# PREDICTING AUTISM SPECTRUM DISORDER FROM GENOME-WIDE ASSOCIATION DATA WITH GENETIC BALANCING GENERATIVE ADVERSARIAL NETWORK

by

RORY MCKENZIE FLYNN

Bachelor of Science in Mathematics

Metropolitan State University of Denver, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Masters of Science

Computer Science Program

November 2020

This thesis for the Masters of Science degree by

Rory McKenzie Flynn

has been approved for the

Computer Science Program

by

**Advisor**:

Ashis Kumer Biswas

**Committee Members**:

Farnoush Banaei-Kashani

Mazen Al Borno

Flynn, Rory McKenzie (MS. Computer Science)

Predicting Autism Spectrum Disorder From Genome-Wide Association Data With Genetic Balancing Generative Adversarial Network

Thesis directed by Assistant Professor, Computer Science and Engineering Ashis Kumer Biswas

## ABSTRACT

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition that affects approximately 1% of the population, and seems to be largely genetic in origin. A genetic diagnosis has the ability to improve treatment of those with ASD, but conventional methods have not been effective enough to make confident predictions from genetic data alone. Genome Wide Association Study (GWAS) data has been proven effective for understanding complex disease like the ASD.

In this thesis, we aim to analyze GWAS data sets using deep learning methods to better understand ASD. However, in order to succeed with GWAS data, deep learning algorithms must overcome some issues common in such data sets: imbalanced phenotypes, high dimensionality, and the need for model interpretability. To this end we introduce a novel algorithm, the Genetic Balancing Generative Adversarial Network (GBGAN) which addresses the issues. GBGAN was tested as a phenotype predictor on a large data set obtained from the UK Biobank, and was compared alongside two other neural network models, and two more interpretable models. Results show evidence in favor of the proposed GBGAN algorithm in terms of interpretability. GBGAN outperformed other models in some metrics and appeared to address high dimensionality and class imbalance problems.

The form and content of this abstract are approved. I recommend its publication.

Approved: Ashis Kumer Biswas

Dedicated to the coffee roasters and purveyors of Denver, CO.

## ACKNOWLEDGEMENTS

Thank you to my advisor, my committee members, and the rest of the faculty at the University of Colorado Denver. Special thanks to Howard Holmes for advising on medical research, and Dr. Subrata Paul for advising on Genome Wide Association Study data.

## TABLE OF CONTENTS

# LIST OF TABLES

TABLE

## LIST OF FIGURES

FIGURE

# 1 INTRODUCTION

## 1.1 Autism Spectrum Disorder

Autism Spectrum Disorder (ASD) is a developmental disorder that affects roughly 1% of the population [1–3] amounting to millions of individuals around the world. However, ASD is not well understood. It is a complex condition to diagnose, quantify and even to define. Early intervention is currently the best tool to address long term hardship experienced by many with ASD, and since the symptoms can take years to manifest, a genetic approach to screening is highly desirable. In the past few decades, science has slowly begun to solve the mysteries of how genetics affect human psychology, largely thanks to next generation sequencing tools. One such tool, the Genome Wide Association Study (GWAS), has proven to be particularly popular and effective. There is strong evidence that ASD is affected by genetics, and there has been a great deal of scientific effort directed at understanding the mechanisms in the human genome responsible. Using GWA studies and other methods, many candidate genes and genetic variations have been associated with ASD, but no variant has been able to explain the condition. The data suggests that ASD is a complex and polygenic condition, and may require more advanced modeling to understand. Deep learning has proven effective in complex applications where traditional techniques fail, and could be a game changing tool for ASD research if it can overcome the challenges inherent in GWAS data.

## 1.2 Deep Learning and Genome Wide Association Studies

Deep learning techniques such as convolutional and feed-forward Neural Networks(NNs) have demonstrated considerable success with large data sets on complex estimation and classification tasks. These techniques have contributed significantly to medicine, and deep learning is now used to understand medical data as varied as MRI scans, Xray images, and EEG. It was hoped that these techniques would have similar success in genetics, and enable a new age for genetic screening and personalized medicine. However, in predicting phenotype from genetic data alone, success has been very limited. Data sets created by Genome Wide Association Studies(GWAS) have proven to be particularly challenging for deep learning algorithms in many cases [4].

Aspects of GWAS data sets would seem to have advantages for machine learning. For example, GWAS data-sets now have many participants, and the sample sizes of these data sets is constantly increasing [5]. However, the large sample size is deceptive. A typical GWAS study will measure a huge number of genotypes. Each genotype is a feature and each participant is an observation from the perspective of a machine learning model, making GWAS data extremely high dimensional. The curse of dimensionality is a famous problem in machine learning, one that is usually addressed by training the model on a larger number of observations. At a minimum, it is desirable to have more observations than variables. Unfortunately, in an imputed GWAS data set there could easily be over 90 million variants, and it would be difficult to find a GWAS with more than half a million participants. This frequently leads to over-fitting, as the model has insufficient observations to determine what input variables are truly predictive of the response and which are merely correlated in the sample.

At the same time the response variable in GWAS data, often a medical condition or phenotype, is likely to be imbalanced. Some phenotypes may be present in millions of people but still be a small proportion of the total population. ASD is one such phenotype. As a result, a large sample size from the general population may not result in a large sample of ASD positive individuals. This further exacerbates the challenges posed by the curse of dimensionality, and makes it difficult for models to prioritize the minority phenotype.

The enigmatic nature of deep learning is another obstacle to this application. Interpretability is a major concern in medical diagnostics and screening. Models that can't be understood are difficult to trust, and understanding the genetic nature of a condition is extremely important to future research. Genetic studies are also limited in their scope, often sampling groups with shared heredity. The findings on such groups require additional scrutiny, and may not be applicable to the general human population. Interpretability is one of the great struggles of deep learning, but it is a necessary hurdle to overcome in the study of genetics.

## 1.3 Objective

In order to succeed with GWAS data, deep learning will need to overcome imbalanced phenotype, high dimensionality resulting in over-fitting, and the interpretability problem. Our goal is to use convolutional NNs with advanced machine learning architectures to overcome these obstacles, and develop a better understanding of the ASD genotype.

## 1.4 Approach

We propose a novel algorithm, the Genetic Balancing Generative Adversarial Network (GBGAN) to address these challenges. Inspired by methods for image analysis and generation, and adapted for GWAS data, this architecture will be able to both classify individuals by their phenotype, and generate class representative samples to be used in further analysis. The problem of imbalanced phenotypes is addressed by applying a simple form of meta learning, and the use of sample weighting. High dimensionality is addressed by using dropout layers, and a GWAS specific data preprocessing pipeline. The GBGAN provides tools for model interpretation that are not available in traditional deep NN techniques, in the form of artificial output and latent coefficients. We hope that this method gives insight into how future deep learning algorithms can also address the challenges detailed above.

## 1.5 Thesis Organization

Chapter 2 explains basics of ASD, the data used in the study, introduces machine learning models, and other relevant background information. In Chapter 3, we present our proposed architecture of the GBGAN, and in Chapter 4 we outline the data processing pipeline, and introduce the model testing methodology including model evaluation metrics. Chapter 5 presents the results of experiments with GBGAN, and a comparative analysis with existing state-of-the-art methods. Finally, Chapter 6 concludes the thesis with a summary of the project, its findings, limitations, and future research scope.

# 2 BACKGROUND

## 2.1 Definitions

Phenotype: The observable traits of an individual, such as height, eye color, and blood type. The genetic contribution to the phenotype is called the genotype [6].

Genotype: The version of a DNA sequence that an individual has, typically in relation to a phenotype [6].

Linkage Disequilibrium: The non-random association of alleles at two or more loci in a general population. Linkage disequilibrium between two alleles is related to the time of the mutation events, genetic distance, and population history [7].

Heritability: A statistic that estimates the degree to which differences in people's genes account for differences in their traits.

De-Novo Mutation: A mutation that originates in the individual and is not inherited.

Polygenic Trait: A polygenic trait is one whose phenotype is influenced by more than one gene. Many polygenic traits are also influenced by the environment [8].

Epistasis: AKA gene-gene interaction. The effect of one gene on a disease modified by another gene or several other genes [7].

## 2.2 Autism Spectrum Disorder

Autism spectrum disorder is the name given to a neurodevelopmental disorder with diverse expressions. ASD is characterized by persistent deficits in social communication with restricted and repetitive patterns of behavior. It is believed that approximately 1% of the general population is affected by ASD [1–3], but reliable counts are difficult to determine. Because ASD takes a wide variety of forms and levels of severity it is suspected to be under-reported. The definition of ASD has changed regularly since it was first described as a condition in the early 1900s, a trend that continues in recent history. The current definition of ASD was adopted as recently as 2013, and is a broader definition than previously used. For example, Asperger syndrome is no longer a separate disorder but considered to be a form of ASD. Language from previous definitions is still found in many data sets, including the data sets used in this study, and must be accounted for.

### 2.2.1 Need For Genetic Prediction

Regardless of how it is defined or its exact prevalence, ASD is a common condition, and one that noticeably impacts the lives and families of children who are diagnosed. The best method to mitigate the cost and hardship is to provide early education and therapeutic interventions to those born with the condition, but to do so requires early screening, preferably before symptoms become apparent. It follows that there is significant enthusiasm in the medical community to find and understand the genetic origins of ASD. Such an understanding would enable early screening, hereditary risk prediction, and possibly inform alternative treatment. Traditional approaches have made significant progress, and many custom techniques have been applied as well. However, the predictions made by these techniques, while useful, have not yet been sufficiently accurate to be used for early interventions. Neural Network (NN) approaches have not been fully explored in this application, and could offer a viable solution.

### 2.2.2 The ASD Genotype

The outlook is optimistic for a genetic approach to ASD diagnosis in the near future. There is solid evidence that genetics at least influence a person's risk of ASD. Studies that measure heritability make up the most well understood evidence for genetically influenced ASD. Twin studies that have shown strong heritability, some put the estimate as high as 91% [1]. Larger studies of siblings suggest that there is 7-19% recurrence risk of developing ASD in the siblings of an ASD positive individual [1,3]. Past analysis of whole genome genotyping studies estimate heritability to be (31-71%) [1], and this is the most likely range.

It is worth noting that while many conditions are also associated with ethnic groups, it is not known if this is the case in ASD. Studies focusing on the frequency of ASD diagnoses in ethnic groups have reached different conclusions [9–11]. Researchers are also more likely to attribute differences to well documented disparities in economic status, ASD awareness, and attitudes towards ASD generally [9,10,12]. There is a known association between ASD and sex, with the male being 2 to 4 times more likely to be diagnosed compared to females. However, the cause of this association is unknown, and could be the result of genetics, or under-reporting in the female population [13]. There is a general call for better diagnostics, and more widespread adoption of the tools already available [9–12].

Heredity is probably not the only way that genotype influences ASD, de-novo mutations have also been shown to influence the condition. De-novo mutations, unlike heredity, can only be measured with genetic sequencing, and now can be found effectively using GWAS data [14,15]. Research into de-novo variants is valuable for understanding the macro causes of ASD. By studying de-novo mutations that effect genes, the contribution of said genes can be inferred. However, this form of mutation has direct relevance to only a small proportion of cases [1]. There is also evidence to suggest that structural factors such as copy number variation could be the root cause of ASD [16]. Despite these many studies, most suggesting genes that could be associated with ASD, little is known with certainty about its root causes. Results are consistent with a highly polygenic trait, meaning that there are likely many legitimate genetic causes for ASD.

### 2.2.3 Measurements Correlated With ASD

Given that ASD is likely under-reported, it would be attractive if we could study it indirectly using traits that are associated with ASD, but easier to measure. Traits known to be associated with ASD include higher fluid intelligence scores [17, 18], and worse outcomes in employment [19] and education [19, 20]. Data is highly available for these types of measurements in GWAS studies, and they may provide insight into those who are undiagnosed, or who have symptoms associated with ASD, but not with the severity that would be formally diagnosed. Obviously, many other variables are associated with these traits as well, and they cannot replace high quality ASD diagnosis data. Rather, these associated measurements should be considered to supplement ASD data when appropriate.

## 2.3 Genome Wide Association Studies

A Genome Wide Association Study (GWAS) is a fast and effective tool to study genetics in many organisms, and the primary source of genetic data in this study. Over the past two decades GWAS have become a standard method for human genome research [21], and have already been used to estimate risk for several conditions [5, 21, 22]. Compared to most other methods GWAS are faster, less expensive, and more targeted. If an ASD related pattern is to be found in human genes, then odds are good that GWAS data will be able to measure it. There is more to know about GWAS analysis then can be covered in this document, but it is necessary to describe the structure of GWAS data and the methodology of GWA studies in a simple sense.

