPICH. In this paper, the thread-safe Open MPI implementation employed. Open MPI is an open source project that is developed and maintained by a consortium of industry partners, academics and researchers. Thus, it can combine the expertise, resources, and technologies from all the HPC communities. It offers advantages for computer science researchers, application developers, and system and software vendors. It is the library that is used by many Top500 supercomputers. C and C++ programming languages used to write the systems which are acknowledged languages for HPC applications besides Fortran (Umbarkar et al., 2015).

2.5. Brain-Computer Interface (BCI)

BCI is the communication and control system that is used to translate monitored brain activity into commands and messages to introduce its user's intentions. It is designed to help paralyzed people who cannot use or cannot depend on their brain normal output pathways of peripheral nerves and muscles; hence, they cannot use their voluntary muscles usually and lost some or all of their ability to dominate their external environment.

Like any communication system, BCI consist of an input part, which is the brain activity, where command and messages considered as the output part. Those commands and messages expressed by electrophysiological signals generated within the brain instead of muscle contractions. These signals are recorded from users as they respond to external stimuli or perform a mental activity using a brain imaging technology which involves specified sensors (Lan, 2011). The most commonly used method in BCI studies is electroencephalographic (EEG) recordings obtained from a defined number of electrodes placed on the scalp (Smith, 2004). EEG is a favorite signal acquisition technique as its devices are inexpensive, easy to use, and the preparation of the measurement takes a brief time. It also has an excellent temporal resolution compared to its simplicity and availability.

BCI systems classified into invasive and non-invasive and the described one here is a non-invasive one. Different applications of non-invasive BCI systems using electrodes placed on the scalp: spellers, virtual reality, gaming, and wheelchair controlling. The invasive class of BCI systems requires an intracranial surgery since it involves attaching electrodes directly to the brain tissue. Hence, non-invasive BCI systems preferred over invasive ones due to safety and cost (Hoffmann, 2008; Luck, 2005; Croux et al., 2008; Blankertz, 2010).

Current non-invasive BCIs categorized into separate groups based on the type of EEG signals that are used to control and guide the BCI operations (Luck, 2005). One of these groups uses P300, which is an

event-related potential (ERP) that measures the brain response to a stimulus (Alani &Trad, 2010). P300 is a wave that is a significant positive deflection in the voltage that starts about 300 milliseconds after the target stimulus presented to the user (Luck, 2005). It usually produced when a "rare" but expected stimulus presented. Each stimulus and its resulting observation EEG signals referred to as trials. Trials containing the expected stimulus are called target trials, where other trials are non-target trials.

The process of utilizing EEG signals produce the required messages and commands for controlling and communicating decomposed into several phases. These phases form the structure of any BCI system, see Figure 3, including the P300 based one. It starts with signal acquisition where the brain activity is recorded and digitalized. Then, the data obtained from this phase processed; this data represent all the features of the signals produced by the brain. These features are on a massive number which arises the problem of high dimensionality with many irrelevant or redundant features as well. Irrelevant and redundant features would decrease the performance of the following phase. Therefore, feature selection or feature extraction is an effective method to exceed the curse of dimensionality problem. The features reduced are lastly going through the classification phase which will lead to meaningful messages and commands to control and communicate the user's surrounding environment. (Kamel & Anas, 2014; Smaith, 2004; Luck, 2005; Wolpaw et al., 2002; Cabrera, 2009; Hoffmana, 2008; Borikar, 2014).

