A Deep Learning Approach to Detect SNP Interactions

Suneetha Uppu*, Aneesh Krishna*, Raj P. Gopalan
Department of Computing, Curtin University, Bentley 6102, Western Australia, Australia.

* Corresponding author. Email: Suneetha.uppu@postgrad.curtin.edu.au, A.Krishna@curtin.edu.au
Manuscript submitted May 15, 2016; Accepted August 11, 2016.
doi: 10.17706/jsw.11.10.960-975

Abstract: The susceptibility of complex diseases are characterised by numerous genetic, lifestyle, and environmental causes individually or due to their interaction effects. The recent explosion in detecting genetic interacting factors is increasingly revealing the underlying biological networks behind complex diseases. Several computational methods are explored to discover interacting polymorphisms among unlinked loci. However, there has been no significant breakthrough towards solving this problem because of biomolecular complexities and computational limitations. Our previous research trained a deep multilayered feedforward neural network to predict two-locus polymorphisms due to interactions in genome-wide data. The performance of the method was studied on numerous simulated datasets and a published genome-wide dataset. In this manuscript, the performance of the trained multilayer neural network is validated by varying the parameters of the models under various scenarios. Furthermore, the observations of the previous method are confirmed in this study by evaluating on a real dataset. The experimental findings on a real dataset show significant rise in the prediction accuracy over other conventional techniques. The result shows highly ranked interacting two-locus polymorphisms, which may be associated with susceptibility for the development of breast cancer.

Key words: Deep feedforward neural networks, SNP-SNP interactions, two-locus polymorphisms, and machine learning techniques.

1. Introduction

Complex diseases often do not have a clear pattern of biological inheritance in a familial or a population level. There are no individual factors responsible for a disease manifestation. Recent reviews in the literature focus on several genetic, environmental and life style factors acting together or independently which contributes to a complex disease. Over the past decade, a number of genome-wide association studies (GWAS) have predicted numerous genetic variants to unravel the genetic architecture. Almost the entire human genome is identical (99%) with only 1% of variations between any two individuals [1]. These genetic variants that are caused by a change in a single nucleotide of a base pair (adenine and thymine, and guanine and cytosine) are termed as single nucleotide polymorphisms (SNPs). On average, a SNP occur almost one in every 300 nucleotides of a DNA sequence. Roughly, 12 million SNPs are estimated to be present in the human genome that may be useful in identifying their associations with complex diseases or traits [2]. GWAS predominantly focused on single-locus approaches to identify disease causing SNPs, leading to the problem of "missing heritability" [3]. However, multi-locus interaction studies emerged as a step forward in GWAS to improve the understanding of underlying complex architectures of diseases [4], [5].

Multifactor Dimensionality Reduction (MDR) is a non-parametric and model free technique, which exhaustively searches for multi-locus SNP interactions [6]. A constructive induction approach is used to re-

duce multi-dimensional data to a single-dimensional data. The MDR is implemented in java and publically available to download [7]. However, the model eliminates some of the useful SNPs due to the dimensionality reduction. It does not scale up for a large number of SNPs due to exhaustive searching. The model further does not assess the degree to which the level of the risk is distributed. The power of MDR is much affected by the existence of noise. Various extensions and modifications are suggested and applied over general algorithm of MDR to address the drawbacks. Some of the extended pioneer works are Velez [8], Lee [9], Pattin [10], Namkung [11] , Model Based MDR (MB-MDR) [12], Generalized MDR (GMDR) [13], Odds Ratio based MDR (OR-MDR) [14], MDRAC [15] and Robust MDR (RMDR) [16]. Gola reviewed a chronological overview and a vast extensions of MDR as a roadmap [17]. Random forest (RF) [18] is an ensemble learning for classification and regression. RF has been successfully applied over genetic data to uncover the interactions between SNPs [19]. However, at least one of the interacting SNP pair requires a marginal effect which reduces the performance of the model. Some of the strategies addressing the limitations of RF are Random Jungle (RJ) [20], SNPInterForest [21], RFCouple [22], EpiForest [23], TRM [24], Stratified sampling RF (SRF) [25], and SNPInterForest [21].

Support Vector Machines (SVMs) [26] are supervised learning methods that learn from sets of feature vectors. They are used in classification and regression. Interacting and non-interacting SNP pairs are separated in high-dimensional space using a hyperplane [27]. These models produce unacceptable type I errors, and are implemented on a small set of SNPs. Furthermore, the performance of the models is reduced in presence of noise (such as, missing data and genetic heterogeneity). Chen [28], Ozgur [29], Shen [30], Fang [31], Grammatical SVM (GESVM) [32], and Zhang [33] report some of the extended research used to identify SNP interactions using SVMs. Neural Networks (NNs) are represented as a graph in which, nodes (computational units) denote SNPs and arcs denote multi-locus SNP interactions [34]. Some of the pioneering works based on NNs are Grammatical Evolution NN (GENN) [34], Tomita [35], Keedwell [36], Hardison [37], and Genetic programming NN (GPNN) [38]. Some of the pioneering works using regression based models are Park [39], PLINK [40], Monte Carlo Logic regression (MCLR) [41], Genetic programming for association studies (GPAS) [42], Logic Feature selection (LogicFS) [43], and Modified logic regression-gene expression programming (MLR-GEP) [44], and Full Bayesian logic regression (FBLR) [45]. Fast epistatic interactions detection using Markov blanket FEPI-MB [46], WinBUGS [47], Bayesian Network based epistatic association studies (bNEAT) [48], and Bayesian epistasis association mapping (BEAM) [49], are some of the Bayesian methods used in SNP-SNP interaction studies.

However, detecting these SNP interactions still remains one of the biggest challenges due to the inherent mathematical and computational complexities of the problem. Some of these challenges include, the problem of curse-of-dimensionality, genetic heterogeneity, missing heritability, computational complexity and absence of marginal effects [5], [21], [50]. There is no single method that can identify multi-locus interacting SNPs effectively by revealing their associations to a disease manifestation. The previous paper proposes a deep learning approach to detect SNP interactions associated to a complex disease [51]. The proposed deep learning model was trained to discover two-locus interacting SNPs. The method was validated using simulated datasets. The performance of the algorithm is observed in terms of execution time, accuracy, training speed, log loss, and classification error. In this paper, the model is evaluated on a real (sporadic breast cancer) dataset to confirm the findings of the previous study. The result ranks top twenty interactions among ten SNPs, which are included in five different estrogen-metabolism genes. The model is evaluated by changing the parameters to identify the best model with low test set error. Furthermore, it is compared with existing methods, such as MDR, RF, logistic regression (LR), and Gradient Boosted Machines (GBM). The evaluation results using the best model show the improvements in predicting two-locus SNP interactions associated to the disease manifestation over the previous study.

Rest of this manuscript is organised as follows: Section 2 gives an introduction to the chronological overview of the previously proposed deep learning method. Furthermore, the section introduces the extension of multifactor combination to the proposed method. Sections 3 elaborates and discusses several experimental evaluations performed on the method. An outline of future work is described in Section 4.

