Ensemble supervised learning for genomic selection

Sheikh Jubair and Michael Domaratzki
Department of Computer Science
University of Manitoba
Winnipeg, Canada
{jubairs@myumanitoba.ca, Mike.Domaratzki@cs.umanitoba.ca}

***Abstract*—To meet the world's growing food and nutrition demands, agricultural breeders need to grow crops with improved phenotypes and create varieties that allow increased production. Genomic selection enables the breeders to select individuals with improved phenotypes even before growing them. Existing genomic selection is made mostly through statistical methods that do not accurately predict complex non-linear traits. Deep learning and other machine learning methods have been applied, but most of the deep learning methods are not specifically designed for genomic selection. Also, there has been relatively little comparison between different machine learning methods. We propose three ensemble learning methods: i) ensemble support vector regression; ii) ensemble deep convolutional neural networks and iii) random forests; to predict phenotype with high accuracy. We also propose a feature selection strategy that identifies important markers and contributes to improved phenotype prediction. The proposed marker selection strategy is independent of machine learning methods; thus, the markers that are selected remain the same when the machine learning model is changed. We employed our methods to Iranian wheat landraces. The result shows that ensemble learning methods are better than the single machine learning methods with the lowest PCC 0.339 for plant height and the highest PCC 0.747 for grain length. Our models are also robust as they rank both top twenty and bottom twenty individuals well with nDCG@20 ranges from 0.188 to 0.712.**

***Index Terms*—genomic selection, deep learning, machine learning, ensemble learning**

I. Introduction

Selection of proper individuals with intended phenotypes from a collection of varieties of a crop is essential to breeders as the right selection can lead to improvements in the crop such as drought resistance, biotic and abiotic stress resistance, yield improvement and disease resistance. While the amount of water, fertilizer, pest control, and good production practices constitutes the environment for the plant, the variety of the plant defines the ability to produce desired phenotypic value within that environment [17]. Thus, if the environmental factors and breeding practices are standardized, it is also vital to create improved varieties for that environment. Genomic selection (GS) is a marker-assisted selection method that uses whole-genome molecular markers to improve the quantitative traits or phenotypes of an organism, such as a crop or livestock, by identifying the top genotypes. That is, GS is a computational tool for choosing the most advantageous individuals from varieties and has the potential to save money and time by accelerating improvements to crops or livestock. Thus GS can solve the two main objectives of the breeders: building variation and selection of leading individuals from the variety that fulfills the breeding objective [1]. Though GS has been successfully applied to livestock, GS for crops is not as well developed [3] and therefore, there is a need for new computational tools of GS for plants. With proper GS software, it is possible to address the problem of feed quality, increased supply needs for food in a growing world population, and adaptation of crops for a specific environment such as drought stress and wet conditions.

GS links traits and the underlying genomic information. A trait is a characteristic or a feature of an organism, such as height, length, yield, and disease resistance. A phenotype is the expression of a particular trait that is determined by the interaction between an organism's genotype and environment. A large number of small effect genes known as polygenes cumulatively contribute towards the final expression of the phenotype. Though many markers contribute to the complex phenotypes of plants, some markers mostly interact with the environment and are responsible for a specific phenotype, and other markers remain stable [18]. Identification of markers that interact with the environment and respond to a phenotype is necessary to understand the crops and build a better variety.

In this paper, we propose three ensemble machine learning models: i) ensemble support vector regression; ii) ensemble deep convolutional neural networks and iii) random forests to predict different phenotypes of Iranian wheat landraces. We also compare the performance of our ensemble models with single machine learning models, such as support vector machines and convolutional neural networks, and a statistical model RR-BLUP. We also combine the concept of binning the continuous values of labels [26] and apply a filter-based method feature selection algorithm [9] to identify important markers for obtaining improved phenotypic values. As the filter-based feature selection method, we use chi-square feature selection. In general, for GS, the features are the genotyped markers and the labels are the traits. In the training phase, each of our models takes genotyped markers and a phenotyped trait as the input, performs feature selection on the marker data, creates several subsets of markers from the selected markers and then trains each subset employing a machine learning algorithm. In the testing phase, the inputs of the trained models are the same subset of markers that are used to train a specific model. The trained models predict the phenotypes and the

978-1–7281-1867-3/19/$31.00 ©2019 IEEE

final output is the average of all the predicted phenotypes. The training data is both genotyped and phenotyped, but the test data is only genotyped but not phenotyped.

