IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 6, 2018, accepted January 5, 2019, date of publication January 17, 2019, date of current version February 6, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2893269*

# Prediction of SNP Sequences via Gini Impurity Based Gradient Boosting Method

**LONGQUAN JIANG[1], BO ZHANG[1], QIN NI[1], XUAN SUN[2], AND PINGPING DONG[1]**

[1]College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China
[2]School of Information Science and Technology, Sanda University of Shanghai, Shanghai 201209, China

Corresponding authors: Bo Zhang (zhangbo@shnu.edu.cn) and Qin Ni (niqin@shnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572326, Grant 61802258, and Grant 61702333, in part by the Natural Science Foundation of Shanghai under Grant 18ZR1428300, and in part by the Shanghai Committee of Science and Technology under Grant 17070502800 and Grant 16JC1403000.

**ABSTRACT** Recent research has witnessed the fostered application of machine learning approaches in analyzing the single nucleotide polymorphisms (SNP) data, which has been proved to be implicated in complex human diseases. In the identification of SNPs responsible for complex diseases, most genome-wide association studies always took single SNP into consideration at one time and ignored diverse interactions between SNPs. One of the major problems is the higher number of features and the relatively small number of individuals, which complicates the task and harms the predictive ability of DNA sequences. In this paper, a novel boosting-based ensemble approach was proposed to study these interactions. An importance scoring strategy based on Gini impurity was introduced for feature selection. We evaluated its efficacy on the SNP genotyping data collected by the Southeastern University of China and compared it with naive Bayes, support vector machine, and random forest. The experimental results have shown its validity and effectiveness on SNP interaction identification. In addition, our approach had an obvious advantage of computational time and resources.

**INDEX TERMS** Single nucleotide polymorphism, data mining, machine learning, interaction detection and genome-wide association studies.

## I. INTRODUCTION

Recent studies have revealed most complex diseases, such as diabetes [1], myocardial infarction [2] and Crohn's disease [3], are mainly caused by the alteration of a single nucleotide (A, T, C and G) at a specific location in DNA molecules, or by genes that contains multiple nucleotide variants. Complex diseases are characterized by the complex genetic architecture and involve several genes and their interactive effects, which makes the task of connecting variants with phenotypic differences being one of the great challenges in genetic research. The single nucleotide polymorphism (SNP) [4], carrying the genetic information of DNA molecules, is the most basic unit of genetic variants. SNPs may pervade the DNA sequence, including regions of coding, non-coding or between genes. In general, the majority of SNPs have minimal impact on biological system. However, rare SNPs or their combinations may cause changes in protein function, further contribute to genetic disorders.

The SNP interaction is equivalent to statistical *epistasis*, which refers to a phenotype influenced by an allele at a locus masking the expression of an allele at another locus. The term *epistasis* [5] was first coined by Bateson in 1909, whose definitions were extended by biologist, statisticians, epidemiologists, and geneticists. Due to our limited knowledge of its various definitions used by different experts and scholars from different areas, however, it is recommended that the term *epistasis* should be constrained to its original sense defined by Bateson [5] and Fisher [6].

The detection and identification of SNPs and their interactions responsible for complex diseases is playing an increasingly vital role in interpreting the genetic epidemiology of a disease. Interactions between SNPs are increasingly being investigated in the context of disease susceptibility. The task, however, is fundamentally difficult since the quantity of variables/SNPs (p) is much larger than the quantity of samples (n), which is also described as the 'small $n$ big $p$' problem in statistics. Over 12 million SNPs are unevenly distributed across the human genome [8]. Computationally, it is almost infeasible to evaluate candidate combinations of SNPs and identify an optimal [4]. In general, this problem can be viewed as a feature/variable selection problem. A subset of all existing features is selected to search the possible combinations of SNPs. However, traditional methods are usually unable to incorporate so many variables in their analysis because of the overfitting problem as well as the lack of interpretability.

The diagnosis, prevention and treatment for complex diseases will highly benefit from better understanding the role of SNPs and their interactions [9]. Plenty of methods are proposed for SNP interaction identification. The simplest way is by exhaustive search using classic statistics such as exact likelihood ratio, the Pearson's $\chi^2$ test. One popular method is Multifactor Dimensionality Reduction (MDR) [10]–[12], in which all possible genotypic combinations were partitioned into n-dimensional subspaces as contingency table using the constructive induction approach. Tree based epistasis association mapping (TEAM) [13] is a model free method, where the minimum spanning tree is used for alleviating the heavy computation while updating the contingency tables without scanning all individuals. Contrary to TEAM, boolean operation based screening and testing (BOOST) [14] is a model based two-stage search approach which tries to discover all pairwise interactions in SNP data. Similar methods are EpiMiner [16] and CINOEDV [15]. Aforementioned methods could successfully detect SNP interactions, but it still suffers from intensive computation and poor scalability to a large number of SNPs.

Over the past decade, machine learning based methods such as Bayesian models, SVM, RF, have been widely used in the genomic prediction of phenotypes. Bayesian models [19], [20], which typically use maximum likelihood to estimate probabilities of phenotypes on genotyping SNP data, have aroused extensive attention due to their simplicity and efficiency. However, the Bayes methods and their modifications suffer from strong independence assumptions, which makes them difficult to capture complex SNP signals and may harm classification performance [22]. Another dominant algorithm for classification problems is the so-called SVM. In spite of the advantage of accounting multiple factors, it lacks of interpretability for resulting classifiers when applied in biological context, especially for which contain the massive SNPs [23]. In order to achieve better interpretation, the strategies that entail some sort of feature selection is highly required in issues with large amounts of features or variables, such as genomic prediction and evaluation. Ensemble methods like boosting might be a good alternative. The main idea is to form a 'committee' with greater potential of prediction than that of any of individual classifiers by combining several weak classifiers. Studies using the original AdaBoost algorithm and its various modifications have proved the ability in classification problems [24]–[26]. One of the most interesting versions is gradient boosting algorithm, which has been proved to have advantages of great robustness to outliers, missing data and numerous relevant or irrelevant variables.

The objectives of this study are: 1) to propose a novel ensemble approach by extending the basic Gradient Boosting algorithm with introducing a new scoring rule-based Gini impurity for feature selection. and 2) to apply this novel ensemble approach in genome-assisted genetic evaluations and show the superiority in classification performance compared with the commonly used methods such as Naive Bayes, SVM and Random Forest. Thus, a new thought in this study was provided to analyze SNP genotyping data and predict its phenotype with respect to disease susceptibility.

The rest of this article is organized as follows: Section II describes the related work about machine learning methods, especially boosting algorithm applied in the bioinformatics; Section III states the problem definition from the mathematical perspective, and illustrates the framework as a solution with the demonstration of the main idea of the proposed SNP-GB algorithm; Section IV provides a theoretical and experimental analysis to show the capability and efficiency of the proposed framework in SNP phenotype prediction and interaction analysis. Moreover, the data processing procedure, the experimental results and performance evaluation of the proposed framework are reported. Section V presents the conclusion and future work.

