DEEP LEARNING FOR EFFICIENT GWAS FEATURE SELECTION

A PREPRINT

Kexuan Li
Global Biometrics and Data Sciences
Bristol Myers Squibb
kexuan.li.77@gmail.com

December 27, 2023

ABSTRACT

Genome-Wide Association Studies (GWAS) face unique challenges in the era of big genomics data, particularly when dealing with ultra-high-dimensional datasets where the number of genetic features significantly exceeds the available samples. This paper introduces an extension to the feature selection methodology proposed by Mirzaei et al. (2020), specifically tailored to tackle the intricacies associated with ultra-high-dimensional GWAS data. Our extended approach enhances the original method by introducing a Frobenius norm penalty into the student network, augmenting its capacity to adapt to scenarios characterized by a multitude of features and limited samples. Operating seamlessly in both supervised and unsupervised settings, our method employs two key neural networks. The first leverages an autoencoder or supervised autoencoder for dimension reduction, extracting salient features from the ultra-high-dimensional genomic data. The second network, a regularized feed-forward model with a single hidden layer, is designed for precise feature selection. The introduction of the Frobenius norm penalty in the student network significantly boosts the method's resilience to the challenges posed by ultra-high-dimensional GWAS datasets. Experimental results showcase the efficacy of our approach in feature selection for GWAS data. The method not only handles the inherent complexities of ultra-high-dimensional settings but also demonstrates superior adaptability to the nuanced structures present in genomics data. The flexibility and versatility of our proposed methodology are underscored by its successful performance across a spectrum of experiments.

Keywords First keyword · Second keyword · More

1 Introduction

Feature selection stands as a critical cornerstone in numerous biological studies, a process pivotal in unraveling the complexities of data-intensive domains such as genome-wide association studies (GWAS), microarray analysis, and mass spectra analysis. The omnipresent challenge lies in datasets characterized by an inherent high-dimensional nature, coupled with a paucity of observations. In the realm of GWAS, a paradigmatic exploration where the identification of genes and the delineation of associations between single-nucleotide polymorphisms (SNPs) and human diseases take center stage, the landscape is riddled with obstacles. The GWAS datasets often manifest a disconcerting dichotomy — an expansive number of SNPs (e.g., $p \geq 10^5$) juxtaposed against a relatively diminutive sample size (e.g., $n \leq 10^3$). Navigating through this ultra-high-dimensional space and extracting a representative set of SNPs emerges as a persistent and formidable challenge in the quest for deciphering genetic underpinnings.

Numerous methodologies have been proposed to address the challenge of feature selection in GWAS data. To analysize GWAS data, the Cochran-Armitage trend test (Cochran, 1954; Armitage, 1955) has become a standard procedure for association testing in large-scale genome-wide association studies. However, various more complicated models had been developed to analize GWAS data as well. For instance, Fan and Lv (2008) introduced the Marginal Sure

Independence Screening procedure (SIS), specifically designed for ultrahigh-dimensional linear models, relying on Pearson correlations. Subsequent efforts in feature screening have yielded diverse procedures tailored to various models and successfully applied to GWAS data (Cui et al., 2015). Another prominent avenue in GWAS feature selection leverages the power of Lasso. In studies such as (Wu et al., 2009) and (Ayers and Cordell, 2010), penalized logistic regression underpinned by Lasso has been employed to unravel associations within GWAS data. Exploring Lasso coefficients, Arbet et al. (2017) investigated alternative, swift, and potent methods, highlighting the efficacy of permutation selection and analytic selection as alternatives to standard univariate analysis in GWAS data. Addressing the intricacies of joint multiple-SNP regression models, Yang et al. (2020) proposed a permutation-assisted tuning procedure within the Lasso framework to discern phenotype-associated SNPs. Beyond these Lasso-based models, a myriad of feature selection methods tailored for GWAS data exists. For example, de Oliveira et al. (2014) proposed a methodology to simultaneously select the most relevant SNPs markers for the characterization of any measurable phenotype described by a continuous variable using Support Vector Regression with Pearson Universal kernel. Li and Huang (2018) harnessed incremental feature selection to unearth novel gene expression patterns in brain tissues associated with early wake-up, drawing insights from GWAS data. Cueto-López et al. (2019) provided a comprehensive comparative study on various machine learning based feasure selection methods on or colorectal cancer. Meanwhile, Chu et al. (2020) proposed a two-step gene-detection procedure embedded in generalized varying coefficient mixed-effects models. For a more exhaustive exploration of these approaches, please refer to the comprehensive review by (Tadist et al., 2019; Pudjihartono et al., 2022). Despite their utility, classical methods encounter challenges in large biological datasets, including:

- **Feature Dependencies and Nonlinear Structures**:
  Traditional feature selection methods, broadly categorized as filter, wrapper, and embedded methods, exhibit limitations that become pronounced in the intricate landscape of GWAS datasets. For instance, filter methods often make assumptions about parametric model forms (e.g., linear models, lasso-based models, logistic regression models) and tend to overlook intricate interactions between features or nonlinear structures (Wu et al., 2009; Yang et al., 2020).
- **Lack of Flexibility**:
  Moreover, the rigidity of many algorithms poses a substantial roadblock. While the spotlight often shines on supervised feature selection, the realm of unsupervised feature selection is equally pivotal in biology. Consider, for example, clustering analysis, where the objective is to unearth new phenotypes by selecting genes devoid of prior phenotype knowledge (Solorio-Fernández et al., 2020; Varshavsky et al., 2006).
- **Dealing with Unbalanced Data and Reconstruction Challenges**:
  The nuances of unbalanced data, an omnipresent challenge in large biological datasets, introduce complexities (Abdulrauf Sharifai and Zainol, 2020). Additionally, the classical methods grapple with the intricate task of reconstruction and imputation, adding layers of intricacy to the feature selection process.