Machine learning methods are known to perform better than statistical methods generally, but for GS, there is no single method that is better than all other methods and the performance of the same method can differ for different traits of the same species [11], [19], [23]. Very recently, Pérez-Enciso and Zingaretti [21] have reviewed deep learning techniques for GS, showing a limited amount of existing research on genomic selection for both plant and animal breeding [13]–[16]. Many of these models show that multi-layer perceptron models and convolution neural networks are comparable to or exceed established statistical techniques such as GBLUP. Ma et al. [15] employed DeepGS, a deep learning model with convolution neural network to predict the phenotypes from the genotypes and to the best of our knowledge, this is the only model that is specifically designed for GS for wheat. This model obtained better performance than existing statistical models, and the authors demonstrated that decreasing the number of markers increases the performance of DeepGS. Though the number of markers plays a crucial role in the prediction of phenotype, the selection of the markers in DeepGS is done randomly, and there is a need to identify the important markers for each trait, which leads to improved phenotype prediction.

As the traits we are going to predict are continuous variables, to select the features, one of the strategies is to make several bins within a range of values and consider each of those bins as a category [26]. After obtaining the categorical labels for each individual, any filter-based feature selection method can be applied to retain the markers that contribute to discriminate the groups [9]. In this paper, we use the chi-square feature selection as the filter method. In general, the filter-based feature selection does not rely on any machine learning algorithm; instead, they consider each feature individually and calculate a value based on some criteria to identify the potential to separate the classes. The features are then ranked based on the calculated values and the top features are used as inputs in any machine learning algorithm [25].

Ensemble learning methods combine the prediction of multiple weak machine learning methods and produce a reliable prediction [8]. There are two types of ensemble learning methods: bagging and boosting [22]. In this paper, we employed bagging to build our models and make predictions of phenotypes. In bagging, each model is trained independently with different subsets of the data. When making the prediction, each model produces an independent outcome, and the final result is the average of all the guesses. As each model is trained with different subsets, they capture the information of only a small part of the data. After combining all the models, they produce an overall picture and make a stronger prediction.

We organize the rest of the papers as follows. In section II of this paper, we described our materials and methods; section III contains the result, and section IV is our conclusion.