## II. RELATED WORK

Ensemble leaning [17], [18], which consists of a collection of single classifiers whose predictions are then combined to produce a final decision, has been proved to yield better performance than that of the individual classifiers like Naive Bayes, SVM, etc. Boosting algorithm is one of ensemble systems that are widely used in classification, regression or other tasks. Recent studies have demonstrated the widely applications of boosting algorithm and its modifications in the field of bioinformatics, such as genomic selection, interaction analysis and genetic disease diagnosis.

### A. WORK ON FAUNA AND FLORA

The boosting algorithm and its modifications were introduced to generate stronger learner by combining simple classifiers. In [25], L2-boosting, a new version with L2 loss function in a recursive fashion, was used to predict the productive lifetime of 4702 Holstein sires and the progeny averages of food conversion rate of 394 broilers. In order to improve its efficiency and reduce computational time on large scale datasets, L2-boosting based random boosting [26] was proposed, in which p SNPs were randomly selected in each iteration for computing loss values. The results on a real data set containing 39714 SNPs from 1797 bulls displayed the outstanding potential in the analysis of effective SNPs in fauna and flora.

A lot of work have been done to compare machine learning methods in breeding research. In the prediction and estimation of genomic breeding values, Gradient Boost Machine (GBM) and Extreme Gradient Boost Method (EGBM) were explored to identify a subset of SNPs, and then used the subset to construct relationship matrices [27]. Three machine learning approaches were examined on a real data set containing 38083 SNPs from

2093 Branhman cattle. The results demonstrated an effective role GBM played in potential candidate genes identification as well as the performance superiority at the expense of computational time. Similarly, Genomic Best Linear Unbiased Prediction (GBLUP) method was proposed in [28], where heritability, number of quantitative trait loci (QTL) and distribution of QTL effects were its vital parameters. Their different combinations were tried when compared with RF, Boosting and SVM. The experimental procedures were conducted on a genome of five chromosomes on which 10000 biallelic SNPs were distributed. The results showed a serious limitation for this algorithm—heavy computational consumption.

### B. WORK ON GENETIC DISEASES

Human complex diseases, such as diabetes and Crohn's, are caused by a number of genetic factors. The boosting algorithm has been applied in the pathogenesis exploration. Detecting and identifying the gene interactions responsible for complex diseases was one of the main aims of genome wide association studies (GWAS). Gene interaction is also known as epistasis in a broad sense. Decision tree has the ability to capture interactions due to its tree structure, but it still suffered from some unavoidable limitations including data fragmentation and representation problem, which may harm the detection performance. In [29], gradient tree boosting method followed by an adaptive iterative SNP search was proposed to identify groups of interacting SNPs that contribute the most to the breast cancer risk, which can capture complex non-linear SNP interactions. In [30], the capability of feature selection for AdaBoost was carefully examined in the context of epistasis detection. A novel strategy of ranking candidate SNPs using importance score to study gene interactions between Alzheimer's and Parkinson. Experimental results have showed the higher sensitivity of AdaBoost on parameter settings of weak learners. In [31], permutation-based Gradient Boosting Machine (pGBM) was proposed, which detected pure epistasis and uncovered more complex disease pathogenesis by estimating the power of a GBM classifier that was influenced by permuting SNP pairs. The experimental results demonstrated that this method had high success rate in balanced and unbalanced simulation and real data. In [32], multivariate component-wise boosting method, capable of modeling non-linear associations, was explored under the setting of high dimension and low sample size. It greatly differed from other two excellent methods—recursive partition [33] and low rank feature learning [20], but it still got similar performance. Among those selected genes, five unknown genes (GFAP, GRB7, ALOX12, MFAP4, and HOXB2) relating to breast cancer were found. Various machine learning approaches have been successfully used to predict individual risk to polygenic diseases. They greatly differed from a large degree on performance or suitability. In [34], it paid more attention to comparative studies between several predictive models in the task of risk prediction, and on the basis of well-designed experiments suggested that the boosting algorithm may be more suitable for modeling individual predisposition to Type 1 diabetes and rheumatoid arthritis and should be considered for more in-depth research. In [38], random forests and gradient boosting machine were used to explore SNPs responsible for Colorectal cancer (CRC). Because of the genetic complexity of cancer [39], the non-additive interaction effects among multiple genetic variants have gained a superiority in the explanation of the missing heritability in GWAS.

In [34]–[37], it has been experimentally proved the superiority of the boosting algorithm over other machine learning algorithms, such as logistic regression, random forest and support vector machine, especially on the task of epistatic effect detection. In this article, SNP-GB, a predictive model based on gradient boosting algorithm, was proposed for SNP sequences and interaction analysis.

## III. METHOD

### A. PROBLEM DESCRIPTION

In general, the pathogenesis is quite complicated for complex diseases that entail multiple genes, and it is often caused not only by a single factor but also by a variety of genetic factors such as SNPs. SNP interactions are quite common in which the effect of a particular genetic variant was influenced by a variant at another locus. Therefore, variant of phenotype with genotype at one locus was only apparent amongst those with certain genotypes at the second locus. A growing body of studies have shown that many phenotypic traits in the human body as well as the susceptibility to drugs and diseases may be closely associated with certain loci or genes that contain multiple loci [3]. The problem can be formulated as follows:

Given a set of $N$ individuals $X = \{X_1, X_2, \cdots, X_N\}$ with the corresponding binary phenotypes $Y = (y_1, y_2, \cdots, y_N)$, $y_i \in \{-1, +1\}$, and each individual $X_i$ with $D$ markers. In general, the genotypes of each individual $X_i$ are three-valued categorical data (AA, AB, BB), denoted as a sequence of $\{x_{i1}, x_{i2}, \cdots, x_{iD}\}$. Supposed that $R = (r_1, r_2, \cdots, r_D)$ corresponds to $n$ SNPs, its goal is to select an optimal subset $R_s = (r_{s1}, r_{s2}, \cdots, r_{sd})$ $(d < D)$ satisfying the following conditions:

$$\max p_s = f(R_s) \tag{1}$$

The framework of the proposed predictive model is depicted in Fig (1). **T** represents the data set, $w_i$ is the weight matrix, $T_i$ represents the subset re-sampled after rearranging $w$ in the m-th iteration, $\beta_i$ and $\gamma_i$ is the model parameters to be iteratively optimized, $g_i$ is the gradient of loss function, $h_m$ is the weak learner trained in the m-th iteration. $f(x)$ is the final classifier.