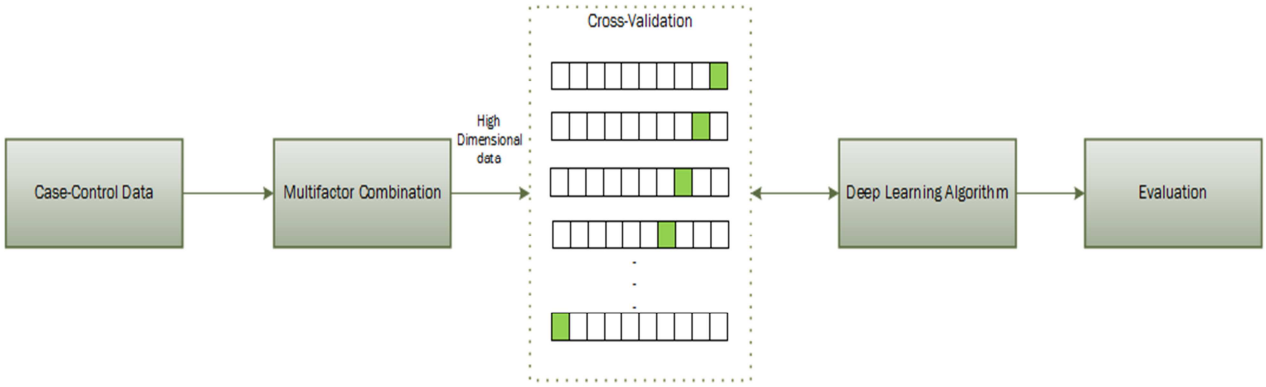

Fig. 1. Workflow representation of the deep learning neural networks [51].

2. Methods

In the current era of genetic epidemiology, conventional machine learning techniques are increasingly used to reveal underlying architecture behind complex diseases. However, none of the models have truly solved the problem of detecting or classifying the patterns in the genomic data. Deep learning is an emerging field that allows systems to learn the data by portraying in hierarchical abstractions. They allow the computational models to identify the representations required for the classification using general-purpose learning procedures [52]. These deep structured learning models provides stability, generalization, and scalability to big data by providing high prediction accuracy in a number of diverse problems [53]. Among these, deep learning has been a breakthrough in image recognition [54, 55], and speech recognition [56]. It has also produced promising results in language translation [57], reconstructing brain circuits [58], question answering [59], and natural language understanding [60]. Many researchers believe that these methods will have tremendous success in many other domains such as bioinformatics [52, 61]. This motivated the exploration of training a deep learning method to detect two-locus interactions between SNPs in the previous research [51]. The following section briefly introduces the method.

2.1. Deep Learning Method

The proposed deep learning method is illustrated in Fig. 1 [51]. Stage one comprises of case-control based data input. Various simulated scenarios are generated. In total 27,600 datasets are simulated using GAMETES tool [51], [62]. The findings are confirmed on a real (sporadic breast cancer) dataset [6]. Combinations of SNPs at various locations are combined together in stage two. It is performed to improve the prediction accuracy of the models such that none of the SNPs are left. Ten-fold cross validation is conducted in stage three to analyse the predictive power of the method. Stages four and five consist of the deep learning algorithm and its evaluation respectively. The deep learning algorithm classifies high-risk genotype combinations, and discovers multi-locus SNP interactions associated to a disease manifestation. Finally, the model is evaluated by varying various parameters. The best model is identified as one with low test set error.

2.1.1 Multifactor combinations and cross validation

Multifactor genotype combinations at different loci are combined together to improve the prediction accuracy of the method. Consider h genetic factors where g_k are selected with k = 1, 2,…., h, levels in X factors

[17]. During training, all the factors are combined together in h-dimensional space. This is represented as a function given by:

$$f(X) = \sum_{k=1}^{h} g_k \tag{1}$$

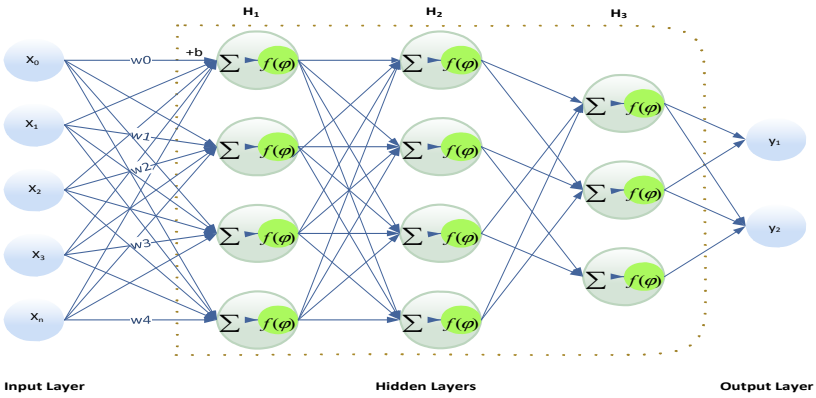

Fig. 2. Basic structure of a four-layer feedforward network.

Ten-fold or five-fold Cross Validation (CV) are the most successful internal model validation methods used in both simulated and real data in genetic epidemiology [63]. The data is divided into m equal portions. Training data is represented by m-1 portions, and the remaining one portion represents testing data. In ten-fold CV, the models are built ten times by leaving one tenth of data for every run (training). The method is tested on the rest of the one tenth of data that was left for testing. This process is repeated for ten folds. Hence, the deep learning algorithm executes m times for m-fold CV. The model with high cross-validation consistency (CVC), maximum classification accuracy, and low test set error is chosen as the best model.

2.1.2 Deep learning feedforward algorithm

The method is based on deep feedforward neural networks [52], [53], [64] proposed in the previous work [51]. A feedforward network comprises of single input and output layers along with three hidden layers, an example of which is represented in figure 2. The basic unit of the deep networks are neurons. The neurons are inter-linked in multiple layers. The output of input layer forms the input to hidden layers, and the output of last (third) hidden layer is fed into the output layer as an input. The inputs are combined with weights to denote the importance of the inputs to the output. The output of each neuron is transmitted by weighted sum $\sum_j w_{ij} x_i$ less than or greater than a threshold level. The generalised output transmitted by a neuron is given by $a = b + wx$, where b is bias ($b \equiv -threshold$), w is a weight assigned to the input x for each neuron. Bias is included for all the neurons in the network excluding the neurons in the output layer. The performance of the network depends on width, depth, weights and biases of the network. The weighted combinations of a neuron transmitts an output function $f(a)$ [52], were non-linear f is an activation function of a neuron.

