Fergus, P, Montanez, C, Abdulaimma, B, Lisboa, P, Chalmers, C and Pineless, B

Utilising Deep Learning and Genome Wide Association Studies for Epistatic-Driven Preterm Birth Classification in African-American Women

Utilising Deep Learning and Genome Wide Association Studies for Epistatic-Driven Preterm Birth Classification in African-American Women

Paul Fergus, Casimiro C. Montañez, Basma Abdulaimma, Paulo Lisboa, Carl Chalmers, and Beth Pineles
Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK

Abstract—Genome-Wide Association Studies (GWAS) are used to identify statistically significant genetic variants in case-control studies. The main objective is to find single nucleotide polymorphisms (SNPs) that influence a particular phenotype (i.e. disease trait). GWAS typically use a p-value threshold of $5 * 10^{-8}$ to identify highly ranked SNPs. While this approach has proven useful for detecting disease-susceptible SNPs, evidence has shown that many of these are, in fact, false positives. Consequently, there is some ambiguity about the most suitable threshold for claiming genome-wide significance. Many believe that using lower p-values will allow us to investigate the joint epistatic interactions between SNPs and provide better insights into phenotype expression. One example that uses this approach is multifactor dimensionality reduction (MDR), which identifies combinations of SNPs that interact to influence a particular outcome. However, computational complexity is increased exponentially as a function of higher-order combinations making approaches like MDR difficult to implement. Even so, understanding epistatic interactions in complex diseases is a fundamental component for robust genotype-phenotype mapping. In this paper, we propose a novel framework that combines GWAS quality control and logistic regression with deep learning stacked autoencoders to abstract higher-order SNP interactions from large, complex genotyped data for case-control classification tasks in GWAS analysis. We focus on the challenging problem of classifying preterm births which has a strong genetic component with unexplained heritability reportedly between 20%-40%. A GWAS data set, obtained from dbGap is utilised, which contains predominantly urban low-income African-American women who had normal and preterm deliveries. Epistatic interactions from original SNP sequences were extracted through a deep learning stacked autoencoder model and used to fine-tune a classifier for discriminating between term and preterm births observations. All models are evaluated using standard binary classifier performance metrics. The findings show that important information pertaining to SNPs and epistasis can be extracted from 4666 raw SNPs generated using logistic regression (p-value=$5 * 10^{-3}$) and used to fit a highly accurate classifier model. The following results (Sen=0.9562, Spec=0.8780, Gini=0.9490, Logloss=0.5901, AUC=0.9745, and MSE=0.2010) where obtained using 50 hidden nodes and (Sen=0.9289, Spec=0.9591, Gini=0.9651, Logloss=0.3080, AUC=0.9825, and MSE=0.0942) using 500 hidden nodes. The results were compared with a Support Vector Machine (SVM), a Random Forest (RF) and a Fishers Linear Discriminant Analysis classifier, which all failed to improve on the deep learning approach.

Index Terms—Preterm Birth, GWAS, Epistasis, Classification, Stacked Autoencoders, Deep Learning, Machine Learning

1 Introduction

Preterm birth (PTB) is the delivery of live babies born before 37 weeks of gestation [1]. In contrast, term births are the live delivery of babies born between 37 and 42 weeks. In 2010, the World Health Organisation (WHO) declared that preterm deliveries accounted for 1 in 10 births worldwide [1]. Compared with Caucasians, the risk of preterm birth in African-Americans is 1.5 times higher. This group also has an even greater risk of giving birth before 32 weeks gestation [2]. Population-specific risk factors include anaemia during pregnancy, low serum folate levels, vitamin D deficiency, poor weight gain during pregnancy, and high pregnancy body mass index (BMI) [3].

PTB has significant adverse effects on newborns. The severity increases the more premature the delivery is. Approximately, 50% of all perinatal deaths are caused by preterm delivery. For those that survive, they often suffer impairments in hearing and vision and from chronic respiratory diseases. Up to 40% of survivors of extremely premature birth can develop chronic lung disease [4]. In other cases, survivors suffer from neuro-developmental or behavioural defects, including cerebral palsy, motor, learning and cognitive impairments.

The precise etiology of PTB remains elusive. However, 30%-35% are known to be medically indicated (i.e. preeclampsia and foetal growth restriction) [5]. Preterm prelabor ruptured membranes (PPROMs - often attributed to infection, placental abruption, and anatomical abnormalities in the mother) account for 25%-30% [5] and spontaneous PTB (sPTB) for the remaining 35%-45% where the cause is unclear [6].

A strong body of evidence, from twin-based studies, has shown that maternal and foetal genetic factors contribute to PTB with heritability between 20%-40% [7], [8], [9]. Though attempts to identify the specific variant(s) of prematurity in genome-wide association studies (GWAS) have failed to produce any reproducible findings [10]. Several GWAS stud-

- *Paul Fergus, Casimiro C. Montanez, Basma Abdulaimma, Paulo Lisboa, and Carl Chalmers are with the Faculty of Engineering and Technology Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK. (E-mail: p.fergus@ljmu.ac.uk)*
- *Beth Pineles is with the University of Maryland Medical Center, 22 S. Greene Street, Baltimore, MD 21201, USA.*

ies have identified notable relationships but meta-analysis has shown that these are often negligible or contained within a particular population [11], [12].

Associations can be measured using Bonferroni correction, which is a highly conservative threshold designed to minimise type 1 errors in multiple testing studies. This leads to missing heritability were single genetic variations cannot fully explain the heritability of phenotypes [13], [14]. Among the many approaches that exist, multifactor dimensionality reduction (MDR) has found that single nucleotide polymorphisms (SNPs) with little individual effect, through their interactions, can account for more variance in phenotypes [15], [16], [17]. Random forest algorithms have also been heavily utilised to detect significant SNPs in large-scale GWAS [18], [19], [20], [21], [22]. However, enumerating the large number of high-order combinations common in genetics is computationally very difficult to implement and a major limitation in these approaches. This issue has been mitigated by applying filters to select groups of SNPs that are relevant to the phenotype of interest [23], for example, using PLINK [24] which also provides two SNP epistatic analysis. Larger combinations are possible using LAMPLINK [25], but scalability issues still persist.

In this paper, we combine the quality control and logistic regression functions in GWAS with deep learning stacked autoencoders (DL) [26] to create a novel framework for extracting epistatic interactions between SNPs [27]. A multilayer feedforward softmax classifier is initialised using the generated deep learning stacked autoencoder model and fine-tuned to classify case and control birth outcomes. The complete network models the epistatic effects of major and minor SNP perturbations.

Deep learning is used in [28] to select regulatory SNPs with functional impact before association analysis is conducted (DeepWAS). Variants (SNPs) that alter functional regulatory elements, i.e. elements that control gene expression and DNA methylation, are identified using DeepSEA [29] before association analysis. This approach differs to the approach presented in this paper, in that, QC and GWAS are conducted using all of the SNPs genotyped in the preterm birth study dataset. Pre-SNP selection, based on functional regulatory effects, is not applied since our aim is to find epistatic interactions between SNPs. While DeepWAS concentrates more on biological outcomes (including regulatory mechanisms in GWAS), this paper focuses on testing new algorithms for epistatic interactions and classification analysis.

The results in this paper are compared with a multilayer feedforward neural network, Support Vector Machine, Random Forest and a Fishers Linear Discriminant Analysis classifier, trained and tested using Bonferroni and suggestive p-value SNPs, to assess their predictive capacity. Our approach shows significant improvements and is the first comprehensive study of its kind that combines GWAS analysis with deep learning stacked autoencoders for epistatic-driven GWAS analysis and case-control classification.

The remainder of this paper is organised as follows. Section 2 describes the Materials and Methods used in the study. The results are presented in Section 3 and discussed in Section 4 before the paper is concluded and future work is presented in Section 5.

