This article has been accepted for publication in IEEE/ACM Transactions on Computational Biology and Bioinformatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3214863

A secure high-order gene interaction detection algorithm based on deep neural network

Yongting Zhang[1,2,4], Yonggang Gao[1,4], Huanhuan Wang[1,4], Huaming Wu[3], Youbing Xia[1,4], Xiang Wu[1,2,4*]

Abstract — Identifying high-order Single Nucleotide Polymorphism (SNP) interactions of additive genetic model is crucial for detecting complex disease gene-type and predicting pathogenic genes of various disorders. We present a novel framework for high-order gene interactions detection, not directly identifying individual site, but based on Deep Learning (DL) method with Differential Privacy (DP), termed as Deep-DPGI. Firstly, integrate loss functions including cross-entropy and focal loss function to train the model parameters that minimize the value of loss. Secondly, use the layer-wise relevance analysis method to measure relevance difference between neurons weight and outputting results. Deep-DPGI disturbs neuron weight by adaptive noising mechanism, protecting the safety of high-order gene interactions and balancing the privacy and utility. Specifically, more noise is added to gradients of neurons that is less relevance with the outputs, less noise to gradients that more relevance. Finally, Experiments on simulated and real datasets demonstrate that Deep-DPGI not only improve the power of high-order gene interactions detection in with marginal and without marginal effect of complex disease models, but also prevent the disclosure of sensitive information effectively.

Index Terms — Genome-Wide Association Studies, gene interaction detection, deep learning, differential privacy

1 INTRODUCTION

The genetic basis of many complex diseases involves multiple genetic variants, such as Single Nucleotide Polymorphism (SNP), and complicated interactions between them [1]. Increasing evidence shows that genes do not function independently, rather, they cross talk with each other, termed the gene–gene interaction [2]. Detecting gene-gene interaction refers to finding the combinations of multiple genes that affect complex diseases to identify the pathogenic causes and genetic mechanism of complex diseases in humans, which has played important role in Genome-Wide Association Studies (GWAS) [3].

Methods for gene–gene interactions have been extensively studied in the literature [4-9]. Attila et al. [4] proposed the exhaustive method, which required scanning all possible combinations in detecting epistatic effect. While this method took comprehensiveness and integrity into account, it did not balance the experimental calculation burden and detection efficiency. Literature [5] constructed the statistical-based approach to estimate the gene combinations. This method could decrease the calculation burden, but still could not increase the power of detection. The swarm intelligence-based method has the advantage of controllable time complexities, heuristic positive feedback search and high detection power, the researches including FHSA-SED [6], IPSO [7], DECMDR [8], AntEpiSeeker [9] and so on. These methods based on non-parametric did not assume specific parametric models, thus, they had certain advantages. However, they could only detect gene interactions of without marginal effect disease model or weak marginal effect and could not estimate the interaction effect in most models, making result interpretation challenging. Moreover, with the exponential growth of the number of SNPs, the detection of *K*-order gene interactions on the basis of these methods, especially, when *K* is greater than 3, is still unable to achieve high performance due to the enormous computational burden.

Deep Learning (DL) has emerged from the advances in high dimensional data by using sophisticated algorithms and the power of parallel computation, solving poor accuracy performance and overcoming the influence of computational burden [10]. However, the majority of current approaches based on DL does not seem suitable for gene interaction process due to the objective of *K*-order gene detection process has the particularity [11]. More importantly, the DL model training process will cause the disclosure of genetic private information in the training data [12]. The previous study has shown that adversary can identify someone only by obtaining 30~80 SNP information [13]. Suppose that one adversary grabs the 75 SNP information with the help of repeat query attack from the published DL model for gene interaction. He can predict the private and sensitive feature of target individual based on the released model and some background information about the target individual, making use of the unknown, sensitive feature and

- ** Huaming Wu and X. Wu are the corresponding author.*
- [1] *The Institute of Medical Information Security, Xuzhou Medical University, Xuzhou 221000, China.*
- [2] *The School of Engineering, Universiti Malaya, Kuala Lumpur, 55100, Malaysia.*
- [3] *The Center for Applied Mathematics, Tianjin University, Tianjin, 300072, China.*
- [4] *The School of Medical Information & Engineering, Xuzhou Medical University, Xuzhou 221000, China.*

model output. Thus, publishing the DL model without privacy protection will increase the risk of private information leakage.

Consequently, it is urgent to develop a rigorous privacy preserving framework that not only resists the attack of adversaries, but also improves the accuracy of high-order gene interactions. In this paper, we propose the high-order gene interactions detection DL framework (Deep-DPGI) based on Differential Privacy (DP). This framework first identifies gene interaction combinations in the DL model and sets multi-loss functions that are more suitable for high-order sites detection. Secondly, Deep-DPGI uses the layer-wise relevance analysis method to measure relevance difference between neurons weight and outputs, and disturbs neuron weight by adaptive noising mechanism to protect the safety of high-order gene interactions, balancing the privacy and utility. The main contributions of Deep-DPGI are the following:

(1) Analyze the particularity of high-order gene interaction process in DL model, design and integrate multi-loss functions to make sure that the detecting process is more reasonable.

(2) Propose the adaptive noising mechanism to protect the security of whole identifying process by layer-wise relevance analysis method to measure relevance difference between neurons weight and outputting results, disturbing neuron weight.

(3) According to the relevance analysis result, add more noise to gradients of neurons that is less relevance with the outputs, less noise to gradients that more relevance, solving the imbalance between privacy and utility.

The remainder of the paper is organized as follows. Section 2 overviews the related literature and lists the preliminaries of this paper, mainly including gene interaction, deep learning and differential privacy. In Section 3, we introduce our proposed method in detail. The experimental evaluations and results are discussed in Section 4. Finally, Section 5 summarizes the paper.

2 PRELIMINARIES AND BACKGROUND

DL constructs the multi-layer structured network to learn the internal development rules of data objects under unsupervised conditions, which has improved training performance. Based on this, this paper identifies high-order gene interactions using DL. However, the training of DL requires private and representative datasets, which contain sensitive personal information probably. Ideally, this training process will not disclose private information. In fact, one adversary can steal sensitive information and infer the key feature by constructing model inversion attack, which will leads to the disclosure of private information [14]. Therefore, integrating privacy protection methods into DL methods is a feasible approach to address privacy threats. This section will introduce the definition of the differential privacy theory, deep neural networks and the Layer-wise relevance analysis algorithm in detail.

Given the dataset $D = \{(X_1, Y_1), (X_2, Y_2), \dots (X_n, Y_n)\}$ (X represents the SNP, and the Y is the class label of SNP association with disease), where $X_i = (x_{i1}, x_{i2}, \dots x_{id})$ (x_{ij} is the gene-type result of SNP). Our objective is to protect the safety of D by an adaptive differential privacy mechanism for deep neural network that takes X_i as input and ensure the accuracy of output Y_i to the greatest extent. Table I summarizes the notations used throughout this paper.

TABLE I SUMMARY OF NOTATIONS OF THIS PAPER

Notations	Description
GWAS	Genome-wide association studies
SNP	Single Nucleotide Polymorphism
DL	Deep Learning
DP	Differential Privacy
D	Input datasets
X_i	The training samples
Y	The output of machine learning model
K	Covolution kernal
t	The epoch time of CNN model
$\mathcal{M}$	Randomized algorithm
$\{h_1, h_1, \dots, h_n\}$	Hidden layers number
e, p	One neuron
$R_e^l(X_i)$	The relevance analysis result between input and one neuron
T_p	An affine transformation of neuron
$f(p)$	The update parameter of each training layer
γ	Regularization parameter
θ	Learning rate
Z_t	Random Laplace noise
MAF	Minor Allele Frequencies

2.1 Deep learning concept

DL learns and extracts internal laws of datasets by multi-layer networks that describe the potential relationships between the inputs and outputs, and has been one of the most used machine learning technologies [15]. There are multiple networks of deep learning frameworks, such as Multi-Layer Perception (MLP) [16], Convolutional Neural Network (CNN) [17], Recurrent Neural Network (RNN) [18] ans so on. Different networks are applied for solving different types of problems. Among these models, CNN is a very common and representative model of deep learning and is used in this paper. Specifically, CNN can share the convolution kernel during layers, and there is no need to manually select features on high-dimensional data processing. The definition of CNN is shown as follows.