The output transmitted by jth neuron of hidden layer 1 (H_1) is given by:

$$f\left(a_j\right) = f\left(\sum_i w_{ij} x_i + b_{\,i}\right) \tag{2}$$

The output transmitted by kth neuron of hidden layer 2 (H_2) is given by:

$$f\ (a_k) = f\left(\sum_j w_{jk} x_j + b_j\right) \tag{3}$$

The output transmitted by l[th] neuron of hidden layer 3 (H3) is given by:

$$f\ (a_l) = f\left(\sum_k w_{kl} x_k + b_k\right) \tag{4}$$

The output transmitted by m[th] neuron of hidden layer 4 (H_4) is given by:

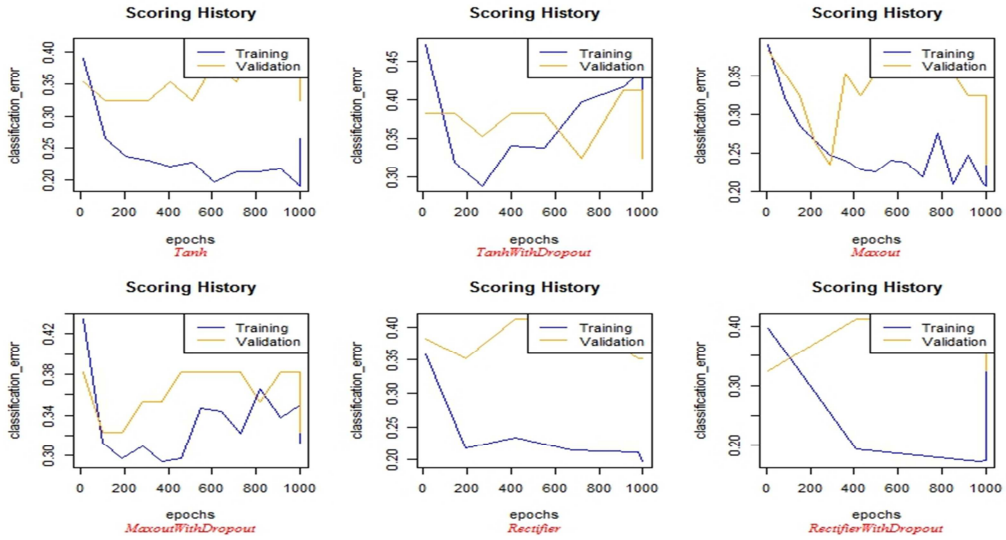

Fig. 3. Scoring history of the model by changing activation functions (Tanh, TanhWithDropout, Maxout, MaxoutWithDropout, Rectifier, and RectifierWithDropout).

$$f\ (a_m) = f\left(\sum_l w_{lm} x_l + b_l\right) \tag{5}$$

The proposed deep feedforward neural networks are trained by three hidden layers. Width of each hidden layer is 50. Tanh, rectified linear, and maxout are the three different non-linear activation functions that can be used by the models [53].

The hyperbolic tangent activation function is represented by:

$$f(a) = \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{6}$$

Rectified activation function is represented by:

$$f(a) = \max(0, a) \tag{7}$$

Maxout is a generalised rectifier activation function and is given by:

$$f(a_1, a_2) = \max(a_1, a_2) \tag{8}$$

Backpropagation is used to adapt the weights by reducing the loss. The output error is estimated by using a cross entropy objective function and is given by [53]:

$$E = -\sum_{u \in L} (\log y_j * t_j + \log(1 - y_j) * (1 - t_j)) \quad (9)$$

where, y denotes actual and t denotes the predicted output respectively. u represents output units and L represents the output layer. The stochastic gradient descent (SGD) is parallelised by using a lock free approach to handle the memory efficiently [65]. The method is scalable and can specify the number of training samples as noted in [66]. It is trained with N epochs (number of passes over training data) per iteration on M nodes. For example, consider the training samples per iteration to be 200,000 running on 8 nodes. Each node processes 25,000 samples per iteration. Hence, if the data has 20 million samples, there will be 80 distributed iterations to process one epoch. Dropout is a regularisation technique used in the algorithm to prevent overfitting [67]. It randomly drops neurons along with their connections to prevent neurons from co-adapting too much in the networks. That is, each neuron in the network prevents its activation with a probability of 0.2 and 0.5 for the neurons in the input and hidden layers respectively. As a result, tanh along with dropout is used to improve generalization of the method by preventing the data from overfitting.

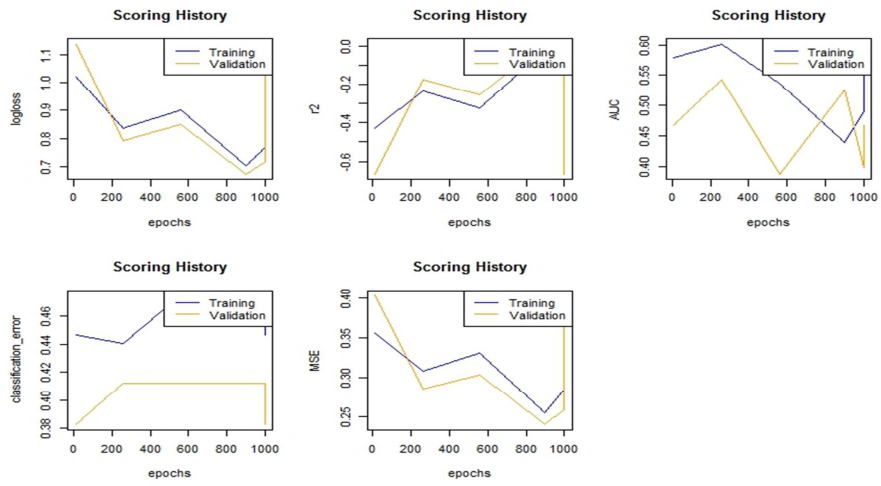

Fig. 6. Scoring history of epochs vs metrics of the model.

3. Experimental Evaluations and Discussions

Multiple experiments are performed to evaluate the performance of the proposed method on simulated datasets under various scenarios, and a real dataset. The preliminary analysis of the method was demonstrated in the previous work, and the results were encouraging [51]. The models are evaluated by splitting the data (randomly into 80%, 10%, and 10% of data for training, validation, and testing respectively) for validation, and performing n-fold cross validation to determine the statistical significance of the models. The findings are confirmed by evaluating the method on a sporadic breast cancer dataset [6]. The aim of this study is to confirm whether the trained model is an effective method to discover the two-locus SNP interactions. Furthermore, this study is evaluated and analysed by varying the parameters to identify the best model with low test set errors. The method is built and analysed in R using the H2o package [66].

2.2 Evaluation and Analysis of the Proposed Method on Real Data

The deep feed forward network trained in the previous study [51] has a single input and output layers along with three hidden layers. Each layer in the hidden layers is trained with 50 computational units. The method processes 1000 epochs per 1000 iterations on 10 compute nodes. By default, the entire data is processed on every node locally by shuffling the training samples in each iteration. The preliminary results of the method illustrated improvements to the prediction accuracy on all simulated datasets. The method is furthermore evaluated on published breast cancer data obtained from Vanderbilt University Medical centre [6]. The model is trained with training samples of 320,000. The model took 17.658 seconds to train the data. The training speed of the model is estimated as 8122.098 samples/second. The validation error of the model is 0.294. Finally, test set error of the model is estimated as 0.661. The model is validated by passing various non-linear activation functions such as, rectifier, tanh, maxout, rectifier with dropout, tanh with drop out, and maxout with drop out. Figure 3 compares the scoring history of the model by varying the activation functions. Among all, tanh with dropout has high prediction accuracy with low classification error. Hence, it is chosen as an appropriate activation function to achieve better approximation. The input drop out ratio is set to 0.2 and hidden dropout ratios for the three hidden layers are each set to 0.5. The model is predicted using test data for the classification. The algorithm is executed ten times as 10-fold cross validation is performed. Each time a different split is omitted for testing the data. The model with a high CVC and low classification error is selected. The best model chosen from n-fold cross validation, predicted the two-locus SNPs and SNPs with main effects that are highly related to breast cancer.