### 2.3.1 Structure of GWAS Data

GWA studies measure the DNA of their participants looking for specific predetermined variations. These variations are recorded as a set of genetic markers called Single Nucleotide Polymorphisms (SNPs), which are a topic of study in themselves. The output of such a study is a data set that can be visualized as having a column for each SNP with a row for each participant. In addition to the directly measured SNPs, most modern GWAS studies will also use genotype imputation to estimate additional SNPs with a high degree of accuracy [23]. The additional SNPsin imputations data provide valuable and often indispensable information. Thus, it is the accepted practice to use imputed data whenever possible [23].

Each SNP has a host of information associated with it, for this study we are most concerned with the major and minor allele. The GWAS data relevant to this study is stored in the PLINK [24] data format. In this format, an integer -1 up to 2 is associated with each SNP for each individual. The integer describes how that person's DNA relates to the SNP's major and minor allele or if it is missing. The details of this encoding can be seen in Table 2.1. For purpose of machine learning, each SNP is a categorical variable with one of 4 values including missing.

Table 2.1: The PLINK format genotype codes. Each code describes a relationship that an individual may have with the major and minor allele of a SNP.

| Integer Value | Relation To The Alleles |
|---|---|
| -1 | Unknown |
| 0 | Homozygous for major allele |
| 1 | Heterozygous |
| 2 | Homozygous for minor allele |

### 2.3.2 Filtering by Ethnicity

Representation has been a subject of great interest in the public and scientific community especially around ethnicity. While the relation of ethnicity to ASD is not known, this remains relevant to any GWAS because filtering by ethnicity is a key choice. Studies that purposely included diverse ethnic groups are successful in many tasks, but data sets with sufficient sizes and diversity are rare [25]. Ethnic filtering is often necessary to limit population stratification effects, which can cause false positives and confounding [25, 26]. For these reasons it is generally appropriate to filter GWAS data by ethnicity.

## 2.4 Importance of Interpretability

The amount that is not known about the ASD genotype makes the interpretability problem more apparent in this discipline. Finding genetic structures and SNPs associated with conditions can influence the definition or treatment of said conditions. More importantly, in order to be trusted as a diagnostic a model should at least be able to identify the genetic variants that it uses to make its decisions. The causal SNPs that create heritable diseases are inherited with many other SNPs that are not causal, but are still correlated with the outcome. Combined with the extreme dimensionality of genetic data, a model can easily pickup on the shared heredity of an ASD population, and miss the variants that actually effect ASD. Hereditary essential guarantees a wealth of confounding variables.

The ability to determine if an SNP is causal is extremely non-trivial, and many applications of deep learning in GWAS analysis focus only on predicting causal SNPs. Such methods include DeepND [27] which uses graph convolution, Bayesian NN methods [28], and Markov blanket based methods [29,30]. These techniques also do not function on raw GWAS data directly, but use relationship data, text data, and other genotype descriptive data in order to select genes that are likely to be casual. While highly advanced, these techniques are also often highly specific, and tend not to focus on diagnostics except in an indirect fashion. These methods are also not easily comparable to GBGAN, which attempts to address both phenotype and genotype simultaneously.

## 2.5 Baseline Algorithms

Many other algorithms have been applied to phenotype prediction with GWAS data. Mostly simpler and better understood algorithms than the deep learning solutions, which focus on causal SNPs. For example, simple and complex regression methods have been applied to GWAS data with variable success [31,32]. Logistic, rigid regression, and other variants have been used with or without being part of a meta-algorithm such as boosting. Regression's biggest contribution in this field may be its role in generating metadata for other analyses, such as Polygenic Risk Scores (PRSs), and linkage disequilibrium clumping. However, as interest grows in utilizing higher dimensional GWAS data, some have called into question whether regression based frameworks are feasible [33].

A Random Forest (RF) is an ensemble method, and one of the most popular matching learning algorithms across disciplines. The structure of random forests is particularly well suited to applications in high dimensional spaces like genetics, and has been applied to GWAS data with success [34–36]. Particularly, random forests have shown the ability to pick up on the linkage disequilibrium and epistasis [33]. The RF algorithm is also one of the more interpretable machine learning algorithms, providing internal measures of variable importance that can be used to rank the importance of SNPs to the model.

However, like regression, RF has serious limitations in the GWAS arena. Solutions have been proposed to tailor RF to GWAS data [37]. To the best of our knowledge these methods have not supplanted the traditional model as of yet. Suggested preprocessing for RF models include sub-sampling, pruning based on linkage disequilibrium (LD), and removing strong effects from RF analyses [35]. It also may be necessary to increase the maximum numbered of features to search at each split to account for the weak predictive power of SNPs [35]. The exact optimum value however would be up to experimentation.

### 2.6 Related Algorithms

The hope of the GBGAN model is to leverage a Generative Adversarial Network (GAN) structure to simultaneously generate genotypes while classifying phenotypes. The intention is to train a deep learning classifier with a generator, so the latter can provide insight into the nature of the classification. Building a classifier through GAN learning is a goal that GBGAN shares with another recent model for a very different data set, OR-AC-GAN [38]. Like OR-AC-GAN, it is hoped that GBGAN will treat outlining observations as false input, and not attempt to apply the same rules to them. However, OR-AC-GAN takes inspiration from the now ubiquitous Auxiliary Classifier GAN (AC-GAN), and GBGAN takes inspiration from the more recent Balancing GAN (BGAN) in an attempt to leverage the autoencoder architecture for pre-training. Although inspired by the original design of BGAN [39], this is a new model with significant modifications to reflect the data and goals of imbalanced GWAS classification. GBGAN also uses some design hints from the gGAN [40] model, which used the GAN architecture for semi supervised learning on GWAS data. GBGAN was also designed with a healthy dose of pragmatism, using variables that worked with the data available for testing. While the final design is sufficient for proof of concept, future work should include the use of one of the many hyperparameter optimization software packages to tune the model.

# 3 ARCHITECTURE

## 3.1 Data Encoding

In order to understand the architecture of GBGAN it is important to know how the input data is formatted. The GBGAN takes in GWAS data with each participant as a row and each SNP as a column. The SNP data should be encoded in the PLINK [24] format described in the background and in Table 2.1. In its original integer format the data does not reflect the categorical nature of each SNP code. Therefore, before training or running the model, each integer line is transformed into a one-hot encoded matrix with 4 binary lines, such that an $M \times N$ array would become an $M \times N \times 4$ array. In the resulting output each row corresponds to one of the three possible values of data as seen in Table 3.1, mapping row index to value, except the unknown entries which are mapped to row 4 instead of -1. The conversion of one line is shown in Table 3.1 for reference. With this data encoding the convolutional layers in GBGAN interpret the data as having 4 channels, one for each of the possible SNP values. Early experimentation showed that this one-hot encoding was more optimal than scaling or normalizing the data.

Table 3.1: GWAS data before and after preprocessing for use in GBGAN.

$$\begin{bmatrix} 2 & 0 & 2 & \dots & 1 & 2 & -1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

## 3.2 GBGAN

GBGAN has a potentially confusing design, three NN structures, are arranged in three separate sub-models, to facilitate the two-step training algorithm. In this section the design and training process of GBGAN will be explained, starting with the training process and moving into finer detail.

### 3.2.1 Training Step 1

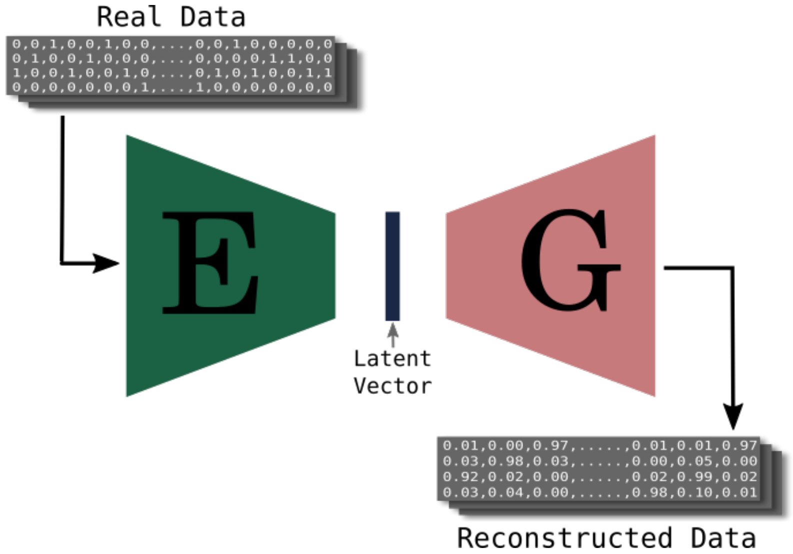

Figure 3.1: The structure of the GBGAN's first training step. The encoder ($E$) and generator($G$) form an autoencoder sub-model in this image.

In order to start training the model only the encoder($E$) and generator ($G$) NN structures are initially required. These structures are stacked to form an autoencoder model as seen in Figure 3.1. $E$ takes in GWAS data, preprocessed as described above with each observation having shape $N \times 4$, and through a series of convolution steps reduces it to a latent vector of size $\lfloor N/8 \rfloor \times 1$. $G$ acts as the decoder in the autoencoder structure; it takes the latent input from $E$ and attempts to recreate the original input. More specifics on the structure of $E$ can be seen in the top portion of Table 3.3, and $G$ is detailed in Table 3.2. The intention in this step is to learn a latent representation of the data before attempting classification.

This process was especially essential in the BGAN model because it was the only opportunity to train on the full data set. The remaining training steps would only expose these structures to data that had been balanced with Random Under Sampling (RUS).

Attempting a similar application of RUS never produced productive results in GBGAN prototypes. GBGAN uses a large sample size with classification weights instead of RUS, so Step 1 serves another purpose in our model. Training with extremely imbalance data results in an uneven training process using gradient descent. The hope is that this autoencoder training will find near optimal weights before moving on to the more volatile GAN-training in Step 2.

### 3.2.2 Intermediate Stage

Transitioning between the first and second training steps requires a brief but important intermediate stage. This stage can not be called a training step because no machine learning takes place. The goal here is to make a set of distributions which model the latent vector learned in Step 1, and which are key to Step 2. In the intermediate stage $E$ is decoupled from $G$. The predictive data set is split into partitions by output class, and each class specific partition is processed through $E$ to create, for each class $i$, a latent data set $l_i$. The latent mean vector $\mu_i$ and latent covariance matrix $\Sigma_i$ is calculated from each $l_i$. Finally, $\mu_i$ and $\Sigma_i$ are uses to characterize a set $Z$ of multivariate normal distributions where $Z_i = \mathcal{N}(\mu_i, \Sigma_i)$. When the autoencoder characterizes the data well, it is hoped that a random sample from the distribution $Z_i$ will look similar to the output of $E$ and the input of $G$ for class $i$.

### 3.2.3 Training Step 2

With $Z$ defined the second training step can start. In this training step $G$ and $E$ take on new roles in a GAN training configuration that can be seen in Figure 3.2. $E$ is combined with the classification NN structure ($C$) to form the discriminator submodel ($D$), while the NN structure $G$ combines with the latent generator for form the generator sub-model. Like with most all NN models GBGAN is trained using batch gradient descent. During each training iteration, a batch of real samples is selected with its corresponding class labels. The class labels are given to the latent generator. The

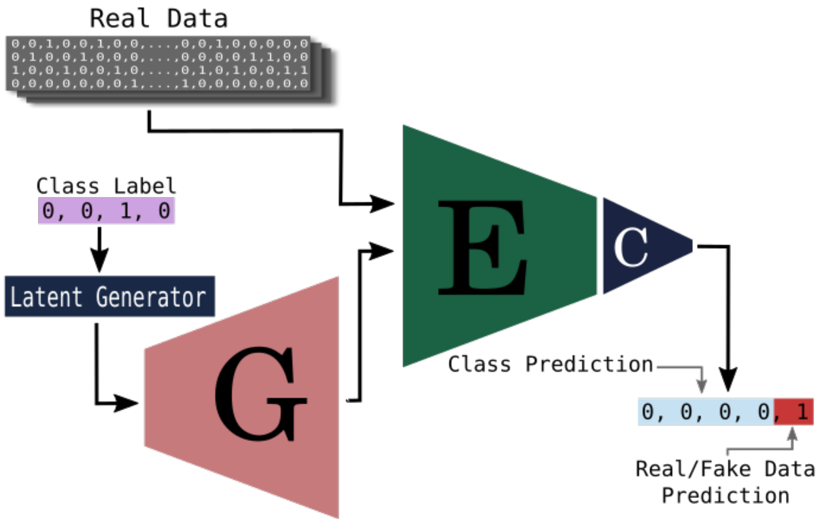

Figure 3.2: GBGAN's second training step. The three NN structures the generator ($G$), the encoder ($E$), and the classifier ($C$) form a GAN training pipeline in this image. $G$ functions as the generator sub-model in this pipeline with the help of the latent generator. The $E$ is combined with the $C$ to form the discriminator ($D$).

latent generator takes a random sample from distribution $Z_i$ for each instance of class $i$ in the batch. The batch of latent data is used as input to $G$ making a batch of false data. $D$ takes in the false data from $G$ with the real data, and the output of $D$ is a classification vector with binary indicators for each class, and one for determining if the input was real or false.