To address these challenges, we build upon the approach introduced by (Mirzaei et al., 2020), originally designed for feature selection but not explicitly tailored for ultra-high-dimensional data. Recognizing the unique demands of ultra-high-dimensional settings, particularly prevalent in GWAS data, we extend their method to enhance its applicability to datasets with a large number of features. Specifically, we introduce a Frobenius norm penalty into the student network, adapting the approach to better navigate the complexities associated with ultra-high-dimensional and small-sample scenarios. Our extended method maintains its flexibility, proving effective in both supervised and unsupervised scenarios, and excelling at uncovering intricate nonlinear structures and interactions within the data. The architecture of our method comprises two neural networks: the first dedicated to dimension reduction through an autoencoder or supervised autoencoder, and the second utilizing a regularized feed-forward network with only one hidden layer. This extension refines the original approach, making it well-suited for the challenges posed by ultra-high-dimensional datasets encountered in GWAS analyses.

The remainder of the paper is structured as follows: Section 2 delineates the problem of interest and offers a comprehensive literature review. Section 3 provides an in-depth exposition of the proposed method. In Section 4, we apply the proposed method and conduct comparisons with alternative approaches across various experiments. Finally, Section 5 delves into a discussion of the proposed method.

2 Problem of Interest and Literature Review

2.1 Problem Formulation

Let's delve into the intricacies of both supervised and unsupervised feature selection within the context of GWAS. Imagine a set of observations $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, \ldots, n$, representing a sample size n with p features, assumed to be independent and identically distributed (i.i.d.) from a distribution $p(\boldsymbol{x})$. In the realm of unsupervised feature selection specific to GWAS, the goal is to discern a subset $\mathcal{S} \subseteq \{1, 2, \ldots, p\}$ comprising the most discriminative and informative features. This subset, with $|\mathcal{S}| = k \leqslant p$, is accompanied by a reconstruction function $f : \mathbb{R}^k \to \mathbb{R}^p$. The critical aspect here is the mapping from a low-dimensional feature space $\mathbb{R}^k$ back to the original feature space $\mathbb{R}^p$. The aim is to minimize the expected loss between $f(\boldsymbol{x}^{(\mathcal{S})})$ and the original input $\boldsymbol{x}$, where $\boldsymbol{x}^{(\mathcal{S})} = (x_{s_1}, \ldots, x_{s_k})^\top \in \mathbb{R}^k$ and $s_i \in \mathcal{S}$ represents the low-dimensional k features. This process is pivotal in GWAS, where selecting relevant genetic features and understanding their intricate relationships contribute significantly to uncovering associations with complex traits and diseases. Moving to supervised feature selection, the complexity increases with the availability of both the sample design matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times p}$ and the label vector $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$. Here, y_i can be continuous or categorical, and the assumption is an unknown true relationship between a subset of features and y, expressed as $y = f(\boldsymbol{x}^{(\mathcal{S})})$. This relationship involves $\mathcal{S} \subseteq \{1, 2, \ldots, p\}$ with $|\mathcal{S}| = k \leqslant p$ and $\boldsymbol{x}^{(\mathcal{S})} \in \mathbb{R}^k$. Traditional linear assumptions, such as $y_i = f(\boldsymbol{x}_i^{(\mathcal{S})}) = \beta_0 + \sum_{j=1}^k \beta_j x_{i,s_j} + \epsilon_i$ for continuous responses or logistic regression $\log \frac{Pr(y_i=1)}{Pr(y_i=0)} = \beta_0 + \sum_{j=1}^k \beta_j x_{i,s_j}$ for categorical responses, are common in GWAS. However, the limitations of such linear models prompt the exploration of more expressive choices like neural networks. In the realm of GWAS, this adaptive approach is crucial for capturing the intricate genetic architecture underlying complex phenotypes.

2.2 Related Works

In recent years, the remarkable success of DNNs has reverberated across a myriad of domains, underscoring their versatility and potency. These domains span from the visually immersive realms of computer vision (He et al., 2016) to the intricate intricacies of deciphering language in natural language processing (Bahdanau et al., 2014). DNNs have left an indelible mark on recommendation systems (Zhang et al., 2019), offering personalized suggestions with unprecedented accuracy, and have even delved into the realms of drug discovery (Jiménez-Luna et al., 2020), spatial data analytics (Li et al., 2023b), computational biology (Angermueller et al., 2016), and the nuanced dynamics of complex systems (Li et al., 2021). Amidst this expansive landscape, the application of DNNs to feature selection has emerged as a fascinating area of exploration, garnering significant attention from researchers. The integration of DNNs with sparse group lasso to tackle problems of heterogeneous feature representations was a pivotal exploration by (Zhao et al., 2015). Meanwhile, Li et al. (2016) introduced Deep Feature Selection (DFS), a novel approach employing regularization techniques to rank feature importance, contributing to the nuanced understanding of feature relevance in complex datasets. In the pursuit of effective feature selection, Liu et al. (2017) proposed a strategy known as deep neural pursuit (DNP). This method strategically selects relevant features by leveraging the averaging out of gradients, employing multiple dropouts to lower variance and enhance robustness. Building upon the foundation of the knockoffs framework, Lu et al. (2018) incorporated this methodology into DNNs, enabling feature selection while maintaining a controlled error rate—an essential consideration in applications where precision is paramount. In the realm of high-dimensional nonlinear variable selection, Chen et al. (2021) established a comprehensive framework utilizing DNNs. They demonstrated the method's selection consistency under the condition of a generalized stable restricted Hessian in the objective function. This breakthrough contributes significantly to the understanding and application of

DNNs in scenarios characterized by complex, nonlinear relationships between variables. Adding to this rich tapestry of DNN applications, Gui et al. (2019) introduced an attention-based mechanism for supervised feature selection. This mechanism harnesses the power of attention in neural networks to dynamically emphasize and de-emphasize features, providing a nuanced approach to selecting relevant variables in a supervised context. Abid et al. (2019); Singh et al. (2023), who ingeniously harnessed the Gumble-Softmax trick (Jang et al., 2016). They introduced a concrete selector layer into the architecture, allowing gradients to seamlessly pass through the network during the feature selection process. Further enriching the landscape, Lemhadri et al. (2021) presented LassoNet, a feature selection network that introduced a residual layer between the input and output layers. Notably, this architecture imposed penalties on parameters within the residual layer while ensuring that the norm of parameters in the first layer remained less than the corresponding norm in the residual layer. Building upon the foundations laid by LassoNet, Li (2022) extended its applicability to censored data. Additionally, Li et al. (2023a) introduced a comprehensive approach that integrates deep learning and feature screening, yielding a framework capable of achieving both supervised and unsupervised feature selection. This hybrid methodology capitalizes on the strengths of both paradigms. Nevertheless, directly applying standard DNNs to GWAS data presents challenges. On one hand, interpretability is paramount in biological data, and many deep learning algorithms operate as "black-boxes," lacking inherent interpretability. On the other hand, the efficacy of most deep learning algorithms hinges on abundant training data, a luxury often unattainable in biological and medical datasets. These domains frequently contend with smaller sample sizes in comparison to the data dimension $(p >> n)$, posing a distinct set of challenges.