#### 1) PRE-PROCESSING

Due to some problems such as data deficiency, data duplication and feature representation, the raw data is difficult to be used as input to machine learning algorithms. Therefore, it is extremely necessary to perform data pre-processing

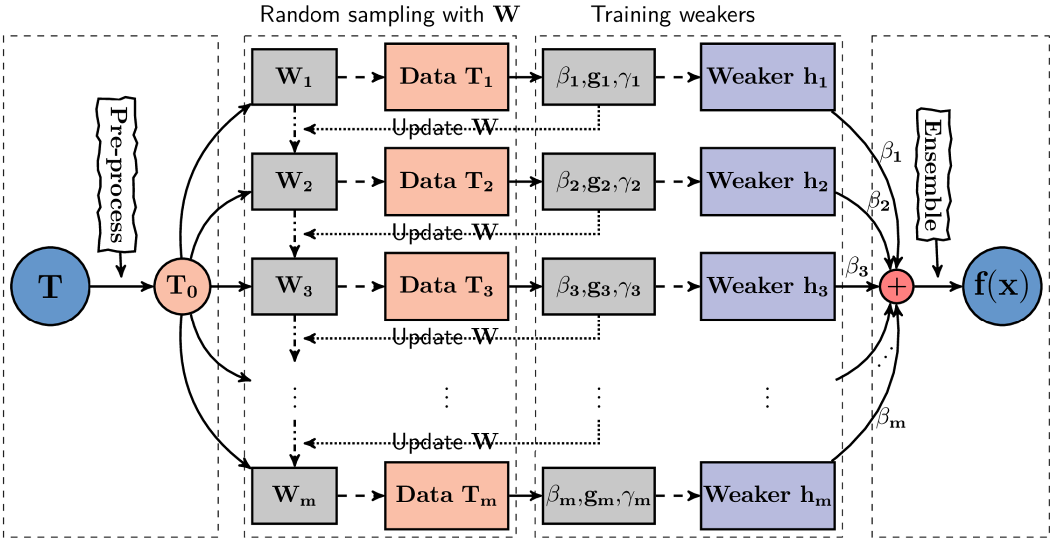

**FIGURE 1.** **The framework of our proposed predictive model.**

procedure prior to data analysis. The data pre-processing procedure is detailed in Section IV.

### 2) RANDOM SAMPLING

Individuals are randomly sampled from the data set according to their weight matrix $W$. Initially, each individual is weighted uniformly, then the weighted data set is trained by a classifier at each step. Afterward, the weights of individuals are updated based on the performance—decreasing the weights of incorrectly classified individuals and increasing the weights of correctly classified individuals. In this way, the subsequent classifiers focus on the individuals that are more difficult to be classified.

### 3) MODEL TRAINING

Given a loss function $L$ that is used to measure the deviation between the real value $y$ and the predicted value $\hat{y}$, the negative gradient $y_m(x)$ is computed to optimize model parameters $\alpha_m$ and $\rho_m$ and further to learn the weak classifier $h_m$ in the m-th iteration according to the probability distribution. As a result, the weight matrix $W$ is rearranged according to whether the individuals are correctly classified or not.

### 4) MODEL ENSEMBLE

In this stage, the trained weak classifiers are combined to form a strong classifier $f(x)$ according to the parameter $\rho_m$, and the final unified decision is obtained as a weighted combination of the predictions of the weak classifiers of the ensemble.

## *B. ALGORITHM*

### 1) OUTLINE OF THE APPROACH

Ensemble methods, a special class of machine learning methods, function on the criterion of the crowding wisdom. Instead of creating only one single strong learner, ensemble methods attempt to build many heterogeneous weak learners. Actually, a weak learner is defined as a simple model with slightly better performance than random guessing. A thorough picture of the whole data set is eventually formulated by combining the decisions made by these weak learners. Here, the basic Gradient boosting algorithm was slightly modified by introducing Gini impurity as a strategy to choose candidate SNPs. Decision tree is chosen as the weak leaner in the light of the characteristics of genotyping SNP data. In decision tree, every subsequent split is conditioned on previous splits, which enables to naturally capture interactions. In the first part of this subsection, we explained the principle of decision tree algorithm and then the formulation of Gini impurity measurement used for feature selection. We implemented efficient experimental designs using Python programming language and paid special attention to the memory space consumption in the process of analyzing SNP genotype data. In the last part, we outlined the main idea of our SNP-GB method.

### 2) DECISION TREE

As one of supervised learning algorithms, decision tree is commonly used in data mining. The goal of decision tree is to build a model that can predict the value of a target variable based on several input variables. Decision tree can be learned in a top-down manner, where each interior node corresponds to one of the input variables, each edge to children denotes one possible value of the input variable and each leaf represents the value of the target variable that is represented by the path from the root to the leaf. Here, we described the implementation details of this algorithm performing on genotyping SNP data. At every node, decision tree chose a SNP (feature) and then splitted individuals at this node into subgroups in accordance with their genotypic values. This splitting process is recursively repeated on each desired subset. The recursion stops when the subset at a node has all the same value as the target, or when splitting no longer contributes to the predictions. Given the subset, its label distribution and an element that was randomly chosen from it, the gain of Gini impurity (GI) measures how often the element would be correctly labeled if it is randomly labeled. Intuitively, the lower impurity score that child nodes can obtain from a split, the purer classification that each node would achieve. Formally, GI is formulated in Equation (2), where $p$ is the tree node, $p_i$ is the fraction of items labeled with class $i$ in the set, $J$ is the total number of classes (there are two classes in the experiment:case and control).

$$\begin{aligned} GI(p) &= \sum_{i=1}^{J} p_i(1-p_i) \\ &= 1 - \sum_{i=1}^{J} p_i^2 \end{aligned} \tag{2}$$

And the gain of GI is computed according to Equation (3), where, $N$ and $GI$ represents the quantity of individuals at a node and its GI, respectively. $p$ is the parent node and $d$ is its child.

$$Gain(p) = GI(p) - \sum_{d \in p} \frac{N_d}{N_p} \times GI(d) \tag{3}$$

### 3) GRADIENT BOOSTING

Gradient boosting is an ensemble learning method, which iteratively adds basic models in a greedy fashion such that each additional basic model further reduces the gradient of the selected loss (error) function. This algorithm is detailed as follows:

$x$ denotes the feature vector and $y$ denotes the corresponding class label. Given some training samples $\{x_i, y_i\}_{i=1}^{N}$, the goal is to find a function $F^*(x)$ that can map $x$ to $y$, such that the expected value of a specific loss function $L(y, F(x))$ is minimized over the joint distribution of $\{x, y\}$ values.