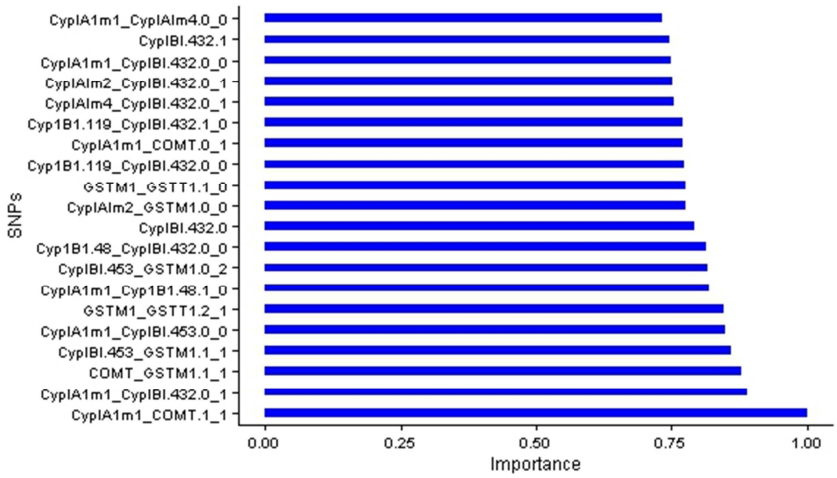

Fig. 4. Top 20 interacting SNPs.

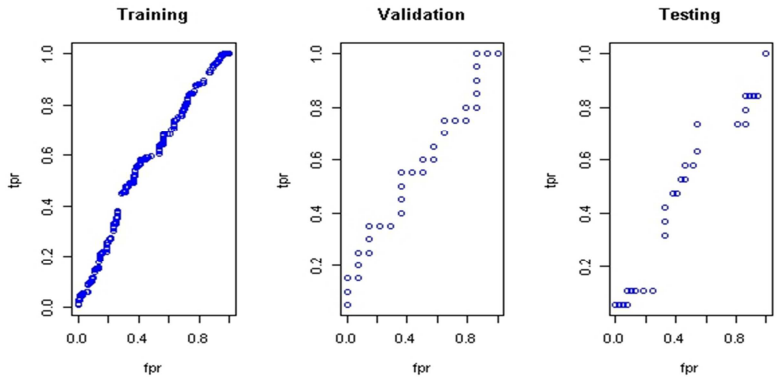

Fig. 5. Performance of the model while training, validating, and testing.

Fig. 4 represents highly ranked top 20 interacting genetic polymorphisms in ascending order. The best model is validated by dividing the entire data into three parts with the probabilities of 0.8, 0.1 and 0.1 for training, validation, and testing respectively. The performance of the model for training, validating, and testing is shown in Fig. 5. Scoring history of the best model is shown in figure 6. The plots are outlined between timesteps on the x-axis (epochs and samples) and metrics on the y-axis (log-loss, classification error, r2, mse, and auc). It is noted that there is a fall in synchronisation and model convergence when training samples/epochs increased. It is also observed that the performance of the model is reduced drastically when the training samples are too small. This occurs due to the dominance of network communication between computational units increase by affecting the execution time of the algorithm.

2.3 Evaluation and Analysis by Changing Parameters

A number of studies are carried out based on [68] to find the best model with improved accuracy and speed, by tuning the parameters. The performance of the method is evaluated in terms of training speed and training time for each set of parameters which is similar to the studies in [68]. In the first study, the model's network topologies (hidden layers) are changed by setting rest of the parameters with their default values. The model is evaluated for one, two, three, and four hidden layers with 100 epochs. It is observed that, the network with three hidden layers (each with 64 neurons), performed better than other models with a training speed 4661.972 samples/seconds. In the second study, all the models are trained with three hidden layers (each with 2048 neurons) by varying scoring selections (score training samples, duty cycle, and interval). The best model identified in this study has a training speed of 41.586 samples/second, whose training time is 8.272 seconds. The third study is a comparative evaluation of manual and adaptive learning rates along with momentum. It is observed that the model with manual learning rate along with no momentum has performed well compared to other models. This is due to less usage of memory and low computational burden. Training samples per iteration is varied in the study four with the same 3 layer network. The model performed well as the number of training samples increased in terms of training speed (48.258 samples/second), and training time (6.631 seconds). Figure 7 shows the performance of the models evaluated under these four studies are plotted in terms of training speed and training time.

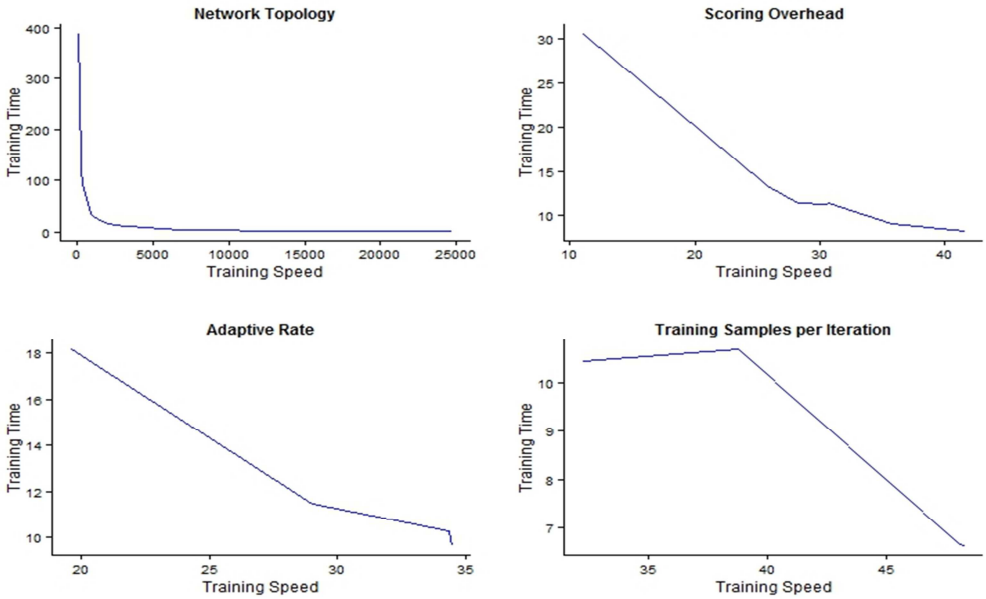

Fig. 7. Training speed vs training time (seconds) of the model by varying parameters which includes, network topology, scoring selections, adaptive learning rate, and training samples per iteration.