Definition 1 (***Convolutional Neural Network***) [17]. CNN generally consists of the input layer, convolution and layer (also called hidden layer), fully connected layer and output layer. CNN employs the notion of convolution that is not matrix multiplication but the mathematical linear operation at least in one of the hidden layers. The contribution of convolution is to determine the feature maps. Given a three-order SNP as input with three-dimensional kernel K, and the outputting Y is expressed as follows:

$$Y[p,q,r] = (S * K)[p,q,r] = \sum_{h} \sum_{i} \sum_{j} S[p-h, q-i, r-j] K[h,i,j] \quad (1)$$

Where p, q, r represent three SNPs respectively. The new feature maps will be obtained by combining the input and learned kernel. And, the non-linear activation function is used for the convolved outputting.

$$Activation\ function = \frac{2}{1+e^{-2Y[p,q,r]}} - 1 \quad (2)$$

This function is as the input in pooling layer, and the activation function in pooling layer is softmax activation function. Softmax activation function computes probability for each class by interpreting its confidence value. The total error of the output layer is calculated by using cross-entropy function as follows:

$$Error =- \frac{1}{N} \sum_{s} (ln^{Y * Y'} + ln^{(1-Y)*(1-Y')}) \quad (3)$$

Where Y' is the desired output and N is the sample of CNN model.

Then, the gradient optimization method is used to compute the partial derivative of cross entropy loss function to search the optimal parameters. The key parameters of the CNN model are updated for every epoch from time t to $t+1$.

2.2 Related work of deep learning in gene interaction

There is no doubt that DL is a popular branch of machine learning techniques. Research [19] extended the DL by proposing the hybrid architecture DNN-RF to improve the presicion to some extent. Li et al. [20] conducted the DL on the basis of clustering for the prediction of gene interactions detection, termed as DPEH. However, it did not validate algorithm performance on real datasets, its actual availability is questionable, and it was only suitable for small datasets. Although, literature [21] showed the performance of its method based on convolution neural network in detecting two-loci of hypertension data, but still used the sorting method to pick out top 20 relevant sites actually, which meant that this method was a false neural network intelligent method. Abdulaimma et al. [22] proposed the framework using the DL to model the cumulative effects of SNPs for the classification of Type 2 Diabetes. After verification, the practicability and generality of this method is poor. Wang et al. [23] studied the marginal epistasis by using DL that combines the one-dimensional convolutional neural network and the Long-short Term Memory. But, literature [24] had concluded that one-dimensional CNN model was disadvantaged for prediction of complex sites. The deep learning in identifying SNP interactions is yet to meet its potential achievements [22]. However, the first drawback of DL methods is that they are highly specialized to a specific domain, and reassessment is needed to tackle issues that do not pertain to that identical domain. These models are unable to understand the expression of the data that they are trained with, which is an issue while interpreting the results [25]. Then, these models do not really apply to high-order gene Interactions. Last but not least, they do not consider the security of input data that usually contain the large amounts of private information of contributor during DL training.

2.3 Differential privacy concept

Differential privacy is as one promising strategy for data privacy protection, and is usually integrated into machine learning and DL algorithms to preserve the privacy of input data [26-27]. Indeed, differential privacy presents the reliable privacy guarantee to ensure that adversaries cannot infer the inclusion or exclusion of records in the database, even if they have information about all records other than the target. The definition of it is shown as follows.

Definition 2 **(*ϵ-Differential Privacy*)** [28]. Given two adjacent databases D_1, $D_2 \in D$, the randomized algorithm $\mathcal{M}: D \rightarrow R$ satisfies ϵ-differential privacy, and if for any subset of output $O \subseteq R$, we have:

$$Pr[\mathcal{M}(D_1) = O] \leq e^{\epsilon} Pr[\mathcal{M}(D_2) = O] \quad (4)$$

Where ϵ is privacy budget and is an important role in affecting the privacy preserving intensity. ϵ represents the protection level of the randomized algorithm $\mathcal{M}$ can provide. In practice, the privacy budget ϵ is always set to a small value because the smaller the ϵ, the stronger the privacy guarantee, and vice versa. ϵ should be greater than 0, however, although in $\epsilon = 0$ the algorithm can provide the strongest guarantee privacy for training data, but for any adjacent datasets, there are two same probability distribution, and also can not reflect any useful information about data. Therefore, the research of the size design of ϵ value has always been a hot direction in the field of differential privacy. Actually, it is difficult to model in utility and seek the trade-off between privacy information and privacy.

2.4 Related work of differential privacy in deep learning

As literature [14, 48] demonstrated, deep neural network model training data (especially some highly sensitive data, such as biological or image data containing personal information, etc.) has the risk of privacy information disclosure, and privacy protection methods must be integrated during the training. Differential Privacy has become one of the most popular methods for preserving privacy for all records due to it has the strict mathematical theory. Many scholars have carried out researches on differential privacy protection for deep learning. Xia et al. proposed the gradient based differential privacy optimizer in [30], which simply combined random sampling, gradient clipping, gradient based on random perturbation and advanced privacy budget statistics methods. Gati et al. [31] expressed the data by tensor and disturbed the tensor matrix to ensure the differential privacy preserving. Chang et al. [32] focused the privacy of neural network and proposed the scheme for solving the privacy disclosure of

This article has been accepted for publication in IEEE/ACM Transactions on Computational Biology and Bioinformatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2022.3214863

centralized and distributed by analyzing the privacy vulnerabilities of the training model. Hao et al. [33] proposed an efficient and privacy-protected joint deep learning protocol by combining the homomorphic encryption with differential privacy. It assumed that third-party servers are honest and secure, but this assumption was unreasonable. Xu et al. [34] studied the secure framework based on differential privacy for edge computing that injected noise into learned features achieving the purpose of bofuscating sensitive information. Liu et al. [35] presented the privacy-protected generative adversary-network model, by adding noise to the training gradient and balanced privacy and utility by controlling the number of training iterations. Cheng et al. [36] proposed a new algorithm, averaging noise stochastic Gradient Descent. However, Yang et al. [37] pointed out that the generally differential privacy SGD algorithm (DP-SGD) added Gaussian noise of a fixed level would cause the accuracy of the model to decrease slowly with the increase of training times. And, Hoefer et al. [38] concluded that the the classification performance would decrease with the development of the pre-training model in differential private data classification under dynamic pre-training model, especially when the datasets themselves were not be considered. Zhang et al. [39] demonstrated that deep neural network with standard differential privacy would not provide quantifiable protection to fend off model reversal attack, by reconstrcuting the training data from the existing model reversal attack. Wang et al. [40] embed differential privacy into specific layers and learning processes to achieve domain adaptation privacy guarantees. Gong et al. [41] aimed to bridge the gap between private and non-private models and proposed the general differential private deep neural network learning framework based on back propagation algorithm. Although this framework improved data availability to some extent, it still led to excessive back propagation gradient and algorithm time complexity.

2.5 Layer-wise relevance analysis concept

Layer-wise relevance analysis is a classical algorithm that calculates the relevance between each input feature x_{ij} and output $\mathcal{F}x_i(O)$ by decomposing the neurons of preceding layers. The definition and process of relevance analysis is illustrated in the following.

Definition 3 **(Layer-wise relevance analysis)** [42]. l hidden layers $\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_l$, given $R_e^{(l)}(x_i)$ is the relevance result between input x_i and neuron e at layer l. Define the process is $R_{e\leftarrow p}^{(l-1,l)}(x_i)$ that neuron e send the message to p. The total relevance of neurons is:

$$R_e^{(l-1)}(x_i) = \sum_{p \in \mathrm{h}_l} R_{e\leftarrow p}^{(l-1,l)}(x_i) \tag{5}$$

The decomposing of layer-wise relevance analysis is:

$$R_{e\leftarrow p}^{(l-1,l)}(x_i) = \{ \begin{matrix} \frac{T_{ep}}{T_p+\theta} R_e^{(l)}(x_i), & T_p \geq 0 \\ \frac{T_{ep}}{T_p-\theta} R_e^{(l)}(x_i), & T_p < 0 \end{matrix} \tag{6}$$

$\theta\ (\theta \geq 0)$ is the predefined stabilizer to resolve the issue of the unboundedness of $R_e^{(l)}(x_i)$. Where T_p is an affine transformation of neuron e, and it can be defined as:

$$T_{ep} = v_e \omega_{ep} \tag{7}$$

$$T_p = \sum_e T_{ep} + u_p \tag{8}$$

Where v_e is the value of neuron e, ω_{ep} is the weight between neuron e and p. u_p is the basic term.