Fake data is treated as an extra class label, and all labels are mutually exclusive, so $D$ is trained to predict the real class label or that the data is false. At the same time $G$ is trained to trick $D$ into labeling the fake data with the target class, rather than labeling it as false. This is achieved by first training $D$ on the true data with class labels and fake data labeled as such. Then $G$ is trained, with the weights of $D$ locked, and the real output labels as targets. BGAN, and by extension GBGAN, differs from AC-GAN in that $G$ is not directly privy to which class is the target. Instead, $G$ must infer the appropriate output to produce for class $i$ from the latent vectors generated from $Z_i$. This false data mimics the imbalanced distribution of the real data, and class weights affect $G$ through $D$.

Table 3.2: The Details of the generator NN structure ($G$) for an input of N elements. Note that // is a stand in for integer division.

| Layer | Activation | Output Shape |
|---|---|---|
| Latent Data | | $N//8$ |
| Reshape | | $(N//8, 1)$ |
| 1 Padding/Cropping | | $(\sim N//8 \pm 1, 1)$ |
| Batch Normalization | | ... |
| UpSampling1D | | $(\sim N//4, 1)$ |
| 1D Convolution | Sigmoid | $(\sim N//4, 64)$ |
| 1 Padding/Cropping | | $(\sim N//4 \pm 1, 64)$ |
| Batch Normalization | | ... |
| UpSampling1D | | $(\sim N//2, 64)$ |
| 1D Convolution | Sigmoid | $(\sim N//2, 8)$ |
| 1 Padding/Cropping | | $(\sim N//2 \pm 1, 8)$ |
| Batch Normalization | | ... |
| UpSampling1D | | $(\sim N, 8)$ |
| 1D Convolution | Sigmoid | $(\sim N, 4)$ |
| 1 Cropping | | $(N, 4)$ |

It is hoped that the GAN training will help stabilize the predictions by teaching $D$ to recognize outlying data, as fake and make predictions accordingly. The classification of real or fake data is discarded when the GBGAN is used as a classifier, and the maximum value of the remaining score is taken as a label. This is an idea that is similar to OR-AC-GAN [38], but with a much simpler application. It is theorized that this will allow $D$ to make different calculations in cases where a real observation, by reason of being an outlier, is recognized as fake. This form of training comes at a massive cost however, essentially doubling the amount of data processed for the same number of epochs. It also relies on the assumption that outlining data will look more like the output from $G$. The validity of this assumption, and the cost benefit analysis of GAN training, will be examined further in the results.

### 3.2.4 Generator Structural Details

The detailed structure of $G$ is shown in Table 3.2. When the generator is used as a decoder in Step 1 it receives input from the encoder in the top part of Table 3.3, forming an autoencoder sub-model seen in Figure 3.1. As a generator sub-model in Step 2 it receives input from a random number generator. The structure of $G$ consists of three sets of batch normalization, upsampling, and 1D convolutional layers. The convolution layers have a kernel size of 6, a stride of 1, and a decreasing number of filters ending at 4. The sigmoid activation functions on the convolutional layers are probably unnecessary on all but the last layer, and in that case probably should be replaced with softmax activations in the future. The batch normalization has a momentum of 0.8, and is included to stabilize the training process.

The padding and cropping layers are conditionally active depending on the input shape of the data, and ensure that the data is the correct shape for the autoencoder to function. The autoencoder requires that the output layer be the same shape as the input layer. Data received as input from the encoder is guaranteed to have $L$ elements where $M$ is the size of the original vector and $L = \lfloor M/8 \rfloor$. Each of the 3 upsampling layers outputs a vector 2 times the size of the input vector, so the output is $L \times 2^3$ or $L \times 8$. If $M$ is not perfectly divisible by 8 then these layers pad and crop the vector as needed. If $M$ is not divisible by 2 the final layer fixes the output by removing one observation from the end.

### 3.2.5 Encoder and Discriminator Structural Details

The encoder structure (E) shown in Table 3.3, consists of 3 1D convolutional layers each with an increasing number of filters, a stride of 2, and a kernel size of 6. The convolution layer uses leaky Rectified Linear Unit (leaky ReLU) activation functions with (alpha=0.2), and each convolution layer is followed by a dropout layer with a 30% drop out rate. The dropout layer is key to prevent over-fitting, and the 30% rate was arrived at through experimentation. The final layer of the encoder is a dense layer $1/8^{th}$ the size of the original input, and has sigmoid activation functions. Sigmoid activations were chosen to try and exaggerate the difference between the classes in the latent vectors.

Table 3.3: Details of the discriminator sub-model ($D$) for an input of N elements. $D$ is made of two NN structures; the encoder structure ($E$) is shown in the top portion of the table and the classifier structure ($C$) is detailed on the bottom.

| Layer | Activation | Output Shape |
|---|---|---|
| Input data | | M by 4 |
| 1D Convolution | Leaky ReLU | $\sim N//2$ by 16 |
| Dropout | | ... |
| 1D Convolution | Leaky ReLU | $\sim N//4$ by 32 |
| Dropout | | ... |
| 1D Convolution | Leaky ReLU | $\sim N//8$ by 64 |
| Dropout | | ... |
| Flatten | | $(\sim N//8) \times 64$ |
| Dense | Sigmoid | $N//8$ |
| | End of Encoder | |
| Dense | Leaky ReLU | (300) |
| Dense | Linear | (150) |
| Dense | Linear | (70) |
| Dense | Linear | (30) |
| Dense | Linear | (10) |
| Dense | Sigmoid | (C) |

The design of the classification structure of GBGAN requires some comment as it is counterintuitive and likely controversial. The classification portion takes the input from the encoder and passes it through several linear layers and a sigmoid activation function. However, this model was arrived at through trial and error to beat out models with saner configurations. In fact, there are theoretical reasons why in this extremely imbalanced scenario this model is more effective.

The problem with liner activations is that they reduce the efficiency of back propagation, and cause the model to behave like a simple linear model [41]. However, in models with hybrid activations, linear activations near output layers are known to have a benefit in some contexts [42]. The reason that these normally sub-optimal activations work well in some cases is not known. In our case the model struggles with over-fitting and learning with weighted samples, so some restrictions may be welcome. It also may be a good fit to treat the output of the model as liner in this case.

At the same time sigmoid activation would allow for multiple classes to be highly likely at once. This is unfortunately necessary in the GBGAN architecture in order for the model to predict class labels and real/fake data labels, but it is a mystery why this also seems to improve the standalone convolution model. Soft max activation functions are usually preferred for multi class scenarios. However, having dueling independent confidences of the sigmoid function may be ideal in this one instance.

The decision to use these layers will be discussed more in the results section, they are not necessarily good design choices.

### 3.3 Alternative Neural Network Models

In order to test the effectiveness of the GAN training, and convolutional layers in GBGAN, two other NN models were created. To test the GAN training a simple Convolutional Neural Network (CNN) model was created with the same shape and design as $D$ in Table 3.3. This CNN model was trained using the optimized default Keras [43] fitting algorithm, with none of the complications of the GBGAN training algorithm.

Table 3.4: The structure of the feed-forward Model made in the image of the discriminator sub-model $D$ in the GBGAN.

| Layer | Activation | (Output Size) |
|---|---|---|
| Input data | | (M, 4 |
| Flatten | | $M \cdot 4$ |
| Dense | Leaky ReLU | $M$ |
| Dropout | | ... |
| Dense | Leaky ReLU | $M//2$ |
| Dropout | | ... |
| Dense | Leaky ReLU | $M//4$ |
| Dropout | | ... |
| Dense | Leaky ReLU | $M//8$ |
| Dense | Leaky ReLU | (300) |
| Dense | Linear | (150) |
| Dense | Linear | (70) |
| Dense | Linear | (30) |
| Dense | Linear | (10) |
| Dense | Sigmoid | (C) |

An even simpler feed-forward NN model was also created in the general image of $D$, and is shown in Table 3.4. The feed-forward model matches the pattern of activations and the number of dropout layers in $D$, but it lacks any convolution layers. Instead, feed-forward layers of decreasing dimensionality feed into an exact copy of $C$. This is not a perfect comparison to a CNN model, NN architecture is too nuanced to easily achieve such a thing, but it should give some indication of how convolution layers interact with the data.

# 4 EXPERIMENTS AND EVALUATION

## 4.1 Data Collection and Prepossessing Pipeline

The UK Biobank is a powerful international resource for GWAS data, and the primary data set of interest in this study. It includes genotype and phenotype data describing just under 500,000 individuals between the ages 40-65, at the time of recruitment, and recruited from across the United Kingdom [44]. The phenotype data includes self reported health and lifestyle data, as well as medical records. There are multiple measurements of ASD, and other measurements believed to be correlated with ASD, such as fluid intelligence and academic qualifications. Regrettably, despite its size the UK Biobank offers only a small number of ASD positive cases, less than 1% by any available measure. The extreme imbalance of the target class proved very difficult to overcome.

From the UK Biobank 3 separate phenotype measures were selected to serve as response variables. Two sets of GWAS data with varying levels of dimensionality were also created to serve as input. This section will describe the pipeline step by step starting with reducing the input dimensionality, and ending with the final encoding of the output variables. It is worth noting that this pipeline was partly inspired by conjecture when a full data set was not available. This is therefore an area that would be appropriate for future research, and is not guaranteed to produce the optimal results.

### 4.1.1 Linkage Disequilibrium Clumping

Reducing the dimensionality of the genetic data was a high order priory from the start of the study, and several options were explored before the full data became available. Linkage Disequilibrium clumping (LD clumping) was identified as the most viable option to apply in the limited testing window. LD clumping is a widely used method for reducing the dimensionality of GWAS data. It allows researchers to use summary statistics and measurements of linkage disequilibrium to build an optimal sub-sample of

SNPs for predicting a given phenotype. Given that the total number of ASD samples in the UK Biobank is small, the ability to use summary statistics from a GWAS with more ASD positive participants was a significant benefit. It was hoped that the use of high quality summary statistics and LD clumping would limit false positives caused by inherited genes unrelated to ASD, in addition to reducing dimensionality.

The summary statistics for the LD clumping algorithm came from the Physiological Genetic Consortium's (PGC's) public data set [1], selected for its unmatched size and public availability. The PGC offers several versions of their summary statistics with different populations included in the calculation. The version of the statistic computed from the European population was used in this case, to better reflect the mostly European population of the UK Biobank.

The UK Biobanks offers two relevant genomic data sets for LD Clumping to be applied to, the genotype data and the imputed genotype data [44]. At the time of writing, the University of Colorado Denver still has limited access to the imputed data set, and only chromosomes 1, Y, and X are available. Despite the potential importance of the sex chromosomes to ASD, their general exclusion from summary statistics, including the PGC statistics, forces their exclusion from LD clumping as well. The most influential chromosomes according to the PGC data are chromosomes 5, 6, and 7, and are unavailable in this data set. Using only chromosome 1 with LD clumping did not result in a large enough data set to draw conclusions, even with the parameters significantly relaxed. Because of these considerations, the non-imputed genotype data was used. This had the added benefit of a smaller size, 805,426 genotypes in the non-imputed verses $\sim$ 90 million in the imputed data. The small size enabled the inclusion of all chromosomes excluding X and Y.

Before clumping, the data was filtered using PLINK2 [45], so that the genotype data did not contain any SNP IDs which did not also appear in the PGC data and vice versa. The LD clumping algorithm makes decisions to include or exclude SNPs based on a set of parameters. One of the most important parameters $p_1$ is the cut off p-value for the SNPs effect in the summary statistics. The LD clumping procedure was applied twice, once with $p_1 = 0.001$ and once with $p_1 = 0.01$, in both cases the default values were used for all other parameters. LD clumping was applied to all chromosomes separately and in parallel, which is the accepted procedure, and reduced the computation time by days. LD clumping with $p_1 = 0.001$ resulted in the selection of 770 SNPs henceforth referred to as the small data set. The same procedure with $p_1 = 0.01$ resulted in the selection of 4489 SNPs henceforth referred to as the large data set.