II. Materials and Methods

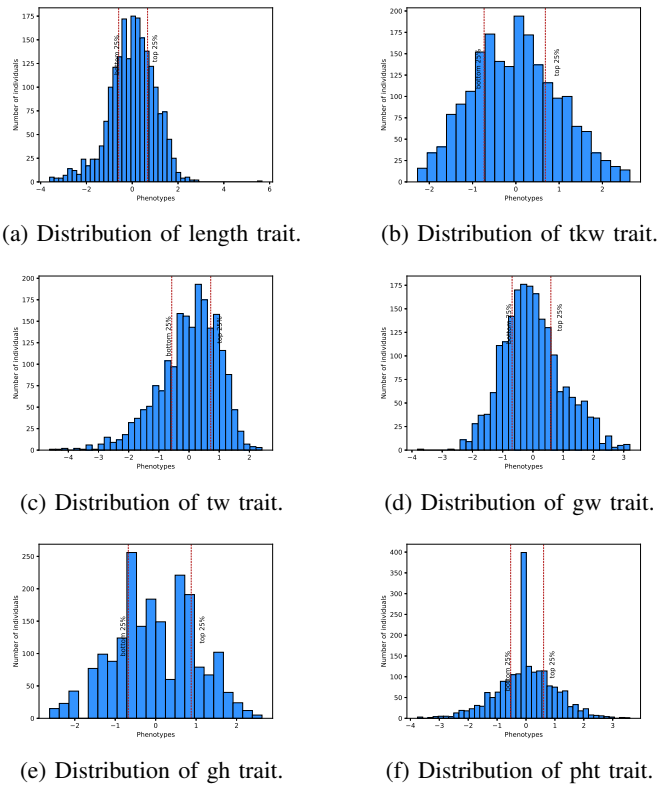
A. Dataset

The Iranian bread wheat (Triticum aestivum) dataset was obtained from the wheat gene bank of CIMMYT [6]. The dataset contains 2000 individuals of Iranian bread wheat and genotyped with genotype by sequencing method (GBS) using $33,709$ DArT (Diversity Array Technology) markers where the values are either 0 or 1 for each marker. All the phenotypic traits were measured in a single standard environment. The traits are thousand-kernel weight (tkw), test weight (tw), grain hardness (gh), grain length (gl), grain width (gw), and plant height (pht). More details can be found in [7].

B. Marker Selection

Previous work [15] on this dataset indicated that not all features are informative for genomic selection. This is consistent with other research that demonstrates that feature selection generally improves phenotype prediction [18].

To select features, we consider each phenotype individually. We created three bins where the first bin contains the top 25% genotypes based on their phenotypic value; in the second bin, the middle 50% were placed, and in the last bin, the bottom 25% were kept. The distribution of all traits is shown in Figure 1. The left side of the left red line shows the bottom 25% and the right side of the right red line indicates the top 25% individuals.



(a) Distribution of length trait.

(b) Distribution of tkw trait.

(c) Distribution of tw trait.

(d) Distribution of gw trait.

(e) Distribution of gh trait.

(f) Distribution of pht trait.

Fig. 1: Distribution of phenotypes of different traits of Iranian Wheat.

We labeled the bins with discrete values (1,2 and 3) and then applied chi-square feature selection [24] to rank the markers where higher values of chi-square indicate a more prominent feature. We retained markers that have a chi-square p-value of less than 0.005.

C. Marker Ensemble

Our genomic selection is an ensemble model that trains N independent machine learning models on subsets of markers. From the training set of the data, we created N subsets of markers where each subset contains m markers. To ensure that all the markers are at least selected once, we selected m markers per subset without replacement. This process was continued until all the markers are selected once. If all the markers are selected, we restarted the process again from the beginning and continue until we get M subsets. Each of the N selected subsets of features is used to train a machine learning model described in section II-D and II-E.

D. Deep Learning Model

Figure 2 shows the architecture of our convolution neural network (CNN). Our CNN model consists of one input layer, five convolution 1D layers, three ReLU layers, two max-pooling layers, three dropout layers, two fully connected layers, and one output layer. The input layer takes an input of M neurons and passes the input neurons to blocks of convolution layers. Each convolution layer has ten filters with kernel size ten and a stride of one. There are two groups of convolution layers. In the first group, there are two convolution layers, and in the second group, there are three convolution layers. Each convolution layer group is followed by a ReLU layer, a max-pool layer where both the kernel size and stride are 2 and a dropout layer. There are two fully connected layers where a ReLU and a dropout layer follow the first layer. The last fully connected layer is the output layer. We use Adam as the optimizer and Mean Absolute Error (mae) as the loss function. The batch size and learning rate are 128 and 0.001, respectively. The CNN is implemented using the Keras package in python [5].

E. Support Vector Regression

Support vector regression (SVR) [10] maps the data from one vector space to another vector space to find out better separability for prediction. We employ Support Vector Regression (SVR) as an alternative to CNNs for both individual and ensemble prediction. In our SVR model, we used the radial basis function (RBF) kernel as RBF can make a non-linear prediction and often performs better than other kernel functions. Two parameters can be optimized for the RBF kernel: cost and gamma. For the non-ensemble model, the cost is optimized in the range of $\{1, 2, 4, 16\}$. The gamma parameter is optimized from 2^{-4} to 2^4 in powers of 2. For the ensemble model, we use the default parameter values. Scikit-learn [20] is used to implement SVR.

F. Overall Architecture

Figure 3 shows the overall architecture of our framework for ensemble CNN and ensemble SVR. The genotyped and phenotyped data are first binned based on their phenotypic value, and then chi-square feature selection [12] [24] is applied to find the most discriminating features, which are chosen as those that have chi-square p-value less than 0.005. After that, an ensemble of N subsets of features are created, and a machine learning model, either CNN or SVR, is independently trained on each subset. The final output is the average of all the predicted outputs of each of the models.