In the last hidden layer, for the output variable o, the relevance is calculated as follows:

$$R_p^{(l)}(x_i) = \{ \begin{matrix} \frac{T_{po}}{T_o+\theta} f_{x_i}(\omega), & T_o \geq 0 \\ \frac{T_{eo}}{T_o-\theta} f_{x_i}(\omega), & T_O < 0 \end{matrix} \tag{9}$$

3 The Proposed Method

At present, epistatic detection studies still focus on identifying 2-order gene interactions. Moreover, few researchers have paid attention to the security of genome-wide association studies analysis based on gene interactions. To remedy these research gaps, this paper proposes a secure high-order gene interactions detection framework (Deep-DPGI). This framework provides privacy guarantee for deep neural networks based on relevance analysis for high-order gene interactions, which not only preserves the private information in the training data effectively, but also keeps the utility of the framework by adaptive disturbance mechanism to gradients. As shown in Fig. 1, Deep-DPGI consists of five steps, including standardizing input data, determining output requirements, CNN training, correlation analysis, and result output. The layer-by-layer correlation analysis method is integrated in the output layer of the convolutional neural network, and the correlation between weighted correlation neurons and classification results is analyzed mainly through back propagation. Small noise is allocated to the parameters of neurons with strong correlation, and large noise is allocated to those with weak correlation. In addition, the size of the noise range is determined by the Laplace distribution of the data and the results of the correlation analysis. The purpose of this is to obtain the trade-off between privacy and availability, and avoid adding too much noise and low data availability, and vice versa. The definition of high-order gene interaction, specific problem definition and method elaboration will be presented in the following sections.

3.1 The definition of high-order gene interaction

The scientific community generally believe that there is almost no phenotype characteristics of an individual is determined only by a single gene, thus gene-gene (or gene-environment interaction) to explain individual characteristics has important theoretical and practical significance, also makes the study of gene-gene are being more and more attention. The process of identifying gene-gene interactions consists of three steps: sequencing of contributed gene data, standard SNP data, and association analysis. The process of high-order gene interaction is defined as the combinations of at least κ SNPs affecting phenotype or disease genes. We express the gene interactions process as $R = \{S, G, A\}$, where $S = \{S_1, S_2 ..., S_j\}$ represent SNP typing, $G = \{G_{11}, G_{12}, ..., G_{ij}\}$ represent interaction between G_i and G_j corresponding genes, and $A = \{A_1, A_2 ..., A_i\}$ represented association results. The κ-order gene interaction represents the recognition of SNP interaction results of the order of 3^n. Among them, when $G_{mn} > \theta$, G_{mn} is called the result with the main effect, and when $G_{ab} < \theta$, G_{ab} is the result of the edge effect.

3.2 Problem statement

Let the set of gene variables $X = \{X_1, X_2, \dots, X_i\}$ includes $S = \{S_1, S_2, \dots, S_j\}$ SNP marker for N individuals. For high-order gene interaction detection algorithms, the temporal $O(f(n))$ and spatial $S(n)$ complexity of the algorithm increases exponentially in 3^n detection demand. Convolutional neural networks reduce complexity and improve detection efficiency by constructing multiple convolutional and pooling layers. But there are three ways in which neural network training data may reveal genetic privacy. Firstly, in the data input phase, one attacker A initiates $AK = \{AK_1, AK_2, ..., AK_n\}$attacks, including repeated queries and so on, obtaining about lots of SNPs information $I_1, I_2, ... I_n$, and combining background knowledge $KN = \{KN_1, KN_1, ..., KN_1\}$ to directly locate in the individual. Secondly, in the data analysis stage, A launches $AK = \{AK_1, AK_2, ..., AK_n\}$ attacks, including model inversion attack and so on, to obtain the T gradient, ω weight, θ learning rate and other key parameters related to the original input data. A may achieve sensitive information by combining the these parameters and $KN = \{KN_1, KN_1, ..., KN_1\}$. Finally, in the outputting results stage, the privacy disclosure process is similar to that of the input scenario. For the training model without integrated privacy protection methods, the output of the model is directly related to the original data. When A obtains a certain number of output results, he may locate an individual based on the $KN = \{KN_1, KN_1, ..., KN_1\}$ to obtain sensitive information of the individual.

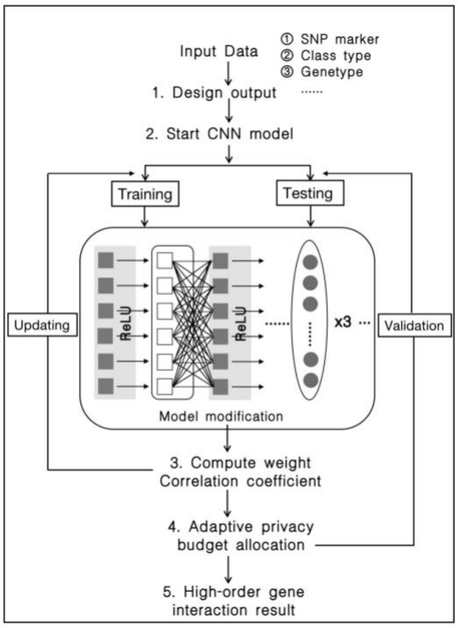


Fig. 1: Flow diagram of detecting high-order gene interaction by deep learning in Deep-DPGI framework

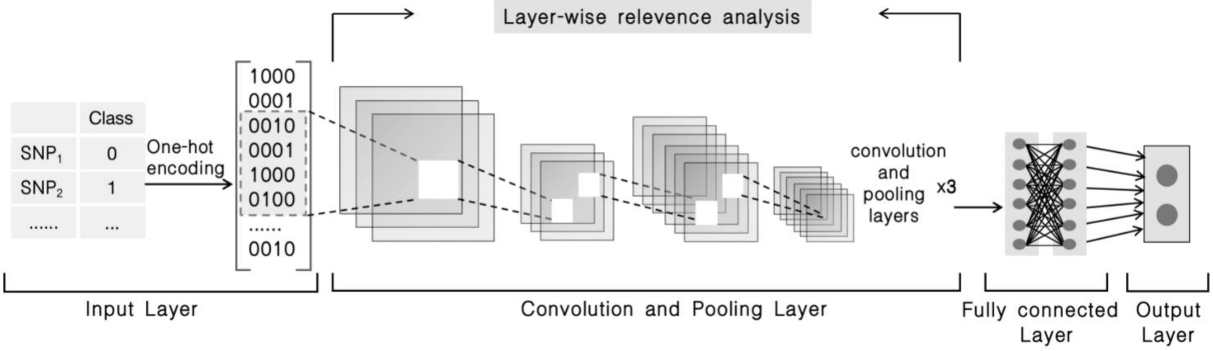


Fig. 2: The architecture of Deep-DPGI proposed in this paper

As shown in Fig. 2, this paper proposes a method to protect the safety of CNN framework, termed as Deep-DPGI. The method determines the importance of each neuron by analyzing the correlation between each layer of neuron and the result during back-propagation, providing each neuron with varying degrees of privacy protection. Next, we will elaborate this method.

3.3 Design integrated loss functions

Loss function is an important part of unsupervised machine learning. A good loss function is critical for successful training of model parameters because it is possible to determine parameters that minimize the mean value of losses for a given training set [43]. The detection efficiency of the current deep learning model for epistatic detection is low due to use the learner for multiple repeat training when detecting multi-order gene interactions. In this paper, We integrate the commonly used cross-entropy and the novel focal loss function [44], which is originally used to resolve the text classification problem. In order to make it suitable for epistatic detection process, we have optimized and improved it, and the specific definition is as follows.