3 Methods

We present a two-stage deep neural network designed to effectively capture the intricate structure of genomic data. Our first-stage neural network, chosen for its complexity, employs a supervised autoencoder to ensure an expressive model capable of extracting the complex manifold inherent in the data. This complexity is essential for capturing nonlinear structures within the features, a task unattainable by simpler models like linear ones. In cases where a response variable is unavailable (unsupervised scenarios), our approach seamlessly transforms the supervised autoencoder into a standard autoencoder. The primary goal of the first stage is dimensionality reduction and feature extraction. Following the training of the first stage, we transform the high-dimensional input into a low-dimensional feature space, which serves as the output for the second stage. The second stage employs a single fully-connected layer with sparsity regularization to reproduce the output from the first stage. Utilizing sparsity regularization on the weight matrix facilitates feature selection by emphasizing the highest feature scores. Comprehensive details for each stage are elaborated in the ensuing sections.

3.1 The First Stage: Dimension Reduction

In the initial stage, a supervised autoencoder is employed to acquire a sophisticated representation of the input data. In scenarios where labels are unavailable (unsupervised situations), a conventional autoencoder is seamlessly substituted. Autoencoders, a distinct class of feed-forward neural networks, specialize in dimension reduction. Notably, many traditional dimension reduction techniques can be considered specific instances of autoencoders. For instance, Principal Component Analysis (PCA) can be conceptualized as an autoencoder when the loss function corresponds to mean square loss without an activation function.

A standard (unsupervised) autoencoder consists of two parts, the encoder and the decoder. Suppose the input space and output space is $\mathcal{X}$, the hidden layer space is $\mathcal{F}$. The goal is to find two maps $\Phi : \mathcal{X} \to \mathcal{F}; \Psi : \mathcal{F} \to \mathcal{X}$ which minimize the loss $\mathcal{L}_r(\Theta|\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2$, where $\mathcal{L}_r(\cdot)$ is the reconstruction loss function and $\Theta = [\Theta_\Phi, \Theta_\Psi]$ are the model parameters. Here, Φ is called the encoder, and Ψ is called the decoder. To better learn the non-linear structure of features, people always assume Φ and Ψ are neural networks. For example, if there is only one layer in both encoder and decoder with mean square loss, then $\Phi(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x} + b) \in \mathcal{F}, \Psi(\Phi(\boldsymbol{x})) = \sigma'(\boldsymbol{W}'\Phi(\boldsymbol{x}) + b')$, and

$$\mathcal{L}_r(\Theta|\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2 = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \sigma'(\boldsymbol{W}'\sigma(\boldsymbol{W}\boldsymbol{x}_i + b) + b')||_2^2, \tag{1}$$

where σ, σ' are nonlinear active functions, $\boldsymbol{W}, \boldsymbol{W}'$ are weight matrices, b, b' are bias vectors, and $|| \cdot ||_2$ is the l_2 norm. The standard autoencoder can be extended to many other forms, such as sparse autoencoder (SAE), denoising autoencoder (DAE), variational autoencoder (VAE).

In this paper, since we focus on the supervised feature selection, we will use a supervised autoencoder instead of the standard (unsupervised) autoencoder. In supervised autoencoder, we add an additional loss on the hidden layer, for example, mean square loss for continuous response or cross-entropy loss for categorical response. Let $\mathcal{L}_s(\cdot)$ be the supervised loss on the hidden layer and $\mathcal{L}_r(\cdot)$ be the reconstruction loss as in Equation (1). The loss in the supervised autoencoder with continuous response is:

$$\mathcal{L}(\Theta_1|\boldsymbol{X}, \boldsymbol{y}) = \mathcal{L}_s(\Theta_\Phi, \Theta_\Upsilon|\boldsymbol{X}, \boldsymbol{y}) + \mathcal{L}_r(\Theta_\Phi, \Theta_\Psi|\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} \left(||y_i - \Upsilon(\Phi(\boldsymbol{x}_i))||_2^2 + \lambda||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2\right), \tag{2}$$

where $\Theta_1 = [\Theta_\Phi, \Theta_\Psi, \Theta_\Upsilon]$ is the model parameters, Φ is the encoder, Ψ is the decoder, Υ is the regressor, and λ is the regularization parameter for the reconstruction loss. The corresponding loss in the supervised autoencoder with categorical response is:

$$\begin{aligned}\mathcal{L}(\Theta_1|\boldsymbol{X}, \boldsymbol{y}) &= \mathcal{L}_s(\Theta_\Phi, \Theta_\Upsilon|\boldsymbol{X}, \boldsymbol{y}) + \mathcal{L}_r(\Theta_\Phi, \Theta_\Psi|\boldsymbol{X}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(-\log\left(\frac{\exp(\Upsilon(\boldsymbol{x}_i)_{y_i})}{\sum_{c=1}^{C}\exp(\Upsilon(\boldsymbol{x}_i)_c)}\right) + \lambda||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2\right),\end{aligned} \tag{3}$$

where Θ_1, Φ, Ψ and λ are the same as Equation (2), C is the number of classes, and Υ is a classifier on the hidden layer with softmax output. The training process in the first stage is an optimization problem to minimize the loss function $\mathcal{L}(\Theta_1|\boldsymbol{X}, \boldsymbol{y})$.

Once the model is trained, we can extract the features by mapping the original input from $\mathcal{X}$ to the low dimension hidden space $\mathcal{F}$. Without loss of generality, we assume $\mathcal{X} = \mathbb{R}^p, \mathcal{F} = \mathbb{R}^h$ where $h \ll p$. Define normalized encoded input

$$\boldsymbol{x}_{\text{encode}} = \frac{\Phi(\boldsymbol{x}) - \min\Phi(\boldsymbol{x})}{\max\Phi(\boldsymbol{x}) - \min\Phi(\boldsymbol{x})} \in \mathcal{F},$$

which generates the abstract features from the original high-dimensional data and will be used in the second stage.