The loss function is used to measure the deviation between the real value $y$ and the predicted value $\hat{y}$.

$$\begin{aligned} F^*(x) &= \arg\min_{F(x)} E_{y,x} L(y, F(x)) \\ &= \arg\min_{F(x)} E_x[E_y L(y, F(x))|x] \end{aligned} \tag{4}$$

The "additive" expansion is expressed in Equation (5) in order to approximate the function:

$$F(x; P) = \sum_{m=0}^{M} \beta_m h(x; \gamma_m) \tag{5}$$

where $P = \{\beta_m, \gamma_m\}_{m=0}^{M}$. The function $h(x; \gamma)$, called 'base learner', is usually simple function of $x$ with parameters $\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_M\}$. The task will become more difficult if $F(x)$ is non-parametrically estimated. Thus, it is needed to transform the function optimization problem to the parameter optimization problem by choosing a model $F(x; P)$ that can be parameterized using $P$. A typical parameter optimization method is a "greedy-stagewise" approach. $\{\beta_m, \gamma_m\}$ is optimized after all the $\{\beta_i, \gamma_i\}$ $(i = 0, 1, \cdots, m-1)$ are optimized. This process can be formulated as follows:

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \beta h(x_i; \gamma)) \tag{6}$$

$$F_m = F_{m-1}(x) + \beta_m h(x; \gamma_m) \tag{7}$$

The steepest-descent method proposed by Friedman [40] that aims to handle the optimization problem is detailed in Equation (5). Based on the steepest-descent method, we proposed the SNP-GB methd for prediction and interaction analysis on genotyping SNP data, which is detailed in Algorithm (1).

```
Algorithm 1 SNP-Gradient boosting
Input:
  The set of all SNPs, $x$;
  The iterative steps, $M$;
Output:
  The final classification function $F_m(x)$;
  initialize $F_0(x) = \arg\min_\rho \sum_{i=1}^{N} L(y_i, \rho)$;
  for $m = 1$ to $M$ do
    Compute the negative gradient
      $\tilde{y}_m = -\frac{\partial L(y_i, F(x_i))}{\partial F_{x_i}}$
    Fit a model
      $\alpha_m = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} [\tilde{y} - \beta h(x_i; \alpha_m)]^2$
    Choose a gradient descent step size as
      $\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i; \alpha))$
    Update the estimation of $F(x)$
      $F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$
  end for
  return $F(x)$;
```

## C. OTHER ALGORITHMS

Random forest, a typical ensemble learning method, was proposed by Breiman [41] in 2001, and has been widely adopted in classification, regression and other tasks. In random forest, a collection of homogeneous decision trees are employed at the same time to obtain a better understanding of the data. Each tree in the ensemble is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. Additionally, RF further introduces a strategy of random attribute selection in the process of training decision tree models. When splitting a node during constructing the

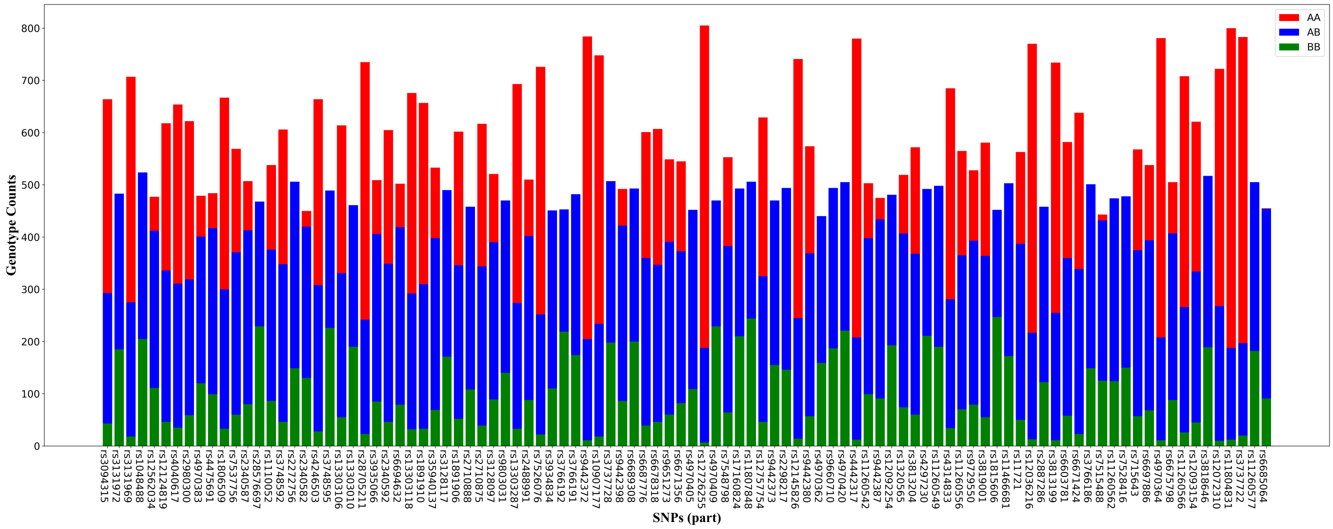

**FIGURE 2.** The distribution of genotypes in each SNP.

tree, traditional decision trees choose an optimal split from a set of attributes at current node (totally $d$ attributes), while random forest firstly picks a subset of $k$ features among all features and then selects an optimal split among a random subset of features. The model parameter $k$ controls the degree of randomness. If $k = d$, the construction of base decision trees would be the same as that of traditional ones, and if $k = 1$ only one attribute would be randomly chosen as a split. Generally, $k = \log_{2d}$ is recommended [41]. RF slightly extended the bagging algorithm by introducing the attribute disturbance derived from the random attribute selection technique, which led to richer diversity of base learners than that in bagging, and further improved the generalization by increasing the differences between these base learners in the final ensembled model. In particular, trees tend to learn highly irrelevant patterns if they grow very deep. That is, they overfit the training sets with low bias but very high variance. In machine learning, overfitting is not avoidable. As a result of this randomness (mainly attribute and sample disturbance), the bias slightly increase, but due to averaging, the variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.

### D. MODEL EVALUATION

The performance of our predictive model is evaluated by:

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{10}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TN + FN)(TP + FN)(TN + FP)}} \tag{11}$$

where $TP$ is the number of positive individuals that are correctly classified, $FP$ is the number of negative individuals that are wrongly classified; $TN$ is the number of negative individuals that are correctly classified, $FN$ is the number of positive individuals that are wrongly classified. Among these four metrics, $MCC$ (Matthews Correlation Coefficient) is the correlation coefficient between the observed and predicted individuals as a measure of the quality of binary classification.

## IV. EXPERIMENTS AND RESULTS

### A. EXPERIMENTAL SETTING

#### 1) DATA AND PRE-PROCESSING

In order to evaluate the performance of *SNP-GB* algorithm, we applied it to the data set containing 1000 individuals (500 cases and 500 controls) with 9000 SNPs each. The data set was collected from patients who suffered from a certain genetic disease by the Southeastern University of China. The statistics of the data set is shown in Fig (2), where the red bar represents the number of AA genotype, the green bar represents the number of BB genotype, the blue bar represents the number of AB genotype, AA and BB are the homozygotic genotypes at some location while AB is heterozygotic. For model training and testing, the data set was randomly splitted into a training set and a testing set with 650 samples and 350 samples respectively. However, in order to validate the ability of our model to learn general patterns from the data set, a subset of the training set was chosen as the validation set.

