DeepCGP: A Deep Learning Method to Compress Genome-Wide Polymorphisms for Predicting Phenotype of Rice

Tanzila Islam, Chyon Hae Kim, Hiroyoshi Iwata, Hiroyuki Shimono, and Akio Kimura

Abstract—Genomic selection (GS) is expected to accelerate plant and animal breeding. During the last decade, genome-wide polymorphism data have increased, which has raised concerns about storage cost and computational time. Several individual studies have attempted to compress the genome data and predict phenotypes. However, compression models lack adequate quality of data after compression, and prediction models are time consuming and use original data to predict the phenotype. Therefore, a combined application of compression and genomic prediction modeling using deep learning could resolve these limitations. A **Deep** Learning **C**ompression-based **G**enomic **P**rediction (DeepCGP) model that can compress genome-wide polymorphism data and predict phenotypes of a target trait from compressed information was proposed. The DeepCGP model contained two parts: (i) an autoencoder model based on deep neural networks to compress genome-wide polymorphism data, and (ii) regression models based on random forests (RF), genomic best linear unbiased prediction (GBLUP), and Bayesian variable selection (BayesB) to predict phenotypes from compressed information. Two datasets with genome-wide marker genotypes and target trait phenotypes in rice were applied. The DeepCGP model obtained up to 99% prediction accuracy to the maximum for a trait after 98% compression. BayesB required extensive computational time among the three methods, and showed the highest accuracy; however, BayesB could only be used with compressed data. Overall, DeepCGP outperformed state-of-the-art methods in terms of both compression and prediction. Our code and data are available at https://github.com/tanzilamohita/DeepCGP.

Index Terms—Deep learning, autoencoder, genomic selection, data compression, genomic prediction

1 Introduction

By 2050, 70% more food production is required to keep pace with the expected increase in food demand and ongoing climate change on a global scale [1]. To achieve this challenge, we need to enhance genetic gains in plant breeding through novel technologies [2], [3]. One such technology is the use of genome-phenotype associations [4], [5], [6]. These include genome-wide association studies (GWAS) [7] and genomic selection (GS) [8]. In GWAS, candidate genes are discovered based on the associations and selecting SNPs has a strong impact on the trait. On the other hand, GS usually does not intend to select important SNPs but to predict genotypic values based on the whole SNPs. While selecting SNPs, the major issue would be correlation among SNPs (linkage disequilibrium). GS is expected to be effective in improving complex traits (e.g., crop yield) controlled by a large number of genes, which have been difficult to improve [9].

The use of genomic data is progressing in various fields, and a massive amount of genomic data has been generated [10] as a resource for plant breeding [6]. Furthermore, with the introduction of high-throughput sequencing technologies, the number of data samples also tends to be large, resulting in challenges for storage and analysis of genomic data in the fields of genomics, bioinformatics, and quantitative genetics [11]. Moreover, the increasing size and dimension of data [12] have led to an intensified need for data compression and compression-based data analysis. The ability to compress genomic data will not only make it easier to store and analyze data, but also aid in streamlining the exchange of data via Web APIs etc. [13], [14].

To effectively analyze high-dimensional data, deep learning (DL) techniques [15] have been introduced in various fields, including genomics, genetics, and breeding. Several DL methods exist [16], [17], [18], [19] that can compress genomic data without compromising model performance. Wang et al. introduced a single sequence based compression method DeepDNA to compress human mitochondrial genome data using hybrid convolutional and recurrent deep neural networks [20]. In DeepDNA, each of compressed

- *Tanzila Islam and Chyon Hae Kim are with the Department of Systems Innovation Engineering, Graduate School of Science and Engineering, Iwate University, Morioka, Iwate 020-8550, Japan. E-mail: tanzilamohita@gmail.com, tenkai@iwate-u.ac.jp.*
- *Hiroyoshi Iwata is with the Department of Agricultural and Environmental Biology, The University of Tokyo, Bunkyo, Tokyo 113-0033, Japan. E-mail: hiroiwata@g.ecc.u-tokyo.ac.jp.*
- *Hiroyuki Shimono is with the Crop Science Laboratory, Faculty of Agriculture, Iwate University, Morioka, Iwate 020-8550, Japan, and also with the Agri-Innovation Center, Iwate University, Morioka, Iwate 020-8550, Japan. E-mail: shimn@iwate-u.ac.jp.*
- *Akio Kimura is with the Department of Systems Innovation Engineering, Graduate School of Science and Engineering, Iwate University, Morioka, Iwate 020-8550, Japan, and also with the Agri-Innovation Center, Iwate University, Morioka, Iwate 020-8550, Japan. E-mail: kimura@cis.iwate-u.ac.jp.*

Manuscript received 10 December 2021; revised 6 October 2022; accepted 12 December 2022. Date of publication 5 January 2023; date of current version 5 June 2023.
This work was supported by the japan society for the promotion of science Grant in Aid for Scientific Research <KAKENHI> Under Grant Jp19H00938.
(Corresponding authors: Chyon Hae Kim and Hiroyoshi Iwata.)
This article has supplementary downloadable material available at https://doi.org/10.1109/TCBB.2022.3231466, provided by the authors.
Digital Object Identifier no. 10.1109/TCBB.2022.3231466

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

sequences can have different dimensions even though the sequences are originally in the same size. Goyal et al. introduced DeepZip, which used recurrent neural networks to compress single sequence based genomics and text data [21]. There have been few recent studies in compressing genomic data using a non-deep learning approach. In a recent paper, Yilmaz et al. introduced Macarons, which is a non-deep learning based SNP selection method that uses the correlations between SNPs to avoid the selection of redundant pairs of SNPs [22]. The SNP selection method of Macarons is fast, but it selects SNPs individually for each trait.

For GS, accurate prediction of phenotypes (strictly speaking, genotypic values) of a target trait is a central and recurring problem in quantitative genetics. Consequently, several genomic prediction methods have been proposed based on machine learning [23], [24], [25], [26], [27], [28] and quantitative genetic models, especially under a Bayesian paradigm [8], [29], [30], [31]. González et al. compared Bayes A and Bayesian LASSO with two machine learning algorithms (boosting and random forests [RFs]) to predict disease occurrence in simulated and real datasets [32]. Although the differences between the methods were small, RF outperformed other methods in most cases. Abdollahi-Arpanahi et al. compared the predictive performance of two deep learning methods (multilayer perceptron [MLP] and convolutional neural network [CNN]), two ensemble learning methods (RF and gradient boosting), and two parametric methods (genomic best linear unbiased prediction [GBLUP] and BayesB) using real and simulated datasets [33]. The authors pointed out that the predictive performance of deep learning methods was marginally better than that of parametric methods for large datasets.

Generally, previously proposed methods in the literature had the following limitations: (i) quality of information after the compression was uncertain, and (ii) original data was utilized for predictions using machine-learning methods. In contrast, in this study, the proposed method predicts phenotypes of target traits based on compressed genome-wide polymorphism data instead of original (i.e., uncompressed) data. Despite the compression of several cycles, the proposed method retains high-quality information, and the prediction accuracy of our method is similar to that of genomic prediction based on the original data, which quantifies the quality of our compression method. Furthermore, we used multiple autoencoder networks, in which the calculation cost of the network increased linearly with the number of genome-wide polymorphisms (i.e., the dimension of genomic data), whereas the calculation cost of other popular methods increased with square order, which is also another novel aspect of the proposed method (Supplementary Section S1, available online). To the best of our knowledge, there are no prediction methods that can predict the phenotypes of a target trait based on compressed genome-wide polymorphism data using Deep Learning in animal and plant breeding.