3.2 The Second Stage: Feature Selection

In the second stage, we train a single-layer neural network with a row-sparse regularization and a weight decay term on the weight matrix to mimic the $\boldsymbol{x}_{\text{encode}}$ from the first stage. The reason why we use a simple neural network is because we want to make sure the gradient can be easily back-propagated to the first layer such that the most important features can be successfully selected. To be more specific, the neural network in the second stage is defined as $\hat{\boldsymbol{x}} = \boldsymbol{W_2}\left(\sigma(\boldsymbol{W}_1\boldsymbol{x} + b_1)\right) + b_2$, where $\boldsymbol{W}_1, \boldsymbol{W}_2, b_1, b_2$ are the weight matrices and biases of the first layer and the output layer and $\sigma(\cdot)$ is the activation function. The loss function we want to optimize is

$$\mathcal{L}(\Theta_2|\boldsymbol{X}, \boldsymbol{X}_{\text{encode}}) = \frac{1}{n}\sum_{i=1}^{n} ||\hat{\boldsymbol{x}} - \boldsymbol{x}_{\text{encode}}||_2^2 + \alpha||\boldsymbol{W}_1||_{2,1} + \frac{\beta}{2}\sum_{i=1}^{2} ||\boldsymbol{W}_i||_F^2, \tag{4}$$

where $\Theta_2 = [\boldsymbol{W}_1, \boldsymbol{W}_2, b_1, b_2]$, α, β are penalty parameters,

$$||\boldsymbol{W}_{m\times n}||_{2,1} = \sum_{j=1}^{n}\left(\sum_{i=1}^{m}|w_{ij}|^2\right)^{\frac{1}{2}}, \ ||\boldsymbol{W}_{m\times n}||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|W_{ij}|^2} = \sqrt{\text{trace}(W^TW)}$$

are the $L_{2,1}$ norm and the Frobenius norm, respectively. We add a row-sparse regularization term ($||\cdot||_{2,1}$) in the first layer for feature selection and weight decay term (Frobenius norm) to avoid overfitting and help convergence. Once the second stage neural network is trained, we can define the feature scores as

$$s = \text{diag}(W_1W_1^T),$$

where s is a p dimensional vector of feature importance. The larger the feature importance is, the more significant role that feature plays. The idea of adding a row-sparse regularization to hidden layers in feature selection has also be investigated in literature (Scardapane et al., 2017; Han et al., 2018; Feng and Duarte, 2018). The algorithm and the architecture of the method are summarized in Algorithm 1 and Figure 3.3.

3.3 IMPLEMENTATION

The minimization of loss functions typically employs stochastic gradient descent (SGD), updating model parameters based on gradients computed from small, randomly drawn batches (usually 32 to 512). These batches significantly reduce the computational cost of gradient calculations. In the context of large-scale GWAS data, we utilize a scalable algorithm to optimize the loss function $\mathcal{L}$ through stochastic gradient descent. Our method encompasses crucial hyperparameters, including the number of layers, neurons per layer, dropout probability, learning rate, regularization parameters, and more. Configuring these hyperparameters accurately is vital for achieving optimal performance. However, determining suitable values can be challenging without domain expertise. Common strategies, such as grid search, random search ((Bergstra and Bengio, 2012)), and Bayesian optimization ((Snoek et al., 2012)), are often employed in practice. In our approach, we adopt a naive search, evaluating the loss for predefined parameter candidates based on a validation set.

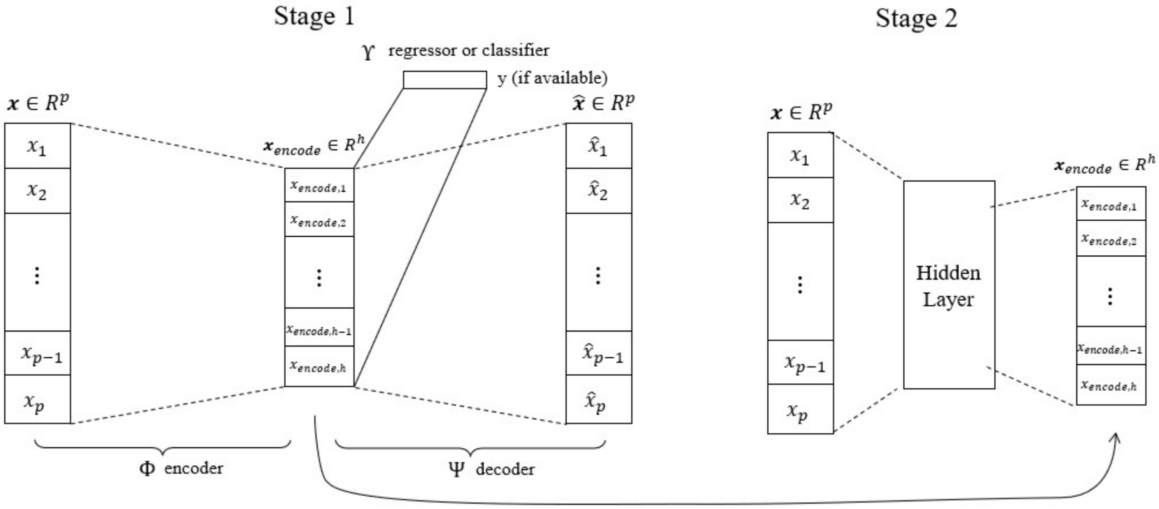

Figure 1: The architecture of the proposed DNN consists of an autoencoder based dimension reduction method for feature extraction (on the left) and a feature selection network (on the right). $\boldsymbol{x}_{\text{encode}} \in \mathbb{R}^h$ is the low-dimensional representation of the original input that captures most of the information of the data in $\mathbb{R}^p$, $h \ll p$, and is used to compute the strength of dependence with each feature. The second stage is a single-layer neural network with a row-sparse regularization and a weight decay term on the weight matrix to mimic the $\boldsymbol{x}_{\text{encode}}$ from the first stage.