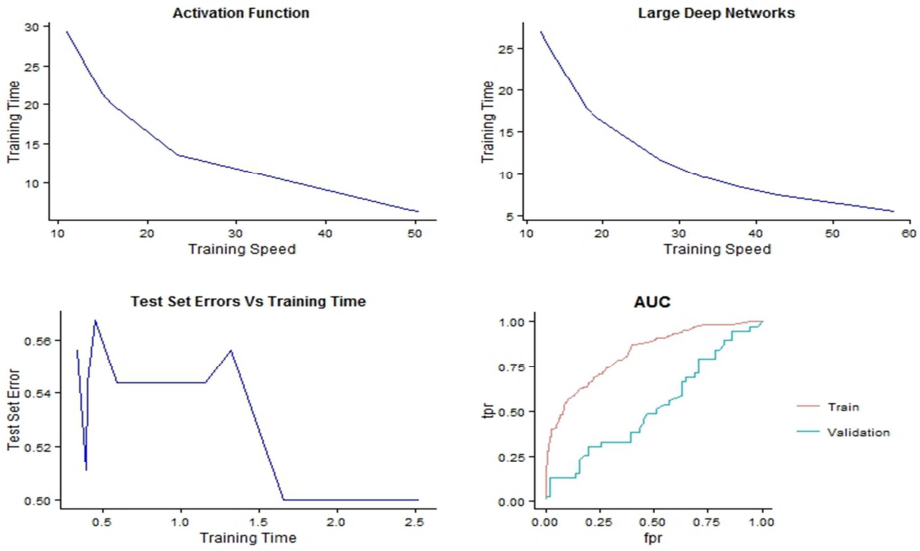

Fig. 8. Training speed vs training time (seconds) of the model by varying parameters (activation function, large deep networks), training time vs test set error to obtain the model with low test set error, and AUC in distributed mode.

Table 1. Metrics of the Deep Learning Method Compared with the Previous Approaches

Methods	Accuracy	MSE	r2	Logloss	AUC	Gini
Deep learning	**68.78**	0.2750	-0.1001	1.192	0.7436	0.4873
RF	55.85	0.3117	-0.2471	1.2019	0.5139	0.0279
LR	67.07	0.2123	0.1508	0.6132	0.7360	0.4720
Naïve Bayes	62.68	0.3092	-0.2369	1.2403	0.6564	0.3128
GBM	65.85	0.2346	0.0616	0.6620	0.7297	0.4593

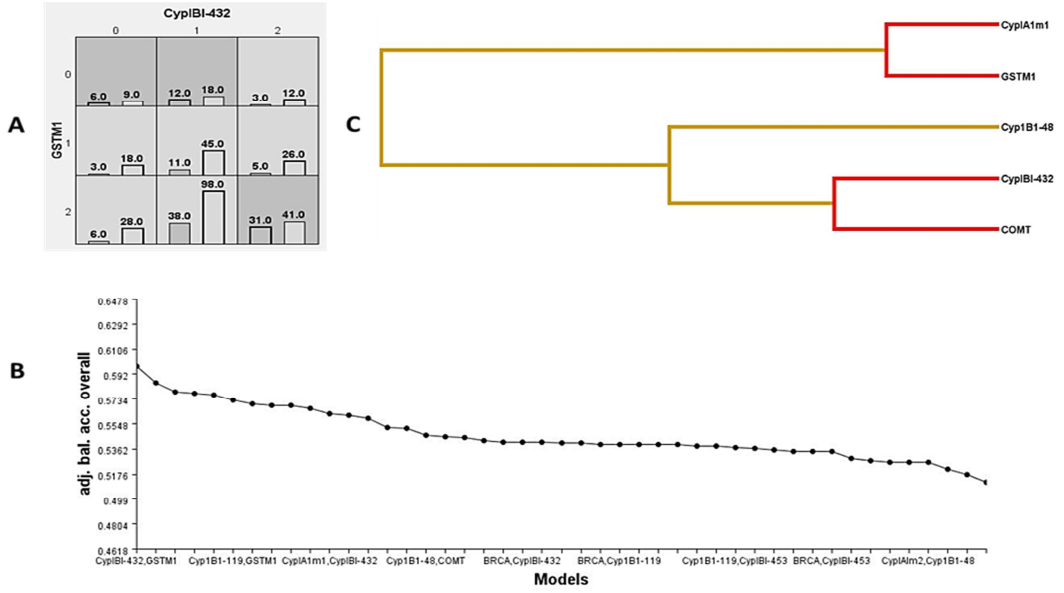

Fig. 9 Performance of MDR. A) Allocation of high-risk and low-risk cells in the two-locus contingency table of genotype combinations. B) The line graph represents the overall adjusted balanced accuracy of top two-locus interacting models. C) An interaction dendogram summarising the information gain associated with constructing pairs of SNPs. Shorter connections among nodes represents stronger synergistic (red lines) interactions.

Study five is performed by varying activation functions (Rectifier, Rectifier with dropout, Tanh, Tanh with dropout, Maxout, and Maxout with dropout) for all the models evaluated in the above four studies. Rectifier, and Rectifier with dropout performed reasonably well compared to other activation functions. In study six, the performance of the method is observed with large deep networks. The networks are trained with four, and five hidden layers, under various parameter settings along with different activation functions. The best model has the highest training speed of 57.982 samples/second, and took 5.519 seconds to train the model. The best model identified by study seven has minimum test set error of 0.5, which took 2.53 seconds to train the two layered network. Study eight validates the benchmark model, and calculates AUC. Fig. 8 illustrates studies five, six, seven, and eight as a line graph.

2.4 Evaluation and Analysis with Respect to Previous Methods

The proposed method is evaluated and compared with few pioneering works, such as Naïve Bayes, RF, GBM, LR, and MDR, proposed to address this research problem. However, some of the preliminary results are demonstrated in the previous work [51]. In depth evaluation is performed on a real dataset in this study. Prediction accuracy of all these approaches is noted and tabulated in Table 1. It is observed that the prediction accuracy of the proposed deep learning method is 68.78 %, which is higher than other current machine learning approaches. The trained model identifies that the interaction between common homozygous CypIBI.453 and recessive homozygous GSTM1 at two different loci has high association to the disease. The MDR tool implemented in java, version-3.0.2, [6] is used to analyse the published sporadic breast cancer data obtained from Vanderbilt University Medical Centre. The best single-locus model detected is GSTM1 with a testing accuracy of 56.82%. MDR identified high interaction between Cyp1B1-432 and GSTM1, and demonstrated as the best two-locus model with default parameters. The balanced accuracy of the two-locus model during testing is observed as 57.1% with the highest CVC (10 out of 10). The experimental results of the best two-locus models are represented in Fig. 9. It is observed that the results were sensitive to the choice of random number seed. The breast cancer dataset is also analysed by using LogicFS, developed for the R environment [43]. The best two-locus interaction model identified by LR is Cyp1B1.119_1 & !GSTT1_1. Furthermore, bagging version of LR is used to compute out-of-bag error (OOB) rate (49.76%). It is observed that searching for SNP interactions among all the possible logic trees became computationally difficult as the number of SNPs grows. It is also observed that variable selection is improved by adopting the simulated annealing algorithm. Figure 10 illustrates the evaluations of LR using LogicFS.