### 4.1.2 Participant Filtering

Having selected an appropriate number of SNPs, we moved to also filter the individuals selected. As discussed in Chapter 2, the nature of GWAS data suggests that a more genetically homogeneous population is desirable in most cases. To achieve this, the data was filtered down to those individuals with British ancestry exclusively. Any individuals with mixed ancestry or non-British ancestry were removed. This process still retained 88% of the data. The participants were also filtered for each of the data sets described in Tables 4.1, 4.2, and 4.3 as needed.

### 4.1.3 Selecting Outcomes of Interest

Selecting the phenotypes, and converting them into mutually exclusive classifications is the final step to create the database. The UK Biobank offers two measures of ASD in the form of the Data-Field 41270 “Summary Diagnosis”, and Data-Field 20544 “Mental health problems ever diagnosed by a professional”. Data-Field 41270 summarizes the distinct diagnosis for study participants recorded across all hospital inpatient records. Data-Field 20544 contains data from an online followup survey, and is the patient’s self reported mental health history. Details on how these data sets were collected can be found in the UK Biobanks excellent public documentation [44]. For brevity, we will call the “Summary Diagnosis” data the recorded data, referencing its origin as hospital records.

Similarly, we will call the “Mental health problems ever diagnosed by a professional” data the Reported data, because it is self reported in origin. Both these data sets were simplified to binary vectors where each individual was labeled as positive or negative for ASD. The number of positive and negative samples in each data set can be seen in Tables 4.2 and 4.1, for the recorded and reported data respectively.

Table 4.1: Number of participants who are ASD positive and ASD negative according to hospital inpatient records, Data-Field 41270.

| Label | Observations | Percentage |
|---|---|---|
| ASD Positive: | 80 | 0.019% |
| ASD Negative: | 430648 | 99.981% |
| Total observations: | 430728 | |

The recorded data is encoded with codes from the ICD10 - WHO International Classification of Diseases. In this encoding there are three conditions that collectively correspond to the modern definition of ASD. The conditions are Childhood Autism, Atypical Autism, and Asperger’s Syndrome. All individuals who had one or more of these conditions were classified as being positive for ASD. There is an argument that the diversity of ASD would be better represented by treating these as separate classes, but the number of observations is small without unnecessary splitting, as is shown in Table 4.1.

The reported data was much simpler to encode. As part of the follow-up survey participants were asked the following question "Have you been diagnosed with one or more of the following mental health problems by a professional, even if you don’t have it currently? (tick all that apply):". Any participants who checked the box labeled “Autism, Asperger’s or autistic spectrum disorder” was considered ASD positive for the purposes of this study. Compared to the recorded data set, the reported data set had many more positive cases of ASD, 181 cases versus 80 cases. This is consistent with the belief that ASD has been under-reported and inconsistently diagnosed. Future work should consider examining the overlap between these two data sets, both for ASD and related conditions such as schizophrenia. Possibly a slightly larger and more effective data set could be created.

Table 4.2: Number of participants who are ASD positive and ASD negative according to self reported diagnosis in Data-Field 20544.

| Label | Observations | Percentage |
|---|---|---|
| ASD Positive: | 181 | 0.042% |
| ASD Negative: | 430547 | 99.958% |
| Total observations: | 430728 | |

In addition to direct measures of ASD the UK Biobank also offers health, lifestyle, and demographic measures, some of which are believed to be associated with ASD. These related measurements include standardized test scores, fluid intelligence, and educational qualifications. Of these many options, time permitted only one to be tested in this study. Data-Field 6138 "Education Qualifications" was selected and used to make the final data set, the qualification data set. There were 4 instances where individuals were asked to self report their qualifications. For this study each individual was assigned the highest qualification value they ever reported; the qualification value being the numerical encoding as seen in Table 4.3. Any individual who refused to answer the survey was dropped from this data. This strategy did not necessarily reflect the complex relationship of ASD to the range of qualifications, but it is a pragmatic solution. The point of this data set was to test GBGAN on data that was less imbalanced, multi class, and not related directly to ASD. A true exploration would give much more attention to the relation of qualifications to the previously mentioned ASD measurements.

Table 4.3: Number of participants who self reported each of the educational qualifications categories in Data-Field 6138.

| Label | Observations | Percentage |
|---|---|---|
| -7 None of the above: | 74715 | 17.511% |
| 1 College or University degree: | 39288 | 9.208% |
| 2 A levels/AS levels or equivalent: | 10690 | 2.505% |
| 3 O levels/GCSEs or equivalent: | 81691 | 19.146% |
| 4 CSEs or equivalent: | 30118 | 7.059% |
| 5 NVQ, HND, HNC or equivalent: | 63220 | 14.817% |
| 6 Other professional qualifications | 126957 | 29.755% |
| Total observations: | 426679 | |

In Chapter 3, the process of one-hot encoding the GWAS data for input into the GBGAN is described whereby an $M \times N$ integer array is transformed into a $M \times N \times 4$ binary array. For the sake of consistency all models were configured to also use categorical data by one-hot encoding the data. All the NN models were configured to take in $M \times N \times 4$ data as described previously. This format is decisive in the convolutional model and in GBGAN, but is not expected to affect the feed-forward model. The random forest and logistic regression models do not support 3 dimensional input, so for these models the data was one-hot encoded in a more traditional fashion; converting each row into 4 binary columns such that an array $M \times N$ would become an $M \times 4N$ array.

## 4.2 Hardware, Models, and Parameters

For the model testing and data processing computations this study utilized a single High Performance Computer(HPC). The HPC in question offered the following hardware:

- 1 20-core Intel® Core™ i9-7900X CPU @3.30GHz
- 4 NVIDIA TITAN Xp graphics cards

- 126 GB of RAM
- 256 GB of Swap Memory
- 11 TB storage capacity supported by 1x 1TB Samsung SSD-860-EVO and 1x 10TB Seagate IronWolf hard drive

In addition, the flowing software was installed on the system:

- Ubuntu Server 18.04 LTS, as the operating system
- Python 3.6, with package versions:
  - Keras(2.3.0)
  - Tensorflow(2.3.1)
  - Imbalanced-Learn(0.7.0)
  - Numpy(1.18.5)
  - Pandas(1.1.3)
  - PyPlink(1.3.5)
  - Scikit-Learn(0.23.2)
  - Scipy(1.4.1)
- both PLINK(1.9) and PLINK(2.0) [24], [45]

Despite this advanced hardware, training of the NN models and executing the data processing pipeline, both frequently utilized the full capacity of the resource. GBGAN especially used a large amount of memory to train the many weights of the generator and discriminator.

We selected logistic regression and random forest models to compare against GBGAN and the other network models. There are more complex models, custom-built for GWAS data, which may outperform these techniques, but these universal models will serve better for comparison. These models are well known and have previously been applied with success to GWAS data [32, 33, 36]. They are also both highly interpretable models. Random forests offers multiple measures of variable importance, and the ways that regression models can be analyzed are too long to list.

From a computational standpoint these algorithms are more accessible as well, the random forest models can be computed by modest hardware at very fast speeds. The logistic regression likewise requires very few resources to train. Consider this in comparison to NN models which have inspired hardware manufacturers to design custom processing units. GBGAN is admittedly complex, but will hopefully pick up on more complex patterns and out perform these simpler solutions.

Both the logistic regression, and random forest models were provided by the scikit-learn python library [46]. Scikit-learn also provided the weight balancing function, which generated class weights for all the models including the NN models.

To help with comparison, the NN based models were run with mostly the same parameters as one another. All NN models used a 30% dropout rate, an Adam optimizer, and a large batch size, $2^{13} \sim 8000$ samples per batch. The large batch size is key when training on weighted samples, and should be large enough that the minority class is well represented in each batch. In this case, the batch size was limited by the system memory, and was made as large as possible given that constraint.

In order to test the models on different output variables and the small and large input data sets, it was necessary to pick a baseline input and output combination. This baseline data is needed to serve as a standard, to which to compare other combinations. The small input data with reported ASD output was the best combination to serve as a baseline. Most models were able to succeed in classification at least to some extent with the lower dimensional small data set, and the reported data set was the largest direct measure of ASD available. To find the appropriate number of epochs for testing, each model was run on the baseline data set until its loss seemed to stabilize. For the feed-forward and CNN models this was about 30 epochs, for GBGAN it was 3 epochs of Step 1 training and 20 epochs of Step 2 training. The faster convergence of GBGAN was seen as a positive sign that the model was benefiting from its more complex training process.

Other than those changes that have been discussed in this and the previous chapter, the models used default parameters. There was an effort to experiment with alternatives in most cases, but in the limited time frame a methodical search was not an option. Hyperparameter optimization techniques would be useful in this situation, but with the size of the data and speed of the models, the time to apply such algorithms is not realistic.

Each model was tested with a 5-fold cross validation resampling procedure stratified by class. This setup was selected to take maximum advantage of the limited ASD positive samples, and to ensure the reproducibility of the results.

### 4.3 Evaluation With Extreme Class Imbalance

Once the models were run the processes of evaluation and analysis could begin, but this process was also affected by the severe imbalance of the data. Imbalanced data has colloquially been defined, in the binary case, to be 1 positive sample for every 100 total samples. The fact that this mimics ASDs prevalence in the population is worth noting. However, the prevalence of ASD in the UK Biobank is less than 0.1%. In this case accuracy scores, and F-scores are typically insufficient to draw conclusions [47].

Area Under the Receiver Operator Characteristic Curve (ROC AUC) score is a viable alternative. ROC AUC is relatively unaffected by imbalance, but still could mask inadequate performance, and may not match the output of other metrics [47, 48].

The geometric mean (G-mean) score is another metric that is worth considering. The geometric mean is the root of the product of class-wise sensitivity in a multi class case, and the root of the product of the sensitivity and specificity in the binary case [48]. This metric is valuable if you want to ensure that the classifier is not ignoring a minority class, and will also prevent false positives. However, this metric can not tell you if the classifier is better than random, and can be deceptive when results are weak.

The best policy is also the common sense approach, which is to use multiple metrics while taking care to note their strengths and weaknesses.

# 5 RESULTS

## 5.1 Discussions

### 5.1.1 ROC AUC Score

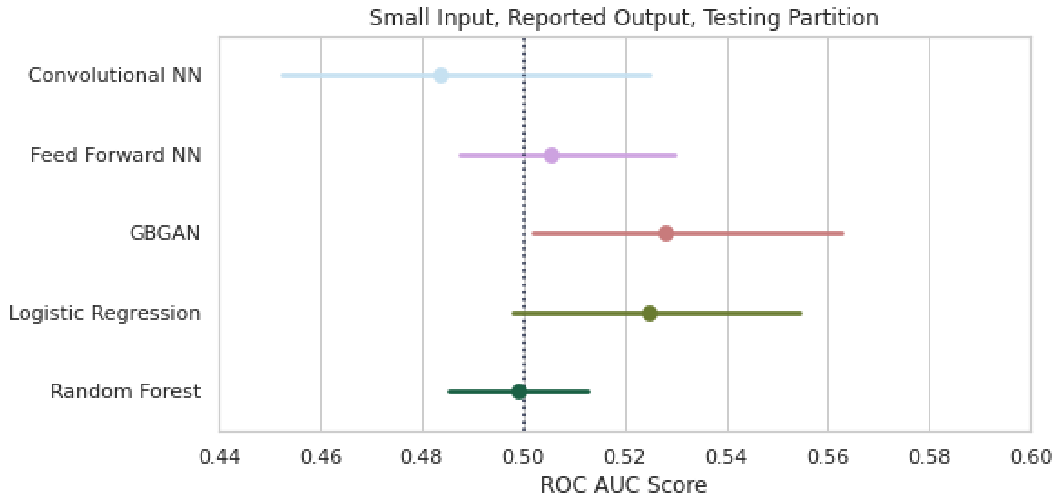

Figure 5.1: The ROC AUC score for models trained on the small input predicting reported output. This table includes only the calculations for the 5 testing data partitions. No model except GBGAN beat the average with perfect reliability, and GBGAN only by a small margin. The logistic regression model also performed above random on average, while the RF and CNN models performed the worst by this metric, averaging no better than random.

The ROC AUC score is a widly used metric, and was hoped to provide a valuable comparison of model performance. Unlike other classification metrics, ROC AUC uses the predicted probabilities to evaluate the models, rather than the predicted class labels. The larger the ROC AUC the better the classification is explained by the probabiliys. The best score by this metric is a 1, and any observation below 0.5 is worse than a random guess. To help with interpretation, a dotted line is drawn on the X axis at the key ROC AUC value of 0.5.