Cross entropy loss, also called logarithmic loss, is the most commonly used classification loss function in neural networks. The outputting prediction is always between 0 and 1 and is interpreted as probability, which is the maximization of logarithmic likelihood between the training data and the corresponding data condition. As the predicted probability deviates from the actual label, the cross entropy loss increases. The definition is shown in Equation (10).

$$L_e(\hat{Y_i}, Y_i) =- \sum_i Y_i log(\hat{Y_i}) \tag{10}$$

Define the f_p to express the probability for the classification:

$$f_p = \begin{cases} p & if\, Y = 1 \\ 1-p & otherwise \end{cases} \tag{11}$$

To balance the classification in Equation (11), focal Loss introduces a regulating factor $(1- f_p)\beta,\ \beta \geq 0$. In this paper, the β in the specific DL framework can be obtained by reversing the frequency and by the parameter cross validation process. Specifically, the focal loss function is defined as follows:

$$L_e(f_p) =- (1- f_p)^{\beta} log(f_p) \tag{12}$$

Deep-DPGI searches for high-order gene interaction results under the cross-entropy and focal function. The two classifiers first search for different epistatic genes and then correlate epistatic combinations to find high-order interaction results. The loss function of this paper is defined as follows:

$$L = \min_{x_i,x_j} \sum_{i=1}^{N} \sum_{j=1}^{\omega} \beta_{i,j}^2 L_{i,j}(Y_{i,j}, \hat{Y}_{i,j}) \tag{13}$$

3.4 Distorting neuron weight based on relevance analysis

In centralized analysis scenarios, sensitive information will be disclosed in the training and sharing stage of epistatic detection research based on machine learning. Differential privacy noise perturbation methods that generally integrated in machine learning include perturbing output result, perturbing gradient, perturbing objective function coefficient and so on. These methods ignore the actual requirement of the data and add inappropriate noise, resulting in poor availability or insufficient protection degree. This paper adds appropriate Laplace noise to the training gradient of neurons on the basis of analyzing the correlation between each neuron layer and the output layer. The core theoretical operation process is shown in Fig. 3. Relevance analysis begins after forward propagation and backward propagation finishing, and total neuronal relevance results of one layer are been calculated. Moreover, the average value can be got between one neuron and outputting results. More importantly, relevance analysis results will adjust with the process of forward and backward propagation, and finish until the propagation process ends.

At the beginning of the training, we define the gradient updating objective function of the general optimization method. In each training step, a group of random training samples L on data set X is used, starting from the initial point f_0 and updating parameter f at t step, we have:

$$f_{t+1} = f_t - \theta_t(\gamma f_t + \frac{1}{L} \sum_{i=1} L(f_t, X_i)) \tag{14}$$

Where θ is the learning rate of step t and γ is the regularization parameter.

In the process of back propagation, we obtain the total correlation value between neurons at layer $j = \{j_1, j_2, \ldots, j_n\}$ and the result through Equation (15).

$$R_j^X = \sum_{i=1} R_{i \leftarrow j}^X \tag{15}$$

Then, the correlation analysis results of individual neurons are obtained by averaging.

$$R_j(X_i) = \frac{1}{N} \sum_{i=1} R_{ij}(X) \tag{16}$$

In order to better combine the correlation result with the noise distribution mechanism, we introduce the correlation coefficient r. Because the stronger the correlation is, the smaller the added noise will be, and vice versa, so the r is expressed as an inverse relationship in this paper.

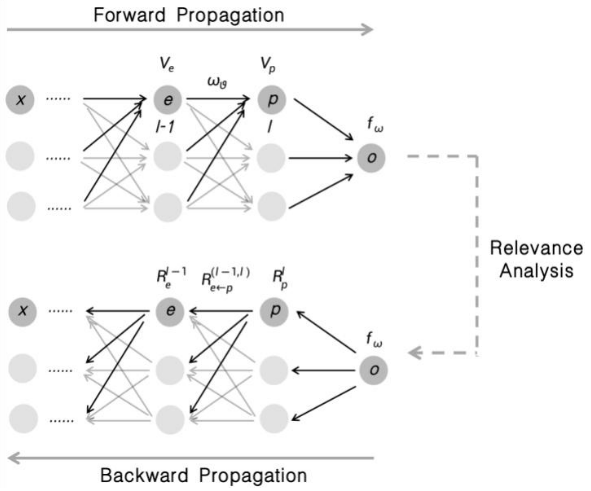


Fig. 3: The process of layer-wise relevance analysis method of Deep-DPGI based on back propagation algorithm

Algorithm 1: DPLRP.

```
Input: SNP datasets $X$, privacy budget $\epsilon$, Learning rate θ, the number of batches $t$, Loss function $L(X_i)$, relevance coefficient $r$.
Output: The optimal and noised gradient of each neuron $f_{t+1}$.
1: Initialize the model parameters.
2: for $j \in [1, i]$ do
3:     Calculate the relevance $R_j(X)$ of each layer in deep neural network.
4:     Get the relevance coefficient $r_j$.
5:     Allocate the adaptive privacy budget $\epsilon_{ij} = r_{ij} \times \epsilon$.
```

```
end for
for $t \in T$ do
    Select the dataset $L_i$ from training samples $L$ on $X$.
    Compute gradient $f(X_i) \leftarrow L(f_i, X_i)$.
    Gradient update after noised $f_{t+1} = f_t - \theta_t(\gamma f_t + \frac{1}{L}(\sum_{i=1} L(X_i) + Z_t))$.
end for
```

$$r_j = \frac{1}{R_j(X_i)} \tag{17}$$

In addition, the adaptive privacy budget allocation is given by:

$$\epsilon_{ij} = r_{ij} \times \epsilon \tag{18}$$

Where ϵ refers to the total privacy budget value calculated from the Laplace distribution of data.

Finally, we disturb the training gradient to ensure the security of SNPs in the training and sharing stages in the centralized scenario. As shown in Equation (19),

$$f_{t+1} = f_t - \theta_t(\gamma f_t + \frac{1}{L}(\sum_{i=1} L(X_i) + Z_t)) \tag{19}$$

Z_t is the Laplace noise. Pseudo-code of adaptive disturb mechanism based on correlation analysis, taking SGD learner as an example, termed as DPLRP, is shown in Algorithm 1.

Next, we will prove that our algorithm satisfies ϵ-differential privacy.

Proof: Given that L and L' are two adjacent batches. The $f_{t+1(L)}$ and $f_{t+1(L')}$ are the parameters of L and L'. The formula is expressed as follows.

$$f_{t+1(L)} = f_t - \theta_t(\gamma f_t + \frac{1}{L}(\sum_{i=1} L(X_i))) \tag{20}$$

$$f_{t+1(L')} = f_t - \theta_t(\gamma f_t + \frac{1}{L}(\sum_{i=1} L(X'_i))) \tag{21}$$

Then, the inequality of two outputs difference is the following:

$$\begin{aligned}\Delta_{ft} &= \frac{\theta_t}{|L|}\sum_{f\in f_t} \| \sum_{X_i\in f_t} L(X_i) - \sum_{X'_i\in f'_t} L(X'_i) \|_1 \\ &\leq \frac{\theta_t}{|L|}\sum_{f\in f_t} \| \sum_{X_i\in f_t} L(X_i) \|_1 \\ &+ \frac{\theta_t}{|L|}\sum_{X_i\in f_t} \| \sum_{X'_i\in f'_t} L(X'_i) \|_1 \\ &\leq 2\frac{\theta_t}{|L|}\max_{X_i\in f_t}\sum_{f\in f_t} \|L(X_i)\|_1\end{aligned} \tag{22}$$

Meanwhile, from the Equation (22) and differential privacy, Δ_{ft} is the sensitivity of neural network ($\Delta_{ft} \leq 2\frac{\theta_t}{|L|}$) . To protect the private information of neural network, we disturb the gradient based on relevance analysis, the noise can be written as:

$$f_{t+1} = f_t - \theta_t(\gamma f_t + \frac{1}{L}(\sum_{i=1} L(X_i) + Lap(\frac{\Delta_{ft}}{\epsilon_i}))) \tag{23}$$