Algorithm 1: Two Stage Deep Feature Selection Network

```
Input: input design matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, labels $y \in \{1, ..., C\}$ (if available), encoder network $\Phi$, decoder
    network network $\Psi$, regressor or classifier $\Upsilon$ (if available), learning rate, penalty term $\lambda$, regularization
    terms $\alpha, \beta$, and number of epochs for each stage $E_1, E_2$.
Output: feature scores
Train the First Stage Neural Network:
Initialize $\Theta_1 = [\Theta_\Phi, \Theta_\Psi, \Theta_\Upsilon]$.
for $e \in \{1, ..., E_1\}$ do
    $\hat{\boldsymbol{x}}_{\text{encode}} = \Phi(x)$
    $\hat{y} = \Upsilon(\hat{\boldsymbol{x}}_{\text{encode}})$ (if labels are available)
    $\hat{\boldsymbol{x}} = \Psi(\hat{\boldsymbol{x}}_{\text{encode}})$
    if response y is not available: then
        $\mathcal{L} = ||\hat{\boldsymbol{x}}||_2^2$
    if response y is continuous: then
        $\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(||y_i - \Upsilon(\Phi(\boldsymbol{x}_i))||_2^2 + \lambda||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2\right)$
    if response y is categorical: then
        $\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(-\log\left(\frac{\exp(\Upsilon(\boldsymbol{x}_i)_{y_i})}{\sum_{c=1}^{C}\exp(\Upsilon(\boldsymbol{x}_i)_c)}\right) + \lambda||\boldsymbol{x}_i - \Psi(\Phi(\boldsymbol{x}_i))||_2^2\right)$
    optimize the loss function using RMSprop.
Finish Training the First Stage Neural Network.
Map input into to hidden space: $\boldsymbol{x}_{\text{encode}} = \Phi(x)$.
Train the Second Stage Neural Network:
Initialize $\Theta_2 = [\boldsymbol{W}_1, \boldsymbol{W}_2, b_1, b_2]$
for $e \in \{1, ..., E_2\}$ do
    $\hat{\boldsymbol{x}} = \boldsymbol{W}_2\left(\sigma(\boldsymbol{W}_1\boldsymbol{x} + b_1)\right) + b_2$
    $\mathcal{L} = ||\hat{\boldsymbol{x}} - \boldsymbol{x}_{\text{encode}}||_2^2 + \alpha||\boldsymbol{W}_1||_{2,1} + \frac{\beta}{2}\sum_{i=1}^{2}||\boldsymbol{W}_i||_F^2$ optimize the loss function using RMSprop.
Finish Training the Second Stage Neural Network.
return $s = \text{diag}(W_1 W_1^T)$.
```

4 Experiments

In this section, we apply the proposed method to several experiments, with a particular emphasis on scenarios where the response variable is binary, representing dichotomous phenotypes. It's important to note that our approach is not limited solely to GWAS data and can be effectively employed in broader contexts, including both supervised feature selection and unsupervised feature selection.

4.1 Experiment 1

We follow the design of a previous GWAS feature selection paper (Yang et al., 2020). Suppose the number of observations is $n = 200, 500, 1000$ with dimension $p = 1000, 2000, 5000$. Let $y \in \{0, 1\}$ be the binary response and $\boldsymbol{x} = (x_1, \ldots, x_p)^T, x_i \in \{0, 1, 2\}, i = 1, \ldots, p$ be the SNPs. The data generation process is described as below. We first generated n independent p-dimensional random variable $\boldsymbol{Z}_i = (Z_{i1}, \ldots, Z_{ip})^T, i = 1, \ldots, n$, following a multivariate Gaussian distribution $\mathcal{N}(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p \times p$ covariance matrix to capture the correlation between SNPs. The design of $\boldsymbol{\Sigma}$ is the same as (Yang et al., 2020). That is

$$\boldsymbol{\Sigma}_{ij} = \begin{cases} 1 & i = j, \\ \rho & i \neq j, i, j \leq p/20, \\ 0 & \textit{eotherwise}, \end{cases}$$

and $\rho = 0, 0.4, 0.8$. This design allows closer SNPs having a stronger correlation. Next, we randomly generate p minor allele frequencies (MAFs) $m_1, \ldots, m_p$ from the uniform distribution $Uniform(0.05, 0.5)$ to represent the strength of heritability. For the i-th observation $(y_i, \boldsymbol{x}_i), i = 1, \ldots, n$, the SNPs are generated be the following rule:

$$x_{ij} = \begin{cases} 0 & Z_{ij} \leq c_1, \\ 1 & c_1 < Z_{i,j} < c_2, \\ 2 & Z_{i,j} \geq c_2, \end{cases}$$

where c_1, c_2 are the $(1 - m_j)^2$-quantile and $(1 - m_j^2)$-quantile of $\{Z_{1j}, \ldots, Z_{nj}\}, j = 1, \ldots p$, respectively. We standardize each SNP to have a mean of zero and a variance of one. It is worth mentioning that this data generation process is the same as (Yang et al., 2020).

We further define a set $\mathcal{J} = \{j_1, \ldots, j_{10}\}$, where $j_k, k = 1, \ldots, 10$, are randomly sampled from $\{1, \ldots, p\}$ without replacement. The phenotype y is generated according to the dichotomous phenotype model

$$\begin{aligned} \log \frac{\pi}{1-\pi} &= -3 + \beta_1 x_{j_1} + \beta_2 \sin(x_{j_2} + x_{j_1}) + \beta_3 \log(x_{j_3}^2 + 1) + \beta_4 x_{j_4}^2 \\ &+ \beta_5 \text{sign}(x_{j_5} - 1) + \beta_6 \max(x_{j_6} + x_{j_8}, 2) + \beta_7 x_{j_7} \text{sign}(x_{j_7} - 1) \\ &+ \beta_8 \sqrt{|x_{j_8} - 1|} + \beta_9 \cos(x_{j_9}) + \beta_{10} \tanh(x_{j_{10}}) \end{aligned}$$

where $\beta_j \sim Uniform(1, 2)$, for $j = 1, \ldots, 10$, and $y \sim Binomial(1; \pi)$. In other words, there are 10 out of p active SNPs associated with phenotype.