G. Random Forest

Random forest (RF) [4] is an ensemble machine learning method that uses a large number of individual decision trees. Each of the decision trees predicts the phenotypes separately, and the final output is the average of all the prediction of the decision trees. To make each tree different from others, RF uses bagging, and the markers of each tree in a random forest are picked from a subset of random markers. We optimize two parameters for RF: i) the number of trees in a forest from 50 to 1000 with an increase of 25 at each iteration and ii) the number of features to consider when looking for the best split from 10 to 200 with an increase of 10 at each iteration. We applied the grid search to optimize these two parameters. Scikit-learn [20] is used to implement RF.

III. Results

For measuring the performance of our models, we used stratified 5-fold cross-validation. This means that the data is divided into five subsets without overlapping and the machine learning algorithms are trained with four subsets and evaluated with one subset. The training and testing were done five times, each time taking a different test-set. In each fold, 20% of the data from each bin are kept as the test-set and the rest of them are kept for training-set.

As the main objective of GS is to identify individuals that will harvest better phenotypes, it is more beneficial to obtain a linear relation between original phenotypes and predicted phenotypes than to predict the phenotypes accurately. If the relationship between the original phenotype and the predicted phenotype is linear or the order of predicted phenotypes and original phenotypes are the same, this means that the machine learning model performs well. We use two performance measures, i) Pearson correlation coefficient (PCC), and ii) normalized discounted cumulative gain at k (nDCG@k) [2] that either consider the linearity or the orders of the predicted phenotypes.

PCC measures the linear relation between the predicted phenotypes and the original phenotypes. Equation 1 shows the formula for calculating PCC. In this paper, x is the original phenotypes and y is the predicted phenotypes. If the original phenotypes and predicted phenotypes are perfectly linear, the PCC is 1. If the relationship is the opposite, the PCC is -1.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

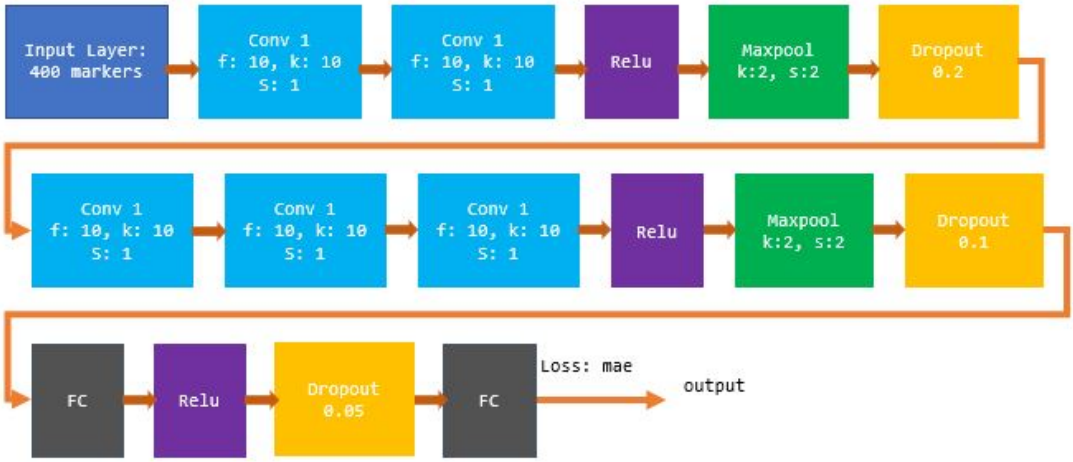


Fig. 2: Architecture of one convolution neural network. In the layers, k indicates the kernel size, f refers to the filter size and s is the stride.