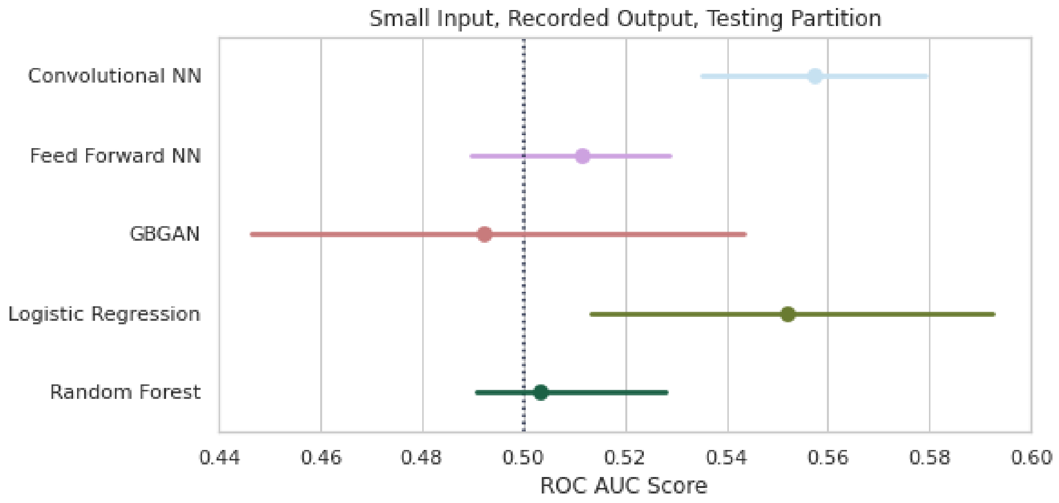

Figure 5.2: The ROC AUC score for models trained on the small input predicting recorded output. This table includes only the calculations for the 5 testing data partitions. The CNN model had better performance in this data set whereas GBGAN did poorly, the exact opposite of Figure 5.1 which used the same data but a different measurement of ASD. This plot brings into question the value of ROC AUC in this scenario. However, it also brings into question how stable the model's performance is.

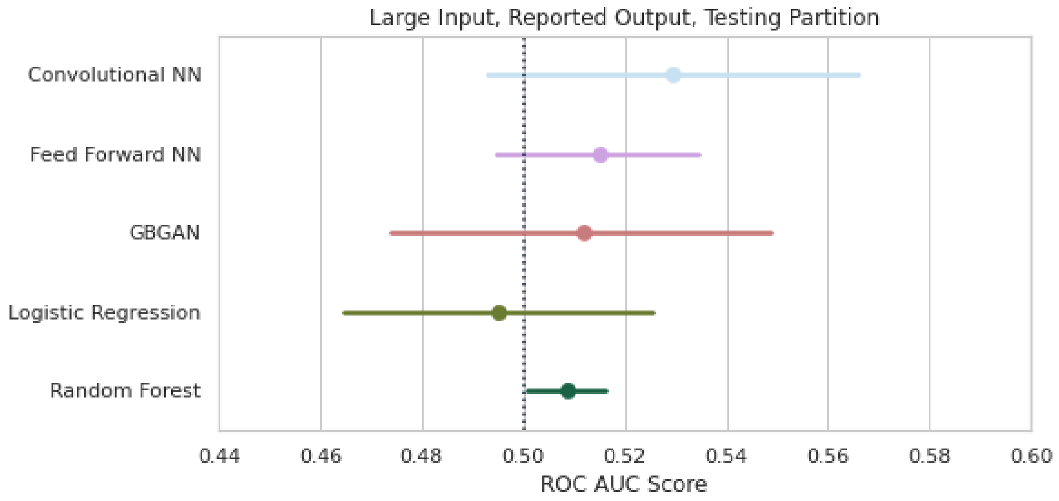

Figure 5.3: The ROC AUC score for models trained on the large input predicting reported output. This table includes only the calculations for the 5 testing data partitions. It is clear that this continues the theme of AUC being a Volatile indicator. Despite predicting on the same outcome data as in Figure 5.1, only with a higher dimensional input, this model shows extremely different results. This metric also masks the extremely poor performance of the Logistic regression and RF models.

Unfortunately this score is inconsistent across the different output and input data sets in a manner not seen in the other metrics. Even worse, the ROC-AUC score doesn't seem to represent the classification performance. For example, Logistic regression performed very well in Figure 5.2, but in that same test it failed to identify a single case of ASD. It is possible that this measurement is not appropriate for evaluating these classifiers, but we can not discount the idea that this variability is somehow a reflection of model uncertainty.

### 5.1.2 Geometric Mean Score

The G-mean is a valuable metric to show the difference in classification between training and testing partitions, differing data sets, and configurations of the model. For this metric, the worst value is 0 and the best is 1; a score below 0.5 indicates a poor predictive performance, but not necessarily worse than random. All the plots of this metric are color coded to show the testing and training data partitions. A large gap between the testing and training scores indicates over fitting.

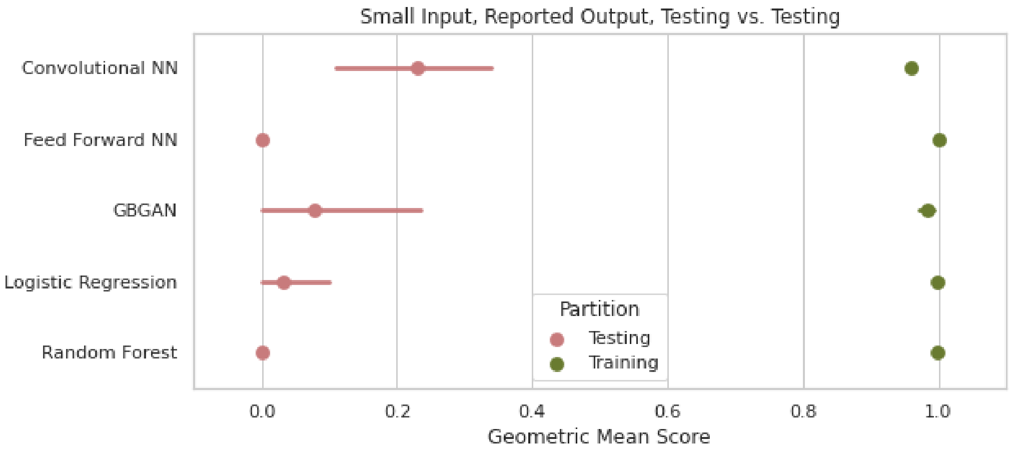

Figure 5.4: The G-mean score for models trained on the small input predicting reported output. It is clear that the data set was severely over fitted in all models, but the condition is less severe for the CNN and GBGAN models.

Although this metric is very well suited to imbalanced classification it is still challenging to interpret when the classifiers are weak. If any model had scored above a 0.5 in this metric, then we could say with confidence that it was performing well. However, G-mean can't be used to compare model performance in relation to probability. For example, consider a case where the minority class is 0.1% of the population. A purely random classifier that correctly classified 50% of the majority and minority classes would have a G-mean of 0.5, but a classifier that always correctly labels 100% of the majority class and 20% of the minority would score 0.44.

The G-mean can indicate a complete failure to predict one of the output classes, because in such a case the G-mean would be 0. It was a frequent occurrence for models to make no minority class predictions in the testing partition, or make only incorrect minority class predictions. This made the G-mean a valuable metric. Plots of G-mean show at a glance if the model made any minority class predictions, and how frequently. To give a visual example, G-mean shows clearly that the feed-forward model failed to correctly classify any minority samples in Figures 5.5, and 5.1, and succeed in somewhat in 5.8.

### 5.1.3 Expected True Positive and Confusion Matrices

Table 5.1: $\mathbb{E}(TP)$ and related values for models trained on the small input predicting the reported output. Models that made no true positive predictions were excluded from this table. The results from the testing partition were used in this table. Models that made no positive predictions were excluded to save space. The tables columns in order are: the name of the model, the fold number of the 5-fold cross-validation, the sum of the True Positives (TP) and False Positives (FP), the TP number alone, and $\mathbb{E}(TP)$.

| Model | Fold | TP+FP | TP | $\mathbb{E}(TP)$ |
|---|---|---|---|---|
| Convolutional NN | 1 | 8053 | 6 | 3.37 |
| | 2 | 5678 | 4 | 2.37 |
| | 3 | 6139 | 1 | 2.64 |
| | 4 | 6800 | 3 | 2.84 |
| | 5 | 6998 | 0 | 2.93 |
| Feed-forward NN | 1 | 12 | 0 | 0.01 |
| | 2 | 12 | 0 | 0.01 |
| | 3 | 16 | 0 | 0.01 |
| | 4 | 8 | 0 | 0.00 |
| | 5 | 27 | 0 | 0.01 |
| GBGAN | 1 | 1236 | 0 | 0.52 |
| | 2 | 1279 | 0 | 0.53 |
| | 3 | 2186 | 0 | 0.94 |
| | 4 | 7024 | 6 | 2.94 |
| | 5 | 1547 | 0 | 0.65 |
| Logistic Regression | 1 | 464 | 1 | 0.19 |
| | 2 | 534 | 0 | 0.22 |
| | 3 | 449 | 0 | 0.19 |
| | 4 | 482 | 0 | 0.20 |
| | 5 | 450 | 0 | 0.19 |

The confusion matrix is an indispensable tool when developing a classification model, but is rarely included in final write-ups. In this case however, values from the matrices like those in Table 5.1 can help us demonstrate a view of model performance specific to this highly imbalanced case. In all our testing no classifier was able to identify all or even the majority of positive ASD cases. However, when viewing the confusion matrices it was clear that some models were performing better than others, finding a smaller False Positive (FP) compared to the True Positive (TP). The limits of G-mean and ROC AUC, made them unable to indicate if the model was better than random. To make a meaningful comparison between these weak models, we turned to simpler laws of probability. We calculated $\mathbb{E}(TP)$, the expected TP under the assumption that the classifier is random. This expected TP number is intended to show how many true positive cases we would expect to see from a random classifier given how many total cases were predicted to be positive. If the algorithm predicted that half the population was ASD positive, and the algorithm was random, then we would expect that the true positive cases would be about half the total positive cases. This calculation for a binary classification in terms of observation and prediction counts is:

$$\mathbb{E}(TP) = \frac{\text{positive observations}}{\text{total observations}} \times (\text{total positive predictions})$$

In terms of a binary confusion matrix with counts of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) the calculation is:

$$\mathbb{E}(TP) = \frac{TP + FN}{TP + FN + TN + FP} \times (TP + FP)$$

For example, Table 5.1 shows that GBGAN only identified 6 ASD Positive cases in the 4th fold of the cross-validation, so it identified 50% more than expected of a random classifier in that fold. GBGAN also identified more total cases over all folds in that test than would be expected of a random classifier.

### 5.2 Analysis

The data set is extreme in a variety of ways making it difficult to comment on how the models would perform generally. Still, there is reason to believe that parts of this model would address the target issues. The combination of high dropout and convolutional layer looks to be robust to high dimensionality. Significant imbalance is likely to remain a fact in GWA studies, and the ability of sample weighting to be effective shows that we do not need to resort to under sampling. Furthermore, the training data from the generator suggests it may offer a viable path forward for interpretability.

### 5.2.1 The Effects of Balancing

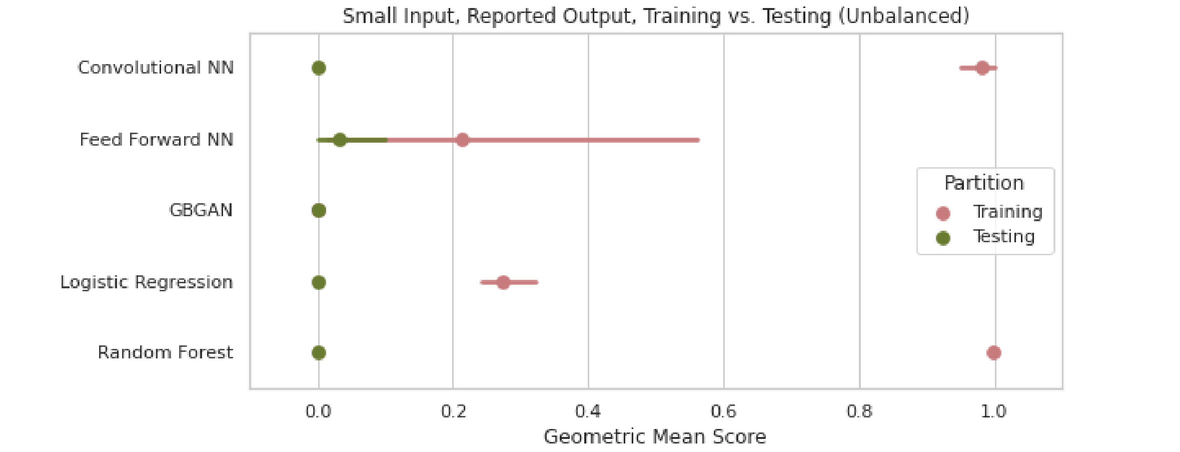

Figure 5.5: The G-mean score for models trained on the small input predicting reported output like in Figure 5.4, but the models were trained without balanced weights. Not using balancing weights resulted in clearly worse performance, whether the models were over fit or not. Only the feed-forward NN model attempted to identify any of the minority cases and probably only because it was under fitted.