In this study, we developed a deep learning approach known as **Deep** Learning **C**ompression-based **G**enomic **P**rediction (DeepCGP) to compress high-dimensional genome-wide polymorphism data and predict phenotypes (estimated genotypic values) of rice agronomic traits from compressed information. DeepCGP consists of two models: (i) an autoencoder model to compress genome-wide polymorphism data; and (ii) a regression model to predict the phenotypes of a target trait based on compressed genome-wide marker data. To demonstrate the usage of DeepCGP, we used two different rice genome datasets, C7AIR, consisting of 7098 SNPs (single-nucleotide polymorphisms) and HDRA, consisting of 700000 SNPs. In this study, we demonstrated that DeepCGP could predict the phenotypes of a target trait based on compressed genome-wide polymorphism data, and achieved almost a similar accuracy to the prediction based on the original genome-wide polymorphism data. Additionally, we also compared the compression-based prediction performance of three genomic prediction methods (GBLUP, BayesB, and RF) to determine the general potential of compression-based genomic prediction over the methods for building a regression model.

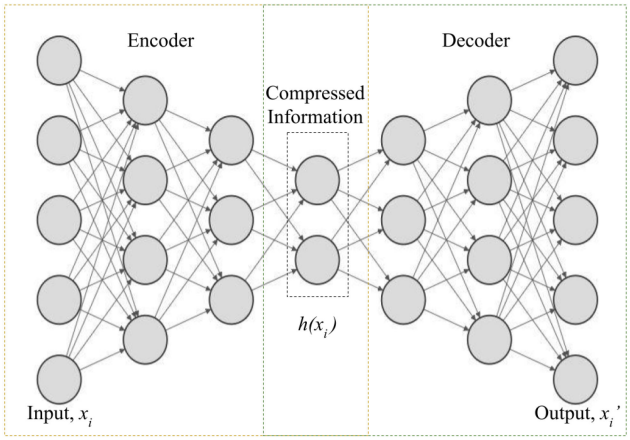

Fig. 1. Basic structure of a Deep Autoencoder.

2 Methodology

2.1 Deep Autoencoder

To compress genome-wide polymorphism data, we used a deep autoencoder [34], [35] (Fig. 1). This autoencoder is composed of two symmetrical deep belief networks with multiple hidden layers: (i) an encoder network $h(x_i)$, where $x_i \in R_d$, an autoencoder first encodes an input x_i to a hidden representation $h(x_i)^{(l+1)}$ based on Equation (1), and (ii) a decoder network x_i', that maps the hidden representation $h(x_i)^{(l+1)}$ back into a reconstruction ${x_i}'^{(l)}$ computed as in Equation (2):

$$h(x_i)^{(l+1)} = f\left(W^{(l)}x_i^{(l)} + b^{(l)}\right) \quad (1)$$

$${x_i}'^{(l)} = g\left(W'^{(l)}h(x_i)^{(l+1)} + b'^{(l)}\right) \quad (2)$$

where f is an en encoding activation function, $W^{(l)}$is an encoding weight matrix, $b^{(l)}$is an encoding bias vector, g is a decoding activation function, $W'^{(l)}$is a decoding matrix, and $b'^{(l)}$is a decoding bias vector of lth input layer to $l+1$th hidden layer.

The activation function of each layer except the middle layer and decoder layer is "ReLU" [36], which scales the negative output value to zero.

$$f(x) = max(x, 0) \quad (3)$$

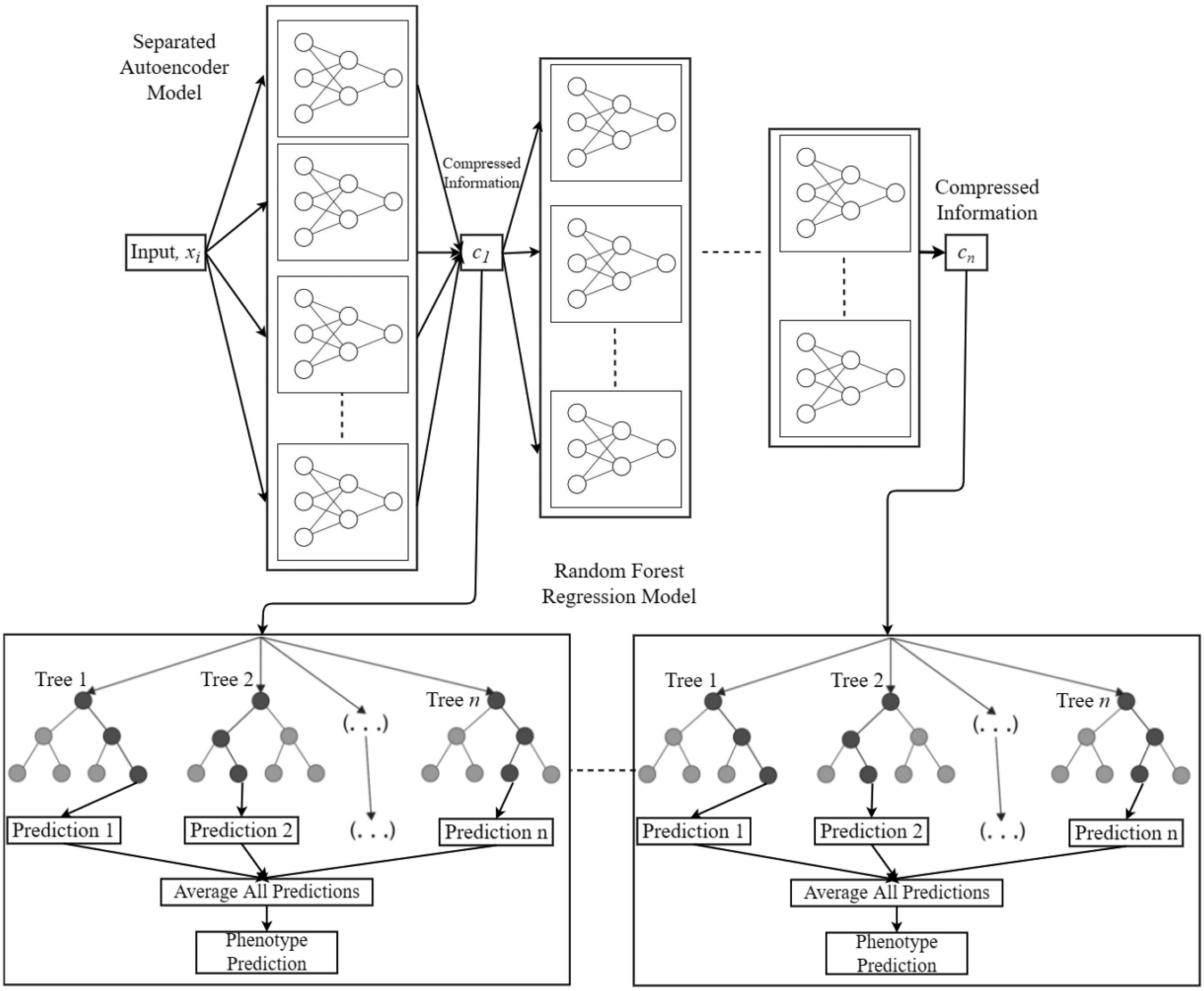

Fig. 2. DeepCGP architecture. The system consists of autoencoder models and a regression model (i.e., Random Forest [RF]). The autoencoder model compresses the genome-wide polymorphism data and the regression model (RF) predicts the phenotypes of traits based on compressed genome-wide polymorphism information.