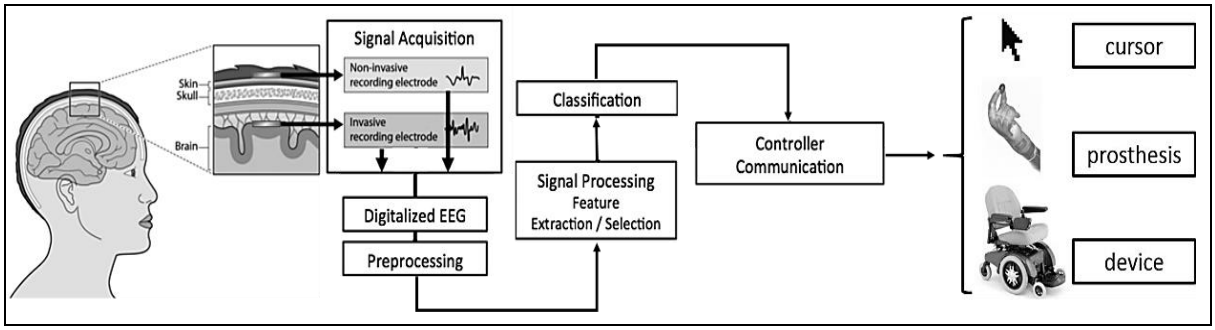

Figure 3. Brain Computer Interface (BCI) Structure

In order to detect the presence of a P300 wave within the trials, a classification process needed. Importantly, these classification methods do require training as the P300 amplitude and latency is different between users and varies with a subject's fatigue level (Hoffmann, 2008; Elshout & Molina, 2009). Classifiers discussed in the upcoming sections. Furthermore, the features describing the BCI trials are noisy, non-stationary, and - as mentioned earlier - high dimensional. They are extracted from several channels at several time segments and have time information because the brain patterns are related to the specific time variation of the EEG. Applying dimension reduction techniques such as feature selection or extraction involving optimization search algorithms over those features enhances the performance of any BCI system significantly (Koprinska, 2010). These techniques along with the suggested search algorithms explored in the next sections.

3. Methods

3.1. Resources and Programming Models

3.1.1. Machine, HPC models and languages

This research runs its code on a high-performance machine that is Fujitsu PRIMERGY CX400, Intel True Scale QDR, Intel Xeon E5-2695v2 12C 2.4GHz. This machine called Aziz and has been launched on June 01, 2015 at King Abdul-Aziz University (Fujitsu, 2015). It considered being one of the top 500 supercomputers (Top 500, 2015). Message Passing Interface (MPI) model applied and the system executed in C and C++ programming languages.

3.1.2. Library used

Boost libraries from boost.org employed in the implementation of this paper proposed model's coding. It is peer-reviewed, free and portable C++ library. It affords support for tasks and structures such as linear algebra, multithreading, regular expressions, pseudorandom number generation, image processing, and unit testing. It contains over eighty individual libraries. Most of the Boost libraries permitted used

with both free and proprietary. In the LDA classifier, Boost used to implement matrices. The library includes the common basic linear algebra operations on vectors and matrices like reductions, addition, subtraction, multiplication with a scalar, inner and outer products of vectors.

3.2. Proposed Model

In this model, Figure 4, a parallel version of GA implemented using a migration method called islands. An island represents one of the computing resources performing the algorithm in parallel. The GA population is distributed equally over the computing resources involved where each of them has its subpopulation. The more computing resources involved in the process, the more significant population initiated.