Figure 5.5 should be compared to Figure 5.4 to understand the importance of balancing the data. G-mean clearly demonstrates that no model performed well without the use of class balancing. The feed-forward model was the only option to predict any of the minority class, and this is mostly likely because it was under trained. It is clear that balancing was necessary, although it made the training process slower.

### 5.2.2 Effect of Phenotype Measure, and Imbalanced Classes

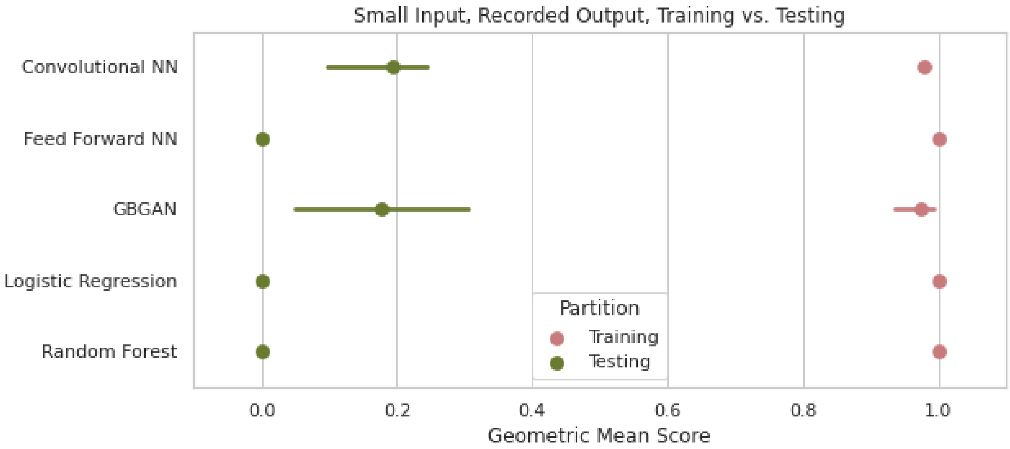

Figure 5.6: The G-mean score for models trained on the small input predicting recorded output. This plot can be compared to Figure 5.4 which used reported output. Using the recorded output increased the imbalance of the data set by a factor of 2. This had a detrimental effect on the ability to target the minority class for all models except for GBGAN and the CNN model.

Table 5.2: $\mathbb{E}(TP)$ and related values for models trained on the small input predicting recorded output. The results from the testing partition were used in this table. Models that made no positive predictions were excluded to save space. The tables columns in order are: the name of the model, the fold number of the 5-fold cross-validation, the sum of the True Positives (TP) and False Positives (FP), the TP number alone, and $\mathbb{E}(TP)$.

| Model | Fold | TP+FP | TP | $\mathbb{E}(TP)$ |
|---|---|---|---|---|
| Convolutional NN | 1 | 4574 | 1 | 0.85 |
| | 2 | 1223 | 0 | 0.23 |
| | 3 | 3988 | 1 | 0.74 |
| | 4 | 5439 | 1 | 1.01 |
| | 5 | 3156 | 1 | 0.59 |
| GBGAN | 1 | 16877 | 3 | 3.14 |
| | 2 | 1439 | 1 | 0.27 |
| | 3 | 802 | 1 | 0.15 |
| | 4 | 1811 | 0 | 0.34 |
| | 5 | 827 | 0 | 0.15 |

While the number of positive samples in the reported data set was extreme, it was still more than twice that of the recorded data set, as seen in Tables 4.2 and 4.1. Despite their small number, ASD observations in the recorded data set were suspected to be more accurate than the self reported diagnosis. This suspicion is hard to confirm without in person follow up, but it does slightly change the perception of the results like Figure 5.6 and Table 5.2. At first glance the models look similar to those trained with the reported data as the response variable. The CNN model beat $\mathbb{E}(TP)$ 3 out of 5 times, and the GBGAN succeeded 2 out of 5. The slight increase suggests that this data may more accurately measure the ASD phenotype, as suspected. The other models failed to classify any ASD cases correctly, possibly the decrease in observations outweighed the higher quality of the labeling.

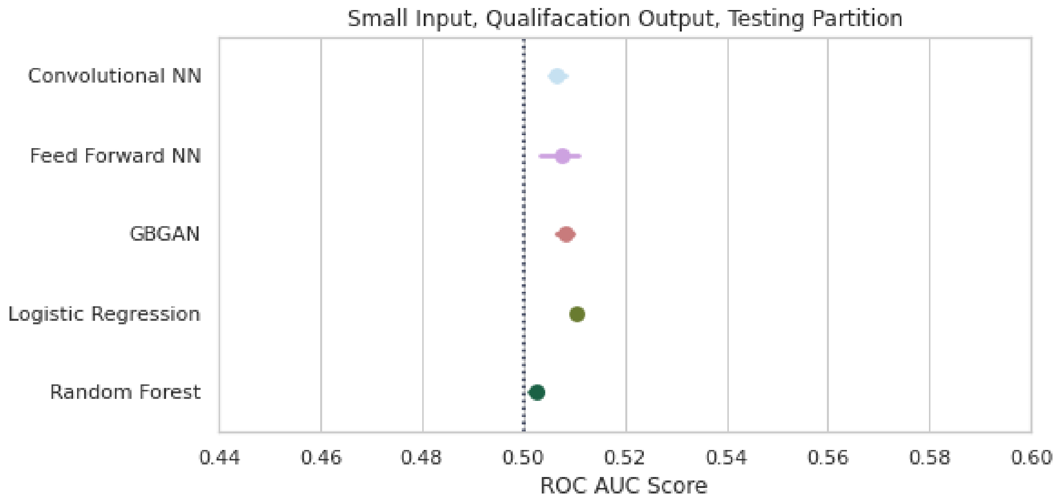

Figure 5.7: This plot shows the ROC AUC score for the small input predicting qualification output. This table includes only the calculations for the 5 testing data partitions. The predictors all show weak but highly consistent performance, very different from comparable Figures 5.1 and 5.7. In some models variance across folds was so small that it can't be seen in the plot. No model performed below 0.5, indicating they are better than random if only barely. The reduction in variability is most likely because the imbalance is also greatly reduced.

If we can study ASD indirectly then we can avoid class imbalance to some extent. The education qualification measure has more classes, but none are as imbalanced as the direct ASD measures. Comparing Table 4.3 to Tables 4.2 and 4.1 shows this extreme contrast. In this case the ROC AUC scores are very consistent across data folds, Figures 5.7 and 5.1 demonstrates this clearly. However, while the relationship between ASD and educational achievement exists, it is complex and is not well understood. Since the data set has been filtered to include genotypes associated with ASD only, it would be likely to exclude other genotypes related to educational achievement. There are also clearly external forces that play a larger role in educational qualifications than genetics. It is possible that no strong genetic classifier exists for this data set. It is therefore not surprising that the predictive power is barely present in Figure 5.7.

### 5.2.3 Effect of Dimensionality

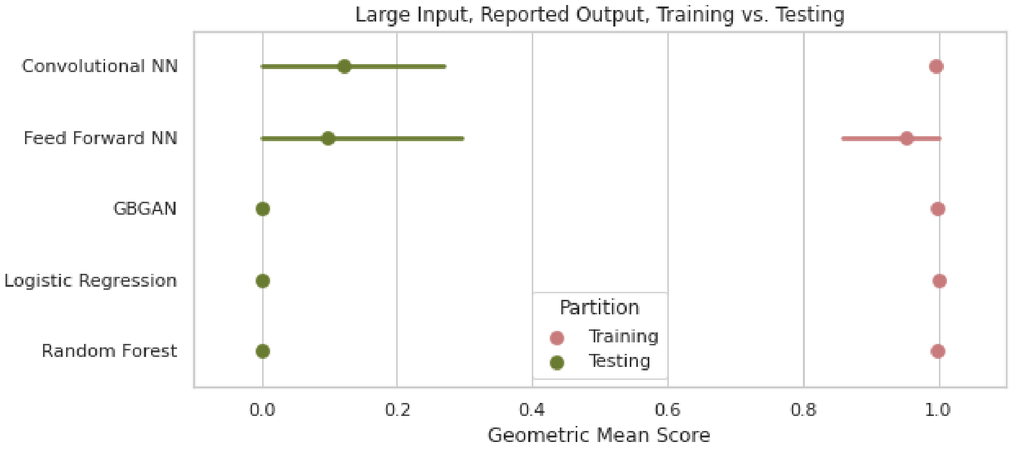

Figure 5.8: This plot shows the G-mean score for large input predicting reported output. This plot when compared to Figure 5.4 suggests that increasing the dimensionality of the data set by a factor of 2 had a severe detrimental effect on the ability to target the minority class for all models except for the feed-forward and the CNN.

The performance of most models suffered when transitioning from the small to the large data set. The CNN and feed-forward model seems to be the only ones to make any positive predictions of ASD. In the case of the convolution model there were now only 2 instances where $\mathbb{E}(TP)$ was exceeded. However, the CNN model performed unambiguously the best in this test, predicting over 5 times its expected value in both rounds where it excited $\mathbb{E}(TP)$, and limiting the number of false positives more effectively than it had in the previous test. The feed-forward had very strange performance in this test and is likely to be a fluke. In the lower dimensional test the feed-forward model never made any correct predictions, only in this test did it beat its $\mathbb{E}(TP)$ by $\frac{1}{12}^{th}$.

The GBGAN model was very cautious in this round of testing, possibly too cautious to make a prediction. Seeing as the discriminator portion of the GBGAN is nearly identical to the CNN model, we should expect that the performance can be transferred. Future work will be focused in part on modifying the training process of GBGAN to get the speed and performance of the standalone CNN model. In the meantime it is preferable that the model be cautious rather than random and inaccurate.

### 5.2.4 Model Analyses

Table 5.3: $\mathbb{E}(TP)$ and related values for models trained on the large input predicting reported output. The results from the testing partition were used in this table. Models that made no positive predictions were excluded to save space. The tables columns in order are: the name of the model, the fold number of the 5-fold cross-validation, the sum of the True Positives (TP) and False Positives (FP), the TP number alone, and $\mathbb{E}(TP)$.

| Model | Fold | TP+FP | TP | $\mathbb{E}(TP)$ |
|---|---|---|---|---|
| Convolutional NN | 1 | 1694 | 5 | 0.71 |
| | 2 | 114 | 0 | 0.05 |
| | 3 | 644 | 0 | 0.28 |
| | 4 | 698 | 2 | 0.29 |
| | 5 | 114 | 0 | 0.05 |
| Feed-forward NN | 1 | 28598 | 13 | 11.96 |
| | 2 | 1 | 0 | 0.00 |
| | 3 | 82 | 0 | 0.04 |
| | 4 | 2 | 0 | 0.00 |
| | 5 | 0 | 0 | 0.00 |
| GBGAN | 1 | 38 | 0 | 0.02 |
| | 2 | 193 | 0 | 0.08 |
| | 3 | 103 | 0 | 0.04 |
| | 4 | 24 | 0 | 0.01 |
| | 5 | 51 | 0 | 0.02 |

Over the combinations of input and output data the performance of all models varied, but patterns in performance still could be observed. To start with, the performance of the models with respect to response variables and high versus low dimensional input, can tell us a lot about how each model approached the classification task.

Logistic regression for example seems to be sensitive to both imbalanced data and high dimensionality. It is important to note that in all tests the logistic regression model failed to converge. The suggested solution is to increase the maximum number of iterations used to fit the data, this was tried but did not result in convergence, and so the default

100 iterations was used to save run time. With these settings, the logistic regression only succeeded in any capacity with the small input data set, and less imbalanced response variables. In future work the logistic regression should be incorporated as part of a meta algorithm, additive logistic regression would be a good place to start.

Regrettably the RF model had even lower performance than its logistic counterpart. The RF model was one of the worst performers in every test, including with qualifications as the outcome variable. RF models have been known to struggle with the weak predictive power of SNPs, a problem that can be addressed by increasing the number of variables necessary for a split. Given enough time more would have been done to optimize this model, which had the distinction of being the fastest model tested.

Of all the models the feed-forward model seemed to be the most inconstant, and it was also the worst performing NN based model. Possibly modifying parameters would increase the performance , but it would be hard to justify such effort when the simpler logistic regression model performed comparably. Feed-forward models have been used in GWAS context, but our data does not support them over CNN models in this case.