2 Materials and Methods

The data, for this study, was obtained through authorised access to dbGap (Study Accession: phs000332.v3.p2) [30]. The dataset includes 722 cases and 1057 controls. Cases were drawn from deliveries at the Boston Medical Center (BMC) that occurred before 37 weeks of gestation irrespective of birth weight. Controls include mothers who delivered term babies after 37 weeks of gestation also from the BMC cohort. Controls were frequency matched with case mothers on race, age ($\pm$ 5 years), parity and the baby's gender. Women were excluded if pregnancies were due to vitro fertilisation, they had multiple pregnancies, or the foetus had chromosomal abnormalities or major birth defects. Further exclusion criteria included mothers who had congenital or acquired uterus lesions, a known history of an incompetent cervix, or previous PTBs caused by maternal trauma. Each subject was interviewed using a standardised questionnaire to gather important epidemiological data, including ultrasound findings, placental pathology reports, laboratory reports, information on pregnancy complications and birth outcomes.

2.1 Data Collection

The GWAS recruited 1000 mothers who delivered preterm and 1000 age-matched mothers who had term births (African-American - 68%; Haitian - 31.5%). The subjects were genotyped in two phases. The first phase was completed in 2011 and the second in 2014. For all study samples, the Qiagen method was used to extract DNA from whole blood. In each phase cases and controls were balanced across 96-well plates and each plate contained between two and four HapMap controls, as well as an average of two study duplicates. Phase 1 was genotyped using the Illumina HumanOmni2.5-4v1 array and using the calling algorithm GenomeStudio version 2-10.2, Genotyping Module version 1.74 and GenTrain version 1.0. Phase 2 was genotyped using the Illumina HumanOmni2.5-8v1 array and using the calling algorithm GenomeStudio version 2011.1, Genotyping Module version 1.9.4 and GenTrain version 1.0. The two phases were merged into a single dataset with 2,369,543 probes common to both arrays.

A total of 1,910 observations (including duplicates) from study subjects were put into genotype production, of which 1,889 were successfully genotyped and passed the Center for Inherited Disease Research (CIDR's) quality control (QC) process. The subsequent quality assurance (QA) procedure removed five observations, and the final set of scans posted to dbGAP included 1,884 study participants and 62 HapMap controls. The 1,884 study observations were derived from 1808 subjects and include 76 pairs of duplicate scans. The 62 HapMap control scans were derived from 24 subjects, all of which were replicated two or more times. The study subjects occur as 1,681 singletons and 60 families of 2-4 members each. The study families were discovered during the analysis of relatedness. The HapMap controls include 8 trios (4 CEU, 4 YRI).

2.2 Quality Control

The dataset was subjected to pre-established QC protocols as recommended in [31], where data QC was applied to

individuals first and then to genetic variants. PLINK v1.9 [24] was used on a Linux Ubuntu machine, version 16.04 LTS, with 16GiB of Memory and an Intel Core I7-7500U CPU @ 2.70HHz x 4, to conduct the required QC and filtering procedures. Before QC, the 24 HapMap controls and the 0 Chromosome were removed from the data.

Individual QC: Individuals with discordant sex information (homozygosity rate between 0.2 and 0.8) were identified using the X-chromosome and ascertained sex. This resulted in eight individuals being removed. Individuals with elevated missing data rates were identified using a genotype failure rate ≥ 0.02 (seven individuals were removed). While, outlying heterozygosity was identified using a heterozygosity rate ± 3 standard deviations from the mean (16 individuals were removed). Pairs of individuals with identity by descent (IBD) > 0.185 were removed (38 individuals). Principle Component Analysis (PCA) was conducted for the identification of outliers and hidden population structure using EIGENSOFT [32]. Individuals were identified with divergent ancestry using thresholds -0.05 and 0.00 for principal component (PC) 1 and 2 respectively. This resulted in 289 individuals being removed using the PC1 threshold and 297 using the PC2 threshold. All unique missing markers were combined and excluded from the data set reducing the total number of individuals to 1527 (Case=632, Control=895) with the genotyping rate in remaining samples equal to 0.992308.

Marker QC: Each individual contains 2,362,044 SNPs. SNPs with a significantly different ($p < 1 * 10^{-5}$) missing data rate between cases and controls were removed (n=22603). SNPs with minor allele frequency (MAF $< 1\%$), call rate <98% and deviations from Hardy-Weinberg equilibrium ($p < 1 * 10^{-5}$) were excluded. The data set following QC resulted in 1527 individuals with 1,927,820 variants each.

2.3 Association Analysis

In this study, association analysis is used to reduce the computationally large number of SNPs (1,927,820) for epistatic analysis and machine learning tasks. Several p-value thresholds are considered that range between $5 * 10^{-3}$ and $5 * 10^{-8}$ inclusive - $5 * 10^{-8}$ being the Bonferroni correction [33]. The resulting groups containing between 3 and 4666 SNPS (depending on the threshold) are used to train and baseline classifier models and assess the predictive capacity in detecting case and control instances. These models are compared with deep learning stacked autoencoder models with progressively smaller layers of hidden nodes and weights that capture epistatic interactions between SNPs using 4666 SNPs (obtained using $5 * 10^{-3}$).

2.3.1 Association Testing

Using a standard association analysis procedure, $\{X_1, \ldots, X_u\}$ describe a set of *U* SNPs for *N* individuals, and $\{y_1, \ldots, y_n\}$ describe the phenotypes. In this study only one phenotype is considered (preterm birth), therefore only $\{y_1\}$ is used. For each SNP, there is a minor allele *a* and major allele *A*. The homozygous major allele is defined as *AA*, the heterozygous allele as *Aa* and the homozygous minor allele as *aa* - 0, 1, and 2 are used to describe these respectively. Therefore, $X_{un} \in \{0,1,2\}$, $(1 \leq u \leq U,\ 1 \leq n \leq N)$. The phenotype is represented as a binary variable, 0 for controls and 1 for cases.

Genotypes are grouped into an additive model. Given A we assume that there is a uniform, linear increase in risk for each copy of the A allele. For example, if the risk is 3x for Aa then the risk is 6x for AA. The additive model is only considered in this study as it has satisfactory power in detecting additive and dominant effects.

2.3.2 Logistic Regression

Logistic regression [34] is used to assess which SNPs increase the odds of a given outcome (in this study a preterm birth). This is performed under an additive model where logistic regression modelling for conditional probability $Y = 1$ is: [35]:

$$\theta(X) = P(Y = 1|X) \tag{1}$$

The logit function [36] which is the inverse of the sigmoidal logistic function, is represented as:

$$logit(X) = ln\frac{\theta(X)}{1 - \theta(X)} \tag{2}$$

The logit is given as a linear predictor function as follows:

$$logit(X)\ \beta_0 + \beta_1 X \tag{3}$$

Utilising logistic regression, while not ideal, enables the number of SNPs with insignificant marginal effects to be reduced to meet the computational needs required for epistatic analysis and machine learning tasks. The remaining SNPs capture the linear associations between SNPs and the phenotype but not the cumulative epistatic interactions that exist between the remaining SNPs. To capture epistatic interactions, we utilise a softmax classifier model pre-initialised with a deep learning stacked autoencoder.

2.4 Multilayer Perceptron

A multilayer perceptron neural network (MLP) is implemented in this study for classification tasks and is based on the work of [37]. A rectifier nonlinear activation function [38] is utilised which provides better training for deep learning networks [39], when compared with logistic sigmoid and the hyperbolic tangent.

The MLP is initialised with small random values and a small learning rate of 0.005 is used with Backpropagation as the learning algorithm and stochastic gradient descent as the optimiser. Small learning rates do increase the training time, however, it reduces the number of oscilations and generates a lower error value which were key performance metrics in this study. Rate annealing is applied to address learning rate freezing in local minima. While rate decay is applied to control learning rate change across layers.

Momentum start is set to 0.5 and momentum ramp and momentum stable to $1 * 10^{-6}$ and 0 respectively to control the amount of momentum at the beginning of training and the amount of learning for which momentum increases. Momentum stable is used to control the final momentum value reached after momentum ramp training examples. Complexity is controlled using weight decay where the decay parameter is optimised through cross-validation. This ensures that a local optimum is found using small-magnitude parameters to avoid overfitting.

Several tests were performed to determine the network topology, in terms of the number of neurons and hidden layers required to provide optimal error rates. For the MLP, the best performance was obtained using a variable number of input neurons (see results for further details), four hidden layers, with ten nodes in each, and 1 output node. It was found that increasing the number of hidden layers or the number of neurons did not improve the performance significantly.