The activation function of the middle layer and decoder layer is a "sigmoid" [36], which scales the output to the range [0, 1].

$$g(x) \;=\; 1/(1+e^{x}) \tag{4}$$

The reconstruction error was calculated as mean squared error (MSE) function, which is calculated as follows:

$$MSE(x,\; x') = \frac{1}{n}\sum_{i=1}^{n}\left(x_i \;- x_i'\right)^2 \tag{5}$$

where x_i and x_i' are the measured and predicted values, respectively, and n is the number of measured values with $i \;\in\; [1, n]$.

2.2 Overview of DeepCGP

DeepCGP (Fig. 2) model can compress genome-wide polymorphism data and use compressed information to predict the phenotype of rice.

DeepCGP consists of two models. The first is an autoencoder model that compresses genome-wide polymorphism data. The second is a regression model that takes the compressed information generated by the autoencoder as input and attempts to predict the genotypic values of a target trait. Regression models such as random forests (RF), GBLUP, BayesB, etc., can be used for prediction.

In the first model, our aim was to compress the genome-wide polymorphism data to the maximum limit. To achieve this, we generated several separated networks and trained the separated autoencoder models. The separated autoencoder models compressed the data, which was defined as Compress_1, c_1. To further compress the genome data, c_1 was used as the input, and the separated autoencoder models were trained for the second compression. The second compression was defined as Compress_2, c_2. In this manner, the separated autoencoder models can compress any genomic data.

After compressing, the regression model will be established to predict phenotypes of a target trait of genotypes (plants/lines). In this study, we used rice germplasm accessions.

In the present study, the models were trained in three steps:

Step 1: The separated autoencoder model was trained to optimize Equation (5) and to compress the genome-wide polymorphism data.

TABLE 1
Selected Traits Id and Name (HDRA Dataset)

Trait Id	Trait Name	Trait Id	Trait Name
1	Flowering time at Arkansas	10	Florets per panicle
2	Flag leaf length	11	Panicle fertility
3	Flag leaf width	12	Seed length
4	Awn presence	13	Seed width
5	Panicle number per plant	14	Seed length width ratio
6	Plant height	15	Blast resistance
7	Panicle length	16	Amylose content
8	Primary panicle branch number	17	Alkali spreading value
9	Seed number per panicle	18	Protein content

Step 2: The regression model mapped the compressed information to phenotypes of the target traits of rice germplasm accessions.

Step 3: After mapping each compression with phenotypes, we trained a regression model.

2.3 Datasets and Data Pre-Processing

In this study, we used two datasets of different sizes to demonstrate the broad applicability of our model. Based on these datasets, we evaluated our model using two metrics: (i) *how precisely we compress data*, where a better compression model is expected to minimize the loss of information in compression, and (ii) *how successfully the compressed genome-wide polymorphism can predict genotypic values of a target trait*, where we expect a prediction performance close to the original data.

C7AIR: The first dataset used was the Cornell-IR LD Rice Array (C7AIR) [37], which offers a second-generation SNP array containing 189 rice accessions for 7098 markers from the Rice Diversity project. The 189 lines had estimated genotypic values for plant height.

HDRA: The second dataset was the high-density rice array (HDRA) [38]. The HDRA dataset consisted of 1568 diverse inbred rice varieties with 700000 SNPs. Among these lines, the genotypic averages of 34 traits were estimated for 388 lines [39], with some missing records. As 29 genotypes had more than or equal to 10 missing data, while 359 had less than 10 missing data, we chose 359 lines in 18 traits (Table 1). Furthermore, the genotype dataset was formatted as a bed matrix in vcf format, in which each entry was scored as 0, 1, or 2, where 1 was identified as heterozygous. Since the accessions used in the study were all inbred lines and were expected to be homozygous in most SNPs, we considered 1 as a missing value and converted 0 and 2 to categorical values A (adenine), C (cytosine), G (guanine), and T (thymine). Subsequently, we saved the output in the csv format. Furthermore, we used the 'gaston' package [40] in R for this conversion.

We pre-processed categorical values (A, C, G, and T) for both datasets by applying one-hot encoding. In addition, all genomes were encoded into one-hot encoding using a 4-bit coding scheme; that is, $x \in R_{d\times 4}$, where d is the length of the genome sequence. "A," "C," "G," and "T" are encoded by "1000", "0100", "0010", and "0001", respectively. The C7AIR and HDRA dataset has ~13% and ~10% missing genotypes, respectively. Therefore, we encoded the missing values "N" by "0000" (Supplementary Fig. S2, available online).

After processing the raw data through one hot encoding, the dimensions of the C7AIR and HDRA data were 189 $\times$ 28392 and 1568 $\times$ 2800000, respectively. As the dimension of the input data was large, an input data splitting technique was applied, which reduced the computational time. We used the NumPy hsplit to split the one-hot encoded array horizontally (axis = 1, i.e., 28392 and 28,00000 for C7AIR and HDRA, respectively). For C7AIR and HDRA, each split contained 189 $\times$ 28 and 1568 $\times$ 28 of data, respectively, that is, an input layer with 28 neurons in each network. Moreover, 1014 and 100000 separated autoencoder networks were employed for the C7AIR and HDRA datasets, respectively (Supplementary Fig. S3, available online).

2.4 Implementation for Compression Modeling

An autoencoder model was utilized to compress the genome-wide polymorphism data. Each dataset was divided into training (60%), testing (20%), and validation sets (20%) using the scikit-learn 'train_test_split' library. To achieve the optimum performance of a compression model, for both datasets, we executed Keras wrapper class 'KerasRegressor', which permitted us to tune hyperparameters (Table 2) using scikit-learn's 'RandomizedSearchCV'. Since the dimension of the C7AIR dataset is low, we tuned the hyperparameters on a whole dataset. For the HDRA dataset, we tuned the hyperparameters on a small subset of training data i.e., first 1000 splits of data where each split contained (1568 $\times$ 28).

For the C7AIR genotype data, the selected model had three hidden layers in both the encoder and decoder networks. In Compress_1, the input layer of a network has 28 nodes, the first hidden layer has 14 nodes, the second hidden layer has 7 nodes, with a code size of 3. For further compressing the data, the first compression data (c_1) was used as an input in Compress_2. In Compress_2, the input layer of a network has 36 nodes, the first hidden layer has 28 nodes, and the second hidden layer has 10 nodes, with a code size of 5. Both compressions were trained with the Adam optimizer using a learning rate of 0.001. ReLU activation was applied to all layers of the encoder and decoder, except the middle and last layers, for which we applied the sigmoid activation function. The model was trained with MSE loss, and the mini-batch size was 52 for Compress_1 and 32 for Compress_2. The epochs were set to 200 for both the compressions.

TABLE 2
Hyperparameters Determined for C7AIR and HDRA Datasets

Hyperparameters	C7AIR		HDRA		
	Compress_1	Compress_2	Compress_1	Compress_2	Compress_3
Neurons	28, 14, 7, 3	36, 28, 10, 5	28, 14, 7, 3	30, 15, 5	25, 14, 5
Batch Size	52	32	52	32	32
Epochs	200	200	200	100	150

Learning rate, batch size, and loss function hyperparameters were considered by tuning the values from one range above and below the default range provided in TensorFlow. An optimum learning rate was searched for the Adam optimizer from often-used logarithmic scale values of {0.01, 0.001, 0.0001}. Furthermore, we experimented with batch sizes {16, 32, 52, 64} and searched for minimum loss from mean squared error, binary cross-entropy, and mean absolute error loss.