We compare our method with other four methods: Armitage trend test (ATT), iterative SIS (Fan and Lv, 2008), Lasso penalized logistic regression (Wu et al., 2009), and PLasso (Yang et al., 2020). To evaluate the performance of our method, In each simulation, we let $\mathcal{J}$ denote the set of true active variables and $\widehat{\mathcal{J}}_i$ the set of selected variables in the i-th replication, $i = 1, 2, \ldots, 500$. The following metric is used to evaluate the performance of each method:

$$\varrho = \frac{1}{500} \sum_{i=1}^{500} \frac{|\widehat{\mathcal{J}}_i \cap \mathcal{J}|}{|\mathcal{J}|}, \tag{5}$$

which measures the proportion of active variables selected out of the total amount of true active variables. Such metric has been widely used in feature selection literature (see, for instance, Fan and Lv (2008) and Yang et al. (2020)).

To validate the comparison, we use the Wilcoxon method to test whether the difference between DNN method and each of the other methods is statistically significant. As such, the following hypotheses are considered:

$$H_0 : \varrho_{\text{DNN}} = \varrho_k \;\text{ versus }\; H_a : \varrho_{\text{DNN}} > \varrho_k, \tag{6}$$

where $\{\varrho_1, \varrho_2, \varrho_3, \varrho_4\}$ are the ϱs associated with ATT, ISIS, LassoWu, andPLasso, respectively. Denote by p_k the p-value of the Wilcoxon test for a comparison between DNN and method $k \in \{1, 2, 3, 4\}$, and let

$$p_{max} = \max_{k=1,2,3,4} p_k. \tag{7}$$

To test the statistical significance, we aim to show p_{max} is less than 1%.

In each replication and for various combinations of ρ, n, and p, we utilize the five methods to conduct feature selection on the simulated data. The results, depicted in Table 1, showcase the average proportion of features selected by each method out of the 10 true active features over 500 replications, as defined in Equation 5. The corresponding standard errors are presented in parentheses. Across different scenarios involving ρ, n, and p, distinctive patterns in performance emerge. Specifically, under the conditions of $\rho = 0, n = 200$, and $p = 1000$, the proposed DNN method exhibits a feature selection rate of 63%, surpassing PLasso, which follows closely, while ATT achieves a rate of 38%. As expected, an increase in the sample size positively influences the performance of all methods, with DNN leading the pack at 83% feature selection when $n = 1000$. However, elevated dimensions in p and/or heightened dependence with ρ among the

Table 1: Results of Simulation 1: the averaged ϱ over 500 replicates (with its standard error in parentheses) of various methods using different combinations of ρ, n, and p. (+) means that $p_{\text{max}} < 0.01$, where p_{max} is the maximum p-values for the six tests defined in (7).

ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0	(200, 1,000)	0.38	0.45	0.51	0.55	**0.63** (+)
		(0.03)	(0.04)	(0.04)	(0.05)	(0.04)
0	(500, 1,000)	0.44	0.52	0.59	0.63	**0.72** (+)
		(0.04)	(0.04)	(0.05)	(0.05)	(0.06)
0	(1,000, 1,000)	0.52	0.60	0.66	0.71	**0.83** (+)
		(0.05)	(0.05)	(0.05)	(0.06)	(0.06)
ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0	(200, 3,000)	0.32	0.39	0.44	0.48	**0.55** (+)
		(0.03)	(0.04)	(0.03)	(0.04)	(0.04)
0	(500, 3,000)	0.39	0.45	0.50	0.55	**0.65** (+)
		(0.04)	(0.04)	(0.03)	(0.04)	(0.04)
0	(1,000, 3,000)	0.42	0.53	0.57	0.61	**0.72** (+)
		(0.04)	(0.05)	(0.05)	(0.05)	(0.06)
ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0.5	(200, 1,000)	0.33	0.41	0.45	0.50	**0.59** (+)
		(0.03)	(0.04)	(0.03)	(0.04)	(0.04)
0.5	(500, 1,000)	0.40	0.47	0.55	0.58	**0.67** (+)
		(0.04)	(0.04)	(0.04)	(0.05)	(0.05)
0.5	(1,000, 1,000)	0.48	0.55	0.60	0.66	**0.74** (+)
		(0.05)	(0.05)	(0.04)	(0.05)	(0.05)
ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0.5	(200, 3,000)	0.28	0.35	0.40	0.43	**0.50** (+)
		(0.03)	(0.03)	(0.03)	(0.04)	(0.03)
0.5	(500, 3,000)	0.35	0.44	0.46	0.51	**0.60** (+)
		(0.04)	(0.04)	(0.03)	(0.04)	(0.04)
0.5	(1,000, 3,000)	0.39	0.49	0.52	0.57	**0.68** (+)
		(0.04)	(0.05)	(0.04)	(0.05)	(0.05)
ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0.8	(200, 1,000)	0.30	0.37	0.42	0.47	**0.55** (+)
		(0.03)	(0.04)	(0.03)	(0.04)	(0.04)
0.8	(500, 1,000)	0.36	0.44	0.52	0.55	**0.62** (+)
		(0.04)	(0.04)	(0.04)	(0.04)	(0.04)
0.8	(1,000, 1,000)	0.43	0.51	0.56	0.61	**0.70** (+)
		(0.04)	(0.05)	(0.04)	(0.05)	(0.04)
ρ	(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
0.8	(200, 3,000)	0.25	0.32	0.36	0.40	**0.46** (+)
		(0.03)	(0.03)	(0.03)	(0.03)	(0.03)
0.8	(500, 3,000)	0.32	0.41	0.41	0.47	**0.55** (+)
		(0.04)	(0.04)	(0.03)	(0.04)	(0.04)
0.8	(1,000, 3,000)	0.33	0.45	0.49	0.52	**0.62** (+)
		(0.04)	(0.05)	(0.04)	(0.04)	(0.04)

SNPs adversely affect the efficacy of feature selection methods. The p-values, consistently below 1%, firmly establish that DNN consistently outperforms the other methods across all evaluated aspects.

4.2 Experiment 2

In this simulation, we utilize the *Breast Cancer Coimbra Data Set* sourced from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra). The dataset comprises 9 quantitative predictors and a binary response, denoting the presence or absence of breast cancer, with a sample size of $n = 116$. While it is reasonable to assume a strong association between these 9 predictors and the response, the specific impact on the

Table 2: Results of Simulation 3: the averaged ϱ over 500 replicates (with its standard error in parentheses) of various methods using different ps. (+) means that $p_{\max} < 0.01$, where $p_{\max}$ is the maximum p-values for the six tests defined in (7).