When viewed across all the tests, it seems that the standalone CNN model performed the best by G-mean and $E(TP)$. It was able to make predictions with high and low dimensionality and for each phenotype measurement, the only model able to do so. An argument could even be made that the model benefited from higher dimensions. The model seemed to beat $\mathbb{E}(TP)$ by more and lose by less cumulatively in Figure 5.3 than it did in Figure 5.1.

The GBGAN was a respectable second to the CNN model across test scenarios. GBGAN made fewer predictions than the standalone convolution model, and suffered fewer losses. The data could suggest that the model is making cautious but informed predictions based on very poor data. However, the possibility that these results are the product of luck can also not be discounted entirely. Statistical testing and a multitude of runs would be able to settle this question, but time prohibits such an analysis.

### 5.2.5 GBGAN Interpretability

In addition to imbalance and over fitting, we also intended to address the interpretability problem with GBGAN. This is an existential problem with deep learning, and will be a subject of research into the foreseeable future. In this case, we proposed that the generator portion of the GBGAN could be used to make class representative samples. To test the feasibility of this idea the learning rate and ROC-AUC scores were saved for each batch of GAN training, and plotted in Figures 5.9, and 5.10.

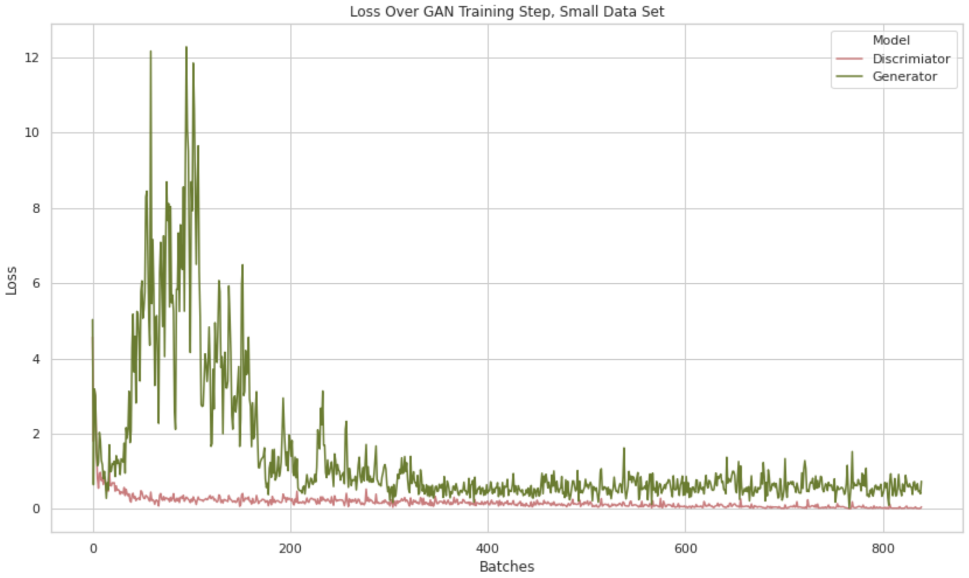

Figure 5.9: Recorded of loss over GAN Training step (Step 2) for models trained on the small input predicting reported output. Only the training on the first fold is shown, but it is similar to all folds. This plot suggests that the Discriminator converged and may have over fitted the input data, while the generator remained volatile.

Figure 5.9 shows the change in loss over the course of training on the first fold of the baseline data set. The plot shows the entire training process of 20 epochs, but contains relatively few batches because of the large batch size. It is clear that the generator in green had a more difficult time converging compared to the discriminator in pink. The generator may not have been in the optimal state even at the end of the 20 epochs of training, and may have benefited from further training. The difference between the two models is especially apparent at the start of training. This suggests that the autoencoder step may need to be adjusted, so that the generator and discriminator benefit more equally from it.

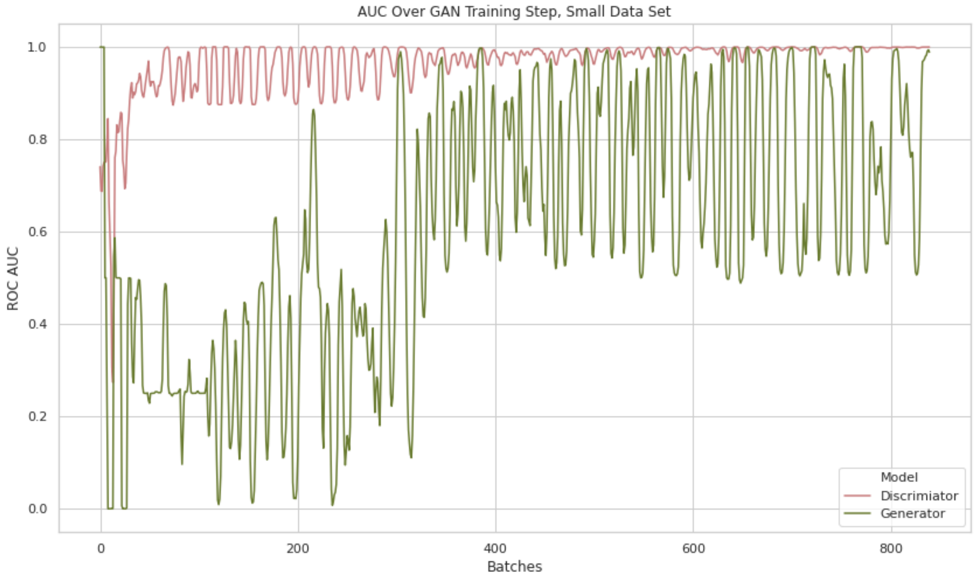

Figure 5.10:
Recorded of ROC AUC score in the GAN Training step (Step 2) for models trained on the small input predicting reported output. Only the training on the first fold is shown, but it is similar to all folds. This plot suggests that the accuracy of the generator is more connected with the discriminators inability to identify ASD positive cases, than it ability to identify false data.

Figure 5.10 shows the ROC-AUC score measured at the same time as the loss shown in Figure 5.9. It is important to note that this ROC-AUC score is an internal statistic calculated by the Keras package by a different method to those used in the rest of this chapter. The ROC-AUC has been shown to be volatile in its application towards the ASD phenotype predictions. In this case, the output of interest is sequences of SNPs, but they are still evaluated using target class labels, so the imbalance is still the same.

The results once again show that ROC-AUC is volatile, but it is still more appropriate than alternative metrics like accuracy.

In this case the ROC-AUC collaborates the implications of the loss. The generator seems to be converging slowly with the discriminator, at least for a time. It appears that at around 400 batches the discriminator became too accurate, and the generator was unable to learn as effectively. Increasing the dropout rate even more in the discriminator may help improve the generator, in addition to preventing overfitting in the discriminator.

The discriminator is most accurate when it identifies the false input of the generator and the correct class label for the input data. The generator is its most accurate when the discriminator identifies its output as the target class. So both benefit from the discriminator being a good classifier of the target class, but would diverge if the discriminator was good at identifying false input. For most of the training batches, the generator's change in performance seems match the discriminator's. This suggests that the accuracy of the generator represents the difficulty that the discriminator has with identifying ASD positive cases, and not that the generator produces clearly false output. This supports using the generator to understand the discriminator, but also suggests the model will struggle to identify phenotype positive individuals and causal SNPs. Clearly there is much work left to be done on GBGAN's interpretability, but there is also reason to think that this goal is within reach.

# 6 CONCLUSIONS

## 6.1 Summary

ASD is a serious condition, one that affects the lives of many people in the United Kingdom and around the world. However, the ASD genotype is not well understood. In particular, there is no method currently that can reliably predict a person's ASD status from their genotype. It is clear that ASD is at least partly genetic in nature, and many risk genes have been found, but the exact variants and mechanisms that cause ASD remain unknown. GWAS data has proven to be a powerful tool to discover risk loci and understand genetic diseases. In combination with deep learning methods, we believe that GWAS data is capable of delivering new insights into ASD. However, the problem of imbalanced phenotype, the curse of dimensionality, and a lack of model interpretability make it difficult to apply many deep learning methods to GWAS data. The Genetic Balancing Generative Adversarial Network (GBGAN) was proposed to address these challenges. The complex design of GBGAN was described with its multiple neural network structures and two-step training process. We outlined the data processing pipeline, including the LD clumping procedure, data encoding, and phenotype selection. Appropriate metrics for imbalanced classification were selected and described in order to evaluate if GBGAN addressed the target concerns. GBGAN was tested using a 5-fold cross validation procedure alongside simpler NN models and baseline algorithms.

## 6.2 Findings

Although no model gave a strong performance there are reasons to be optimistic about GBGAN as a step towards a purely genetic classifier for ASD. The GBGAN performed well on the imbalanced reported response variable and on the even more imbalanced recorded response, one of only two models to do so. When the high dimensional large data set was used GBGAN seemed too cautious, but it is clear that this failing can be addressed. The CNN model based on GBGAN was able to make classifications that were better than random, and was the only model to outperform GBGAN. The performance of the CNN model should be achieved in GBGAN by making only minor alterations. The generator portion of GBGAN started to converge, even in this limited test where it was not the target sub-model. In the future, the generator should be able to shed light into the performance of GBGAN, allowing it to meet the minimum threshold of interpretability. The results certainly support the application of convolutional NN models in the GWAS context. If CNN models can succeed as classifiers, then GBGAN can provide the same performance and the necessary additional features for deep learning in genetics to be highly successful.

### 6.3 Contributions

While it is disappointing that a strong predictor was not found, no high quality predictor has ever been found for ASD. Doing so would be revolutionary, and is certainly worth striving for, but this project can only serve as a small step in the long process towards that goal. The scientific understanding of ASD will continue to change, and it is quite possible that the condition that we call ASD will be redefined, or not exist, in the future. With the ever evolving nature of the ASD phenotype, understanding the genotype is both more challenging and more rewarding. Our results show that relatively simple machine learning techniques such as convolutional neural networks can outperform more established methods like logistic regression. This suggest that a deep learning approach to ASD screening is promising, despite its challenges.

By addressing data with imbalance phenotype, we address the reality that general purpose data sets are naturally imbalanced. The UK Biobank is a unique resource with a great deal of value for researchers, but ASD positive individuals make up a small minority of its population. With data on ASD cases likely to remain in short supply, it is important to make full use of the data that is available. GBGAN offers a start at using more of such data, both by preferring class weights over random under sampling, and by using an autoencoder structure for pre-training.

The generator of the GBGAN offers one of the simplest methods to increase the interpretability of the model. The evidence that the generator was able to train, and make data that the discriminator could accept, supports this approach. Compared to the standalone convolution model the GAN training seemed to make GBGAN more cautious, which is a valuable tool if it can be controlled.

### 6.4 Limitations

In addition to being optimistic for the future of GBGAN, it is important to realize significant limitations of this study. The universally poor performance of all the models is one such limitation. Several models seemed to beat $\mathbb{E}(TP)$ on average with different input and output data sets, but never by a very large amount. Comparing models to a completely random classifier is arguably a very low bar, and would not inspire confidence in anyone looking for a screening or diagnostic tool. Ultimately it is difficult to form strong conclusions when all the available models performed so close to random.

The UK Biobank is a valuable resource with data on an impressive number of individuals, but the Biobank posed serious challenges for the task. While our methods can easily accommodate the imbalance of the data, it is more difficult to get around the sheer lack of total ASD samples. At most, we could identify 181 cases of ASD in over 430,000 individuals, meaning that in each test data set of our 5-fold cross validation we had only $\sim 36$ ASD samples. This could be a good number of samples for a much simpler study, but when considering a minimum of 770 input variables it is hard to be confident. The small number of samples is likely why all results were so variable across evaluation metrics. Poor model performance was exacerbated because there were not enough samples to get consistent results on the small margin of success.

The size of the data also contributed to the slow test cycle. Even with substantial computational power the testing and training time for all models was long, and GBGAN was by far the slowest. The training of the discriminator portion of GBGAN is twice that of the simple CNN model because it trains on the same amount of real data and an equal amount of fake data. However, even the stand-alone CNN model is too slow to be used in many situations. For example, if the full imputed data set became available, the same data pipeline would make a much larger data set. On such a data set, the CNN model would exhaust the system resources or would take days to run. Methods with long run times are not uncommon in the field of genetics, but are typically made of components that are well understood, fully tested, and predictable. This is not compatible with the typical machine learning approach of experimentation and evolution through trial and error, and the convolutional model can not be easily divided into small testable components. Realistically, these models are simply too slow to enable the testing that would guarantee confident results.

The speed issue is also indirectly responsible for the uncertainty around GBGAN's structure. There would be more confidence in the results if the parameters of the models were better understood and optimized. Building models through intuition is fast becoming an obsolete approach. Methodologies for hyperparameter optimization are much more robust. A model that had been fully tested with a clear reproducible strategy would not need to use conjecture to defend design choices. Moreover, a model optimized from the start to perform well on data outside the training set would be less likely to over fit, and would deliver consistent results over cross validation folds.