The architecture selected for HDRA genotype data was very similar, except for the number of compressions, training epochs, and network structure. The data was compressed to the greatest extent as the HDRA dataset had very high dimensions. The number of nodes in each layer was [28, 14, 7, 3], [30, 15, 5], and [25, 14, 5] for Compress_1, Compress_2, and Compress_3, respectively. Compress_1 was trained with 200 epochs and 52 batch sizes, Compress_2 with 100 epochs and 32 batch sizes, and Compress_3 with 150 epochs and 32 batch sizes. Moreover, the remaining parameters were the same as those for the C7AIR network.

The compression model was implemented using Keras functional API [41], which is written in Python and built on top of Tensorflow.

2.5 Random Forests (RF)

In the present study, random forests (RF) [42], [43] were used to predict the phenotypes of a target trait. RF is an ensemble machine learning algorithm consisting of individual decision trees. RF is often a collection of hundreds to thousands of trees, where each tree is built using a bootstrap sample of the original data. The final random forest predictor is computed by averaging the tree predictors over trees, which does not include the given observation in the bootstrap sample. Each tree minimizes the average mean squared generalization error or predictive error, which is used to assess the predictive accuracy. The construction of the RF algorithm can be described in the following steps [44]:

1. Draw $ntree$ bootstrap samples from the original or compressed marker scores.
2. Grow a random forest tree T_b for each bootstrap data set. At each node:
   i. Randomly select $mtry$ variables for splitting.
   ii. Grow the tree so that each terminal node has no fewer than the node size cases.
3. Aggregate the prediction from each tree for prediction by majority voting and assembling the output of trees $\{T_b\}_1^B$.

An RF can be mathematically expressed as:

$$y_i' = \frac{1}{B}\sum_{b=1}^{B} T_b(x_i) \tag{6}$$

where each predictor $T_b(x_i)$ is a decision tree [45] constructed with a bootstrapped sample B of the marker genotype score (or the compressed score) x_i at iteration b(for $b = 1, \ldots\ldots, B$ bootstrap samples).

2.6 GBLUP and BayesB

Moreover, we used GBLUP and BayesB as the commonly used Bayesian regression methods for genomic prediction. The GBLUP model equation is:

$$y = 1\mu + Wu + e \tag{7}$$

where $\boldsymbol{y}$ is the vector of the phenotypes of a target trait, μ is the grand mean, 1 is a vector of ones (all-ones vector), u is the vector of estimated genotypic values, W is the design matrix that relates the genotypic values to samples (i.e., varieties/lines), and e is the vector of residual errors. In this study, we only had one phenotypic record for each variety/line, W is an identity matrix of size n, where n is the number of varieties/lines. The vector u is assumed to follow a multivariate normal distribution $u \sim N(0, G\sigma_g^2)$, where 0 is a vector of zeros (all-zero vector), σ_g^2 is the genetic variance explained by genome-wide polymorphisms, and G is the genomic relationship matrix calculated as ZZ' / m, where Z is the matrix of original or compressed marker scores, and m is the dimension of the original or compressed marker scores. Each column of the matrix of marker scores Z, is scaled to have mean 0 and variance 1 prior to the calculation of G.

The model equation of BayesB is:

$$y = 1\mu + Xa + e \tag{8}$$

where X is the matrix of unscaled original or compressed marker scores, and a is the vector of the original and compressed marker effects. When the marker scores are uncompressed, each element of X represents SNP genotypes, where 0 represents the homozygous genotype of the reference allele and 1 represents the homozygous genotype of the non-reference allele. When the marker scores are uncompressed, each element of X take values of 0 or 1 according to the compressed data. The prior distribution of a marker effect a_k (k-th element of a) is assumed to follow a normal distribution with zero mean and marker specific uncertainty variance $\sigma_{a_k}^2$, and the variance $\sigma_{a_k}^2$ is assumed to follow the same scaled inverse chi-square distribution. A detailed explanation of the BayesB model can be found in [29], [31].

2.7. Implementation for Prediction Modeling

In the present study, three prediction models, such as RF, GBLUP, and BayesB were used. In addition, we used compressed information as input and extracted the compressed data as a matrix from each dataset. Furthermore, we prepared the estimated genotypic values of a target trait

TABLE 3
Compression Analysis for C7AIR and HDRA Datasets

	C7AIR (189, 7098)		HDRA (1568, 700000)		
	Compress_1	Compress_2	Compress_1	Compress_2	Compress_3
Dimension	(189, 3042)	(189, 425)	(1568, 300000)	(1568, 50000)	(1568, 10000)
Training Time	01h:25m:54s	00h:10m:33s	10d:20h:3m:28s	23h:30m:25s	6h:34m:40s
MSE Loss	0.051	0.039	0.0156	0.0749	0.0911
Compression Ratio	57.14%	94.01%	57.14%	92.86%	98.57%

We calculated MSE loss for each autoencoder and then showed the average MSE loss after each compression. We compressed the C7AIR data up to approximately 94.01% and HDRA data up to approximately 98.57%. The compression levels of our model can be adjusted depending on storage requirements3.2 Prediction of phenotypes based on the compressed data

omitting missing entries and arranged them in the same order as the compressed data. A prediction model for each trait was built separately. To evaluate the accuracy of the prediction models and to compare the accuracy among different compression levels, 10-fold cross-validation with five repetitions were applied, and the results were averaged. We used the same folds for all compression levels to ensure that the results were directly comparable. Furthermore, the prediction ability using the correlation coefficient between the estimated and predicted genotypic values was evaluated. Moreover, we evaluated the accuracy of a prediction model based on the original uncompressed genome-wide polymorphism data. Before building the prediction model, we processed the original uncompressed data converting A,T, G,C to 0 and 1 and NA to average values of 0s and 1s, respectively.

A RF model was implemented using the 'ranger' R package [46], which is the fastest and most memory-efficient package to analyze high-dimensional data [42]. To train the RF model, we used default parameter settings of the 'ranger' function (num.trees: 500, mtry: square root of the number of tuning hyperparameters). To implement GBLUP and BayesB, we used the 'BGLR' package [47] in the R language. The MCMC (Markov Chain Monte Carlo) was run for 25000 iterations with a 5000 burn-in period for both GBLUP and BayesB.

All experiments in this study were conducted on a PC with an Intel(R) Core (TM) i9-10980XE, 3.00 GHz CPU, 128 GB RAM, GPU RTX 3090, and a 64 bit Windows 10 pro operating system.

3 Results

3.1 Compression of Genome-Wide Polymorphism Data

The first experiment in this study was aimed to demonstrate the compression ability of DeepCGP for genome-wide polymorphism data. C7AIR and HDRA datasets were used to train separated stacked autoencoders and evaluate the model by calculating the training time and information loss. Furthermore, the compression ratios were calculated for both datasets; compression ratio is defined as the dimension reduction relative to the uncompressed size, and is given as follows:

$$Compression\ Ratio\ =\ 1 - h/x\ \times 100 \tag{9}$$

where h is the dimension after compression and x is the dimension before compression. Table 3 lists the dimensions of the compressed data, training time, MSE loss, and compression ratio for the C7AIR and HDRA datasets.

3.2 Prediction of Phenotypes Based on the Compressed Data

To evaluate the accuracy of the models and to investigate the compressed data, the prediction models were fitted to the compressed data. Fig. 3A and 3B shows the prediction accuracy of RF for different compression levels, including non-compression for both datasets. We considered the compression level according to the compression ratio percentage, which was 0% (original uncompressed data), 57% (57.14%), and 94% (94.01%) for the C7AIR dataset and 0% (original uncompressed data), 57% (57.14%), and 98% (98.57%) for the HDRA dataset. For the C7AIR dataset (Fig. 3A) an accuracy similar to that of the original data (with an average difference of approximately less than 3%) was attained even at 94% compression. For the HDRA dataset (Fig. 3B), the accuracy obtained outperformed that of the original data after 98% compression (with an average difference of approximately 5%) for all the selected 18 traits (Table 1). Moreover, DeepCGP could successfully predict phenotypes even after high-level compression.