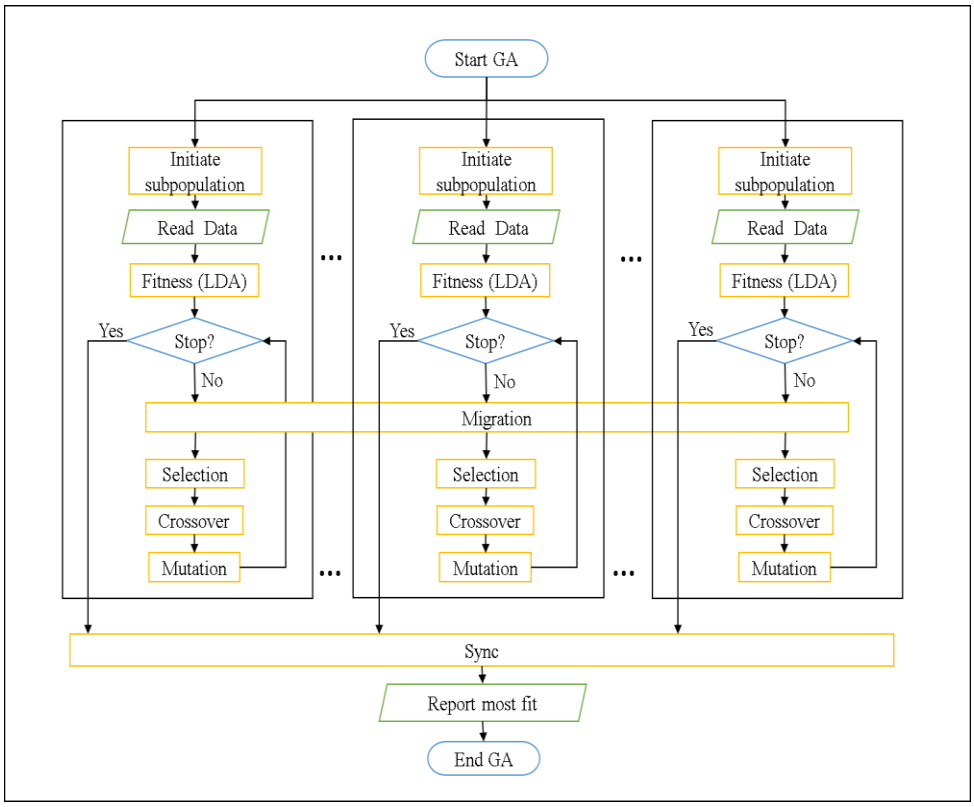

Figure 4. High-Performance GA-LDA Structure

After each generation, some members migrate from one island to its neighbor island in a ring arrangement. The fittest member of the current subpopulation migrates in a counter clockwise to be the fittest one of the neighbors to the left, where the least fit member migrates in a clockwise to be the least fit one of the right neighbors, see Figure 5.

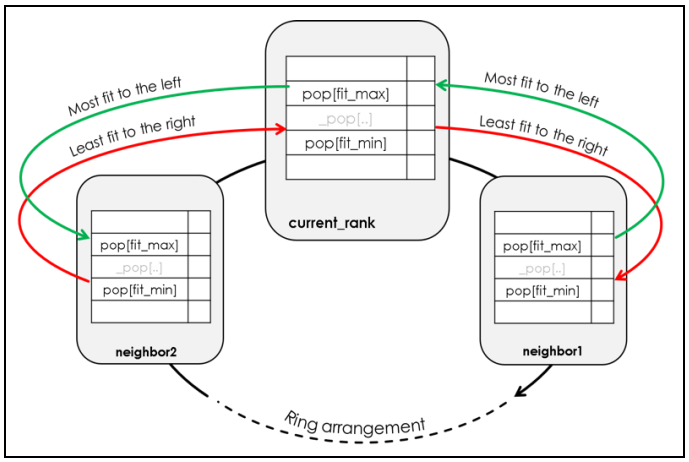

Figure 5. Islands migration method

That concept enables the nodes to communicate; thus, they would not replicate each other which would save the overall time consumed. GA assigns the member that has the highest fitness value as the fittest one where the member has the lowest value as the least one.

Some of the features describe the data excluded by the solution derived from the evolutionary algorithm, GA. Next, a technique called shrinking applied to create a new subset of data out of the original data. By involving this technique, the produced data will have the concerned features only those suggested by the solution derived earlier. Then, the new subset of data sent to the next phase where the classification processed using LDA as a classifier. Within this phase another technique implemented and that is k-folding. It is splitting data into sections in a stated percentage to get different training and testing sets. This procedure repeated for a predefined k-fold number as an input within the model configurations. Consequently, more accurate results attained.