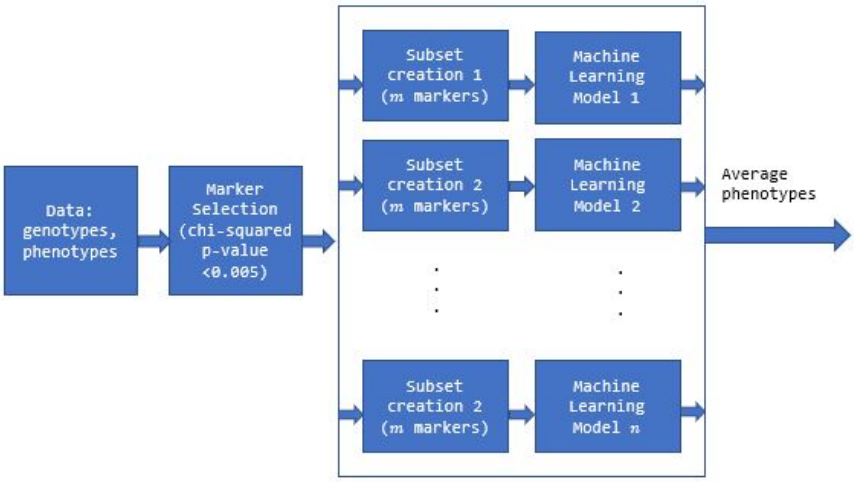


Fig. 3: Full architecture of the framework.

As the new varieties are formed using the top individuals of an existing variety, nDCG@k is a key measure for GS because it measures the quality of the ranking of the predicted phenotypes for the top k individuals. Equation 2 shows formula for calculating nDCG@k.

$$nDCG@k = \frac{DCG@k}{IDDCG@k} \tag{2}$$

In the equation 2, $DCG@k$ means the discounted cumulative gain [2] for the top k individuals and $IDCG@k$ is the ideal DCG for the top k individuals. DCG measures the graded relevance, and in GS, the relevance is the ranking of predicted phenotypes compared to the original phenotypes. In $IDCG$, the relevance is the ranking of original phenotypes. If the ranking is perfect, the nDCG@k is 1 and if the ranking is exactly opposite of the expected ranking, nDCG@k is 0. nDCG was previously used by Ma et al. [15] for evaluating GS.

A. Marker Ensemble

After using chi-square feature selection for each trait, we obtained a reduced set of important markers. Table I shows the number of markers in each trait that have p-value for chi-square ≤ 0.005. Gw has the highest number of markers, and pht has the lowest. A higher value of chi-square indicates better separability and 0 means the marker does not have any effect to predict the phenotypes. Though we use the p-value for chi-square ≤ 0.005, from our result, we observe that there are very few markers that have zero chi-square value.

Table II shows the percentage of common markers between different traits that are selected after feature selection. Most of the traits have $\approx 50\%$ common markers between them except

TABLE I: Number of markers in each trait after using chi squared feature selection.

Traits	# of selected markers.
gl	6,532
gw	11,175
gh	8,028
tkw	6,958
pht	1,023
tw	8,887

pht. Pht has on average 14.598% markers in common with other phenotypic traits. One reason behind this is that pht has only 1023 markers that have chi-square p-value ≤ 0.005, which is approximately six times fewer markers than the second smallest trait.

TABLE II: Percentage of common markers between two traits. The percentage is based on the average number of markers between two traits.

	gl	tkw	tw	gw	gh	pht
gl	100	57.26	54.18	50.75	48.28	17.07
tkw		100	49.21	65.05	56.16	13.74
tw			100	63.74	57.36	15.71
gw				100	69.28	12.65
gh					100	13.82
pht						100

B. Ensemble model vs single model

The single model of SVM and deep CNN are trained with a set that includes all the markers that are selected with feature selection. For the ensemble model, through experiments (results not shown), we selected an ensemble of size $N = 75$ with each machine learning model (SVR and deep CNN) considering $M = 400$ markers. The architecture of the deep CNN is the same for the single model except the input layer takes $11,176$ markers as the input. We chose $11,176$ markers as the input because it covers all the markers that have p-value for chi-square less than 0.005 for all the traits. Table III shows the comparison between the ensemble model and the single machine learning model. From the table, we observed that the ensemble model of SVR performs better than the single model of SVR. This means that the single model is affected by a large number of markers and creates a "large p, small n problem". In the ensemble model of SVR, each model is trained with a small subset of markers from the marker set which solves the "large p, small n" problem. The final output is the average of all the predicted values. The PCC of both deep learning model (ensemble and single) are almost similar though in most of the traits, the ensemble models have slightly higher PCC than the single model.