We have:

$$\begin{aligned}&\frac{Pr[f_{t+1(L)}]}{Pr[f_{t+1(L')}]} \\ &= \frac{\prod_{f\in f_t}\prod_{i=1} n\, exp(\frac{\epsilon_i\frac{\theta_t}{|L|}\|\sum_{X_i\in f_t} L(X_i) - (\sum_{X_i\in f_t} L(X_i) + Lap(\frac{\Delta_{ft}}{\epsilon_i}))\|_1}{\Delta_{ft}})}{\prod_{f\in f_t}\prod_{i=1} n\, exp(\frac{\epsilon_i\frac{\theta_t}{|L|}\|\sum_{X'_i\in f'_t} L(X'_i) - (\sum_{X_i\in f_t} L(X_i) + Lap(\frac{\Delta_{ft}}{\epsilon_i}))\|_1}{\Delta_{ft}})} \\ &\leq \prod_{f\in f_t}\prod_{i=1} nexp(\frac{\epsilon_i\frac{\theta_t}{|L|}}{\Delta_{ft}}\| \sum_{X_i\in f_t} L(X_i) - \sum_{X'_i\in f'_t} L(X'_i)\|_1) \\ &\leq \prod_{f\in f_t}\prod_{i=1} n\, exp(\frac{\epsilon_i\frac{\theta_t}{|L|}}{\Delta_{ft}} 2\max_{X_i\in f_t} \| L(X_i)\|_1) \\ &\leq \prod_{f\in f_t}\prod_{i=1} n\, exp(\epsilon\frac{2\frac{\theta_t}{|L|}\frac{r_j}{R_j(X_i)}}{\Delta_{ft}}) = exp(\epsilon)\end{aligned}$$

4 Experiments

In order to tackle the problem of privacy disclosure of the high-order gene interaction detection and improve the detection efficiency, this paper proposes the Deep-DPGI framework, which integrates the DL training model based on multiple objective functions and adaptive allocation disturbing mechanism based on correlation analysis, achieving the balance between privacy and utility. In this section, we will verify the performance of Deep-DPGI framework with the results of virtual simulation experiments, including the sources of datasets required by simulation experiments and the experimental operating environment.

4.1 Experimental setup

The datasets required by the experiment include simulated and real datasets. More specifically, the simulated datasets are generated by GAMETES 2.0 software [45], the sample size of case and control are 4000 respectively, and the number of SNP changed within 5000, which means that the datasets are at least 4000,000. The real datasets come from Age-related Macular Degeneration (AMD) sequencing results. There are a total of 8 disease models, among which Model 1-4 are the models with marginal effect, which are referred to the literature [46]. Model 5-8 are generated according to the penetrance table without no marginal effect. Furthermore, we adopt the Age-related Macular Degeneration (AMD) [47] datasets to judge Deep-DPGI performance. A 64-bit, Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz processor and 32GB RAM simulation environment is used to train the detection model. Python 3.6 is used as the main programming language, and TensorFlow 1.14 as the machine learning framework in the Windows10 system. In addition, since the interaction results of simulated datasets are the last three SNPs, in order to ensure the actual effect of the framework, the simulated datasets used in the training process are disturbed and the three SNPs are randomly placed in different positions of the datasets.

4.2 Performance of secure DL model for high-order interaction

In DL research, the classification performance of models greatly affects the accuracy of results. Therefore, evaluation of model performance is crucial to timely adjustment of model parameters to improve practical availability. In this experiment, True Positive Rate (TPR), False Positive Rate (FPR), Accuracy and other indicators are used to evaluate the CNN network for identifying high-order gene interactions used in this paper. A total of 300 iterations of training, among which the data in Fig. 4 (c) and (d) are the average values.

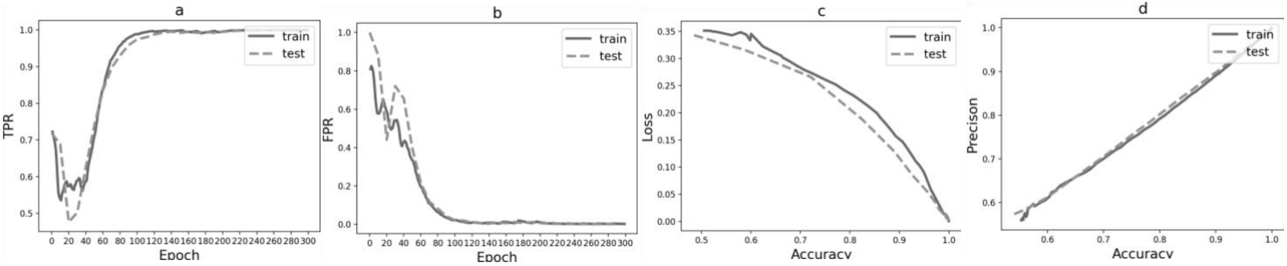


Fig. 4: Performance results of Deep-DPGI method in TPR, FPR, Loss and Precision

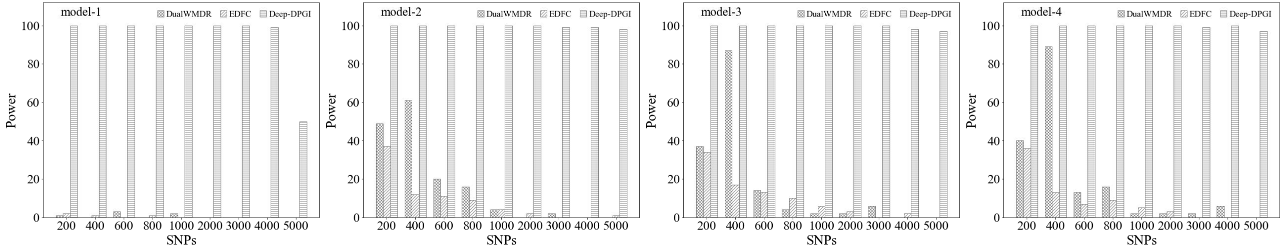
TPR and FPR results are obtained by constructing confusion matrix. TPR refers to the proportion of positive example data correctly identified in the total positive example data, also called recall rate. FPR stands for the percentage of misclassification data predicted to be correct. Ideally, the higher the TPR, the better performance, indicating that the model is more likely to be correctly classified. The smaller the FPR, the better performance, also means that the model is less and less likely to misclassify. Fig. 4 illustrates the performance of Deep-DPGI metrics with respect to accuracy, precision, loss, and classification error for both training and validation. As can be seen from Fig. 4(a), TPR gradually increases with the number of iterations until it approaches 100. It shows that Deep-DPGI can correctly identify positive case data. In Fig. 4(b), FPR gradually decreases with the number of iterations until it approaches 0 and becomes flat. The results show that the capability of Deep-DPGI model to identify error samples increases with the increase of iterations. Fig. 4 (c) shows the relationship between accuracy and loss. In general, the smooth curve indicates that the loss and prediction accuracy change direction is consistent, and the model performance is good. In fact, according to the experimental results, at the beginning of the training iteration, the model still had a high classification loss in identifying high-order gene interaction combinations, but the model after training has a good classification performance. Fig. 4 (d) is the comparison between the model's training precision and testing accuracy. It can be seen from the results that the overall model classification accuracy changes in the same direction, which can reduce the excessive number of iterations and high model complexity caused by training bias. It is worth noting that the number of training iterations of this model is within a reasonable range [48].

4.3 Power of high-order gene interaction based on simulated datasets

Simulated studies are exampled on three-locus epistatic interactions detection. As there are few methods to study high-order gene interaction, DECF [49] and DualWMDR [50] are selected as the compared algorithms. Refer literature [50] to create disease models 1-8 which are influenced by different penetrances and MAF to simulate the real gene state. Especially, MAF varies in [0.05, 0.4] for each epistatic model. Each dataset includes SNPs varying from 200 to 5000, 8000 samples with 4000 cases and 4000 controls. We guess that the number of iterations might have an impact on *Power*, so the average calculated by 200 iterations was taken as the experimental result to ensure the fairness and rationality of the experiment. In addition, *Power* is used to evaluate the performance of high-order gene interaction in simulated experiment, and defined as:

$$Power = \frac{N_T}{N_D} \tag{24}$$

Where N_T is the number of datasets in which specific disease-associated epistasis can be successfully identified, and N_D is the number of generated datasets. The comparison results are shown in Fig. 5.