3.3. Dataset Involved

The BCI system dataset from Hoffmann et al. EPFL (Hoffmann, 2008) is the considered dataset for testing the proposed model. The dataset is the recorded sessions' files. There were eight subjects, and each subject completed four sessions. Each session has in average 2500 samples described in 200 features. Further details about EPFL dataset found in (Hoffmann, 2008).

Since EPFL dataset is large, a small standard dataset used earlier while developing the models. This dataset is the CorrAL (John et al, 2017; UCI, 2017; Alwadei et al., 2017). It has 128 samples each of A0, A1, B0, B1, Irr, R, C. The main concept is (A0^A1) U (B0^B1), irrelevant represented by Irr, and R is an attribute highly correlated with the label C, but it has a 25% of error rate. Figure 6 clarifies the phases of applying GA-LDA over a sample of this dataset.

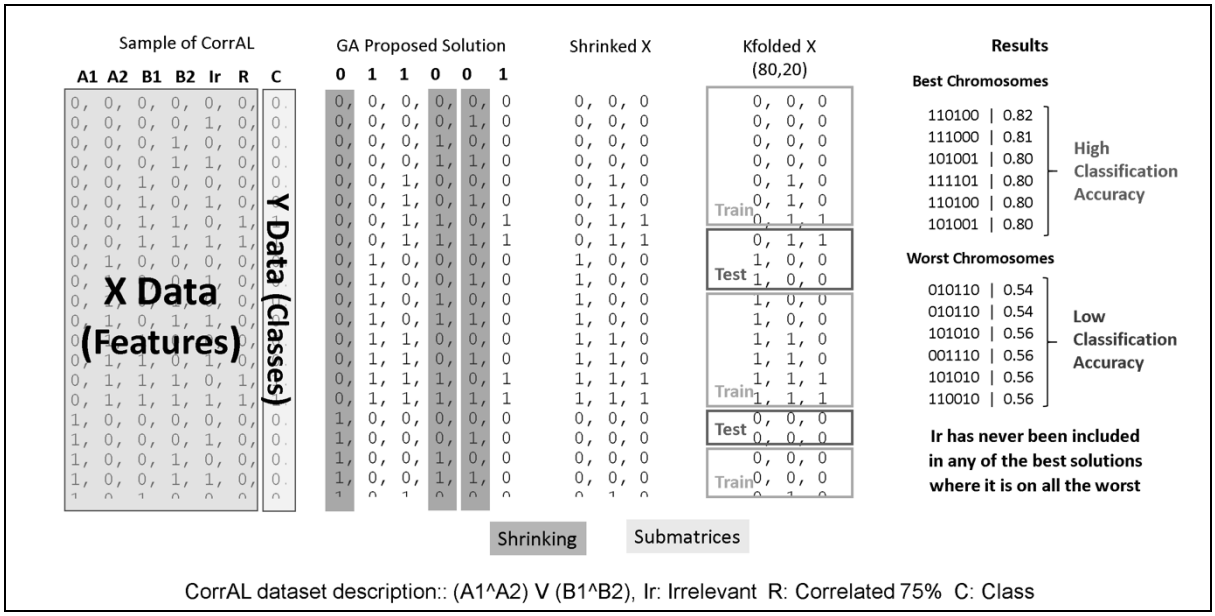

Figure 6. Applying GA-LDA over CorrAL dataset

By starting from the left of the figure, the original data showed then the solution generated by GA applied. This solution formed by a set of zeroes and ones where 1's reflects the considered features while 0's represent the excluded ones. The result of this phase is a dataset that has the considered features only, and that is **shrinking**. Then, the dataset goes through the **k-folding** process within the classification process to define different samples for the named classier training and testing.

3.4. Test Scenario

The proposed model tested on the EPFL dataset. The stopping criterion used was the number of iterations/generations processed: 1, 2 and 4. Different sets of computing resources involved in different test scenarios. The initial set consists of 4 computing resources to investigate the effect of using such a small number compared to the theoretical results discussed earlier in 1.4. Following that, other numbers of computing resources involved such as 128 to represent a large set and 64 to represent an intermediate point between 4 and 128. The initial population size on a single computing resource is 50, and the cross-

rate value is 0.8 where the mutation is 0.1. Considering all the mentioned parameters, there were 288 tests.