One hundred epochs is used to train the MLP (the results show that this number was sufficient for network convergence) with early stopping (when misclassification rate converges) to avoid overfitting - training and validation sets are used to obtain and optimal model and a separate test set is utilised to validate the models performance on unseen data.

2.5 Deep Learning Stacked Autoencoders

A stacked autoencoder (SAE) is implemented based on [27] to further reduce the dimensionality of the subset of SNPs generated using logistic regression (p-value threshold ($5 * 10^{-3}$ - 4666 SNPs). The primary goal is to extract the epistatic interactions between the 4666 SNPs for classifier modelling. The optimal hidden layer units are utilised to achieve this, such that the output $\hat{x}$ is similar to the input x:

$$h_{W,b}(x) \approx x \tag{4}$$

The hidden unit activations a^2 in $\mathbb{R}^h$ aim to reconstruct the input x. If there is structure in the data, the autoencoder will learn it. In fact, very simple autoencoders often learn a low-dimensional representation similar to principal component analysis (PCA).

Nodes in the autoencoder fire when output values are close to 1 and remain inactive when the output is close to 0. The goal is to ensure that nodes remain mostly inactive. Thus, the activation of a hidden unit is represented as $a_j^{(2)}(x)$ when the network receives input x. We let:

$$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right] \tag{5}$$

represent the average activation of hidden unit j and enforce the constraint:

$$\hat{p}_j = p \tag{6}$$

where p is typically a small sparsity parameter close to zero (for example, $p = 0.05$). In order to meet this constraint, the activation of the hidden unit should be mostly 0. To achieve this a penalty term is added that penalizes $\hat{p}_j$ when it deviates significantly from p:

$$\sum_{j=1}^{s_2} p\, log\frac{p}{\hat{p}_j} + (1-p)log\frac{1-p}{1-\hat{p}_j} \tag{7}$$

where s_2 is the number of units in the hidden layer, and j an index used to sum the hidden units in the network. Kullback-Leibler (KL) divergence is used to enforce this penalty term:

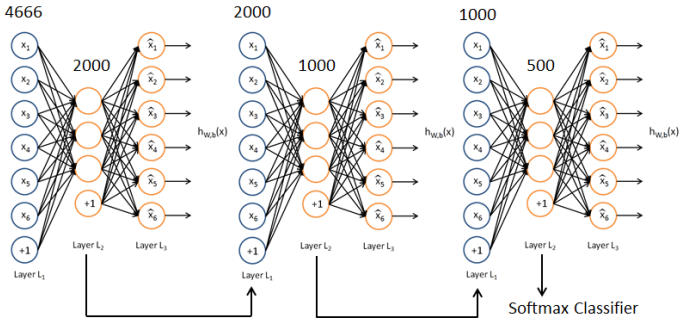


Fig. 1: Stacked Autoencoder that compresses 4666 SNPs to 500 features

.

$$\sum_{j=1}^{s_2} KL(p||\hat{p}_j) \tag{8}$$

where $KL(p||p_j) = \frac{p}{\hat{p}_j} + (1-p)log\frac{1-p}{1-\hat{p}_j}$ is the KL-divergence between two Bernoulli random variables with mean p and $\hat{p}_j$. In this way KL-divergence is used to measure the difference between two distributions. This penalty function is either $KL(p||\hat{p}_j) = 0$ if $\hat{p} = p$, or it increases monotonically as $\hat{p}_j$ diverges from p. The cost function can now be defined as:

$$J_{sparse}(W,b) = J(W,b) + \beta\sum_{j=1}^{s_2} KL(p||\hat{p}_j) \tag{9}$$

where $J(W,b)$ is the same as we previously defined and β is used to control the weight of the sparsity penalty term. The term $\hat{p}_j$ is dependent on W, b, as it is the average activation of hidden unit j, and hidden unit activations are dependent on the parameters W, b.

The KL-divergence term is incorporated into the previously defined derivative calculation and now computed as:

$$\delta_i^{(2)} = \Big(\sum_{j=1}^{s_2} W_{ji}^{(2)}\delta_j^{(3)}\Big)f'(z_i^{(2)}) + \beta\Big(-\frac{p}{\hat{p}_i} + \frac{1-p}{1-\hat{p}_i}\Big) \tag{10}$$

It is important to know $\hat{p}_i$ to compute this term. After computing $\hat{p}_i$, a forward pass on each example is performed to allow backpropagation on that example. Therefore, you compute a forward pass twice on each example in your training set, which does make it computationally less efficient.

A single autoencoder is simple, due to its shallow structure. Consequently, a single-layer autoencoder's representational power is very limited. In this study, autoencoders are stacked to enable greedy layer wise deep learning where the i_{th} hidden layer is used as input to the i+1 hidden layer in the stack. The results produced by the stacked autoencoder are utilized to pretrain (initialize) the weights for our MLP (softmax model), rather than randomly initialising the weights to small values, to classify term and preterm deliveries. Figure 1 shows a simple stacked autoencoder configuration that compresses 4666 SNPs to 500 features by linking hidden layer units to input units in subsequent autoencoders in the stack.

This concludes the methods used in this study and provides the basis for a novel framework that combines GWAS quality control and logistic regression with deep learning stacked autoencoders to abstract higher-order SNP interactions from large, complex genotyped data for case-control classification tasks in GWAS analysis. On these grounds we claim that this is the first comprehensive study of its kind.

2.5.1 Performance Measures

Sensitivity (or Recall), specificity and precision (or Positive Predicted Value) are used in this study to represent the number of correctly identified case and control instances. Sensitivity describes the true positive rate (Controls - term deliveries) and Specificity the true negative rate (Cases - preterm deliveries). Precision on the other hand describes the number of correct predictions among retrieved instances.

The area under the curve (AUC) is the probability that, for each pair of examples, one for each class, the example from the positive class will be ranked highest. This is measured by ranking the estimates of posterior class membership $p(C_i|x)$ in increasing order. If S_0 is the sum of the ranks of values of inferences for test data in class C_1, and similarly for class C_2 the AUC is givn by:

$$\hat{A} = \frac{1}{n_1 n_2}\left(S_0 - \frac{1}{2}n_1(n_1+1)\right) \tag{11}$$

where n_1 and n_2 are the sample sizes in each class [40].

The Gini coefficient is often used in binary classification studies and is closely related to the AUC. The Gini coefficient is defined as being the area between the diagonal and the ROC curve:

$$Gini = 2 * AUC - 1 \tag{12}$$

The Gini coefficient measures statistical dispersion. The SNP(s) with a Gini coefficient of 1, predicts the data perfectly. A coefficient of 0 indicates that the SNP(s) have no predictive capacity.

Log Loss provides a measure of accuracy for a classifier whereby penalties are imposed on classifications that are false. Minimising the Log Loss is correlated with accuracy (as one increases the other decreases). Log loss is calculated by assigning a probability to each class rather than stating what the most likely class would be:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1-y_i)log(1-p_i)]. \tag{13}$$

where N is the number of samples, y_i is a binary indicator for whether j correctly classifies instance i. For models that classify all instances correctly the Log Loss value will be zero. For misclassifications, the Log Loss value will be progressively larger.

The Mean Squared Error (MSE) metric is utilised to measure the average sum of the square difference between actual values and predicted values for all data points. A MSE value of 0 indicates that the model correctly classifies all class instances. Again, for misclassifications, the MSE will be progressively larger.

3 Results

We first present the results using a multi-layer feedforward neural network classification model comprising four hidden layers with 10 neurons in each to provide baseline results. Several association analysis p-value filters are considered for dimensionality reduction - resulting SNP combinations are used to fit our classifier models. The performance of each model is measured using Sensitivity, Specificity, Gini, AUC, LogLoss and MSE values. The data set is split randomly into training (80%), validation (10%) and testing (10%).