Figure 4 shows the comparison of nDCG@20 for deep CNN model, ensemble deep CNN and ensemble SVR model. In this figure, we did not consider the single SVR model as all the other models have very high PCC compared to the single SVR model. Single deep CNN has obtained the highest value of nDCG@20 for three traits. The ensemble SVR outperformed the other two models twice, while the ensemble deep CNN outperformed single CNN three times.

TABLE III: Comparison of PCC between actual and predicted traits, for both single model and ensemble model.

Traits	Deep CNN		SVR	
	Ensemble	Single	Ensemble	Single
gl	**0.738**	0.728	0.732	0.488
tkw	**0.663**	0.661	0.660	0.481
tw	**0.618**	0.614	**0.618**	0.266
gw	0.724	**0.731**	0.724	0.311
gh	0.648	**0.661**	0.660	0.422
pht	0.339	0.323	**0.379**	0.110

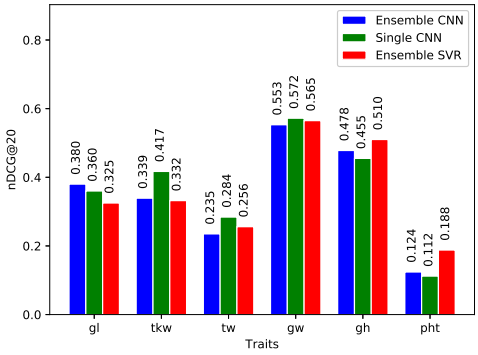


Fig. 4: Comparison of nDCG@20 for ensemble and single models.

Both the performance measures we applied show that the single deep learning model and both the ensemble models provide similar results with minimal improvement in performance among different models. The difference in performance between the single SVR model and the other models are very high due to training with a large number of markers. Though the hyper-parameters of the ensemble models are not optimized, they can produce better or similar performance than the single model. Hyper-parameter optimization of the ensemble models can further improve the performance of the ensemble models.

C. Ensemble Models

We employed three ensemble models: i) ensemble deep CNN, ii) ensemble SVR and iii) RF. Each of these models creates ensembles with subsets of features. Table IV shows the comparison of PCC among these models. From the table, we observe that RF has better PCC in four traits, though the increase in PCC from both ensemble deep CNN and ensemble SVR is small. In two traits, ensemble SVR is better than the other two models.

Though the PCC of the models on the pht trait is lower compared to other traits, all the traits have good PCC for all ensemble models. We compare the result of our best model with the performance of RR-BLUP that was reported in DeepGS [15]. We observe from Table V that we obtained

TABLE IV: Comparison of PCC between actual and predicted traits.

Traits	Ensemble DL	RF	Ensemble SVR
gl	0.738	**0.747**	0.732
tkw	0.663	**0.672**	0.660
tw	0.618	**0.624**	0.618
gw	0.724	**0.738**	0.724
gh	0.648	0.653	**0.660**
pht	0.339	0.352	**0.379**

improvement of 1.013 times to 1.021 times than the RR-BLUP for all the traits except gh. Though the performance of Ensemble SVR on the gh trait is better in RR-BLUP, PCC of 0.660 is a good score.

TABLE V: Comparison of PCC between RR-BLUP and the best model for each trait.

Traits	RR-BLUP	Best Ensemble Model	Improvement
gl	0.735	0.747	1.016
tkw	0.658	0.672	1.021
tw	0.614	0.624	1.016
gw	0.728	0.738	1.013
gh	0.685	0.660	0.963
pht	0.327	0.379	1.15

As it is essential to measure the accuracy of the ranking of individuals to build a better variety, we also measure the nDCG@20 for each trait. Table VI shows the comparison of nDCG@20 for top individuals among different models. Though PCC is better in four traits with RF, only gw with RF has better nDCG@20 than PCC. As we are only considering the top 20, the scores are satisfactory for all the traits except pht.

TABLE VI: Comparison of nDCG@20 for the top individuals.