The predictive performance was compared between RF and two quantitative genetic models, BayesB [8] and GBLUP [48] (Supplementary Tables S1 and S2, available online). Both models are commonly used in genomic prediction; Figs. 4A and 4B display the predictive performance of BayesB, GBLUP, and RF for the C7AIR and HDRA datasets.

We evaluated the predictive performance of the original uncompressed data to the compressed data for both datasets. In the C7AIR dataset, the largest predictive performance was achieved by RF (0.72), followed by GBLUP (0.68) and BayesB (0.67), despite 94% compression. Contrarily, after 98% compression, the largest predictive performance of the HDRA dataset was delivered by BayesB (0.64) followed by GBLUP (0.63) and RF (0.60). The results suggest that RF yielded the highest accuracy of prediction for both original uncompressed data and compressed data of low-dimensional datasets (i.e., C7AIR). In contrast, it is difficult to apply BayesB to a high-dimensional original uncompressed dataset (i.e., HDRA) owing to computational requirements. Therefore, we avoided calculating the prediction accuracy for the original uncompressed HDRA dataset, which is considered as N/A in Fig. 4B. However, BayesB was applied to compressed HDRA

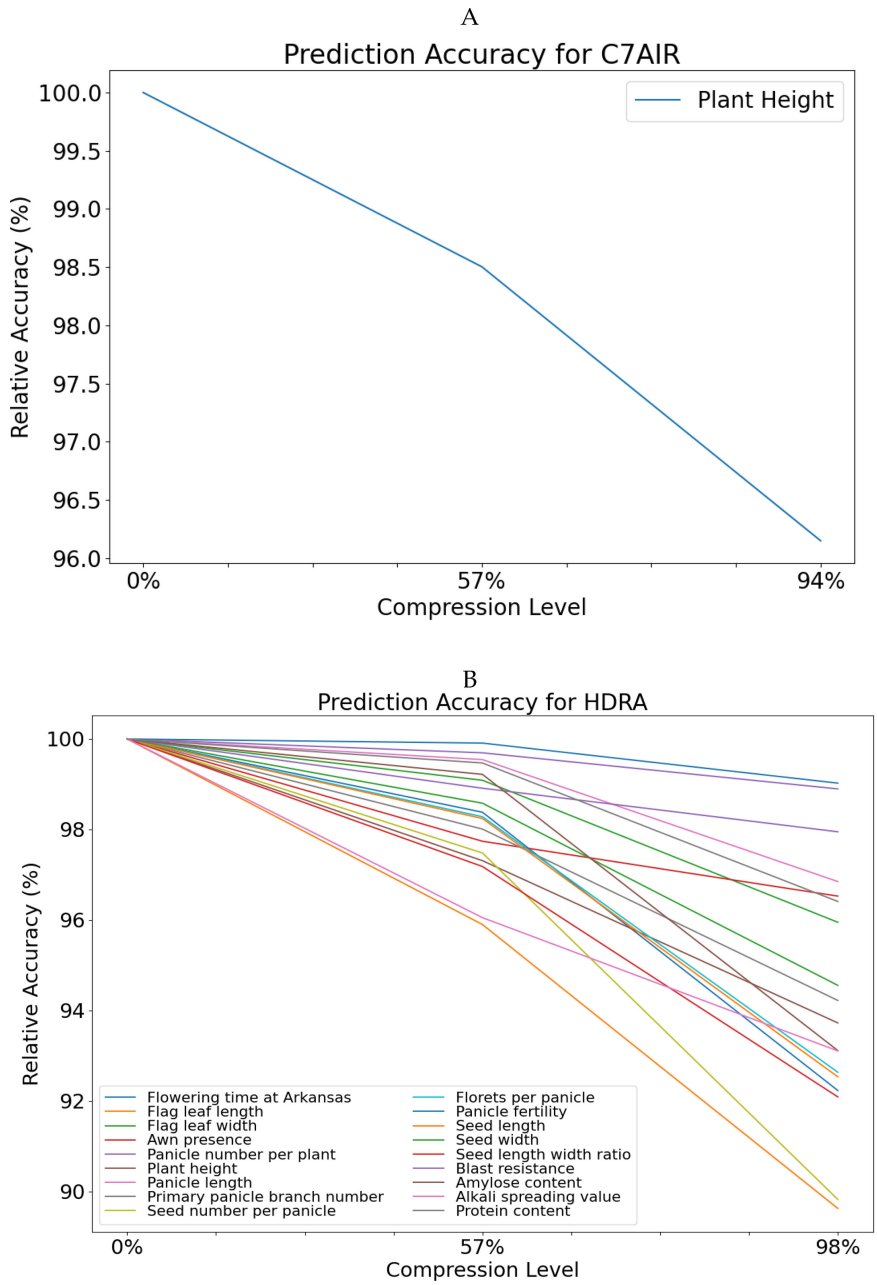

Fig. 3. RF prediction relative accuracy (A) C7AIR (B) HDRA datasets.

data, and its prediction accuracy outperformed both GBLUP and RF. After 98% compression, the predictive performances of BayesB model were 0.01 and 0.04 higher than that of GBLUP and RF, respectively.

Figs. 5A, 5B, and 5C show the prediction accuracies of RF, GBLUP and BayesB models, respectively, for the selected 18 traits (Table 1) of the HDRA dataset. After 98% compression, the predictive accuracy of trait id 16 was higher than that of low compression levels for GBLUP. The prediction times for RF, GBLUP, and BayesB were shorter at higher compression levels (Table 4). RF demonstrates the lowest time for both datasets at all compression levels compared to the other methods. For the HDRA dataset, BayesB takes a longer time for predicting even after applying compression; BayesB cannot be applied to the original data (i.e., 0%) owing to computational requirements, hence it is considered as N/A.

3.3 Compare With Other Compression Methods

We compared the compression performance of DeepCGP with Macarons which is a SNP selection method that takes into account the correlations between SNPs to avoid the selection of redundant pairs of SNPs. For comparing Macarons with our method DeepCGP, first, we selected SNPs using Macarons by setting the k values to 300000 (57%), 50000 (93%) and 10000 (98%). Then, we predicted the accuracy of phenotype using the Random Forest regression method. For predicting the phenotype, we used the same cross validation id as of DeepCGP to ensure that the results are directly comparable.

Fig. 6 shows the prediction performance of DeepCGP and Macarons. The methods are compared for three different levels of compressions (57%, 93% and 98%) for the HDRA dataset. The y-axis shows the averaged prediction

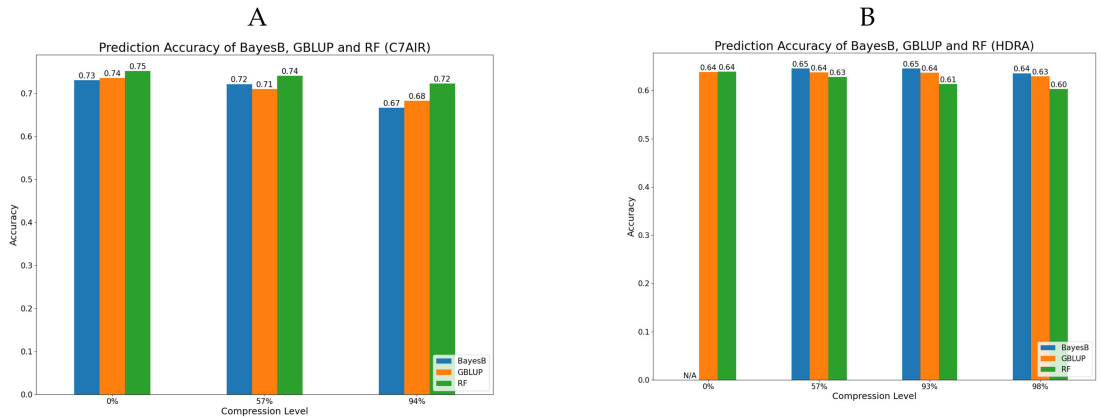

Fig. 4. Comparison of BayesB, GBLUP, and RF model prediction accuracies (A) C7AIR (B) HDRA datasets.