3.1 Baseline Multi-Layer Feedforward Neural Network

3.1.1 Classifier Performance

Table 1 provides the performance metrics for the validation set. Metric values for association analysis p-values $5*10^{-3}$, $5*10^{-4}$, $5*10^{-5}$, $5*10^{-6}$, $5*10^{-7}$, and $5*10^{-8}$) were obtained using optimized F1 threshold values 0.6840 (resulting in 4666 SNPs), 0.6039435 (419 SNPs), 0.2799 (51 SNPs), 0.4471 (11 SNPs), 0.4107814 (11 SNPs) and 0.4450064 (three SNPs), respectively.

TABLE 1: Performance for Validation Set

p-value	Sens	Spec	Gini	LogLoss	AUC	MSE
$5*10^{-3}$	0.9848	1.0000	0.9993	0.1002	0.9996	0.0150
$5*10^{-4}$	0.9696	0.9285	0.9700	0.2988	0.9850	0.0597
$5*10^{-5}$	0.7121	0.7959	0.6020	0.5679	0.8010	0.1928
$5*10^{-6}$	0.9242	0.3673	0.3766	0.6669	0.6883	0.2369
$5*10^{-7}$	0.9393	0.3265	0.3722	0.6507	0.6861	0.2290
$5*10^{-8}$	0.8484	0.2959	0.2719	0.6745	0.6359	0.2407

Table 2 shows the performance metrics obtained using the trained models and the test data. The network comprised four hidden layers each containing 10 nodes. Based on empirical analysis this configuration produced the best results. Metric values for association analysis p-values $5*10^{-3}$, $5*10^{-4}$, $5*10^{-5}$, $5*10^{-6}$, $5*10^{-7}$, and $5*10^{-8}$ were again obtained using an optimized F1 threshold with values 0.7350, 0.3144, 0.2799, 0.4546975, 0.42307, and 0.4534978 respectively. The results are lower than those obtained by the validation set but in some cases not by much.

TABLE 2: Performance for Test Set

p-value	Sens	Spec	Gini	LogLoss	AUC	MSE
$5*10^{-3}$	1.0000	0.9882	0.9996	0.0960	0.9998	0.0128
$5*10^{-4}$	0.9000	0.9411	0.9388	0.3038	0.9694	0.0673
$5*10^{-5}$	0.9666	0.5411	0.6709	0.5581	0.8354	0.1913
$5*10^{-6}$	0.9333	0.3673	0.3766	0.6669	0.6883	0.2369
$5*10^{-7}$	0.9166	0.4352	0.4833	0.6374	0.7416	0.2225
$5*10^{-8}$	0.8833	0.4117	0.3572	0.6679	0.6786	0.2374

Figure 2 demonstrates that overfitting is appropriate managed (p-values $5*10^{-7}$ and $5*10^{-8}$ were omitted as the figure demonstrates a sharp deterioration in performance as p-value thresholds increase). Epochs represent the inflection points where performance on the validation set starts to decrease while performance on the training set continues to improve as the model starts to overfit. An optimised

loss function is adopted to train the models. The AUC plots provide useful information about early divergence between the training and validation curves and highlight if overfitting occurs. From Figure 2, clearly there is a small amount of overfitting but nothing in excess.

3.1.2 Model Selection

The ROC curve in Figure 3 shows the cut-off values for the false and true positive rates using the test set. In this first evaluation, Figure 3 shows a clear deterioration in performance as p-value thresholds increase. In this instance, machine learning demonstrates that highly ranked SNPs do not have sufficient predictive capacity to make distinctions between case-control observations.

3.2 Deep Learning Stacked Sparse Autoencoder

In comparison, the following evaluation uses SNPs generated with a p-value $5 * 10^{-3}$. Latent features are extracted from 4666 SNPs with a deep learning stacked autoencoders that capture information about important SNPs and the cumulative epistatic interactions between them. This is achieved layer-wise by stacking simpler autoencoders that each contain a single hidden layer with 2000, 1000, 500, 200, 100 and 50 hidden nodes respectively as shown in Figure 1. Softmax classifiers (multilayer perceptron) are initialized with each of these layers and fine-tuned to classify case-control instances in the validation and test sets using four hidden layers with 10 nodes each.

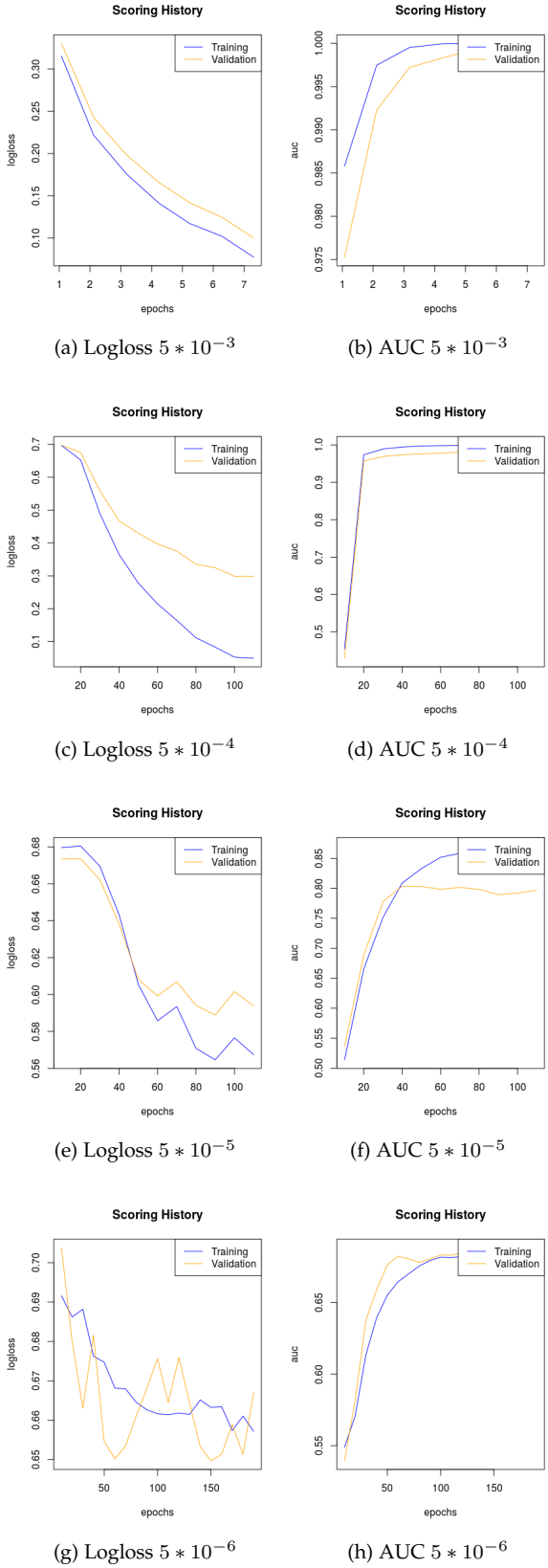
3.2.1 Classifier Performance

With the first layer (2000 neurons) a softmax classifier model is initialised and then fine-tuned. The learning rate is set to $1 * 10^{-3}$ and an optimized F1 value of 0.7374 is used to extract metric values for the validation set as shown in Table 3. Subsequent layers are used to initialise and fine tune the remaining models with 1000, 500, 200, 100 and 50 hidden units respectively (note that the hidden layers are linked to form a stack as illustrated in Figure 1). Metrics were obtained from these models using optimised F1 values 0.2979, 0.0769, 0.5881, 0.4996, and 0.6178 respectively. The learning rate for each of the layers is set to $1*10^{-3}$, $1*10^{-4}$, $1 * 10^{-5}$, $1 * 10^{-5}$, and $1 * 10^{-6}$. The full results are shown in Table 3.