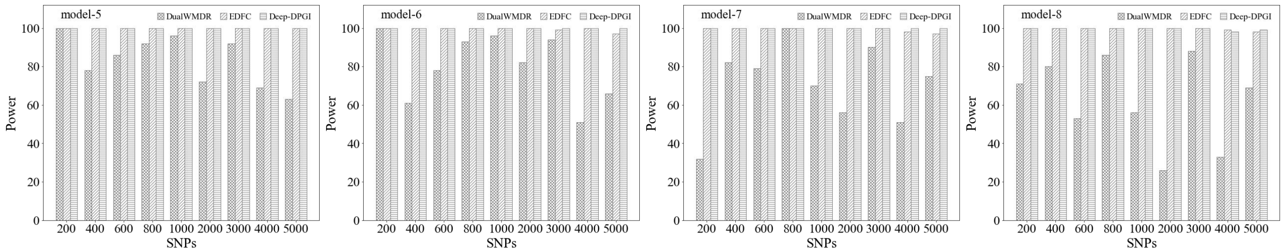


Fig. 5: Power performance comparison results between Deep-DPGI, EDCF and DualWMDR algorithms

The *Power* of these methods on eight three-locus epistatic models are shown in Fig. 5. For all cases, Deep-DPGI frequently outperforms EDCF and DualWMDR. The difference between Deep-DPGI and other methods is that Deep-DPGI can identify epistatic model with marginal effect. The obvious comparison results between them in Fig. 5 model 1-4 show that the Deep-DPGI is more accurate than EDCF and DualWMDR to evaluate the interaction effect of genes, and the *Power* of DualWMDR is high than EDCF. The reason is that Deep-DPGI uses DL method to seek the laws of epistatic model and can meet the needs of different scenarios with and without marginal effects under the guidance the idea of migration, furthermore, DualWMDR improves *Power* based on filtering and exhaustion, which is better than EDCF which only uses clustering method. In model 5-8 (epistatic model with no marginal effect) and SNP>1000 experiment, the overall performance of DualWMDR is inferior to that of EDCF and DeeP-DPGI. The reason is that the computational performance of this algorithm decreases with the increase of experimental data for high-order detection objects. The detection accuracy of EDCF is close to the Deep-DPGI method in this paper, which can meet the requirements of large-scale data analysis. However, after testing, it is found that the running time and space complexity of this method is high, and it is not suitable for large-scale practical application scenarios.

4.4 Power of high-order gene interaction based on real datasets

We apply Deep-DPGI on AMD real datasets [47]. AMD is the leading cause of blindness in middle-aged and elderly people and is a common eye disease. We downloaded AMD data from the official website of WTCCC, which contained 96 case individuals and 50 control individuals with 103611 SNPS. Through quality control, the number of SNP is 96607. Klein et. al [51] reported two interaction results most relevant to AMD, rs380390 and rs1329428. After the initialization parameters, the deep-DPGI framework took these two results as the main effect SNPS to search for the corresponding third-order gene interaction results in AMD. The results are shown in Table II.

TABLE II THREE-ORDER EPISTATIC DETECTION RESULTS OF DEEP-DPGI METHOD IN AMD DATASETS

Gene	SNP	Location	P-value
CFH, NPAT, PCDH9	rs380390, rs3781868, rs1036995	11q22, 13q21	8×10^{-18}
NRG3	rs1458402, rs2207768, rs4901408	11p15	8×10^{-18}
NXPH1, PTPRD	rs1476623, rs6967345, rs1408120	7p22, 9p23-p24	3.2×10^{-24}
KANK1	rs595113, rs1569651, rs2031175	9p24	4.9×10^{-24}
CFH, NPAT	rs132948, rs3781868, rs3781868	1p32, 11p22-23	6.78×10^{-10}
NAMPT, KCNH7	rs10487833, rs10495593, rs1740752	10p13	3.24×10^{-18}

Above are the results of the third-order gene interaction test for AMD. These SNPS are located in some important genes and perform important functions. For example, CFH gene on chromosome 1 encodes a protein that plays a key role in the regulation of complement activation. PCDH9 gene encodes a cadherin associated neuronal receptor and we assume that it involves in specific neuronal connections and signal transduction. In addition, the other SNP combinations also are found that would associate with AMD, but their biological explanation needs further research.

4.5 Privacy protection effect comparison of deep learning-oriented methods

This paper presents a DL differential perturbation method (DPLRP) based on correlation analysis. First, the correlation between each layer of neurons and the results is analyzed, and then the noise is allocated adaptively according to the characteristics of the data. We evaluate the DPLRP compared with Adam, DPAdam and Gaussian. Adam [52] has become the most commonly used neural network gradient optimizer because of its ability to address non-convex problems. Literature [29] demonstrated that the gradient of deep neural network model may disclose the private information. DPSGD [53] adds uniform Laplace noise to the gradient to ensure the security of the sensitive information in the training data. In addition, Gaussian [54] is also selected for comparison that added the equal amount of Gaussian noise to the gradient to prevent sensitive information leakage. All experimental results are obtained after 200 iterations. The experimental results are shown in Fig. 6.

Fig. 6 (a) is a combination diagram of training accuracy based on different training batches and different models. First, the line chart shows the change of accuracy of different learners with the increase of iteration times. Adam is the standard reference without noise disturbance. It can be seen from the line chart that compared with other methods, DPLRP method can greatly improve the accuracy at the beginning of training iteration, and its accuracy iteration speed is better than DPSGD and Gaussian. The reason is that DPLRP can avoid the interference of excessive noise by adding noise intelligently through the correlation between neurons and results, so as to quickly improve the accuracy of training. DPSGD and Gaussian lines have the same change trend, and the change direction of accuracy is consistent with the improvement speed. The reason is that the two methods ignore the characteristics of data and add the same amount of noise to the gradient, resulting in excessive noise and low initial accuracy of training. It can also be seen from the histogram below (a) that DPLRP is slightly better than other methods in detecting high-order gene interaction. Fig. 6 (b) is also a combination diagram of training accuracy based on different training batches and different models. The comparison algorithm includes only the learner with integrated gradient disturbance mechanism. First, the line chart shows the change of accuracy of DPSGD, Gaussian and DPLRP with the change of privacy budget size. As can be seen from the line chart results, the other two methods can always ensure high accuracy, while the accuracy of the DPLRP method in this paper changes along with the privacy budget value. In general, the smaller the privacy budget, the more noise you add and the lower the data availability. DPLRP changes in accordance with the standard law, while DPSGD and Gaussian add the same amount of noise to data each time, creating the illusion of high accuracy and ignoring the law of data distribution and development. As can be seen from the histogram below (b), DPLRP is slightly better than other methods in detecting high-order gene interaction.

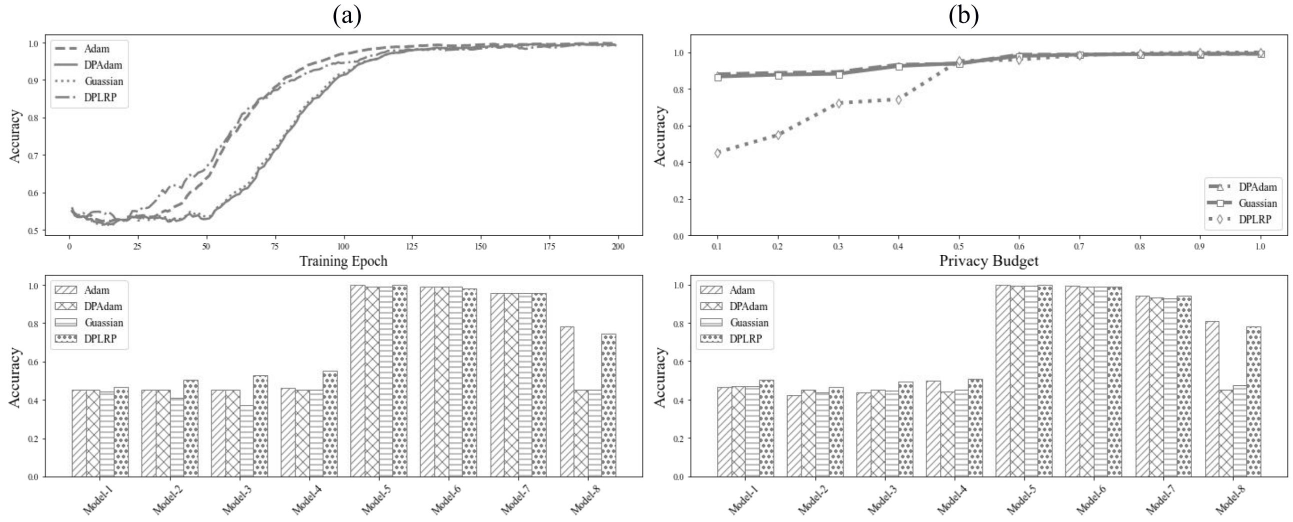


Fig. 6: Accuracy comparison results between Adam, DPSGD, Gaussian and DPLRP methods