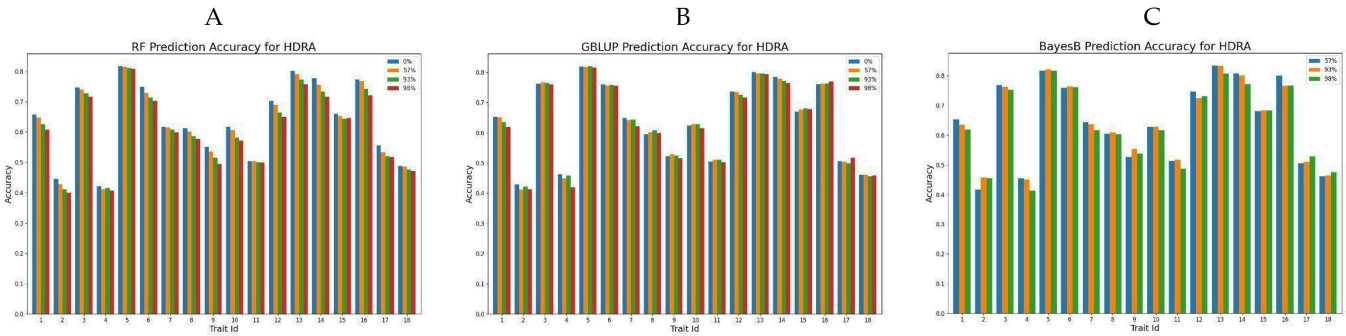

Fig. 5. Prediction accuracy of HDRA dataset (A) RF (B) GBLUP and (C) BayesB.

TABLE 4
Prediction Times of RF, BayesB, and GBLUP for C7AIR and HDRA Datasets

Methods	C7AIR			HDRA			
	0%	57%	94%	0%	57%	93%	98%
RF	9.26s	5.36s	4.29s	01h:53m: 50s	55m:24s	12m:39s	3m:31s
GBLUP	38.79s	41.1s	37.5s	2h:16m: 37s	2h:13m: 5s	2h:12m: 34s	2h:12m: 21s
BayesB	12m:43s	3m:29s	49.03s	N/A	7d:5h: 1m:31s (Average time per trait)	1d:21h: 39m:43s	08h:57m: 20s

accuracy across all the 18 traits (Supplementary Fig. S4, S5, available online). Although the selection method of Macarons is fast, the downside of this approach is to select SNPs for each trait. On the other hand, in DeepCGP, we can compress the data for all the traits using a single task. And the prediction accuracy of our deep learning based method

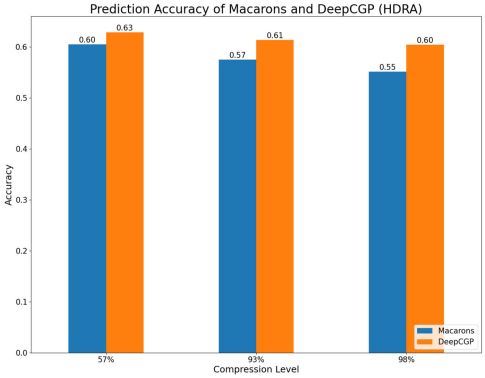

Fig. 6. Comparison of prediction performance with Macarons.

DeepCGP is higher than Macarons, which proved that a deep learning based compression method would be better able to learn meaningful information compared to non-deep learning based compression method.

4 Discussion

High dimensional genome-wide polymorphism data are extensively utilized for plant and animal breeding; this necessitates for the development of innovative platforms that can considerably reduce the resources required for storage and processing. Studies have shown that the intrinsic biological patterns found in genomic data provide a unique opportunity for researchers to compress high-dimensional genome-wide polymorphism data. Several individual studies have been conducted to compress the genome data and predict phenotypes. However, in most studies, there is uncertainty regarding the quality of data after compression and compressed data are not used during the prediction method. For instance, a fast reference-free genome compression method [16] used an autoencoder to compress genome

data, which could maintain the compression ratio at an acceptable level, while reducing the compression time for a small part of the gene. However, they did not include any information about the quality of the data obtained after compression. In contrast, the proposed method in this study was scalable for high-dimensional data owing to its design that uses a large number of autoencoders in parallel and iteratively, and retains high-quality information from compressed data that can be used for any kind of data analysis instead of the original data. Montesinos-López et al. suggested that DL prediction performance was higher for high-density data sets when compared to conventional genomic prediction models [49]. Li et al. provided an integrated framework to conduct GWAS and GS in crops, with an environmental dimension that enhanced prediction performance in breeding for future climates [50]. However, to the best of our knowledge, to date no research has been conducted on the combined application of a compression and a prediction model. In this study, we developed a DL based compression-based genomic prediction model, DeepCGP, which substantially improved breeding and crop yields while providing a considerable reduction in storage requirements related to DNA sequence data.

The most prominent advantage of using DL for compression is its ability to learn meaningful information from the underlying genetic architecture. This method is capable of modeling complex patterns with less intense computer requirements than other algorithms. The experimental results obtained in this study are extremely promising as we were able to provide phenotype predictions by evaluating the robustness of compressed data. The compression levels of our model can be adjusted depending on the storage requirements or prediction accuracy level. In addition, we investigated the predictive performance of three popular prediction methods, RF, BayesB, and GBLUP, to evaluate the potential of compression-based analysis. The results showed that the predictive performance of BayesB was slightly higher than that of GBLUP and RF. However, application of BayesB to the original uncompressed HDRA data was not possible, as the method was extremely time-consuming for analyzing high-dimensional data (Table 4). For this reason, it is important to compress high-dimensional genomic data to apply methods, such as the BayesB method. Furthermore, it is important to compress data to address the computational challenges for managing large-scale genomic data, including storage, processing, complex data analyses, visualization, retrieval, and sharing [51]. Transporting large genome-wide marker data from one database to another (via the Internet), and sharing data among multiple databases using API (e.g., Breeding APIs BrAPI) [13] requires transportation efficiency as well as computational efficiency. These can be achieved by compressing the genome-wide marker data.

In addition, Deep Learning is still on the way of improvement and currently is not suitable to make suggestions for SNP sets. In another word, finding SNP sets using Deep Learning can be an important and a large research theme although we did not try it in this paper. Future work includes analyzing gradients on each element of the neural network that predicts phenotypes from SNP data.

A potential limitation of our approach is that we used diverse rice germplasm data to predict phenotype from the compressed data. We have not yet conducted experiments to different datasets such as soybean or human genome data. Hence, researchers have to use this new method with caution as DeepCGP's information loss can occur when applying it to the other datasets.