TABLE 3: Performance Metrics for Validation Set

Comp	Sens	Spec	Gini	LogLoss	AUC	MSE
2000	0.9482	0.9772	0.9764	0.1273	0.9882	0.0331
1000	0.9827	0.9659	0.9698	0.1246	0.9849	0.0270
500	0.9827	0.8750	0.9674	0.3059	0.9837	0.0962
200	0.8965	0.9545	0.9365	0.3752	0.9682	0.1098
100	0.9482	0.7840	0.8475	0.6059	0.9237	0.2068
50	0.9655	0.9545	0.9518	0.5854	0.9759	0.1988

Table 4 shows the performance metrics obtained using the test set. Layers containing 2000, 1000, 500, 200, 100, and 50 were again used with optimized F1 values 0.5836, 0.1695, 0.2348, 0.6036, 0.5061, and 0.5457 respectively. The learning rate values from the validation set were retained. The results



(a) Logloss $5 * 10^{-3}$ (b) AUC $5 * 10^{-3}$

(c) Logloss $5 * 10^{-4}$ (d) AUC $5 * 10^{-4}$

(e) Logloss $5 * 10^{-5}$ (f) AUC $5 * 10^{-5}$

(g) Logloss $5 * 10^{-6}$ (h) AUC $5 * 10^{-6}$

Fig. 2: (a) to (h) Logloss and AUC plots against epochs for p-value $5 * 10^{-3}$ to $5 * 10^{-6}$.

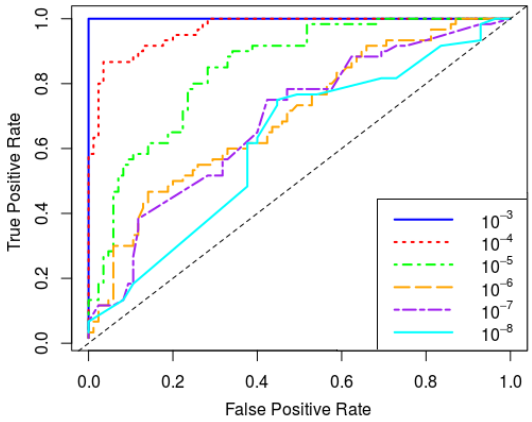


Fig. 3: ROC curves for test set using p-value between $5{*}10^{-7}$ and $5 * 10^{-8}$.

TABLE 4: Performance Metrics for Test Set

Comp	Sens	Spec	Gini	LogLoss	AUC	MSE
2000	0.9672	0.9771	0.9939	0.0850	0.9969	0.0226
1000	0.9781	0.9505	0.9736	0.1352	0.9868	0.0335
500	0.9289	0.9581	0.9651	0.3080	0.9825	0.0942
200	0.8797	0.8897	0.9055	0.3920	0.9527	0.1178
100	0.8907	0.8593	0.8544	0.5970	0.9272	0.2024
50	0.9562	0.878	0.9490	0.5901	0.9745	0.2010

are lower than those achieved with the validation set but in some cases not by much.

Early stopping was again adopted to avoid overfitting as shown in Figure 4 (for brevity only layers 2000 to 200 are illustrated to show overfitting is appropriately managed). Again, there is a small amount of overfitting but nothing significant.

3.2.2 *Model Selection*

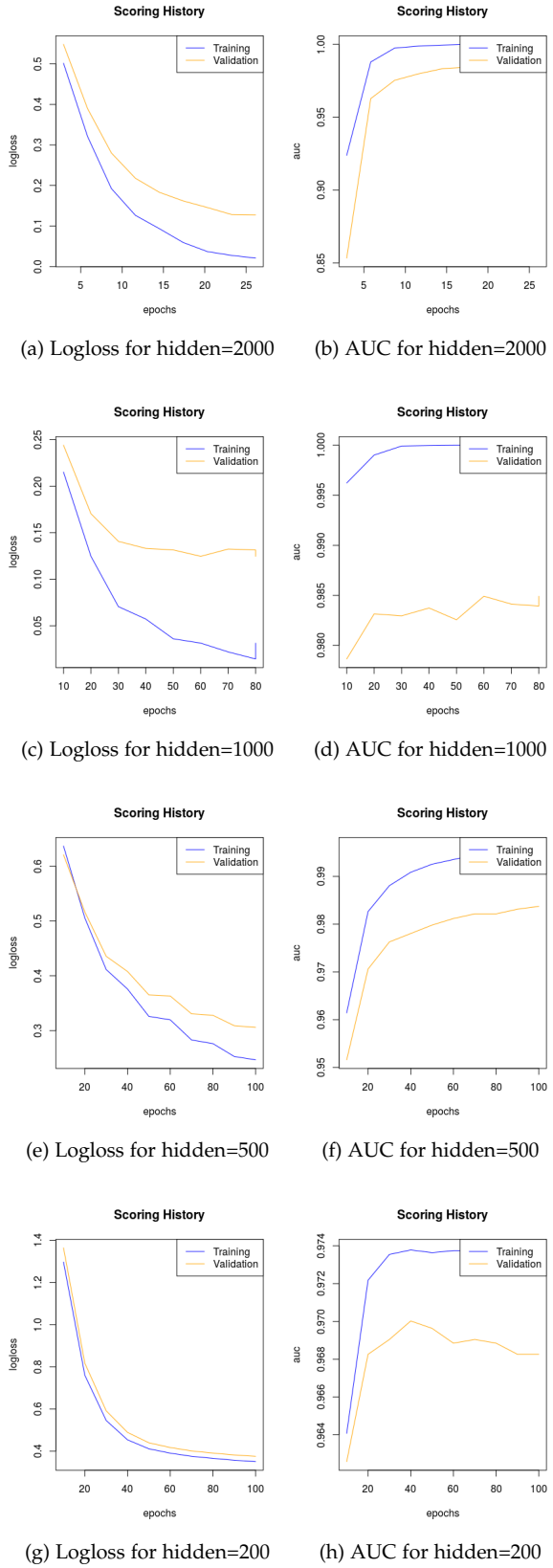
This time the ROC curve in Figure 5 shows significant improvements using the latent information captured in the hidden layers. There is obvious deterioration, however, the results still remain high at 50 and only slightly worse that the results produced when 2000 hidden units are used.

3.3 Comparison with Support Vector Machine, Random Forest and Linear Discriminant Analysis Classifiers

This section compares the results with traditional classifier models to determine whether simpler and less computationally expensive machine learning models are able to improve on or match the results previously obtained.

3.3.1 *Classifier Performance*

In this first comparison, an SVM probability model is used for classification which fits a logistic distribution using maximum likelihood to the decision values of any binary classifier. The same data splitting strategy is adopted; training (80%), validation (10%) and testing (10%). A radial kernel function is used with tuned gamma and cost parameters 0.3333 and 1 respectively.



(a) Logloss for hidden=2000 (b) AUC for hidden=2000

(c) Logloss for hidden=1000 (d) AUC for hidden=1000

(e) Logloss for hidden=500 (f) AUC for hidden=500

(g) Logloss for hidden=200 (h) AUC for hidden=200

Fig. 4: (a) to (h) Logloss and AUC plots against epochs for 2000 to 200 Compression.

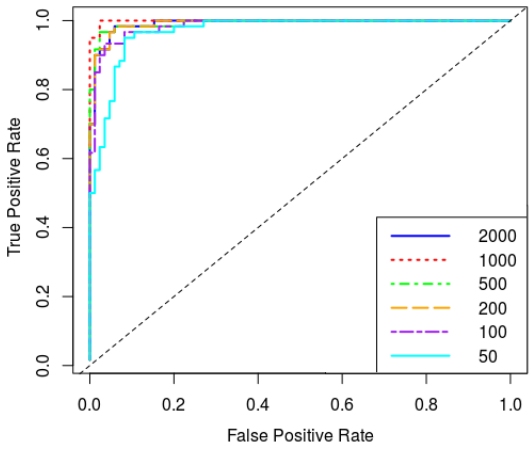


Fig. 5: Performance ROC curve for test set using hidden nodes between 2000 and 50.

Table 5 provides the performance metrics for the test set with p-value thresholds $5*10^{-3}$, $5*10^{-4}$, $5*10^{-5}$, $5*10^{-6}$, $5*10^{-7}$, and $5*10^{-8}$.

TABLE 5: SVM Performance Metrics for Test Set

p-value	Sens	Spec	AUC	Sens	Spec	PrAUC
$5*10^{-3}$	0.9761	0.9720	0.9741	0.9831	0.9721	0.9525
$5*10^{-4}$	0.8603	0.8571	0.8587	0.8953	0.8603	0.7758
$5*10^{-5}$	0.8603	0.3571	0.6087	0.6553	0.8603	0.5352
$5*10^{-6}$	0.7580	0.4017	0.5798	0.6296	0.7580	0.5012
$5*10^{-7}$	0.7688	0.3445	0.5566	0.6119	0.7688	0.4800
$5*10^{-8}$	0.7866	0.3388	0.5627	0.6172	0.7866	0.4856

The second comparison uses Breiman's RF ensemble learning classifier that decorrelates trees generated using bootstrapped training samples. In this evaluation 500 trees are grow.