The proposed model runs on a record of a defined subject (1-8) in a specified session (1-4). Sets of computing resources and a different number of iterations employed in different scenarios. Therefore, a single test name formed as declared in the fifth column of the table, Table 1., S: Subject, s: session, p: processors in reference to computing resources and i: iteration/s. For instance, BCI_1_2_64P_4G declares that this test is running on the first subject record for the second session using 64 processors (computing resources) and having 4 generations to produce as a stopping criterion.

Subject	Session	No. of computing resources	No. of iteration	Test
8 S	4 s	3 p	3 i	288 tests
1-8	1-4	4, 64, 128	1, 2, 4	BCI_S_s_pP_iG
1	1	4	1	BCI_1_1_4P_1G
1	2			BCI_ 1_2_4P_1G
…	…	…	…	…
3	3	64	2	BCI_3_3_64P_2G
…	…	…	…	…
8	4	128	4	BCI_8_4_128P_4G

Table 1. EPFL dataset tests on High-Performance GA-LDA

4. Results

This section will show the results of testing the GA-LDA model over EPFL dataset. The following charts brief 288 tests those of the scenario explained earlier and present the average of their results. This model scores an accuracy of 72.5 % based on 93 features on average, that is 46.2% out of 200 features. Figure 7 shows that higher accuracies achieved by involving a higher number of processors (computing resources). Even though, there is no notable improvement going from 64 to 128 processors. It is clear that a small set of processors, 4 for example, takes longer time than it does running on a more extensive set, see Figure 8.

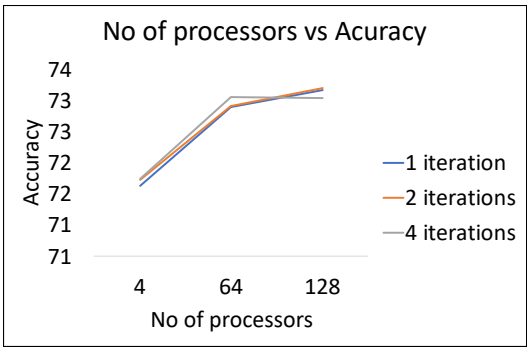

Figure 7. High-Performance GA-LDA model accuracy results

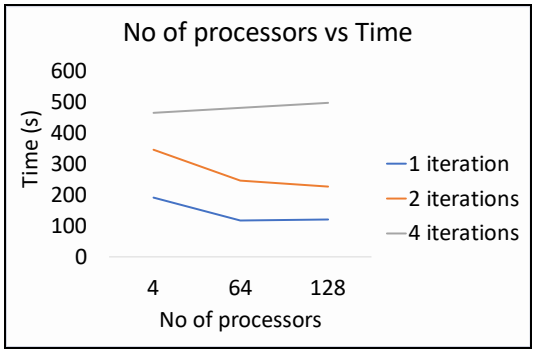

Figure 8. High-Performance GA-LDA model time results

The number of features considered by the model affected where it selects less than half of the original number of features and get even fewer numbers when more processors get involved (See Figure 9). Again, the number of processors getting higher, if it is already has been high, is not reducing the number of features significantly as it has not improved the accuracy earlier. Figure 10 and Table 2 show the vast gap between the running time on one processor and any other set of processors.