Similarly, breast cancer data is evaluated and analysed on RF, GBM, and Naïve Bayes machine learning approaches by using H2o package implemented for the R environment [66]. The classification accuracy of RF is 55.85%. It is observed that RF analysis allowed the models to decide the importance for the variables that can have high possibility of Interactions. However, a high classification error during testing is observed compared with other methods. As noted in the previous work, prediction accuracy of RF analysis on a real dataset also degrades the performance of the models, as it required that at least one SNP in the SNP pair should have marginal effect. However, it has been noted that when there is no interrelationship between trees in the forest, the prediction accuracy is significantly high. The classification accuracy of GBM is 65.85%, which is higher than all other previous methods excluding the proposed method. Furthermore, the Naïve Bayes classifier is evaluated and analysed using the same dataset. The prediction accuracy is observed as 62.68%, which is higher than for RF and MDR. The scoring history of RF, GBM, and LR is illustrated in Fig. 11 and compared with the proposed method. Fig. 12, summaries the prediction accuracy of all the models presented as a bar chart.

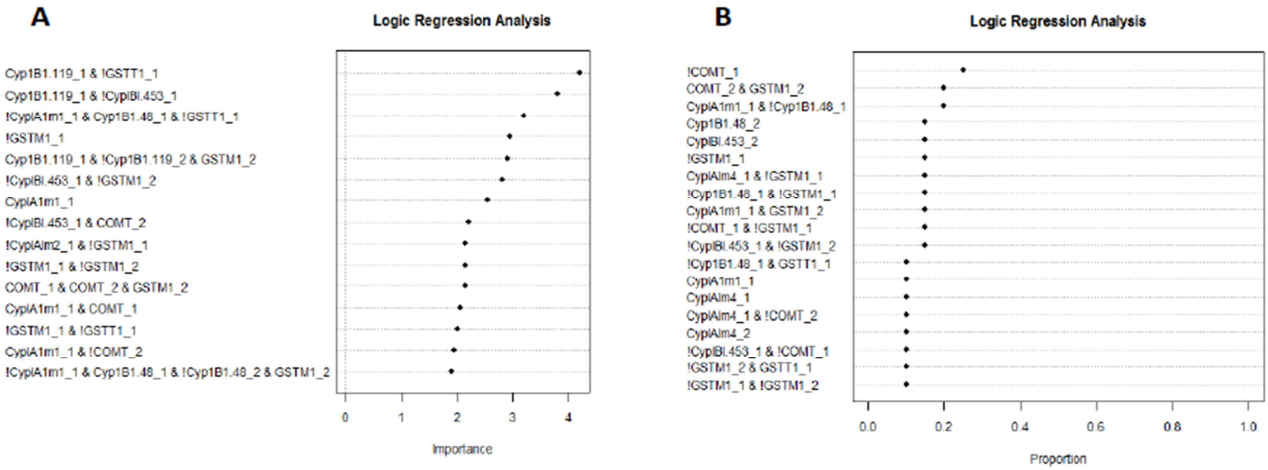

Fig. 10 Performance of LR. A) The graph plots the variable importance of sporadic breast cancer data using LR analysis. B) The graph shows the proportions of models that contain the interactions of interest C) The top 5 important interactions were identified by Logic Regression (LR). The best interaction model identified by LR was Cyp1B1.119_1 & !GSTT1_1, where !GSTT1_1 stands for NOT GSTT1_1 (representing the complement of GSTT1_1). Hence, the logical expression of the top model is interpreted as: Cyp1B1.119_1 is of the homozygous variant genotype and !GSTT1_1 is of the homozygous reference genotype.

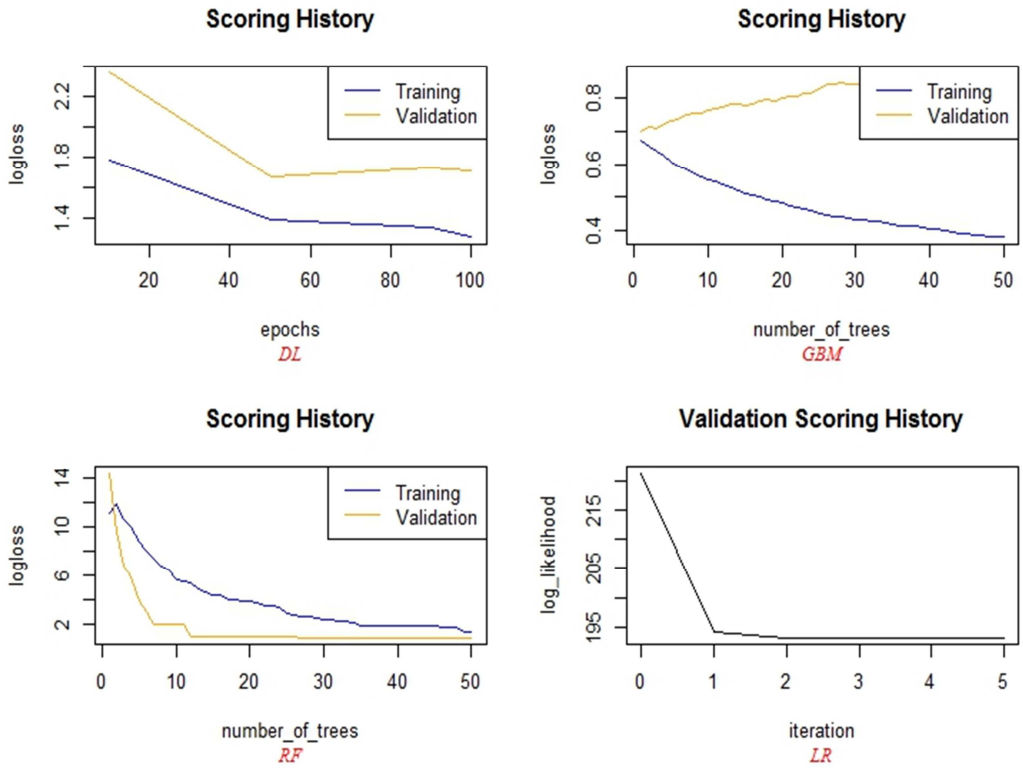

Fig. 11. Scoring history of deep learning method, RF, GBM, and LR.

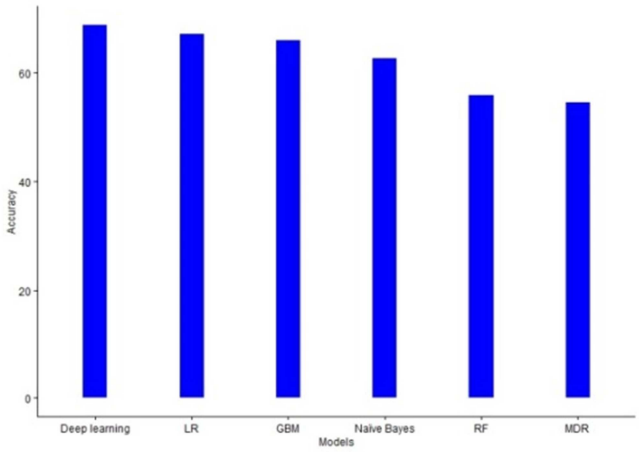

Fig. 12. Prediction accuracy compared with other machine learning methods.

4. Conclusion

A detailed study of the previously proposed deep learning method is carried out in this paper. Findings from preliminary results using simulated datasets in the previous research are confirmed by evaluating the models on real data. The method identifies top twenty highly ranked two-locus SNP interactions, which are highly related to sporadic breast cancer. The performance of the model is predicted and analysed over training, validation and testing. In depth evaluations are performed by varying parameters of the model to identify the best model. Furthermore experiments on the method demonstrated improved prediction accuracy over other previous methods. Current studies are performed to identify only two-locus interactions. Hence, further studies will be performed to validate the performance of the method for higher-order interactions (three-locus or above) in high-dimensional data. The method will be extended for unsupervised learning. Furthermore, future studies will explore different techniques to reduce the data dimensionality, as a preprocessing step to the method.