Table 6 provides the performance metrics for the test set with p-values $5*10^{-3}$, $5*10^{-4}$, $5*10^{-5}$, $5*10^{-6}$, $5*10^{-7}$, and $5*10^{-8}$.

TABLE 6: RF Performance Metrics for Test Set

p-value	Sens	Spec	AUC	Sens	Spec	PrAUC
$5*10^{-3}$	0.9944	0.4603	0.7274	0.7236	0.9944	0.7519
$5*10^{-4}$	0.9274	0.5079	0.7177	0.7281	0.9274	0.6839
$5*10^{-5}$	0.8827	0.3095	0.5961	0.6449	0.8827	0.5274
$5*10^{-6}$	0.7771	0.4530	0.6115	0.6559	0.7771	0.5369
$5*10^{-7}$	0.8187	0.2773	0.5480	0.6037	0.8187	0.4762
$5*10^{-8}$	0.7744	0.3554	0.5649	0.6195	0.7744	0.4864

The final comparison uses the FLDA algorithm to find linear combinations of features to determine the direction along which the two classes are best separated. The criterion used is the ratio of between-class to within-class variances. The data is projected onto a line, and classification is performed in this one-dimensional space where the projection maximizes the distance between the means of the two classes while minimizing the variance within each class.

Table 7 provides the performance metrics for the test set. As in the previous two experiments, p-value thresholds $5*10^{-3}$, $5*10^{-4}$, $5*10^{-5}$, $5*10^{-6}$, $5*10^{-7}$, and $5*10^{-8}$ were used.

TABLE 7: LDA Performance Metrics for Test Set

p-value	Sens	Spec	AUC	Sens	Spec	PrAUC
$5*10^{-3}$	0.8826	0.8174	0.8501	0.8729	0.8827	0.9079
$5*10^{-4}$	0.8491	0.7936	0.8214	0.8539	0.8492	0.8991
$5*10^{-5}$	0.7877	0.4126	0.6002	0.6558	0.7877	0.5300
$5*10^{-6}$	0.7452	0.4103	0.5777	0.6425	0.7898	0.5321
$5*10^{-7}$	0.8375	0.3866	0.5671	0.6381	0.8375.	0.6283
$5*10^{-8}$	0.8475	0.3305	0.5891	0.6318	0.8476	0.5278

4 Discussion

In this paper, we proposed a novel framework that combines GWAS quality control and logistic regression with deep learning stacked autoencoders to abstract higher-order SNP interactions from large, complex genotyped sequence data for case-control classification tasks in preterm birth GWAS analysis. The findings are encouraging. One important advantage deep learning has is its ability to abstract large, complex and unstructured data into latent representations that capture important information about SNPs and the epistatic interactions between them. This offers a powerful way to analyse GWAS data. Feature extraction is performed as a single unified process using stacked autoencoders where multiple layers capture nonlinear dependencies and epistatic interactions between SNPs. These features do not differ when presented with small input changes. Consequently, this has the effect of eliminating noise and increasing robustness within the feature extraction process.

While, GWAS is useful for locating common variants of small effect and identifying very rare variants of much larger effect, they fail to classify phenotypes using suggestive or Bonferroni significance genome-wide associations. This is primarily caused by the fact that highly ranked SNPs are often false positives. Therefore, it is generally agreed that it may be possible to increase the proportion of variation captured in GWAS by incorporating information from rarer SNPs. Methods, such as MDR have attempted this, but they have been plagued by computational challenges.

Using a multilayer perceptron classifier model and a p-value threshold of $5*10^{-3}$ (4666 SNPs) it was possible to obtain good results (Sens=1, Spec=0.9882, Gini=0.9996, Log Loss=0.0960, AUC=0.9998, MSE=0.0128) using the test set. However, when the Bonferroni threshold ($5*10^{-8}$ - 4 SNPs) is used, the results significantly drop (Sens=0.8833, Spec=0.4117, Gini=0.3572, Log Loss=0.6679, AUC=0.6786, MSE=0.2374). Clearly analysing single loci and their effect on the polygenic phenotype fails to capture the accumulative effects of less significant SNPs and their contribution to the outcome.

Investigating this idea further, a deep learning stacked autoencoder was utilised to extract the latent information from the 4666 SNPs through progressively smaller layers (2000, 1000, 500, 200, 100 and 50 nodes). The results using the test set showed significant improvement in classification accuracies. The best result was achieved using 2000 features (Sens=0.9672, Spec=0.9771, Gini=0.9939, Log Loss=0.0850,

AUC= 0.9969, MSE=0.0226). These results are comparable to those produced using the 4666 SNPs extracted using logistic regression and p-value ($5 * 10^{-3}$). The worst results were (Sens=0.9562, Spec=0.8780, Gini=0.9490, Log Loss=0.5701, AUC= 0.9745, MSE=0.2010) when 50 features were used. Nonetheless, the results were significantly better than using the SNPs generated using logistic regression and a p-value of $5*10^{-5}$ (51 SNPs - Sens=0.9666, Spec=0.5411, Gini=0.6709, Log Loss=0.5581, AUC= 0.8354, MSE=0.1913). The Sensitivity value was slightly lower. However, Specificity increased by 34%, Gini by 28%, while LogLoss remained broadly the same. The AUC increased by 14% and the MSE was slightly less. The results when the input set was compressed to 1000 features produced comparable results to those when logistic regression was used with a p-value threshold of $5 * 10^{-3}$ (4666 SNPs).

In the final set of evaluations several traditional machine learning algorithms were considered, more specifically an SVM, RF and FLDA classifier, to determine whether these less complex models could outperform the framework posited in this paper. The same test protocol as the multi-layer feedforward neural networks was adopted with the same complement of p-value thresholds. The results show that the best performing classifier was the SVM using the p-value threshold $5 * 10^{-3}$ with (Sens=0.9761, Spec=0.9720, and a AUC= 0.9741) - we also used precision-recall metrics to accommodate for slight class imbalance with values (Prec=0.9831, Rec=0.9721, and a PrAUC= 0.9525). These results were more or less comparable with all other evaluations performed. The best results obtained using the p-value threshold $5 * 10^{-8}$ was achieved with the FLDA classifier with (Sens=0.8475, Spec=0.3305, and a AUC= 0.5891) - the class imbalance is more pronounced in these results therefore precision-recall metrics are provided with (Prec=0.6318, Rec=0.8476, and a PrAUC= 0.5278). The results are poor and overall worse that those produced using the multi-layer feedforward neural network. More importantly the SVM, RF or LDA were not able to produce results anywhere near those using the deep learning stacked autoencoder and 50 nodes. These findings are in line with other studies that have shown that deep learning models perform better than traditional classifier models like an SVM, in genomic-based studies, i.e. please refer to [41] for an example of such a comparison.

This paper placed a strong emphasis on classification tasks using epistatic interactions between SNPs that were extracted from high dimensional genomic data, which in the present context corresponds to whether a mother will have a normal or premature delivery. This is clearly important for mitigating risk to the mother and unborn foetus. Furthermore, it provides a new and viable way to capture epistatic interactions between SNPs. However, from an extensive literature review the extraction of identified patterns from deep learning neural networks and the classification of phenotypes using GWAS data have received little attention within the research community.

One possible reason could be directly attributed to the fact that deep learning models are difficult to interpret [42]. Compressing the input set to 50 nodes shows reasonably good predictive capacity. However, it is difficult to identify what information from the 4666 SNPs contribute to those 50 hidden nodes. Consequently, deep learning approaches are characterised as black boxes where it becomes difficult to explain good results or modify models to address misclassification issues.