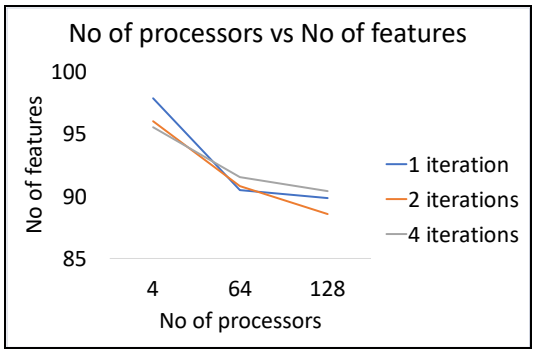

Figure 9. High-Performance GA-LDA number of features results

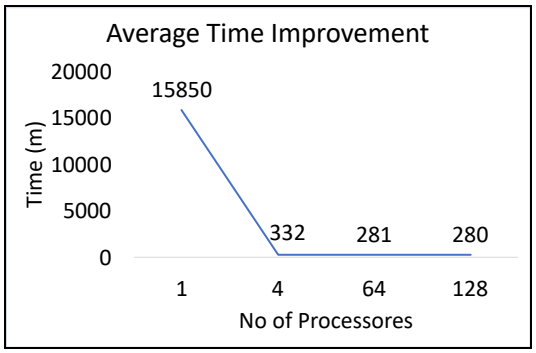

Figure 10. High-Performance GA-LDA average time results

No of computing Resources (p)	Time (m)	Improvement rate (%)
1	10 – 11 days (15500 – 16000 m)	-
4	332	97.906
64	281	98.228
128	280	98.233

Table 2. The proposed model average time results on EPFL

5. Conclusion

A model proposed in this paper to solve the curse of dimensionality problem utilizing the feature selection methods and HPC techniques. GA as a search evolutionary algorithm and LDA as a supervised learning classifier formed the feature selection model. MPI was the library employed to perform the parallelism concept of HPC. The model implemented in C/C++ programming languages. Following, the model tested on a standard dataset called CorrAL then on the main dataset considered as the BCI data, EPFL. The model saves 98.1% of the original consumed time and maintains an accuracy rate of 72.5% selecting 46.2% of the original features only. Even though HPC techniques score significant results but the number of processors used should be rational to the aimed problem size. If not, the implementation of those techniques could be of no use or even reduces the process performance significantly.

Larger sizes of data and higher numbers of processors could be tested and employed. Similar or even better results expected compared to the accomplished ones in this paper running over considerably large sets of data. By following the same strategy employed to build this model other feature selection models

can be formed. Classifiers of the same category selected, linear supervised learning, other categories, or uncategorized classifiers as well could be involved. On the other side, search algorithms of evolutionary ones or other types could be used instead of the one operated in the discussed model earlier. Different models of parallelism and methods of utilizing high performance computing facilities could be another aspect of developing such strategies of feature selection and dimension reduction field in general.

Acknowledgments

The authors would like to thank the members of Aziz support team and the High-Performance Computing Centre (HPCC) for the continuous support and guidance.

Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

References

Al-ani T. and Trad D., (2010.), Signal Processing and Classification Approaches for Brain-computer Interface. INTECH Open Access Publisher.

Alharbi, A. N., and Dahab, M., (2018) Comparative Study on Fast Feature Selection. International Journal of Information Technology and Language Studies.

Alwadei, Sahar & Dahab, Mohamed & Kamel, Mahmod, (2017) A Feature Selection Model based on High-Performance Computing (HPC) Techniques. International Journal of Computer Applications. 180. 11-16.

Atom, Yanina, Gareis, Iván, Gentiletti, Gerardo, Acevedo, Rubén, and Rufiner, Leonardo, (2010), Genetic Feature Selection to Optimally Detect P300 in Brain-Computer Interfaces, 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina.

Blankertz, B., Tangermann, M., Vidaurre, C., Dickhaus, T., Sannelli, C., Popescu, F., Fazli, S., Danóczy, M., Curio, G., and Müller, Klaus-Robert, (2010), Detecting Mental States by Machine Learning Techniques: The Berlin Brain-Computer Interface, B. Graimann et al. (eds.), Brain-Computer Interfaces, The Frontiers Collection, Springer-Verlag Berlin Heidelberg.