The model design is the main focus of this thesis, but the data processing pipeline also limits the scope of this research. Choices in how the data is filtered, selected, and encoded are capable of changing the results for any models. For example, LD clumping is one of many methods to reduce the dimensionality of GWAS data, and each alternative method could favor different models. Summary statistics are desirable if samples are few, but such summary statistics are likely to shape the outcome of experiments based on their own possibly flawed assumptions. There are also multiple ways to process and express GWAS data. Even with the same subject and same SNPs, switching the SNP data from the predicted value to the confidence scores could change the results in surprising ways. In short, it is impossible to know how much of the model's final design and performance is tied to this pipeline, or any of its specific components.

### 6.5 Future Work

The options for long term future research in this field are vast, but it is clear what must be done in the near future to improve the GBGAN model and the power of its results. One of the first priorities should be reducing the time and resources necessary to train GBGAN. This model is intended to work with large high dimensional data sets, such datasets will require significant time to process. It currently takes GBGAN three times as long to complete its second training step compared to the time it takes a CNN model without any GAN training. The performance of GBGAN as a classifier is not commensurate with this increase. The difference is primarily from training the generator sub-model and training on the discriminator on the generated output. Clever tactics must be applied to reduce the time that the generator spends training. There is no theoretical reason for the generator to be trained with the same setup and class distribution as the discriminator. It may also be wise to take a design hint from OR-AC-GAN [38], and train a small separate portion of the discriminator to recognize fake data, rather than the full model. Or the design could adopt a form of batch random under sampling, and strike a balance between weighting and under sampling. Non-random under sampling techniques, like near-miss under sampling, are also options that could be explored. Additional speed increases will be made with the more integrated GAN training tools currently being implemented in the Keras library [43].

While it is important to speed up the second step of GBGAN training, it would be equally desirable to reduce the number of epochs the model needs overall. One way to reduce the total epoch, and especially the costly GAN epochs, is by making the first training step more effective. The current use of latent space is as simple as possible, but has similarities to Bayesian techniques and especially to variational autoencoders. Replacing the simple autoencoder in the first training step with a variational autoencoder would allow GBGAN to benefit from the latter's more integrated design and focus on latent space optimization. A conditional variational autoencoder would learn the coefficient of the latent distributions directly, rather than learning to generate latent vectors, and could be guided to ensure class recognition. This would remove the need for any intermediate steps between the first and second training steps, and could also reduce the model's reliance on GAN training to a more manageable number of epochs.

In addition to speeding up GBGAN with changes to the training algorithm, the sub models should be optimized for speed without sacrificing performance. The current design of GBGAN was created for the best possible performance as a classifier, and so options to increase the speed were not fully explored. It may well be possible to increase the stride length of the convolutional layers, add max-pooling layers, or remove layers completely, and retain the same performance while reducing the time and resources necessary.

If the speed of the model can be sufficiently improved then more experimentation will be possible. This may well start with the optimization and analysis of GBGAN's generator sub model. Confounding via heredity is a major limitation of focusing on the discriminator. A strong generator should be able to address heredity and shed light on which SNPs are meaningful to the model. Analysis could start with simple techniques, such as comparing the SNP frequency and correlation for different generated data sets. Contrasts could be made between ASD positive vs ASD negative generated samples, and samples that are accepted vs rejected by the discriminator. The latent vectors could also be tested to determine if they are indeed representative of multivariate normal distributions, and if they are sufficiently distinct. All tests would benefit from the large balanced data set that could be generated. The generators results are currently promising, but could be improved especially in the case of the minority class. In the more distant future the generator could even be modified to address concerns in the discriminator. For example, the generator could be conditioned on some measure of heredity so the discriminator can't rely on non-causal inherited SNPs. If the speed of the model can be increased it will be much easier to tune and optimize the generator along with the discriminator.

Both the generator and discriminator would benefit from the application of a hyperparameter tuning algorithm. The process of finding the optimal model design should not be left to chance when so many methods are available and integrated with Keras. Hyperparameter tuning will still be a long process even if the model is improved substantially, but it will allow for more confidence in the models going forward. The difference between ASD positive and negative cases is expected to be small and difficult to detect. The results presented here only support that assumption. The current design of GBGAN is unexpected, with liner and sigmoid activations that seem to improve performance. There is reason to think these activation layers may help the model by reducing over-fitting, but in a restrictive manner that ultimately weakens the model's ability to learn complex relations. Hyperparameter optimization can determine if some other set of parameters is capable of better performance, or if this is indeed the optimal model, as could well be the case.

At the same time as hyperparameter tuning, or possibly before, there should be a renewed search for the best possible ASD genotype and phenotype data. In addition to acquiring the imputed data, the genotype intensities, confidences, and SNP-posteriors should be explored. This exploration will be much easier with a faster and more optimized model, but may require that the parameters be changed to reflect the altered input. In terms of output variables, fluid intelligence is the most valuable score that is not already included in this study. If the association with ASD and fluid intelligence is strong then it could even be used as part of a meta learning pipeline.

A full understanding of ASD will require a monumental effort across scientific disciplines. The field of deep learning is still young, and researchers are still discovering what it may be capable of. There is a lot of work to be done in the short term to make deep learning models that meet the requirements for genetic research, but in the long term the possibilities are endless. A better understanding of how human genetics affect human physiology is a challenging goal, but one that can be achieved with sufficient effort and ingenuity. This thesis is one small step in that long road.

# REFERENCES

[1] R. J. L. Anney et al. Meta-analysis of gwas of over 16,000 individuals with autism spectrum disorder highlights a novel locus at 10q24.32 and a significant overlap with schizophrenia. *Molecular Autism*, 8(1):21, May 2017.

[2] O. Pain et al. Novel insight into the etiology of autism spectrum disorder gained by integrating expression data with genome-wide association statistics. *Biological Psychiatry*, 86(4):265 – 273, 2019. Autism Spectrum Disorder: Mechanisms and Features.

[3] T. K. Grønborg et al. Recurrence of Autism Spectrum Disorders in Full- and Half-Siblings and Trends Over Time: A Population-Based Cohort Study. *JAMA Pediatrics*, 167(10):947–953, 10 2013.

[4] S. Szymczak et al. Machine learning in genome-wide association studies. *Genetic Epidemiology*, 33(S1):S51–S57, 2009.

[5] M. C. Mills and C. Rahal. A scientometric review of genome-wide association studies. *Communications Biology*, 2(1):9, Jan 2019.

[6] C. P. Austin. Genotype.

[7] E. L. Goode. *Linkage Disequilibrium*, pp. 2043–2048. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[8] L. C. Brody. Polygenic trait.

[9] D. S. Mandell et al. Racial/ethnic disparities in the identification of children with autism spectrum disorders. *American journal of public health*, 99(3):493–498, Mar 2009. 19106426[pmid].

[10] T. A. Becerra et al. Autism spectrum disorders and race, ethnicity, and nativity: a population-based study. *Pediatrics*, 134(1):e63–e71, Jul 2014. PMC4067639[pmcid].

[11] T. Korioth. Autism prevalence now 1 in 68, varies by sex, race/ethnic group. *AAP News*, 2014.

[12] M. R. Donohue et al. Race influences parent report of concerns about symptoms of autism spectrum disorder. *Autism : the international journal of research and practice*, 23(1):100–111, Nov 2017.

[13] A. K. Halladay et al. Sex and gender differences in autism spectrum disorder: summarizing evidence gaps and identifying emerging areas of priority. *Molecular Autism*, 6(1):36, Jun 2015.

[14] W. Wang et al. De novo mutations from whole exome sequencing in neurodevelopmental and psychiatric disorders: From discovery to application. *Frontiers in Genetics*, 10:258, 2019.

[15] L. C. Francioli et al. A framework for the detection of de novo mutations in family-based sequencing data. *European Journal of Human Genetics*, 25(2):227–233, Feb 2017.

[16] C. R. Marshall et al. Structural variation of chromosomes in autism spectrum disorder. *The American Journal of Human Genetics*, 82(2):477 – 488, 2008.

[17] M. Hayashi et al. Superior fluid intelligence in children with asperger's disorder. *Brain and Cognition*, 66(3):306 – 310, 2008.

[18] F. Chen et al. Superior nonverbal intelligence in children with high-functioning autism or asperger's syndrome. *Research in Autism Spectrum Disorders*, 4(3):457 – 460, 2010.

[19] P. T. Shattuck et al. Postsecondary education and employment among youth with an autism spectrum disorder. *Pediatrics*, 129(6):1042–1049, 2012.

[20] A. Migliore et al. Predictors of employment and postsecondary education of youth with autism. *Rehabilitation Counseling Bulletin*, 55(3):176–184, 2012.

[21] V. Kuleshov et al. A machine-compiled database of genome-wide association studies. *Nature Communications*, 10(1):3341, Jul 2019.

[22] A. Korte and A. Farlow. The advantages and limitations of trait analysis with gwas: a review. *Plant Methods*, 9(1):29, Jul 2013.

[23] Y. Li et al. Genotype imputation. *Annual review of genomics and human genetics*, 10:387–406, 2009. 19715440[pmid].

[24] S. Purcell and C. Chang. Plink 1.9.

[25] Y. R. Li and B. J. Keating. Trans-ethnic genome-wide association studies: advantages and challenges of mapping in diverse populations. *Genome Medicine*, 6(10):91, Oct 2014.

[26] J. N. Hellwege et al. Population stratification in genetic association studies. *Current protocols in human genetics*, 95:1.22.1–1.22.23, Oct 2017. 29044472[pmid].

[27] I. Beyreli et al. Multitask learning on comorbid disorders improves gene risk prediction for autism spectrum disorder and intellectual disability. *bioRxiv*, 2020.

[28] A. L. Beam et al. Bayesian neural networks for detecting epistasis in genetic association studies. *BMC Bioinformatics*, 15(1):368, Nov 2014.

[29] B. Han et al. DASSO-MB: detection of epistatic interactions in genome-wide association studies using markov blankets. In *2009 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2009, Washington, DC, USA, November 1-4, 2009, Proceedings*, pp. 148–153. IEEE Computer Society, 2009.

[30] Y. Zhang and J. Liu. Bayesian inference of epistatic interactions in case-control studies. *Nature genetics*, 39:1167–73, 10 2007.

[31] D. Yoon et al. Phenotype prediction from genome-wide association studies: application to smoking behaviors. *BMC Systems Biology*, 6(2):S11, Dec 2012.

[32] J. J. Lee et al. Gene discovery and polygenic prediction from a genome-wide association study of educational attainment in 1.1 million individuals. *Nature genetics*, 50(8):1112–1121, Jul 2018. 30038396[pmid].

[33] X. Chen and H. Ishwaran. Random forests for genomic data analysis. *Genomics*, 99(6):323 – 329, 2012.

[34] Y. V. Sun. 4 - multigenic modeling of complex disease by random forests. In J. C. Dunlap and J. H. Moore, editors, *Computational Methods for Genetics of Complex Traits*, volume 72 of *Advances in Genetics*, pp. 73 – 99. Academic Press, 2010.

[35] G. BA et al. An application of random forests to a genome-wide association dataset: methodological considerations & new findings. *BMC Genet.*, 2010.

[36] A. Bureau et al. Mapping complex traits using random forests. *BMC genetics*, 4 Suppl 1:S64, 02 2003.

[37] V. Botta et al. Exploiting snp correlations within random forest for genome-wide association studies. *PLOS ONE*, 9(4):1–11, 04 2014.

[38] D. Wang et al. Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (or-ac-gan). *Scientific Reports*, 9(1):4377, Mar 2019.

[39] G. Mariani et al. BAGAN: data augmentation with balancing GAN. *CoRR*, abs/1803.09655, 2018.

[40] C. Davi and U. Braga-Neto. A semi-supervised generative adversarial network for prediction of genetic disease outcomes. *arXiv preprint arXiv:2007.01200*, 2020.

[41] J. Feng and S. Lu. Performance analysis of various activation functions in artificial neural networks. *Journal of Physics: Conference Series*, 1237:022030, jun 2019.

[42] C. Özkan and F. S. Erbek. The comparison of activation functions for multispectral landsat tm image classification. *Photogrammetric Engineering & Remote Sensing*, 69(11):1225–1234, 2003.

[43] F. Chollet et al. Keras. https://keras.io, 2015.

[44] C. Bycroft et al. Genome-wide genetic data on  500,000 uk biobank participants. *bioRxiv*, 2017.

[45] S. Purcell and C. Chang. Plink 2.0.

[46] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] L. A. Jeni et al. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 245–251, 2013.

[48] G. Lemaître et al. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.