At this stage of development, the findings in this paper demonstrate that a GWAS classification system could provide an early screening tool for medical practitioners (general practitioners, gynaecologists and nursing and midwifery professionals) to identify women with a genetic disposition to preterm birth. Currently in gynaecology and obstetrics, screening for patients with an increased risk of preterm birth is performed by assessing patient risk factors and ultrasound of the cervix. Neither of these methods has good sensitivity or specificity and there is a great need to identify patients at high risk of spontaneous preterm birth, both for potential interventions in the disease process which include medications (progesterone) and surgical procedures (cervical cerclage) and for interventions to decrease neonatal morbidity (antenatal steroids). Therefore, the results in this paper suggest that commercialising an assay of the 4666 SNPs identified in this study (processed through the deep learning stacked autoencoder and classification models) would provide sufficient information about a mother's predisposition to deliver term or preterm. This would eventually lead to an automated, therapeutic intervention to direct medical attention toward high-risk pregnant mothers and help to reduce morbidity and mortality associated with preterm deliveries. The current protocol used in gynaecology and obstetrics does not routinely include genetic screening. This approach has the potential to deliver significant impact within preterm birth treatment and care.

5 Conclusion

We presented a novel framework that combines GWAS quality control, logistic regression and deep learning stacked autoencoders for epistatic-driven classification of preterm birth using SNP genomic data in a predominantly African-America population. Using our data set of 1,567 pregnant mothers, we achieve classification results (Sen=0.9562, Spec=0.8780, Gini=0.9490, Logloss=0.5901, AUC=0.9745, MSE=0.2010) using 50 hidden nodes compressed from 4666 SNPs. Minimizing the MSE below 10% we achieved classification results (Sen=0.9289, Spec=0.9591, Gini=0.9651, Logloss=0.3080, AUC=0.9825, MSE=0.0942) using 500 nodes. Figure 4 e and f show the main results in the paper and highlight that there is no significant evidence of overfitting when comparing the training and validation data sets and Figure 5 demonstrates that our framework has good predictive capacity.

These results are encouraging. However, the study needs further research to find more sophisticated strategies for mapping SNP inputs to hidden layer nodes. SNPs are symbolic, and they mean something in the context of GWAS analysis. The minute non-linear transformations of the input space occur it is very difficult to trace the amount of variance they contribute from case-control data. This is a common problem in neural network modelling that seriously hinders genomic analysis.

In future work, we will look at several alternative extensions to this work. It may be interesting to model the

SNPs of mothers who deliver term only and implement anomaly detection using autoencoders [43] to identify pregnant mothers with genetic differences - they do not necessarily have to deliver prematurely. This would provide clear groupings and act as a basis for more in-depth analysis of these genomic differences. In future work we will conduct research using association rule mining (ARM). Exploring the intrinsic relationships in the data and extracting rules to better understand SNP behaviour and their subsequent interactions between each other are important tasks that can be performed using frequent pattern mining. Investigating the correlation between SNPs used to describe rules and their use in deep learning stacked autoencoders may provide an interpretable model. In other words, if models perform well in separating case-control instances in classification tasks (based on SNPs extracted from ARM analysis) then it may be possible to use the rules to interpret the model and provide biological findings and associated insights.

Rather than using logistic regression as a way to reduce the number of SNPs it would be useful to explore analysis of variance based techniques, particularly those reported in [44], and whether this can improve the results further. For example, the accumulative analysis of variance would allow us to capture those SNPs that account for very little variance, but overall, help to better explain the phenotype. Again, this is something we will look at in future work.

Overall, the results in this paper highlight the benefits of using deep learning stacked autoencoders to detect epistatic interactions between SNPs in higher-order genomic sequences and classify term and preterm observations. This contributes to the computational biology and bioinformatics field and provides new insights into the use of deep learning algorithms when analysing GWAS that warrants further investigation. While work exists in biological analysis of variants that alter functional regulatory elements (i.e. elements that control gene expression and DNA) using deep learning methods [28], to the best of our knowledge the study presented in this paper is the first comprehensive study of its kind that combines GWAS quality control and logistic regression with deep learning stacked autoencoders for epistatic-drive GWAS analysis and case-control classification.

ACKNOWLEDGEMENT

The dataset(s) used for the analyses described in this manuscript were obtained from the database of Genotype and Phenotype (dbGaP) found at http://www.ncbi.nlm.nih.gov/gap through dbGaP accession number phs000332.v3.p2. Samples and associated phenotype data for the CIDR Preterm Birth Boston Cohort were provided by Xiaobin Wang, M.D.

REFERENCES

[1] H. Blencowe, S. Cousens, D. Chou, M. Oestergaard, L. Say, A.-B. Moller, M. Kinney, and J. Lawn, "Born too soon: The global epidemiology of 15 million preterm births," *Reprod Health*, vol. 10, no. Suppl 1, pp. S2–S2, 2013.

[2] Z. A. F. Kistka, L. Palomar, K. A. Lee, S. E. Boslaugh, M. F. Wangler, F. S. Cole, M. R. DeBaun, and L. J. Muglia, "Racial disparity in the frequency of recurrence of preterm birth," *American Journal of Obstetrics & Gynecology*, vol. 196, no. 2, pp. 131.e1–131.e6, 2007.

[3] E. A. Anum, E. H. Springel, M. D. Shriver, and J. F. Strauss, "Genetic contributions to disparities in preterm birth," *Pediatr Res*, vol. 65, no. 1, pp. 1–9, 2009.

[4] A. Greenough, "Long term respiratory outcomes of very premature birth (¡32 weeks)," *Seminars in Fetal and Neonatal Medicine*, vol. 17, no. 2, pp. 73 – 76, 2012.

[5] R. L. Goldenberg, J. F. Culhane, J. D. Iams, and R. Romero, "Epidemiology and causes of preterm birth," *The Lancet*, vol. 371, no. 9606, pp. 75 – 84, 2008.

[6] J.-M. Moutquin, "Classification and heterogeneity of preterm birth," *BJOG: An International Journal of Obstetrics and Gynaecology*, vol. 110, pp. 30–33, 2003.

[7] S. A. Treloar, G. A. Macones, L. E. Mitchell, and N. G. Martin, "Genetic influences on premature parturition in an australian twin sample," *Twin Research*, vol. 3, no. 2, pp. 80–82, 2000.

[8] B. Clausson, P. Lichtenstein, and S. Cnattingius, "Genetic influence on birthweight and gestational length determined by studies in offspring of twins," *BJOG: An International Journal of Obstetrics and Gynaecology*, vol. 107, no. 3, pp. 375–381, 2000.

[9] A. C. Svensson, S. Sandin, S. Cnattingius, M. Reilly, Y. Pawitan, C. M. Hultman, and P. Lichtenstein, "Maternal effects for preterm birth: A genetic epidemiologic study of 630,000 families," *American Journal of Epidemiology*, vol. 170, no. 11, p. 1365, 2009.

[10] T. P. York, L. J. Eaves, M. C. Neale, and J. F. Strauss, "The contribution of genetic and environmental factors to the duration of pregnancy," *American Journal of Obstetrics and Gynecology*, vol. 210, no. 5, pp. 398 – 405, 2014.

[11] L. A. Hindorff, P. Sethupathy, H. A. Junkins, E. M. Ramos, J. P. Mehta, F. S. Collins, and T. A. Manolio, "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits," *Proceedings of the National Academy of Sciences*, vol. 106, no. 23, pp. 9362–9367, 2009.

[12] E. DeFranco, K. Teramo, and L. Muglia, "Genetic influences on preterm birth," *Semin Reprod Med*, vol. 25, no. 01, pp. 040–051, 2007.

[13] B. Maher, "Personal genomes: The case of the missing heritability," *Nature*, vol. 456, pp. 18–21, 2008.

[14] W.-H. Wei, G. Hemani, and C. S. Haley, "Detecting epistasis in human complex traits," *Nat Rev Genet*, vol. 15, no. 11, pp. 722–733, 2014.

[15] M. L. Calle, V. Urrea, N. Malats, and K. Van Steen, "mbmdr: an r package for exploring gene–gene interactions associated with binary or quantitative traits," *Bioinformatics*, vol. 26, no. 17, p. 2198, 2010.