Prior to data analysis, preprocessing procedures are greatly required, which contain data cleansing, encoding, completion of missing values, removal of duplicate values, etc. There have been many methods for genotype representation. In [31], four-dimensional vector representation aimed to express different bases (**A**, **C**, **T**, **G**) with a four dimensional one-hot vector. However, it made these bases

**TABLE 1. Numerical representation of example SNPs.**

| Phenotype | Name, genotype and its representation | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | rs100015 | | rs56341 | | rs21132 | | rs7526311 | | rs665691 | | rs294223 | | rs3131972 | | rs9729550 | |
| 1 | TT | 1 | CA | 2 | GT | 2 | AT | 1 | GC | 1 | GA | 1 | CT | 2 | AA | 1 |
| 0 | TT | 1 | CC | 1 | GG | 0 | AA | 0 | GC | 1 | AA | 0 | CT | 2 | AA | 1 |
| 1 | TC | 2 | CC | 1 | GG | 0 | AA | 0 | GG | 2 | GA | 1 | TT | 0 | AC | 2 |
| 1 | TC | 2 | CA | 2 | GG | 0 | TT | 2 | GG | 2 | GA | 1 | CC | 1 | AA | 1 |
| 0 | CC | 0 | CC | 1 | GG | 0 | AT | 1 | CC | 0 | GA | 1 | CT | 2 | AA | 1 |
| 0 | TT | 1 | CC | 1 | GG | 0 | TT | 2 | CC | 0 | GG | 2 | CC | 1 | CC | 0 |

mutually independent and irrelevant. Additionally, another approach was two-dimensional vector representation [21] (i.e. **A** = 00, **T** = 01, **C** = 10, **G** = 11), which easily led to linear relevance and blurred differences in spite of computational reduction. Generally, each of given SNPs have three discrete states (**AA**, **AB**, **BB**) indicating the possible homozygotic and heterozygotic genotypes for this location. As a result of this trait, we just need a three-dimensional one-hot vector to represent each genotype. For example, these three discrete states **AA**, **AB**, **BB** are encoded into 0, 1, 2 respectively (as shown in Table 1). Compared with those methods described above, this method has advantages of simplicity, relatively less computational time and the ability to ignore specific types of each base, which makes it pay more attention on SNP compositionality and further extract more diverse information in order to improve flexibility.

### 2) IMPLEMENTATION

In this study, the proposed approach was implemented in Python, a popular high level programming language with fast prototyping and easy transferability. In addition, numerous useful libraries are accessible in the repository. *Numpy*, an effective scientific library, enables Python to access quick multidimensional array that can be used to store the SNP data efficiently. In this implementation, we realized our predictive model based on *XGBoost* [42], an optimized distributed gradient boosting library designed for high efficiency, flexibility and portability. It supports CUDA accelerated tree construction algorithm, which is vital for model convergence acceleration, computational time reduction and efficiency improvement with the help of the computational power of GPUs on mathematical calculations. It is enabled by default in XGBoost. Source code is deployed on a server machine equipped with two Nvidia GeForce 1080ti GPUs and memory space of total 140 GB.

### 3) PARAMETER FINE TUNING

Effective modeling is usually the first step to perform an exhaustive analytic procedure, but fine tuning parameters appears to be more important to ensure desirable optimal results. In addition to basic parameters including sample size and random state, there are some other parameters to be optimized for Gradient boosting algorithm, such as *n_estimators*, *eta* and *early_stopping_rounds*. *n_estimators* is the number of boosted trees to fit and *eta* is the step size used in update. Intuitively, the larger the values for both parameters are, the better the performance will become. Firstly, for *n_estimators*, its values range from 50 to 400, and for *eta*, its values range from 0.0001 to 0.3. The performance under different values of *n_estimators* and *eta* seemed to vary very much. For example, with *n_estimators* = 400 and *eta* = 0.3, the predictive accuracy decreased by 20%. However, with *n_estimators* = 100 and *eta* = 0.01, the number rose by 36%. Based on the performance and computational resources, thus, the parameters *n_estimators* and *eta* were set to 100 and 0.01, respectively. When constructing each tree and splitting, the subsample ratio of columns *colsample_bytree* and the subsample ratio of column for each split in each level *colsample_bylevel* play an important role. After several careful attempts, we chose *colsample_bytree* = 0.8 and *colsample_bylevel* = 1.

## B. SNP INTERACTIONS ANALYSIS

Tree, a type of structure that is capable of displaying complex inner relationships between features in a visual way, plays an important part in exploring interactions or higher-level effect in genetics. Here we tried to construct a tree for analyzing SNP interactions. To perform the classification of features (SNPs) and produce the possible response value, SNP-GB algorithm constructs a type of hierarchical-structure tree (as shown in Fig (3)), in which every node represents a variable and every edge represents different attributes of the parent node. According to these attributes, every node is splitted into child nodes. Different paths indicate the diverse combinations of predictors (features) that imply their interactions. The variable at the top of structured tree represents the strongest splitting variable and the subsequent nodes are built on the upper node. It is clearly seen from Fig (3) that the locus *rs1006147* is the strongest splitting feature among features from a subset of the entire data set, all of the latter child nodes are built on it. Furthermore, more complex interactions can be seen from Fig (3). For example, loci that interact with *rs1006147* via *rs2253372* are *rs12139487* and *rs6701316*. The number on the leaf nodes is the gradient of the selected loss function calculated based on previous base learners. The goal of gradient boosting algorithm is to further reduce the gradient value in an iteratively additive way. Fig (4) shows the gain of Gini impurity of a randomly selected element being incorrectly labeled. As depicted in previous section, the lower impurity score that child nodes can obtain from a