5 Conclusion

In order to address the problem of privacy disclosure in the field of epistatic detection, improve the detection performance and reduce the detection burden high-order gene interaction, a secure detection framework for high-order gene interaction is proposed in this paper. This framework provides the intelligent protection mechanism that is an adaptive differential privacy preserving learning framework for deep neural networks based on relevance analysis. Our approach adds noise to the gradient adaptively according to the relevance results between neurons and outputs. Specifically, the more noise is added to the gradients that are less relevance to the outputs, and on the contrary, the less noise is added to the gradients of neurons which has more relevance to the outputs. In addition, we also identify network requirements for high-order gene interactions and optimized the convolutional neural network structure. A high order convolutional neural network method based on multiple loss functions is designed. Experimental evaluations are constructed on simulated and real datasets validate the accuracy of our framework. Currently, the Deep-DPGI algorithm also has certain limitations. For example, the execution time need to be improved. In our experience, one possible reason is that the more algorithmic components of Deep-DPGI. Then, we should compare Deep-DPGI with other state-of-the-arts in larger datasets, after all, we will have new challenges in the big data era.

In the future, our work will be extended towards the following aspects. On the one hand, an intelligent gradient clipping method is designed to accelerate the convergence of training and improve the effectiveness of the model. On the other hand, some other noising mechanisms based on differential privacy need to be studied to protect the security of sensitive information from multiple perspectives and ensure the security of model training and sharing. Finally, the research of Deep-DPGI on large-scale datasets is also our future research directions.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant No. 2020YFC2006600, the National Natural Science Foundation of China under Grant No. 62003291, National Science and Technology Foundation Project under Grant No. 2019FY100100, and the QingLan Project of Jiangsu Province of China.

DATA AVAILABILITY STATEMENT

Python 3.6 is the main programming language, and TensorFlow 1.14 is the machine learning framework. In addition, if we have to share the codes, we will consider uploading the key part of Deep-DPGI.

REFERENCES

[1] V. Milad, S. Siavash, K. Karim, et al., “Weighted Single-Step GWAS for Body Mass Index and Scans for Recent Signatures of Selection in Yorkshire Pigs”, Journal of Heredity, vol. 3, no. 3, pp. 1-15, 2022.

[2] N. Hao, Y. Feng, H. Zhang, “Model selection for high-dimensional quadratic regression via regularization”, JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION, vol. 113, pp. 615-625, 2018.

[3] M. Mills, C. Rahal, “The GWAS Diversity Monitor tracks diversity by disease in real time”, Nature Genetics, vol. 52, no. 3, pp. 242-243, 2020.

[4] G. Attila, M. Jonathan, “High-throughput analysis of epistasis in genome-wide association studies with BiForce”, BIOINFORMATICS, vol. 28, no. 15, pp. 1957-1964, 2012.

[5] T. Nguyen, J. Huang, Z. X. Wu, Y. Qing, “Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests”, BMC GENOMICS, vol. 16, pp. 1-11, 2015.

[6] S. Tuo, J. Zhang, X. Yuan, “FHSA-SED: Two-Locus Model Detection for Genome-Wide Association Study with Harmony Search Algorithm”, PLOS ONE, vol. 11, no. 3, pp. 1-27, 2016.

[7] L. Chuang, Y. Lin, H. Chang, “An Improved PSO Algorithm for Generating Protective SNP Barcodes in Breast Cancer”, PLOS ONE, vol. 7, no. 5, pp. 1-9, 2012.

[8] C. Yang, L. Chuang, Y. Lin, “CMDR based differential evolution identifies the epistatic interaction in genome-wide association studies”, BIOINFORMATICS, vol. 33, no. 15, pp. 2354-2362, 2017.

[9] Y. Wang, X. Liu, K. Robbins, “AntEpiSeeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm”, BMC research notes, vol. 3, pp. 117-129, 2010.

[10] M. Xiang, J. Yu, Z. Yang, H. yu, H. He, “Probabilistic power flow with topology changes based on deep neural network”, INTERNATIONAL JOURNAL OF ELECTRICAL POWER & ENERGY SYSTEMS, vol. 117, pp. 1-13, 2020.

[11] F. Cristovao, S. Cascianelli, A. Canakoglu, M. Carman, L. Nanni, P. Pinoli, et al., “Investigating Deep Learning Based Breast Cancer Subtyping Using Pan-Cancer and Multi-Omic Data”, IEEE-ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, vol. 19, no. 1, pp. 121-134, 2022.

[12] X. Liu, J. Zhao, J. Li, B. Cao, Z. Lv, “Federated Neural Architecture Search for Medical Data Security”, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, vol. 18, no. 8, pp. 5628-5636, 2022.

[13] Z. Lin, A. Owen, R. Altman, “Genomic research and human subject privacy”, SCIENCE, vol. 305, no. 5681, pp. 183-183, 2004.

[14] M. Fredrikson, S. Jha, T. Ristenpart, “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures”, in the 22th ACM SIGSAC CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, pp. 1322-1333, 2015.

[15] W. Serrano, “Genetic and deep learning clusters based on neural networks for management decision structures”, NEURAL COMPUTING & APPLICATIONS, vol. 32, no. 9, pp. 4187-4211, 2020.

[16] Y. Zhang, B. Di, J. Lin, L. Song, “HetMEC: Heterogeneous Multi-Layer Mobile Edge Computing in the 6 G Era”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, vol. 69, no. 4, pp. 4379-4391, 2020.

[17] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, “Deep-Learning-Enhanced Multitarget Detection for End-Edge-Cloud Surveillance in Smart IoT”, IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12588-12596, 2021.

[18] X. Zhou, Y. Li, and W. Liang, “CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support”, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 3, pp. 912-921, 2021.

[19] S.Uppu, A. Krishna, “A deep hybrid model to detect multi-locus interacting SNPs in the presence of noise”, INTERNATIONAL JOURNAL OF MEDICAL INFORMATICS, vol. 119, pp. 134-151, 2018.

[20] X. Li, L. Liu, J. Zhou, C. Wang, “Heterogeneity Analysis and Diagnosis of Complex Diseases Based on Deep Learning Method”, SCIENTIFIC REPORTS, vol. 8, pp. 1-8, 2018.

[21] S.Uppu, A. Krishna, “Convolutional Model for Predicting SNP Interactions”, NEURAL INFORMATION PROCESSING, vol. 11305, pp. 127-137, 2018.

[22] X. Zhou, W. Liang, K. Wang, and L. T. Yang, “Deep Correlation Mining Based on Hierarchical Hybrid Networks for Heterogeneous Big Data Recommendations”, IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 171-178, 2021.

[23] L. Abdollahi, D. Gianola, F. Penagaricano, “Deep learning versus parametric and ensemble methods for genomic prediction of complex phenotypes”, GENETICS SELECTION EVOLUTION, vol. 52, no. 1, pp. 1-15, 2020.

[24] P. Bellot, G. delos Campos, M. Pérez-Enciso, “Can deep learning improve genomic prediction of complex human traits?”, Genetics, vol. 210, pp. 809-819, 2018.

[25] A. Chattopadhyay, T. Lu, “Gene-gene interaction: the curse of dimensionality”, ANNALS OF TRANSLATIONAL MEDICINE, vol. 7, no. 24, pp. 1-5, 2019.