[16] X. Wan, C. Yang, Q. Yang, H. Xue, X. Fan, N. L. S. Tang, and W. Yu, "Boost: A fast approach to detecting gene-gene interactions in genome-wide case-control studies," *The American Journal of Human Genetics*, vol. 87, no. 3, pp. 325–340, 2010.

[17] F. V. Lishout, J. M. Mahachie John, E. S. Gusareva, V. Urrea, I. Cleynen, E. Théâtre, B. Charloteaux, M. L. Calle, L. Wehenkel, and K. V. Steen, "An efficient algorithm to perform multiple testing in epistasis screening," *BMC Bioinformatics*, vol. 14, no. 1, p. 138, 2013.

[18] A. Bureau, J. Dupuis, K. Falls, K. L. Lunetta, B. Hayward, T. P. Keith, and P. Van Eerdewegh, "Identifying snps predictive of phenotype using random forests," *Genetic Epidemiology*, vol. 28, no. 2, pp. 171–182, 2005.

[19] R. Jiang, W. Tang, X. Wu, and W. Fu, "A random forest approach to the detection of epistatic interactions in case-control studies," *BMC Bioinformatics*, vol. 10, no. 1, p. S65, 2009.

[20] D. F. Schwarz, I. R. König, and A. Ziegler, "On safari to random jungle: a fast implementation of random forests for high-dimensional data," *Bioinformatics*, vol. 26, no. 14, p. 1752, 2010.

[21] M. Yoshida and A. Koike, "Snpinterforest: A new method for detecting epistatic interactions," *BMC Bioinformatics*, vol. 12, no. 1, p. 469, 2011.

[22] M. B. Kursa, "Robustness of random forest-based gene selection methods," *BMC Bioinformatics*, vol. 15, no. 1, p. 8, 2014.

[23] B. J. Grady, E. S. Torstensen, P. J. McLaren, P. I. D. Bakker, D. W. Haas, G. K. Robbins, R. M. Gulick, R. Haubrich, H. Ribaudo, and M. D. Ritchie, *Use of Biological Knowledge to inform the Analyssi of Gene-Gene Interactions Involved in Modulating Virologic Failure with Efavirenz-Containing Treatment Regimens in Art-Naive ACTG Clinical Trials Participants*, pp. 253–264. 2012.

[24] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. de Bakker, M. J. Daly, and P. C. Sham, "Plink: A tool set for whole-genome association

and population-based linkage analyses," *The American Journal of Human Genetics*, vol. 81, no. 3, pp. 559–575, XXXX.
[25] A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese, "Statistical significance of combinatorial regulations," *Proceedings of the National Academy of Sciences*, vol. 110, no. 32, pp. 12996–13001, 2013.
[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[27] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.
[28] G. Eraslan, J. Arloth, J. Martins, S. Iurato, D. Czamara, E. B. Binder, F. J. Theis, and N. S. Mueller, "Deepwas: Directly integrating regulatory information into gwas using deep learning supports master regulator mef2c as risk factor for major depressive disorder," *bioRxiv*, 2016.
[29] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature Methods*, vol. 12, pp. 931 EP –, 2015.
[30] K. Hao, X. Wang, T. Niu, X. Xu, A. Li, W. Chang, L. Wang, G. Li, N. Laird, and X. Xu, "A candidate gene association study on preterm delivery: application of high-throughput genotyping technology and advanced statistical methods," *Human Molecular Genetics*, vol. 13, no. 7, pp. 683–691, 2004.
[31] C. A. Anderson, F. H. Pettersson, G. M. Clarke, L. R. Cardon, A. P. Morris, and K. T. Zondervan, "Data quality control in genetic case-control association studies," *Nat. Protocols*, vol. 5, no. 9, pp. 1564–1573, 2010.
[32] A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich, "Principal components analysis corrects for stratification in genome-wide association studies," *Nature Genetics*, vol. 38, pp. 904 EP –, 2006.
[33] O. J. Dunn, "Estimation of the medians for dependent variables," *Ann. Math. Statist.*, vol. 30, no. 1, pp. 192–197, 1959.
[34] J. B. M.D., "Application of the logistic function to bio-assay," *Journal of the American Statistical Association*, vol. 39, no. 227, pp. 357–365, 1944.
[35] X. Wang, C. Baumgartner, D. C. Shields, H. W. Deng, and J. S. Beckmann, *Application of Clinical Bioinformatics*. Springer, 2016.
[36] J. Berkson, "Why i prefer logits to probits," *Biometrics*, vol. 7, no. 4, pp. 327–339, 1951.
[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533 EP –, 1986.
[38] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, pp. 947 EP –, 2000.
[39] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, pp. 315–323, PMLR, 2011.
[40] R. A. Webb and D. K. Copsey, *Statistical Pattern Recognition*, p. 666. Wiley, 2011.
[41] R. Cao, D. Bhattacharya, J. Hou, and J. Cheng, "Deepqa: improving the estimation of single protein model quality with deep belief networks," *BMC Bioinformatics*, vol. 17, no. 1, p. 495, 2016.
[42] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016.
[43] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.
[44] H. Yang, H. Tang, X.-X. Chen, C.-J. Zhang, P.-P. Zhu, H. Ding, W. Chen, and H. Lin, "Identification of secretory proteins inMycobacterium tuberculosisUsing pseudo amino acid composition," *BioMed Research International*, vol. 2016, pp. 1–7, 2016.

Dr Paul Fergus is a Reader (Associate Professor) in Machine Learning. Dr Fergus's main research interests include machine learning for detecting and predicting preterm births. He is also interested in the detection of foetal hypoxia, electroencephalogram seizure classification and bioinformatics (polygenetic obesity, Type II diabetes and multiple sclerosis). He is also currently conducting research with Mersey Care NHS Foundation Trust looking on the use of smart meters to detect activities of daily living in people living alone with Dementia by monitoring the use of home appliances to model habitual behaviours for early intervention practices and safe independent living at home.

Casimiro Curbelo Montañez is a PhD candidate of the Applied Computing Research Group at Liverpool John Moores University (LJMU), UK, under the supervision of Dr. Paul Fergus. He received his B.Eng. in Telecommunications in 2011 from Alfonso X el Sabio University, Madrid (Spain). In 2014, Casimiro Aday obtained an MSc in Wireless and Mobile Computing from Liverpool John Moores University. His research interests include various aspects of data science, machine learning and their applications in Bioinformatics"

Basma Abdulaimma received a BSc (Hons) in Computer Science from Baghdad Technology University, Iraq in 1999, an MSc in Computing and Information Systems from Liverpool John Moores University (LJMU), UK in 2013. She is currently a PhD candidate at Liverpool John Moores University. Her research interests include data science, machine learning, and artificial intelligence. She is especially interested in bioinformatics and computational biology at a molecular level particularly genetics.

Prof. Paulo Lisboa is Professor and Head of Department of Applied Mathematics at Liverpool John Moores University. His research focus is advanced data analysis for decision support. He has applied data science to personalised medicine, public health, sports analytics and digital marketing. In particular, he has an interest in rigorous methods for interpreting complex models with data structures that can be validated by domain experts.

Dr Carl Chalmers is a Senior Lecturer in the Department of Computer Science at Liverpool John Moores University. Dr Chalmers's main research interests include the advanced metering infrastructure, smart technologies, ambient assistive living, machine learning, high performance computing, cloud computing and data visualisation. His current research area focuses on remote patient monitoring and ICT-based healthcare. He is currently leading a three-year project on smart energy data and dementia in collaboration with Mersey Care NHS Trust. As part of the project a six month patient trial is underway within the NHS with future trials planned. The current trail involves monitoring and modelling the behaviour of dementia patients to facilitate safe independent living. In addition he is also working in the area of high performance computing and cloud computing to support and improve existing machine learning approaches, while facilitating application integration.

Dr Beth Pineless is a chief redient in obstetrics and gynacology at the University of Maryland Medical Center. As an undergraduate and for two years after college, she worked on a variety of subjects including neonatal pain response and placental microRNAs. She chose to complete an M.D./Ph.D. with an emphasis on epidemiology to gain more experience in research and public health that offered by the M.D. degree. She is planning to continue her work as a fellow in maternal-fetal medicine in 2019.