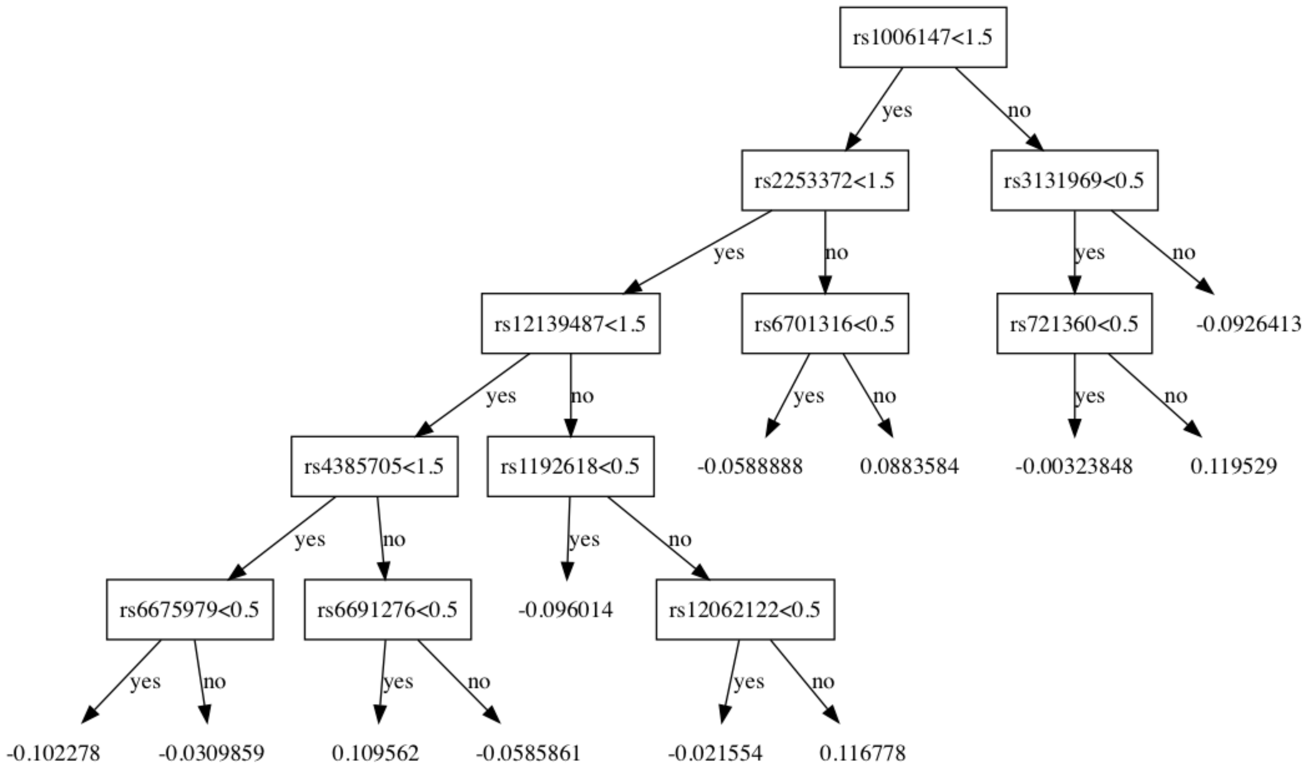

**FIGURE 3.** Prediction tree produced by SNP-GB algorithm.

split, the purer classification that each node would achieve. This means that the locus *rs1006147* have the lowest gain of Gini impurity so that it enables the best classification performance among the data set in m-th iteration.

### C. PERFORMANCE

This section aims to compare baseline methods (Naive Bayes, SVM, Random Forest) with SNP-GB on the predictive performance of SNP sequences. Fig (5) shows ROC curves by 10-fold cross validation. The predictive model based on gradient boosting contributes the best, with the AUC up to 0.9519, followed by Random Forest with 0.9028 AUC scores. The worst is Naive Bayes and SVM is slightly better than it.

**TABLE 2.** Performance of the predictive models based on different classification algorithms.

| Algorithms | AUC | $F_1$ | MCC | Recall | Precision |
|---|---|---|---|---|---|
| Naïve Bayes | 0.8476 | 0.6222 | 0.5442 | 0.6241 | 0.6243 |
| SVM | 0.8815 | 0.8182 | 0.6598 | 0.7761 | 0.8662 |
| Random forest | 0.9028 | 0.8179 | 0.7821 | 0.8218 | 0.9018 |
| SNP-GB | 0.9519 | 0.8979 | 0.8046 | 0.8693 | 0.9292 |

Table 2 shows the classification performance on the independent test data set. Among them, all of performance metrics of SNP-GB and Random Forest is higher than that of Naive Bayes and SVM. It is easily concluded that ensemble methods generally perform better than non-ensemble methods like SVM. Overall, the predictive model based on gradient boosting achieves the best performance.

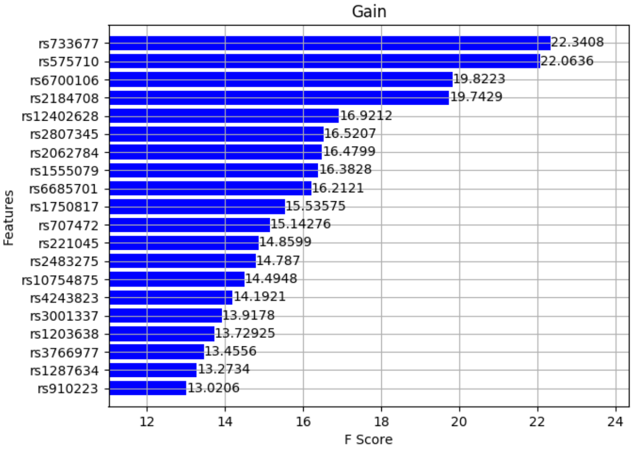

**FIGURE 4.** Feature importance.

### D. COMPUTATIONAL TIME AND RESOURCES

We deployed implementations on a cluster of four PowerEdge R730 servers at the Big Data Laboratory of Shanghai Normal University. We equipped each machine with Intel Xeon multi-core CPU and memory space of total 140 GB, and one with two Nvidia GeForce 1080ti GPU. The two most

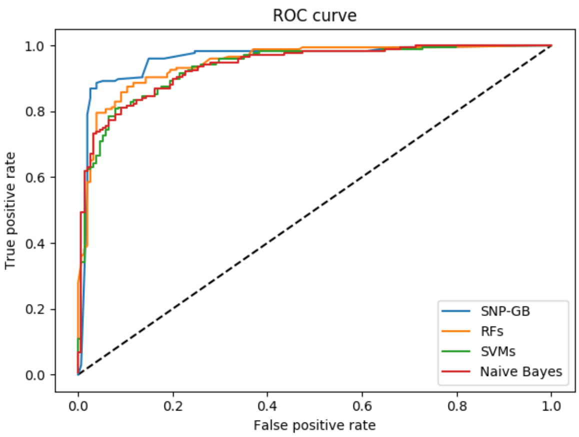

**FIGURE 5.** ROC curves for different models by 10-fold cross validation.

**TABLE 3.** Statistics of time and resources when training models.

| | NB | SVM | RF | SNP-GB |
|---|---|---|---|---|
| Time (min) | 1.4 | 1.8 | 2.4 | 2.3 |
| Memory (MB) | 20.2 | 21.8 | 38.6 | 30.3 |

**TABLE 4.** Statistics of time and resources when cross validating models.

| | NB | SVM | RF | SNP-GB |
|---|---|---|---|---|
| Time (min) | 3.2 | 3.1 | 5.4 | 4.1 |
| Memory (MB) | 21.1 | 21.6 | 40.1 | 38.2 |

important features of *XGBoost* library that are provided to improve performance is the GPU acceleration strategy and multithreading support, which allow us to efficiently use all of computational resources in our system during the training stage to speed up model convergence. In our implementation, the parameter *predictor* indicating which method is used to predict, CPU or GPU predictor, is set to *gpu_predictor* so as to enable faster construction of individual trees. Meanwhile, statistics of different models when training and cross validating is shown in Table 3 and Table 4. It is clearly seen that the proposed predictive model contributes the best performance, especially in comparison to RF, with around 2.3min running time (versus 2.4min by RF) and 30MB memory space (versus 38MB by RF) during training, and with approximately 4.1min (versus 5.4min by RF) and 38.2MB (versus 40.1MB by RF) during cross validating. Those models, such as Naive Bayes, SVM, ran for much less time and used much less memory space but they generally had worse performance than ensemble systems. The reason may be that it would take a long time to construct individual trees for tree structured models in addition to mathematical calculations.