Acknowledgment

Our sincere appreciation to Dr.John Wallace, who works for Ritchie laboratory, Pennsylvania State University, for sharing with us the published breast cancer data, and assisting us with his expert knowledge in this research. We thank Dr. Jason Moore and his team members at the Dartmouth Medical School for developing the MDR tool and making the source code available publically from www.epistasis.org. We appreciate the generosity of H2o.ai team for making the h2o package available for the R environment. We also appreciate Dr. Holger Schwender for his generosity in building up logicFS package and making it available for the users of R. We also thank the anonymous reviewers for helping us to improve this manuscript.

References

[1] Koenen, K. C. (2007). Genetics of posttraumatic stress disorder: review and recommendations for future studies. *Journal of Traumatic Stress, 20*, 737-750.

[2] Medicine., N. L. O. (2016). *Genetics Home Reference.* Retrieved from: http://ghr.nlm.nih.gov/

[3] Steen, K. V. (2012). Travelling the world of gene–gene interactions. *Briefings in bioinformatics, 13*, 1-19.

[4] Gusareva, E. S., Carrasquillo, M. M., Bellenguez, C., Cuyvers, E., Colon, S., Graff-Radford, N. R., Petersen, R. C., Dickson, John, D. W., & Bessonov, K. (2014). Genome-wide association interaction analysis for Alzheimer's disease. *Neurobiology of aging, 35*, 2436-2443.

[5] Cordell, H. J. (2009). Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics, 10*, 392-404.

[6] Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., & Moore, J. H. (2011). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics, 69*, 138-147.

[7] Hahn, L. W., Ritchie, M. D., & Moore, J. H. (2003). Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. *Bioinformatics, 19*, 376-382.

[8] Velez, D. R., White, B. C., Motsinger, A. A., Bush, W. S., Ritchie, M. D., Williams, S. M., & Moore, J. H. (2007). A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. *Genetic Epidemiology, 31*, 306-315.

[9] Lee, S. Y., Chung, Y., Elston, R. C., Kim, Y., & Park, T. (2007). Log-linear model-based multifactor dimensionality reduction method to detect gene–gene interactions. *Bioinformatics, 23*, 2589-2595.

[10] Pattin, K. A., & Moore, J. H. (2008). Exploiting the proteome to improve the genome-wide genetic analysis of epistasis in common human diseases. *Human Genetics,* 124, 19-29.

[11] Namkung, J., Kim, K., Yi, S., Chung, W., Kwon, M.-S., & Park, T. (2009). New evaluation measures for multifactor dimensionality reduction classifiers in gene–gene interaction analysis. *Bioinformatics, 25*, 338-345.

[12] Calle, M., Urrea, V., Vellalta, G., Malats, N., & Steen, K. (2008).Improving strategies for detecting genetic patterns of disease susceptibility in association studies. *Statistics in Medicine, 27*, 6532-6546.

[13] Lou, X.-Y., Chen, G.-B., Yan, L., Ma, J. Z., Zhu, J., Elston, R. C., & Li, M. D. (2007). A generalized combinatorial approach for detecting gene-by-gene and gene-by-environment interactions with application to nicotine dependence. *The American Journal of Human Genetics, 80*, 1125-1137.

[14] Chung, Y., Lee, S. Y., Elston, R. C., & Park, T. (2007). Odds ratio based multifactor-dimensionality reduction method for detecting gene–gene interactions. *Bioinformatics, 23*, 71-76.

[15] Uppu, S., Krishna, A., & Gopalan, R. P. (2015). A multifactor dimensionality reduction based associative classification for detecting SNP interactions. *Neural Information Processing*, 328-336.

[16] Gui, J., Andrew, A. S., Andrews, P., Nelson, H. M., Kelsey, K. T., Karagas, M. R., & Moore, J. H. (2011). A robust multifactor dimensionality reduction method for detecting gene–gene interactions with application to the genetic analysis of bladder cancer susceptibility. *Annals of Human gEnetics, 75*, 20-28.

[17] Gola, M. J. J. D., Van, S. K., & König, I. R. (2015). A roadmap to multifactor dimensionality reduction methods. *Briefings in Bioinformatics.*

[18] Breiman, L., Random forests. *Machine Learning, 45*, 5-32.

[19] Qi, Y. (2012). Random forest for bioinformatics. *Ensemble Machine Learning*, 307-323.

[20] Schwarz, D. F., König, I. R., & Ziegler, A. (2010). On safari to random jungle: A fast implementation of random forests for high-dimensional data. *Bioinformatics, 26*, 1752-1758.

[21] Yoshida, M., & Koike, A. (2011). SNPInterForest: A new method for detecting epistatic interactions. *BMC Bioinformatics, 12*, 469.

[22] Lobel, L. D., Geurts, P., Baele, G., Castro-Giner, F., Kogevinas, M., & Steen, K. V. (2010). A screening methodology based on random forests to improve the detection of gene–gene interactions. *European Journal of Human Genetics, 18*, 1127-1132.

[23] Jiang, R., Tang, W., Wu, X., & Fu, W. (2009). A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics, 10*, S65.

[24] Lin, H. Y., Chen, Y. A., Tsai, Y. Y., Qu, X., Tseng, T. S., & Park, J. Y. (2012). TRM: A powerful two-stage machine learning approach for identifying SNP-SNP interactions. *Annals of Human Genetics, 76*, 53-62.

[25] Wu, Q., Ye, Y., Liu, Y., & Ng, M. K. (2012). SNP selection and classification of genome-wide SNP data using stratified sampling random forests. *IEEE Transactions on NanoBioscience, 11*, 216-227.

[26] Han, J., Kamber, M., & Pei, J. (2006). *Data Mining: Concepts and Techniques*.

[27] Koo, C. L., Liew, M. J., Mohamad, M. S., & Salleh, A. H. M. (2013). A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology. *BioMed Research International.*

[28] Chen, S. H., Sun, J., Dimitrov, L., Turner, A. R., Adams, T. S., Meyers, D. A., Chang, B. L., Zheng, S. L., Grönberg, H., & Xu, J. (2008). A support vector machine approach for detecting gene-gene interaction. *Genetic Epidemiology, 32*, 152-167.

[29] Özgür, A., Vu, T., Erkan, G., & Radev, D. R. (2008).Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics, 24*, i277-i285.

[30] Shen, Y., Liu, Z., & Ott, J. (2012). Support vector machines with L 1 penalty for detecting gene–gene interactions. International *Journal of Data Mining and Bioinformatics, 6*, 463-470.

[31] Fang, Y. H., & Chiu, Y. F. (2012). SVM-based generalized multifactor dimensionality reduction approaches for detecting gene-gene interactions in family studies. *Genetic Epidemiology, 36*, 88-98.

[32] Marvel, S., & Motsinger-Reif, A. (2012). Grammatical evolution support vector machines for predicting human genetic disease association. *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation* (pp. 595-598).