(n, p)	ATT	ISIS	LassoWu	PLasso	DNN
(116, 1,000)	0.17	0.20	0.19	0.18	**0.99** (+)
	(0.02)	(0.02)	(0.2)	(0.03)	(0.2)
(116, 5,000)	0.16	0.18	0.16	0.15	**0.98** (+)
	(0.02)	(0.03)	(0.03)	(0.03)	(0.03)
(116, 10,000)	0.14	0.13	0.0.14	0.13	**0.98** (+)
	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)

response remains unknown. To evaluate our method, we adopt the following procedure: incorporating an additional $p - 9$ irrelevant predictors from a standard normal distribution, independent of the response, with p taking values of $1000, 5000, 10000$. Essentially, out of all p predictors, only 9 are active variables correlated with the response, rendering the rest irrelevant. The manner in which these 9 variables influence the response remains unspecified. The results presented in Table 2 encompass the statistic ϱ based on 500 replications and its corresponding standard error. Additionally, we assess whether the maximum p-value for the six tests defined in 7 is less than 1% for $p = 1000, 5000$, and 10000. An intriguing observation is that, irrespective of the dimensionality p, the DNN method consistently identifies all features. As the feature dimension p expands, the selection precision of other methods diminishes. In contrast, our proposed method exhibits superior capability in selecting the correct active predictors, outperforming others across varied dimensionalities.

4.3 Experiment 3

In this simulation, we leveraged the robust capabilities of the GWAsimulator tool (Li and Li, 2008), a proficient C++ program meticulously designed to simulate genotype data originating from genomic SNP chips widely employed in GWAS. Renowned for its efficiency, GWAsimulator implements a rapid moving-window algorithm (Durrant et al., 2004), allowing for the simulation of comprehensive genome datasets catering to both case-control and population samples. Notably, this versatile program offers the added flexibility of simulating specific genomic regions as per user specifications, enhancing precision and customization in simulations. Specifically designed for case-control simulations, GWAsimulator empowers users to define various disease model parameters, including disease prevalence, the count of disease loci, and specific details for each locus such as location, risk allele, and genotypic relative risk. The program's adaptability extends further to enable users to focus simulations on particular genomic regions, providing a tailored and nuanced approach. In the context of our simulation, GWAsimulator played a pivotal role in generating the GWAS data. A comprehensive summary of the simulated data is presented in Table 3. Subsequently, we applied five distinct approaches to analyze this dataset, and the outcomes are succinctly encapsulated in Figure 2. Notably, the results underscore the superior selection accuracy achieved by the DNN approach in comparison to the other methods.

5 Conclusion

In summary, our extended deep neural network approach emerges as a robust and adaptive solution for ultra-high-dimensional feature selection within GWAS data. By refining Mirzaei et al.'s (2020) method, we have effectively addressed the unique challenges associated with ultra-high-dimensional and small-sample setups prevalent in genomics research. The integration of a Frobenius norm penalty into the student network serves as a pioneering enhancement, significantly broadening the method's applicability to GWAS datasets characterized by intricate structures and limited sample sizes. Beyond its technical advancements, our approach stands out for its flexibility, demonstrated through comprehensive experiments across diverse genomic scenarios. The method's proficiency in unraveling complex relationships embedded in GWAS datasets positions it as a promising solution for gaining valuable insights into genetic associations with diseases. This work makes a substantial contribution to advancing feature selection methodologies,

Table 3: The disease locus position is the position in the input phased file, not the physical position on a chromosome. For example, the disease locus in chromosome 19 is the 2885th SNP, with allele 1 as the disease-risk allele. The first relative risk, RR_1, is the risk ratio of the genotype with one copy of the risk allele versus that with zero copy of the risk allele. Similarly, the second relative risk, RR_2, is the risk ratio of the genotype with two copies of the risk allele versus that with zero copy of the risk allele. If it is "M", then the multiplicative effect is assumed and $RR_2 = RR_1^2$. If it is "D", then the dominance effect is assumed, and $RR_2 = RR_1$. For the recessive effect, specify 1.0 for RR_1. All relative risks should be ≥ 1.

Chromosome Number	Position	Disease Variant Allele	The First Genotypic Relative Risks	The Second Genotypic Relative Risks	Start Position	End Position	Dimensionality
2	10714	0	1.1	D	10000	12000	25215
6	4322	1	1.0	1.1	3000	5600	20269
11	9067	1	1.5	M	8000	10000	14520
18	9659	1	1.1	M	6000	10000	10441
19	2885	1	1.5	M	1000	4000	5789
20	3357	0	1.1	2.0	1000	5000	7802
23	7607	0	1.5	1.5	7000	9000	9120

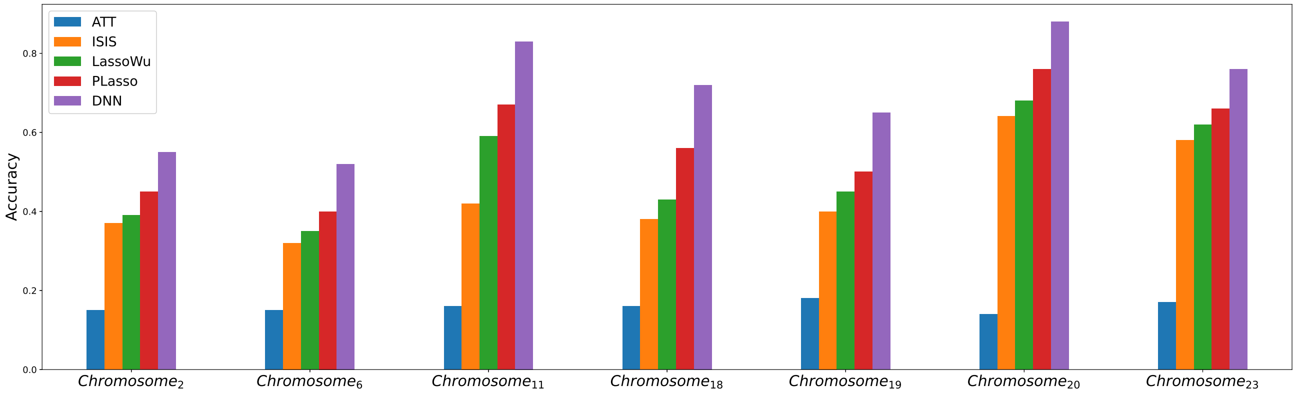

Figure 2: The results of Experiment 3.

explicitly tailored to the specific demands of ultra-high-dimensional genomics research. As big genomics data continues to evolve, our extended deep neural network approach emerges as a potent tool for researchers seeking accurate and interpretable feature selection in the complex landscape of GWAS.