## V. CONCLUSION AND FUTURE WORK

In this work, we presented a novel algorithm, SNP-GB, for classifying SNPs as well as for identifying potential SNP interactions. SNP-GB is a version of the famous gradient boosting algorithm, which was slightly modified by introducing a new rule based on Gini impurity to feature scoring and selection. Using decision tree as weak classifier, we created a strong assembled classifier. We evaluated SNP-GB on the genotyping data collected by the Southeastern University of China. Comparative results with Naive Bayes, SVM and Random Forest showed that ensemble methods obviously outperformed simple single models, since ensemble methods operate on the principle of crowding wisdom. Random Forest didn't outperform the proposed SNP-GB algorithm. The reason may be that in SNP-GB, classification errors of the first single classifier are compensated as good as possibly by the second, etc., thus selection of the next classifier is biased in favor of previously misclassified data points. This trait of SNP-GB algorithm makes it more suitable for tasks of this kind than Random Forests. When allele frequencies are low, however, the Gini impurity may suffer some disadvantage because of inadequate samples of high effect nodes. Overall, SNP-GB is a very fast and memory efficient algorithm and can be used as a promising tool for feature selection, interaction analysis and classification on SNP data.

In association studies, Linkage Disequilibrium (LD) is one important factor when investigating interactive effects. In this work, however, we chose not to consider it for the following reason: In general, high LD usually exists in those SNPs around each disease locus, which may negatively influence the interactive effect and further cause the noise signal enhancement. Moreover, the situation where any SNPs will have high LD with the regulatory SNPs is impossible. A marker could be typed, but the potential rSNPs might not be. Noises in the model may be increased and the predictive power may be reduced. We will test it in our future work.

Deep neural networks, like CNNs, RNNs, etc., have achieved great success in the field of object detection, machine translation and speech recognition. Also, many researchers have applied deep learning techniques to successfully solve specific genetic problems. However, both a larger data set and greater computational power are usually required for deep learning based models. Thus, our approach still has great applicability due to its competitive advantage of resource efficiency. In the future, on one hand, the efficacy of deep learning techniques will be tested; on the other hand, their strengths and weaknesses will be compared and summarized on larger data set.

## REFERENCES

[1] W. Xu, J. Zhang, Q. Zhang, and X. Wei, "Risk prediction of type II diabetes based on random forest model," in *Proc. 3rd Int. Conf. Adv. Elect., Electron., Inf., IEEE Commun. Bio-Inform.*, Feb. 2017, pp. 382–386.

[2] J. Allen, E. Zacur, E. Dall'Armellina, P. Lamata, and V. Grau, "Myocardial infarction detection from left ventricular shapes using a random forest," in *Statistical Atlases and Computational Models of the Heart: Imaging and Modelling Challenges*. New York, NY, USA: Springer-Verlag, 2015, pp. 180–189.

[3] W Mao and S. Kelly, "An optimum random forest model for prediction of genetic susceptibility to complex diseases," in *Proc. Pacific–Asia Conf. Adv. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2007, pp. 193–204.

[4] W. S. Bush and H. J. Moore, "Genome-wide association studies," *PLoS Comput. Biol.*, vol. 8, no. 12, p. e1002822, 2012.

[5] W. Bateson, ''Mendel's principles of heredity,'' *Amer. J. Med. Sci.*, vol. 148, no. 5, p. 747, 1914.

[6] R. A. Fisher, ''XV.—The correlation between relatives on the supposition of mendelian inheritance,'' *Trans. Roy. Soc. Edinburgh*, vol. 52, no. 2, pp. 399–433, 1919.

[7] S. Wernicke, *On the Algorithmic Tractability of Single Nucleotide Polymorphism (SNP) Analysis and Related Problems*. Tübingen, Germany: Diplomarbeit, WSI für Informatik, Univ. Tübingen, 2003.

[8] R. De, W. S. Bush, and J. H. Moore, *Bioinformatics Challenges in Genome-Wide Association Studies (GWAS). Clinical Bioinformatics*. New York, NY, USA: Springer, 2014.

[9] Human Genome Project. U.S. Dept. Energy Genome Program's Environmental Research Information System (BERIS). Jul. 2010. [Online]. Available: http://www.ornl.gov/sci/techresources/Human_Genome/

[10] M. D. Ritchie *et al.*, ''Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer,'' *Amer. J. Hum. Genet.*, vol. 69, no. 1, pp. 138–147, 2001.

[11] D. R. Velez *et al.*, ''A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction,'' *Genetic Epidemiol.*, vol. 31, no. 4, pp. 306–315, 2010.

[12] M. D. Ritchie and A. A. Motsinger, ''Multifactor dimensionality reduction for detecting gene-gene and gene-environment interactions in pharmacogenomics studies,'' *Pharmacogenomics*, vol. 6, no. 8, pp. 823–834, 2006.

[13] X. Zhang, S. Huang, F. Zou, and W. Wang, ''TEAM: Efficient two-locus epistasis tests in human genome-wide association study,'' *Bioinformatics*, vol. 26, no. 12, pp. i217–i227, 2010.

[14] X. Wan *et al.*, ''BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies,'' *Amer. J. Hum. Genet.*, vol. 87, no. 3, pp. 325–340, 2010.

[15] J. Shang, Y. Sun, J.-X. Liu, J. Xia, J. Zhang, and C.-H. Zheng, ''CINOEDV: A co-information based method for detecting and visualizing n-order epistatic interactions,'' *BMC Bioinformat.*, vol. 17, no. 1, p. 214, 2016.

[16] J. Shang, J. Zhang, Y. Sun, and Y. Zhang, ''EpiMiner: A three-stage co-information based method for detecting and visualizing epistatic interactions,'' *Digit. Signal Process.*, vol. 24, no. 1, pp. 1–13, 2014.

[17] W. Luo *et al.*, ''Guidelines for developing and reporting machine learning predictive models in biomedical research: A multidisciplinary view,'' *J. Med. Internet Res.*, vol. 18, no. 12, p. e323, 2016.

[18] D. M. D. Farid, A. Nowe, and B. Manderick, ''An ensemble clustering for mining high-dimensional biological big data,'' *Int. J. Des., Nature Ecodyn.*, vol. 11, no. 3, pp. 328–337, 2016.