Borikar, Shweta Sanjay, Snehal Ramesh Kochre, and Yugandhara D. Zade. (2014), Brain-Computer Interface. National Conference on Engineering Trends in Medical Science (NCETMS).

Cabrera, Alvaro R. Feature Extraction and Classification for Brain-Computer Interfaces. (2009), Ph.D. Thesis, Brain-Computer Interface Laboratory Center for Sensory-Motor Interaction (SMI), Department of Health Science and Technology Aalborg University, Denmark.

Croux, C., Filzmoser P., and Joossens K., (2008), Classification efficiencies for robust linear discriminant analysis, Statistica Sinica, 18(2): p. 581-599.

Cunningham, Pádraig. (2008), Dimension reduction. Machine learning techniques for multimedia. Springer Berlin Heidelberg, 91-112.

Datasets from UCI. SGI, Silicon Graphics International Corp, Retrieved 22 October 2017 from https://www.sgi.com/tech/mlc/db/

Elshout J.and Molina G. G., (2009), Review of brain-computer interfaces based on the p300 evoked potential," tech. rep., Koninklijke Philips Electronics.

Fodor, Isola K. (2002), A survey of dimension reduction techniques.

Fujitsu Supports King Abdul-Aziz University Research Capabilities with New Supercomputing System. Press release on June 01, 2015. King Abdul-Aziz University, Fujitsu Limited. Jeddah and Tokyo,

Hoffmann, Ulrich, Vesin, Jean-Marc, Ebrahimi, Touradj, and Diserens, Karin, (2008), An efficient P300-based brain-computer interface for disabled subjects, Journal of Neuroscience Methods 167, pp. 115–125.

John, Kohavi, and Pfleger, Irrelevant features and the subset selection problem. Machine Learning: Proceedings of the Eleventh International Conference, 1994, Retrieved 22 October 2017 from http://robotics.stanford.edu/~ronnyk . Last access: 10/22/2017.

K Y Lee, M.A. El-Sharkawi, (2008), Modern Heuristic Optimization Techniques, IEEE press and Wiley – InterScience, New Jersey.

Kachitvichyanukul, Voratas. (2012), Comparison of Three Evolutionary Algorithms. Industrial Engineering & Management Systems 11.3 215-223.

Kamel, Mahmoud I., and Anas A. Hadi. (2014), Improving P300 Based Speller by Feature Selection. Journal of Medical Imaging and Health Informatics 4.4: 469-487.

Koprinska, Irena. (2010), Feature selection for brain-computer interfaces. New frontiers in applied data mining. Springer Berlin Heidelberg, 106-117.

Lan, Tian. (2011), Feature extraction feature selection and dimensionality reduction techniques for brain-computer interface. Doctor of Philosophy in Electrical Engineering examined and approved thesis. Oregon Health & Science University, OHSU Digital Commons, Scholar Archive, Paper 706.

Luck, Steven J. (2005), An Introduction to the Event-Related Potential Technique Cognitive Neuroscience, Cambridge, Mass. MIT Press.

Rody P S Oldenhuis, (2010), Trajectory Optimization of a mission to the Solar Bow shock and minor planets," MSc thesis report, Delft University of Technology, Netherlands.

Smith, Raymond C. (2004), Electroencephalograph based Brain-Computer Interfaces, Master Thesis, University College Dublin, Dublin, Ireland.

Top 500, The List retrieved 22 October 2017 from http://www.top500.org/site/50585.

Umbarkar, A. J., M. S. Joshi, and P. D. Sheth. OpenMP Dual Population Genetic Algorithm for Solving Constrained Optimization Problems. International Journal of Information Engineering and Electronic Business (IJIEEB) 7.1: 59.

Wolpaw, Jonathan R., et al. (2002), Brain-computer interfaces for communication and control. Clinical neurophysiology 113.6.

Xin-She Yang, (2010), Engineering Optimization – An Introduction to Metaheuristic Applications. John Wiley & Sons, Hoboken, New Jersey.