References

Abdulrauf Sharifai, G. and Zainol, Z. (2020). Feature selection for high-dimensional and imbalanced biomedical data based on robust correlation based redundancy and binary grasshopper optimization algorithm. *Genes*, 11(7):717.

Abid, A., Balin, M. F., and Zou, J. (2019). Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology*, 12(7):878.

Arbet, J., McGue, M., Chatterjee, S., and Basu, S. (2017). Resampling-based tests for lasso in genome-wide association studies. *BMC genetics*, 18:1–15.

Armitage, P. (1955). Tests for linear trends in proportions and frequencies. *Biometrics*, 11(3):375–386.

Ayers, K. L. and Cordell, H. J. (2010). Snp selection in genome-wide and candidate gene studies via penalized logistic regression. *Genetic epidemiology*, 34(8):879–891.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Chen, Y., Gao, Q., Liang, F., and Wang, X. (2021). Nonlinear variable selection via deep neural networks. *Journal of Computational and Graphical Statistics*, 30(2):484–492.

Chu, W., Li, R., Liu, J., and Reimherr, M. (2020). Feature selection for generalized varying coefficient mixed-effect models with application to obesity gwas. *The annals of applied statistics*, 14(1):276.

Cochran, W. G. (1954). Some methods for strengthening the common χ 2 tests. *Biometrics*, 10(4):417–451.

Cueto-López, N., García-Ordás, M. T., Dávila-Batista, V., Moreno, V., Aragonés, N., and Alaiz-Rodríguez, R. (2019). A comparative study on feature selection for a risk prediction model for colorectal cancer. *Computer methods and programs in biomedicine*, 177:219–229.

Cui, H., Li, R., and Zhong, W. (2015). Model-free feature screening for ultrahigh dimensional discriminant analysis. *Journal of the American Statistical Association*, 110(510):630–641.

de Oliveira, F. C., Borges, C. C. H., Almeida, F. N., e Silva, F. F., da Silva Verneque, R., da Silva, M. V. G., and Arbex, W. (2014). Snps selection using support vector regression and genetic algorithms in gwas. *BMC genomics*, 15:1–15.

Durrant, C., Zondervan, K. T., Cardon, L. R., Hunt, S., Deloukas, P., and Morris, A. P. (2004). Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes. *The American Journal of Human Genetics*, 75(1):35–43.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(5):849–911.

Feng, S. and Duarte, M. F. (2018). Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation. *Neurocomputing*, 312:310–323.

Gui, N., Ge, D., and Hu, Z. (2019). Afs: An attention-based mechanism for supervised feature selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3705–3713.

Han, K., Wang, Y., Zhang, C., Li, C., and Xu, C. (2018). Autoencoder inspired unsupervised feature selection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2941–2945. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jiménez-Luna, J., Grisoni, F., and Schneider, G. (2020). Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584.

Lemhadri, I., Ruan, F., Abraham, L., and Tibshirani, R. (2021). Lassonet: A neural network with feature sparsity. *The Journal of Machine Learning Research*, 22(1):5633–5661.

Li, C. and Li, M. (2008). Gwasimulator: a rapid whole-genome simulation program. *Bioinformatics*, 24(1):140–142.

Li, J. and Huang, T. (2018). Predicting and analyzing early wake-up associated gene expressions by integrating gwas and eqtl studies. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1864(6):2241–2246.

Li, K. (2022). Variable selection for nonlinear cox regression model via deep learning. *arXiv preprint arXiv:2211.09287*.

Li, K., Wang, F., Liu, R., Yang, F., and Shang, Z. (2021). Calibrating multi-dimensional complex ode from noisy data via deep neural networks. *arXiv preprint arXiv:2106.03591*.

Li, K., Wang, F., Yang, L., and Liu, R. (2023a). Deep feature screening: Feature selection for ultra high-dimensional data via deep neural networks. *Neurocomputing*, 538:126186.

Li, K., Zhu, J., Ives, A. R., Radeloff, V. C., and Wang, F. (2023b). Semiparametric regression for spatial data via deep learning. *arXiv preprint arXiv:2301.03747*.

Li, Y., Chen, C.-Y., and Wasserman, W. W. (2016). Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336.

Liu, B., Wei, Y., Zhang, Y., and Yang, Q. (2017). Deep neural networks for high dimension, low sample size data. In *IJCAI*, pages 2287–2293.

Lu, Y., Fan, Y., Lv, J., and Stafford Noble, W. (2018). Deeppink: reproducible feature selection in deep neural networks. *Advances in neural information processing systems*, 31.

Mirzaei, A., Pourahmadi, V., Soltani, M., and Sheikhzadeh, H. (2020). Deep feature selection using a teacher-student network. *Neurocomputing*, 383:396–408.

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., and O'Sullivan, J. M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics*, 2:927312.

Scardapane, S., Comminiello, D., Hussain, A., and Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89.

Singh, D., Climente-González, H., Petrovich, M., Kawakami, E., and Yamada, M. (2023). Fsnet: Feature selection network on high-dimensional biological data. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Solorio-Fernández, S., Carrasco-Ochoa, J. A., and Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948.

Tadist, K., Najah, S., Nikolov, N. S., Mrabti, F., and Zahi, A. (2019). Feature selection methods and genomic big data: a systematic review. *Journal of Big Data*, 6(1):1–24.

Varshavsky, R., Gottlieb, A., Linial, M., and Horn, D. (2006). Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513.

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721.

Yang, S., Wen, J., Eckert, S. T., Wang, Y., Liu, D. J., Wu, R., Li, R., and Zhan, X. (2020). Prioritizing genetic variants in gwas with lasso using permutation-assisted tuning. *Bioinformatics*, 36(12):3811–3817.

Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Zhao, L., Hu, Q., and Wang, W. (2015). Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. *IEEE Transactions on Multimedia*, 17(11):1936–1948.