[19] Z. Ma and A. E. Teschendorff, ''A variational Bayes beta mixture model for feature selection in DNA methylation studies,'' *J. Bioinform. Comput. Biol.*, vol. 11, no. 4, p. 1350005, 2013.

[20] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, ''Variational Bayesian matrix factorization for bounded support data,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, Apr. 2015.

[21] W. Chen, Z. Chen, Z. Chen, Z. Wang, and H. Qiu, ''Digital coding of the genetic codons and DNA sequences in high dimension space,'' *Acta Biophys. Sinica*, vol. 16, no. 4, pp. 760–768, 2000.

[22] D. Gianola, A. Gustavo, W. G. Hill, E. Manfredi, and R. L. Fernando, ''Additive genetic variability and the Bayesian alphabet,'' *Genetics*, vol. 183, no. 1, pp. 347–363, 2009.

[23] J. O. Ogutu, H.-P. Piepho, and T. Schulz-Streeck, ''A comparison of random forests, boosting and support vector machines for genomic selection,'' *BMC Proc.*, vol. 5, no. 3, pp. S3–S11, 2011.

[24] J. Li, B. Horstman, and Y. Chen, ''Detecting epistatic effects in association studies at a genomic level based on an ensemble approach,'' *Bioinformatics*, vol. 27, no. 13, pp. i222–i229, 2011.

[25] O. González-Recio, K. A. Weigel, D. Gianola, H. Naya, and G. J. M. Rosa, ''$L_2$-Boosting algorithm applied to high-dimensional problems in genomic selection,'' *Genet. Res.*, vol. 92, no. 3, pp. 227–237, 2010.

[26] O. González-Recio, J. A. Jiménez-Montero, and R. Alenda, ''The gradient boosting algorithm and random boosting for genome-assisted evaluation in large data sets,'' *J. Dairy Sci.*, vol. 96, no. 1, pp. 614–624, 2013.

[27] B. Li, N. Zhang, Y.-G. Wang, A. W. George, A. Reverter, and Y. Li, ''Genomic prediction of breeding values using a subset of SNPs identified by three machine learning methods,'' *Frontiers Genet.*, vol. 9, p. 237, Jul. 2018.

[28] F. Ghafouri-Kesbi, G. Rahimi-Mianji, M. Honarvar, and A. Nejati-Javaremi, ''Predictive ability of random forests, boosting, support vector machines and genomic best linear unbiased prediction in different scenarios of genomic evaluation,'' *Animal Prod. Sci.*, vol. 57, no. 2, pp. 229–236, 2017.

[29] H. Behravan *et al.*, ''Machine learning identifies interacting genetic variants contributing to breast cancer risk: A case study in Finnish cases and controls,'' *Sci. Rep.*, vol. 8, no. 1, p. 13149, 2018.

[30] L. A. Assareh, G. Volkert, and J. Li, ''Feature selections using AdaBoost: Application in gene-gene interaction detection,'' in *Proc. IEEE Int. Conf. Bioinf. Biomed. Workshops (BIBMW)*, Oct. 2012, pp. 831–837.

[31] K. Che, X. Liu, M. Guo, J. Zhang, L. Wang, and Y. Zhang, ''Epistasis detection using a permutation-based gradient boosting machine,'' in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2016, pp. 1247–1252.

[32] L. Xiong, P.-F. Kuan, J. Tian, S. Keles, and S. Wang, ''Multivariate boosting for integrative analysis of high-dimensional cancer genomic data,'' *Cancer Inform.*, vol. 13, p. 16353, Jan. 2014.

[33] E. A. Houseman *et al.*, ''Model-based clustering of DNA methylation array data: A recursive-partitioning algorithm for high-dimensional data arising as a mixture of beta distributions,'' *BMC Bioinf.*, vol. 9, no. 1, pp. 1–15, 2008.

[34] V. Potenciano, M. M. Abad-Grau, A. Alcina, and F. Matesanz, ''A comparison of genomic profiles of complex diseases under different models,'' *BMC Med. Genomics*, vol. 9, no. 1, p. 3, 2015.

[35] G. H. Lubke *et al.*, ''Gradient boosting as a SNP filter: An evaluation using simulated and hair morphology data,'' *J. Data Mining Genomics Proteomics*, vol. 4, 2013, doi: 10.4172/2153Ű0602.1000143.

[36] A. Mikhchi, M. Honarvar, N. E. J. Kashan, S. Zerehdaran, and M. Aminafshar, ''Comparison of three boosting methods in parent-offspring trios for genotype imputation using simulation study,'' *J. Animal Sci. Technol.*, vol. 58, no. 1, p. 1, 2016.

[37] Q. Xu *et al.*, ''PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm,'' *J. Theor. Biol.* vol. 417, pp. 1–7, Mar. 2017.

[38] F. Dorani, T. Hu, M. O. Woods, and G. Zhai, ''Ensemble learning for detecting gene-gene interactions in colorectal cancer,'' *PeerJ*, vol. 6, p. e5854, Oct. 2018.

[39] F. Firoozbakht, I. Rezaeian, A. Ngom, L. Rueda, and L. Porter, ''A novel approach for finding informative genes in ten subtypes of breast cancer,'' in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Aug. 2015, pp. 1–6.

[40] J. H. Friedman, ''Greedy function approximation: A gradient boosting machine,'' *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[41] L. Breiman, ''Random forests,'' *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[42] T. Chen and C. Guestrin, ''XGBoost: A scalable tree boosting system,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[43] A. Jalali and N. Pfeifer, ''Interpretable per case weighted ensemble method for cancer associations,'' *BMC Genomics*, vol. 17, no. 1, p. 501, 2016.

**LONGQUAN JIANG** received the master's degree in computer science from the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China, in 2018. His research interests include data mining and analysis, intelligent information processing, and deep learning.

**BO ZHANG** received the Ph.D. degree in computer science from the College of Electronics and Information Engineering, Tongji University, in 2009. In 2012, he completed the Postdoctoral research work at Tongji University. He is currently a Professor with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. He is also the Director of the Trust Computation Project, which is funded by the National Nature Science Foundation of China. His current research interests include data mining, trust computation, and social network analysis.

**QIN NI** received the Ph.D. degree from the Universidad Politécnica de Madrid, Spain, in 2016. She is currently a Lecturer with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, China. Her research interests include pervasive computing, smart environment, activity modeling and recognition, smart home, and healthcare system development. Contact her at niqin@shnu.edu.cn.

**XUAN SUN** received the master's degree in computer science from the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China, in 2018. She is currently a Lecturer with the School of Information Science and Technology, Sanda University of Shanghai. Her research interests include data mining, text classification, and machine learning.

**PINGPING DONG** is currently pursuing the master's degree with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China. Her research interests include machine learning, data mining, and recommendation systems.