5 Conclusion

In conclusion, a novel deep learning model DeepCGP as a new paradigm was introduced to compress genome-wide polymorphism data that successfully predicts phenotype from the compressed information. DeepCGP methodology can potentially consider complex modeling into account. For example, lower-dimensional compressed data allow us to explicitly include interactions among polymorphisms (epistasis) in BayesB owing to the smaller dimensions (i.e., a smaller variable number) of the compressed data. Another novelty of the proposed method is that it provides a combinatorial application using DL for genomic prediction, which may substantially improve the computational efficiency of DL by using compressed data as input variables. The proposed method also provides a strong alternative for compressing high-dimensional genomic data and predicts phenotypes from compressed data, which is beneficial for saving storage as well as computational time.

References

[1] T. He and C. Li, "Harness the power of genomic selection and the potential of germplasm in crop breeding for global food security in the era with rapid climate change," *Crop J*, vol. 20, no. 5, pp. 688–700, Oct. 2020, doi: 10.1016/j.cj.2020.04.005.

[2] M. Tester and P. Langridge, "Breeding technologies to increase crop production in a changing world," *Science*, vol. 327, no. 5967, pp. 818–822, Feb. 2010, doi: 10.1126/science.1183700.

[3] M. Qaim, "Role of new plant breeding technologies for food security and sustainable agricultural development," *Appl. Econ. Perspectives Policy*, vol. 42, pp. 129–150, Apr. 2020, doi: 10.1002/aepp.13044.

[4] M. T. Hamblin, E. S. Buckler, and J.-L. Jannink, "Population genetics of genomics-based crop improvement methods," *Trends Genet*, vol. 27, no. 3, pp. 98–106, Mar. 2011, doi: 10.1016/j.tig.2010.12.003.

[5] C. Kole et al., "Application of genomics-assisted breeding for generation of climate resilient crops: Progress and prospects," *Front. Plant Sci.*, vol. 6, Aug. 2015, Art. no. 563, doi: 10.3389/fpls.2015.00563.

[6] M. Thudi et al., "Genomic resources in plant breeding for sustainable agriculture," *J. Plant Physiol.*, vol. 257, Feb. 2021, Art. no. 153351, doi: 10.1016/j.jplph.2020.153351.

[7] P. K. Gupta, P. L. Kulwal, and V. Jaiswal, "Association mapping in crop plants: Opportunities and challenges," *Adv. Genet.*, vol. 85, pp. 109–147, 2014, doi: 10.1016/B978-0-12-800271-1.00002-0.

[8] T. H. Meuwissen, B. J. Hayes, and M. E. Goddard, "Prediction of total genetic value using genome-wide dense marker maps," *Genetics*, vol. 157, no. 4, pp. 1819–1829, Apr. 2001, doi: 10.1093/genetics/157.4.1819.

[9] J.-L. Jannink, A. J. Lorenz, and H. Iwata, "Genomic selection in plant breeding: From theory to practice," *Brief. Funct. Genomic.*, vol. 9, no. 2, pp. 166–177, Mar. 2010, doi: 10.1093/bfgp/elq001.

[10] Z. D. Stephens et al., "Big data: Astronomical or genomical?," *PLoS Biol.*, vol. 13, no. 7, Jul. 2015, Art. no. e1002195, doi: 10.1371/journal.pbio.1002195.

[11] J. Fan, F. Han, and H. Liu, "Challenges of Big Data analysis," *Nat. Sci Rev.*, vol. 1, no. 2, pp. 293–314, Jun. 2014, doi: 10.1093/nsr/nwt032.

[12] R. Bhukya, S. Yadav, J. K. Sharma, B. lal Sharma, and A. Kumar, "Compression for DNA sequences using huffman encoding," in *Information and Communication Technology For Sustainable Development*, M. Tuba, S. Akashe, and A. Joshi, Eds. Singapore: Springer, 2020, pp. 615–624.

[13] P. Selby et al., "BrAPI-an application programming interface for plant breeding applications," *Bioinformatics*, vol. 35, no. 20, pp. 4147–4155, Oct. 2019, doi: 10.1093/bioinformatics/btz190.