[26] X. Wu, H. Wang, Y. Wei, Y. Mao, S. Jiang, “An Anonymous Data Publishing Framework for Streaming Genomic Data”, Medical Imaging and Health Informatics, vol. 8, no. 3, pp. 546–554, 2018.

[27] H. Wang, X. Wu, “IPP: An Intelligent Privacy-Preserving Scheme for Detecting Interactions in Genome Association Studies”, IEEE/ACM transactions on computational biology and bioinformatics, 2022.

[28] X. Wu, Y. Zhang, A. Ming, “MNSSp3: Medical big data privacy protection platform based on Internet of things”, Neural Computing & Application, 2020.

[29] X. Zhou, W. Liang, K. Wang and S. Shimizu, “Multi-Modality Behavioral Influence Analysis for Personalized Recommendations in Health Social Media Environment”, IEEE Transactions on Computational Social Systems, vol. 6, no. 5, pp. 888-897, 2019.

[30] J. Xia, W. Huang, H, Li, “Gradient-Based Differential Privacy Optimizer for Deep Learning Model Using Collaborative Training Mode”, in the 7th IEEE International Conference on Computer Science and Network Technology (ICCSNT), pp. 208-215, 2019.

[31] N. Gati, L. Yang, Z. Ren, “Differentially Private Tensor Deep Computation for Cyber-Physical-Social Systems”, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, vol. 8, no. 1, pp. 236-245, 2021.

[32] S. Chang, C. Li, “Privacy in Neural Network Learning: Threats and Countermeasures”, IEEE NETWORK, vol. 32, no. 4, pp. 61-67, 2018.

[33] M. Hao, H. Li, H. Yang, “Towards Efficient and Privacy-preserving Federated Deep Learning”, in the 2019 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS (ICC), 2019.

[34] C. Xu, J. Ren, K. Ren, “EdgeSanitizer: Locally Differentially Private Deep Inference at the Edge for Mobile Data Analytics”, IEEE INTERNET OF THINGS JOURNAL, vol. 6, no. 3, pp. 5140-5151, 2019.

[35] Y. Liu, J. Peng, Y. Wu, “PPGAN: Privacy-preserving Generative Adversarial Network”, in the 25th IEEE International Conference on Parallel and Distributed Systems (IEEE ICPADS), pp. 985-989, 2019.

[36] H. Cheng, P. Yu, Y. Chen, “Towards Decentralized Deep Learning with Differential Privacy”, in the International Conference on Cloud Computing (CLOUD) held as Part of the Services Conference Federation (SCF), pp.130-145, 2019.

[37] J. Yang, J. Wu, X. Wang, “Convolutional neural network based on differential privacy in exponential attenuation mode for image classification”, IET IMAGE PROCESSING, vol. 14, no. 15, pp. 3676-3681, 2020.

[38] N. Hoefer, S. Monroy, N. Abe, et. al, “Performance Evaluation of a Differentially-private Neural Network for Cloud Computing”, in the 2018 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), pp. 2545, 2018.

[39] Q. Zhang, J. Ma, J. Lou, “Broadening Differential Privacy for Deep Learning Against Model Inversion Attacks”, in the IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), pp. 1061-1070, 2020.

[40] Q. Wang, Z. Li, S. Wang, “Deep Domain Adaptation With Differential Privacy”, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, vol. 15, pp. 3039-3106, 2020.

[41] M. Gong, K. Pan, Z. Tang, “Preserving differential privacy in deep neural networks with relevance-based adaptive noise imposition”, NEURAL NETWORKS, vol. 125, pp. 131-141, 2020.

[42] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, ”Siamese Neural Network Based Few-Shot Learning for Anomaly Detection in Industrial Cyber-Physical Systems”, IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5790-5798, 2021.

[43] D. Tian, F. Yang, Y. Niu, “Loss function of SL (sekiguchi lesion) in the rice cultivar Minghui 86 leads to enhanced resistance to (hemi)biotrophic pathogens”, BMC Plant Biology, vol. 20, no. 1, pp. 507-517, 2020.

[44] T. Lin, P. Goyal, R. Girshick, et. al, “Focal Loss for Dense Object Detection”, IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 99, pp. 2999-3007, 2017.

[45] R. Urbanowicz, J. Kiralis, A. Sinnott, T. Heberling, J. Fisher, J. Moore, “Gametes: A fast, direct algorithm for generating pure, strict, epistatic models with random architectures”, Bio Data Mining, vol. 5, no. 16, pp. 5-16, 2012.

[46] Y. Zhang, J. Liu, “Bayesian inference of epistatic interactions in case-control studies”, Nature Genetics, vol. 39, no. 9, pp. 1167-1173, 2007.

[47] X. Wan, “Detecting two-locus associations allowing for interactions in genome-wide association studies”, Bioinformatics, vol. 26, pp. 2517–2525, 2010.

[48] S. Lin, K. Wang, Y. Wang, D. Zhou, “Universal Consistency of Deep Convolutional Neural Networks”, IEEE TRANSACTIONS ON INFORMATION THEORY, vol. 68, no. 7, pp. 4610-4617, 2022.

[49] M. Xie, J. Li, “Detecting genome-wide epistasis based on the clustering of relatively frequent items”, Bioinformatics, 2012.

[50] Cao, G. Yu, W. Ren, “DualWMDR: Detecting epistatic interaction with dual screening and multifactor dimensionality reduction”, Human Mutation, vol. 41, pp. 1-15, 2020.

[51] R. Klein, “Complement factor polymorphism in age-relatedmacular degeneration”, Science, vol. 308, pp. 385-389, 2005.

[52] D. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization”, Computer Science, 2014.

[53] M. Abadi, A. Chu, I. Goodfellow, H. Mcmahan, “Deep learning with differential privacy” in ACM SIGSAC conference on computer & communications security, 2016.

[54] H. Liu, Z. Wu, “Privacy-Preserving Data Aggregation Framework for Mobile Service Based Multiuser Collaboration”, INTERNATIONAL ARAB JOURNAL OF INFORMATION TECHNOLOGY, vol. 17, no. 4, pp. 450-460, 2020.

Yongting Zhang received the BMS degree in Nanjing University of Chinese Medicine Hanlin College, Taizhou, China, in 2019. She received the MM degree in Xuzhou Medical University, Xuzhou, China, in 2022. She is pursuing the Ph.D degree in Universiti Malaya. Her research interest includes privacy protection and information security.

Email: 301910911596@stu.xzhmu.edu.cn

Yonggang Gao received the B.S. degree in Xuzhou Medical University, Xuzhou, China, in 2019. He is pursuing the M.S. degree in medical informatics with Xuzhou Medical University, Xuzhou, China. His research interest includes medical information security.

Email: 300109110840@stu.xzhmu.edu.cn

Huanhuan Wang received the Master's degree in computer technology in Anhui University of Technology, in 2015. She is working toward PhD degree at China University of Mining and Technology. Her research interests include medical privacy protection, data mining methods in medical informatics.

Email: whh@xzhmu.edu.cn.com

Huaming Wu received the BE and MS degrees from Harbin Institute of Technology, China, in 2009 and 2011, respectively, both in electrical engineering, and the PhD degree with the highest honor in computer science from Freie Universitat Berlin, € Germany, in 2015. He is currently an assistant professor in the Center for Applied Mathematics, Tianjin University. His research interests include model-based evaluation, wireless and mobile network systems, mobile cloud computing, and deep learning. He is a member of the IEEE.
Email: whming@tju.edu.cn.

Youbing Xia received Ph.D in Philology of Traditional Chinese Medicine in Nanjing University of Chinese Medicine, Nanjing, China. He is currently the Secretary of the party committee of Xuzhou Medical University, China. His research interests include acupuncture treatment of infertility and study on acupuncture schools.
Email: 110403@njucm.edu.cn

Xiang Wu received the B.Eng. degree in Information Engineering, the M.S. and Ph.D. in communication and information system all in China University of Mining and Technology, Xuzhou, China, in 2007, 2010 and 2014, respectively. He is currently the deputy dean of the School of Medical Information and Engineering and the director of the Institute of Medical Information Security, Xuzhou Medical University, China. He is also a visiting professor and doctoral supervisor of Universiti Malaya. His research interests include privacy protection and information security.
Email: wuxiang@xzhmu.edu.cn