[33] Zhang, H., Wang, H., Dai, Z. M.-s. Chen., & Yuan, Z. (2012).Improving accuracy for cancer classification with a new algorithm for genes selection. *BMC Bioinformatics, 13*, 298.

[34] Motsinger-Reif, A. A., Dudek, S. M., Hahn, L. W., & Ritchie, M. D. (2008). Comparison of approaches for machine-learning optimization of neural networks for detecting gene-gene interactions in genetic epidemiology. *Genetic epidemiology, 32*, 325-340.

[35] Tomita, Y., Tomida, S., Hasegawa, Y., Suzuki, Y., Shirakawa, T., Kobayashi, T., & Honda, H. (2004). Artificial neural network approach for selection of susceptible single nucleotide polymorphisms and construction of prediction model on childhood allergic asthma. *BMC bioinformatics, 5*, 120.

[36] Keedwell, E., & Narayanan, A. (2005). Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2*, 231-242.

[37] Hardison, N. E., & Motsinger-Reif, A. A. (2011). The power of quantitative grammatical evolution neural networks to detect gene-gene interactions. *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (pp. 299-306).

[38] Ritchie, M. D., Motsinger, A. A., Bush, W. S., Coffey, C. S., & Moore, J. H. (2007). Genetic programming neural networks: A powerful bioinformatics tool for human genetics. *Applied Soft Computing, 7*, 471-479.

[39] Park, M. Y., & Hastie, T. (2008). Penalized logistic regression for detecting gene interactions. *Biostatistics, 9*, 30-50.

[40] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., & Daly, M. J. (2007). PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics, 81*, 559-575.

[41] Kooperberg, C., & Ruczinski, I. (2005). Identifying interacting SNPs using monte carlo logic regression. *Genetic* epidemiology, vol. 28, pp. 157-170, 2005.

[42] Nunkesser, R., Bernholt, Schwender, T. H., Ickstadt, K., & Wegener, I. (2007). Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics, 23*, 3280-3288.

[43] Schwender, H., & Ickstadt, K. (2008). Identification of SNP interactions using logic regression. *Biostatistics, 9*, 187-198.

[44] Chen, C. C., Schwender, H., Keith, J., Nunkesser, R., Mengersen, K., & Macrossan, P. (2011). Methods for identifying SNP interactions: A review on variations of logic regression, random forest and bayesian logistic regression. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8*, 1580-1591.

[45] Fritsch, A., & Ickstadt, K. (2007). Comparing logic regression based methods for identifying SNP interactions. *Bioinformatics Research and Development*, 90-103.

[46] Han, B., Chen, X.-W., & Talebizadeh, Z. (2011). FEPI-MB: Identifying SNPs-disease association using a Markov Blanket-based approach. *BMC Bioinformatics, 12*, S3.

[47] Lunn, D. J., Whittaker, J. C., & Best, N. (2006). A bayesian toolkit for genetic association studies. *Genetic Epidemiology, 30*, 231-247.

[48] Han, B., & Chen, X.-W. (2011). bNEAT: A bayesian network method for detecting epistatic interactions in genome-wide association studies. *BMC Genomics, 12*, S9.

[49] Zhang, Y., & Liu, J. S. (2007). Bayesian inference of epistatic interactions in case-control studies. *Nature Genetics, 39*, 1167-1173.

[50] Uppu, S., Krishna, A., & Gopalan, R. P. (2015). Rule-based analysis for detecting epistasis using associative classification mining. *Network Modeling Analysis in Health Informatics and Bioinformatics, 4*, 1-19.

[51] Uppu, S., Krishna, A., & Gopalan, R. P. (2016). Towards deep learning in genome-wide association interaction studies. *Proceedings of the Pacific Asia Conference on Information Systems*.

[52] Cun, Y. L., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436-444.

[53] Candel, A., Parmar, V., Dell, E. L., & Arora, A. (2015). *Deep Learning with H2O*.

[54] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097-1105.

[55] Tompson, J. J., Jain, A., Cun, Y. L., & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in Neural Information Processing Systems*.

[56] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, A. N., Vanhoucke, V., Nguyen, P., & Sainath, T. N. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, 29*, 82-97.

[57] Sutskever, I., Vinyals, O., & V. Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 3104-3112.

[58] Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., & Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature, 500*, 168-174.

[59] Bordes, A., Chopra, S., & Weston, J. (2014). Question answering with subgraph embeddings.

[60] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research, 12*, 2493-2537.

[61] Min, S., Lee, B., & Yoon, S. (2016). Deep learning in bioinformatics.

[62] Urbanowicz, R. J., Kiralis, J., Sinnott-Armstrong, N. A., Heberling, T., Fisher, J. M., & Moore, J. H. (2012). GAMETES: A fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData Mining, 5*, 1-14.

[63] Gory, J. J., Sweeney, H. C., Reif, D. M., & Motsinger-Reif, A. A. (2012). A comparison of internal model validation methods for multifactor dimensionality reduction in the case of genetic heterogeneity. *BMC Research Notes, 5*, 623.

[64] Bengio, Y., Goodfellow, I. J., & Courville, A. (2015). *Deep Learning*. An MIT Press Book in Preparation.

[65] Recht, B., Re, C., Wright, S., & Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*, 693-701.

[66] Aiello, S., Kraljevic, T., & Maj, P. (2015). Package 'h2o'."

[67] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*, 1929-1958.

[68] Arno, C. (2015). The definitive performance tuning guide for H2O deep learning. Retrieved February 26, from http://blog.h2o.ai/2015/02/deep-learning-performance/

Suneetha Uppu is a PhD student in computer science at Curtin University, Australia. She received an MS by research degree in computing from University of Technology, Sydney and a B.E in electronics engineering from Madurai Kamaraj University, India. Her research interests include bioinformatics, big data analytics and deep learning.

Aneesh Krishna is currently an senior lecturer of software engineering with the Department of Computing, Curtin University, Australia. He holds a PhD in computer science from the University of Wollongong, Australia, an M.Sc. (Eng.) in electronics engineering from Aligarh Muslim University, India and a B.E. degree in electronics engineering from Bangalore University, India. He was a lecturer in software engineering at the School of Computer Science and Software Engineering, University of Wollongong, Australia (from February 2006 - June 2009). His research interests include software engineering, requirements engineering, conceptual modelling, agent systems, formal methods, data-driven software engineering, data mining and bioinformatics. His research is (or has been) funded by the Australian Research Council and various Australian government agencies as well as companies such as Woodside Energy, Amristar Solutions, Andrew Corporation, NSW State Emergency Service, Western Australia Dementia Study Centre and Autism West. He serves as assessor (Ozreader) for the Australian Research Council. He has been on the organising committee, served as invited technical program committee member of many conferences and workshop in the areas related to his research.

Raj P. Gopalan is currently an adjunct senior research fellow in the Department of Computing, Curtin University, Australia. He has a PhD in computer science from the University of Western Australia, a MSc by research in computer science from the National University of Singapore, a post graduate diploma in management from the Indian Institute of Science, Bangalore and a BSc in mechanical engineering from Kerala University. His research interests include database systems, data mining, bioinformatics and data-driven software engineering.