[14] R. Swaminathan, Y. Huang, S. Moosavinasab, R. Buckley, C. W. Bartlett, and S. M. Lin, "A review on genomics apis," *Comput. Struct. Biotechnol. J.*, vol. 14, pp. 8–15, 2016, doi: 10.1016/j.csbj.2015.10.004.
[15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
[16] Z. N. Absardi and R. Javidan, "A fast reference-free genome compression using deep neural networks," in *Proc. Big Data Knowl. Control Syst. Eng.*, 2019, pp. 1–7, doi: 10.1109/BdKCSE48644.2019.9010661.
[17] R. Wang, T. Zang, and Y. Wang, "Human mitochondrial genome compression using machine learning techniques," *Hum. Genomic.*, vol. 13, no. Suppl 1, Oct. 2019, Art. no. 49, doi: 10.1186/s40246-019-0225-3.
[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.
[19] M. Silva, D. Pratas, and A. J. Pinho, "Efficient DNA sequence compression with neural networks," *Gigascience*, vol. 9, no. 11, Nov. 2020, Art. no. giaa119, doi: 10.1093/gigascience/giaa119.
[20] R. Wang et al., "DeepDNA: A hybrid convolutional and recurrent neural network for compressing human mitochondrial genomes," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2018, pp. 270–274, doi: 10.1109/BIBM.2018.8621140.
[21] M. Goyal, K. Tatwawadi, S. Chandak, and I. Ochoa, "Deepzip: Lossless data compression using recurrent neural networks," in *Proc. Data Compression Conf.*, 2019, pp. 575–575, doi: 10.1109/DCC.2019.00087.
[22] S. Yilmaz, M. Fakhouri, M. Koyutürk, A. E. Çiçek, and O. Tastan, "Uncovering complementary sets of variants for predicting quantitative phenotypes," *Bioinformatics*, Dec. 2021, doi: 10.1093/bioinformatics/btab803.
[23] S. Szymczak et al., "Machine learning in genome-wide association studies," *Genet. Epidemiol.*, vol. 33, Suppl 1, pp. S51–S57, 2009, doi: 10.1002/gepi.20473.
[24] 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, May 2011, Art. no. S11, doi: 10.1186/1753-6561-5-S3-S11.
[25] J. O. Ogutu, T. Schulz-Streeck, and H.-P. Piepho, "Genomic selection using regularized linear regression models: Ridge regression, lasso, elastic net and their extensions," *BMC Proc.*, vol. 6, May 2012, Art. no. S10, doi: 10.1186/1753-6561-6-S2-S10.
[26] S. Okser, T. Pahikkala, A. Airola, T. Salakoski, S. Ripatti, and T. Aittokallio, "Regularized machine learning in the genetic prediction of complex traits," *PLoS Genet.*, vol. 10, no. 11, Nov. 2014, Art. no. e1004754, doi: 10.1371/journal.pgen.1004754.
[27] O. González-Recio, G. J. M. Rosa, and D. Gianola, "Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits," *Livest. Sci.*, vol. 166, pp. 217–231, Aug. 2014, doi: 10.1016/j.livsci.2014.05.036.
[28] M. Blondel, A. Onogi, H. Iwata, and N. Ueda, "A ranking approach to genomic selection," *PLoS ONE*, vol. 10, no. 6, Jun. 2015, Art. no. e0128570, doi: 10.1371/journal.pone.0128570.
[29] D. Gianola, G. de los Campos, W. G. Hill, E. Manfredi, and R. Fernando, "Additive genetic variability and the Bayesian alphabet," *Genetics*, vol. 183, no. 1, pp. 347–363, Sep. 2009, doi: 10.1534/genetics.109.103952.
[30] D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick, "Extension of the bayesian alphabet for genomic selection," *BMC Bioinf.*, vol. 12, May 2011, Art. no. 186, doi: 10.1186/1471-2105-12-186.
[31] D. Gianola, "Priors in whole-genome regression: The bayesian alphabet returns," *Genetics*, vol. 194, no. 3, pp. 573–596, Jul. 2013, doi: 10.1534/genetics.113.151753.
[32] O. González-Recio and S. Forni, "Genome-wide prediction of discrete traits using Bayesian regressions and machine learning," *Genet. Sel. Evol.*, vol. 43, Feb. 2011, Art. no. 7, doi: 10.1186/1297-9686-43-7.
[33] R. Abdollahi-Arpanahi, D. Gianola, and F. Peñagaricano, "Deep learning versus parametric and ensemble methods for genomic prediction of complex phenotypes," *Genet. Sel. Evol.*, vol. 52, no. 1, Feb. 2020, Art. no. 12, doi: 10.1186/s12711-020-00531-z.
[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series), Illustrated*. Cambridge, MA, USA: The MIT Press, 2016.
[35] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: 10.1002/aic.690370209.
[36] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*, 1st ed.. Sebastopol, CA, USA: O'Reilly Media, 2017.
[37] K. Y. Morales et al., "An improved 7K SNP array, the C7AIR, provides a wealth of validated SNP markers for rice breeding and genetics studies," *PLoS ONE*, vol. 15, no. 5, May 2020, Art. no. e0232479, doi: 10.1371/journal.pone.0232479.
[38] S. R. McCouch et al., "Open access resources for genome-wide association mapping in rice," *Nat. Commun.*, vol. 7, Feb. 2016, Art. no. 10532, doi: 10.1038/ncomms10532.
[39] K. Zhao et al., "Genome-wide association mapping reveals a rich genetic architecture of complex traits in Oryza sativa," *Nat. Commun.*, vol. 2, Sep. 2011, Art. no. 467, doi: 10.1038/ncomms1467.
[40] H. Perdry and C. Dandine-Roulland, "gaston: Genetic data handling (QC, GRM, LD, PCA) & linear mixed models," R package version 1.5.4, 2018. [Online]. Available: https://CRAN.R-project.org/package=gaston
[41] N. Ketkar, "Introduction to Keras," in *Deep Learning with Python*, Berkeley, CA, USA: Apress, 2017, pp. 97–111.
[42] L. Breiman, *Random Forests*, Berlin, Germany: Springer, 2001, doi: 10.1023/a:1010933404324.
[43] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, pp. 18–22, 2002. [Online]. Available: http://CRAN.R-project.org/doc/Rnews/
[44] X. Chen and H. Ishwaran, "Random forests for genomic data analysis," *Genomics*, vol. 99, no. 6, pp. 323–329, Jun. 2012, doi: 10.1016/j.ygeno.2012.04.003.
[45] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2016.
[46] M. N. Wright and A. Ziegler, "ranger : A fast implementation of random forests for high dimensional data in*C++* and *R*," *J. Statist. Softw.*, vol. 77, no. 1, pp. 1–17, 2017, doi: 10.18637/jss.v077.i01.
[47] P. Pérez and G. de los Campos, "Genome-wide regression and prediction with the BGLR statistical package," *Genetics*, vol. 198, no. 2, pp. 483–495, Oct. 2014, doi: 10.1534/genetics.114.164442.
[48] P. M. VanRaden, "Efficient methods to compute genomic predictions," *J. Dairy Sci.*, vol. 91, no. 11, pp. 4414–4423, Nov. 2008, doi: 10.3168/jds.2007-0980.
[49] O. A. Montesinos-López et al., "A review of deep learning applications for genomic selection," *BMC Genomic.*, vol. 22, no. 1, Jan. 2021, Art. no. 19, doi: 10.1186/s12864-020-07319-x.
[50] X. Li et al., "An integrated framework reinstating the environmental dimension for GWAS and genomic selection in crops," *Mol. Plant*, vol. 14, no. 6, pp. 874–887, Jun. 2021, doi: 10.1016/j.molp.2021.03.010.
[51] T. Nepolean, J. Kaul, G. Mukri, and S. Mittal, "Genomics-enabled next-generation breeding approaches for developing system-specific drought tolerant hybrids in maize," *Front. Plant Sci.*, vol. 9, Apr. 2018, Art. no. 361, doi: 10.3389/fpls.2018.00361.

Tanzila Islam received the master's degree in computer science from Jahangirnagar University, Bangladesh, in 2017. She is currently working toward the PhD degree with the Department of Systems Innovation Engineering, Graduate School of Science and Engineering, Iwate University. During her PhD, she was a PhD research student with the University of Tokyo for 1.5 years (Apr 2020 - Sep 2021). She was a lecturer with the Computer Science and Engineering Department, Southeast University, Bangladesh (2017-2018). She was a software engineer with Edu-smart, Bangladesh (2016-2017). Her research focuses on Artificial Intelligence specially deep learning and bioinformatics. She was awarded by the Japanese Government (Monbukagakusho: MEXT) Scholarship (2018-2022) for doing her PhD with Iwate University, Japan. Her recent work "A Deep Learning Method to Impute Missing Values and Compress Genome-wide Polymorphism Data in Rice" got awarded as "Best Poster" in Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies - BIOINFORMATICS.

Chyon Hae Kim received the doctor degree in engineering from Waseda University, in 2008. He is a director CTO of Sky Ocean Technology Co., Ltd. He is a visiting associate professor with Iwate University (2020-). He is technical adviser of AISing Ltd (2020-). He is an invited researcher (2008-) of Waseda University. He was assistant researcher of 21st Center of Excellence Project with Waseda University (2005-2008). He was researcher with Honda Research Institute Japan Co., Ltd. (2008-2013). He was an Invited Researcher of RIKEN (Japan) (2008-2013). He was associate professor of Iwate University (2013-2020). He was adjunct lecturer with Waseda University (2013-2014). He received the NISTEP Award (The Researchers with Nice Step) from the Ministry of Education as 11 top young researchers in Japan (2017,11). He received the Minister of Economy, Trade and Industry Award from University Originated Venture Awarding in Japan. His research focuses on Intelligent robot, Learning algorithm, Sequential learning, Underwater image processing, Artificial intelligence, Sensor network, Robotics, Agricultural and Environmental Biology.

Hiroyoshi Iwata received the PhD degree from the University of Tokyo, in 1998. He is associate professor with the Department of Agricultural and Environmental Biology, Graduate School of Agricultural and Life Sciences, The University of Tokyo, Tokyo, Japan. He received the Encouragement Award from Japanese Society of Breeding in 2009, the Best Paper Award from Japanese Society of Breeding in 2007, 2011, 2012, 2014, and 2015, and the Best Author Award from Japan Society for Simulation Technology in 2016. His research interests include the application of data science and statistics to plant genetics and breeding.

Hiroyuki Shimono received the doctor degree from Hokkaido University, in 2003. He is professor with the Faculty of Agriculture, Iwate University. Japan Prize in Agricultural Sciences, Achievement Award for Young Scientists (2010) & Award for Young Scientists of Japanese Society of Crop Science (2010) were received. His research interests include focuses on agronomy, phenotyping technologies, stress physiology, and simulation modeling.

Akio Kimura received the master's degree in computer and information sciences from the Graduate School of Engineering, Iwate University, in 1993, and joined Sony Corporation. While with Sony, he was engaged in research and development of magnetic recording. In 1995, he joined Iwate university as an assistant professor, and is now an associate professor of the department of Systems Innovation and Engineering. He is engaged in research related to image processing, computer vision, and machine learning. He holds a Dr. Eng. degree and is a member of IEICE, IPSJ, the Institute of Image Electronics Engineers of Japan, and the Society for Art and Science.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.