Traits	Ensemble CNN	RF	Ensemble SVR
gl	**0.380**	0.355	0.325
tkw	**0.339**	0.325	0.332
tw	0.235	0.230	**0.256**
gw	0.553	**0.612**	0.565
gh	0.478	0.473	**0.510**
pht	0.124	0.125	**0.188**

To find the robustness of our ensemble models, we also measure the nDCG@20 for bottom individuals. Table VII shows the result. From the table, we observe that all the traits achieve nDCG@20 ≥ 0.3 except pht and tw.

When we perform feature selection, pht has very few features compared to other traits that can separate the top and bottom individuals. Figure 1f shows that for the pht trait, the dataset contains a large number of individuals in a specific range of phenotypic values and in other ranges, the number of individuals is very low. This distribution of phenotypes may cause machine learning models to overfit. Both PCC and nDCG@20 showed that pht is the hardest trait to predict in this dataset. As in complex crops like wheat, a lot of markers contribute cumulatively to the final expression of phenotypes,

TABLE VII: Comparison of nDCG@20 for the bottom individuals.

Traits	Ensemble CNN	RF	Ensemble SVR
gl	0.668	**0.712**	0.577
tkw	0.314	0.319	**0.335**
tw	0.236	0.276	**0.287**
gw	0.370	**0.395**	0.323
gh	**0.325**	0.290	0.265
ph	0.198	0.233	**0.235**

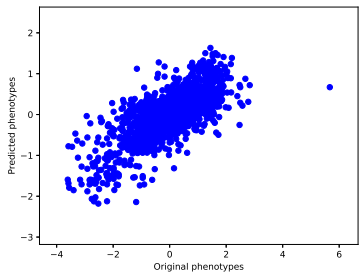
the machine learning models do not have sufficient information from the markers from which it can predict the pht.

The distribution of phenotypes plays the role in selecting fewer markers and obtain poor performance. Figure 5 shows the original vs. predicted phenotypes plot for gl, tw and pht. From the figure, we observe that the predicted phenotypes of gl are almost linear to the original phenotype; hence, it has high PCC and nDCG@20. The predicted phenotypes of trait tw are also close to linear to the original phenotypes though the nDCG@20 is low. The reason behind is there are many individuals with phenotypes that have a very close value between -2 to 1, making it difficult for the machine learning model to maintain the ranking while predicting, despite having a very high linear relationship with the original value. Thus we can consider that our model worked well for predicting tw. The predicted phenotypes of pht have a very little linear relation with the original phenotypes and thus results in low PCC. Gl has the best PCC and pht is the worst one we obtained with our models. The comparison of performance measure shows that there is no single method that outperforms the other models and there is no significant difference in performance with different ensemble models. Though RF obtained better PCC than others, RF models are optimized while other ensemble models are not.

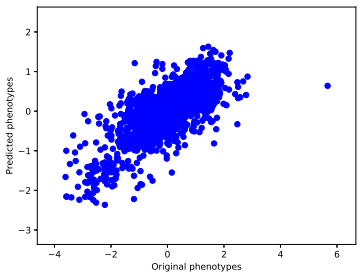
The marker selection method of Ma et al. [15] is random. Thus the performance of the model can vary when the subset of markers differs. In our model, we have a defined marker selection technique that gives the same subset of markers. Ma et al. reported a PCC of 0.742 for the DeepGS model for grain length only, where the model hyperparameters are appropriately tuned. In the random forest and ensemble deep learning model, we obtained PCC of 0.747 and 0.738, respectively, where the parameters of our ensemble deep learning model are not optimized. From this, we can observe that our proposed ensemble models are competitive with DeepGS, and that optimization of ensemble methods is a promising area to improve the results of the deep learning models.

IV. Conclusion

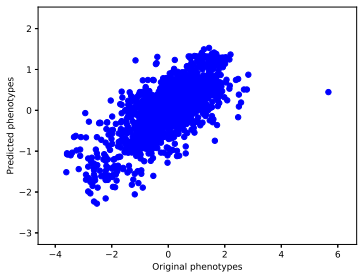
In this paper, we proposed three ensemble learning methods: i) ensemble SVR ii) ensemble CNN and iii) random forest for GS in wheat. We also proposed a binning approach by applying chi-square feature selection to the identification of essential markers for a specific trait. We showed that the ensemble models on a wheat dataset are competitive with both DeepGS [15] and single models. The performance of different



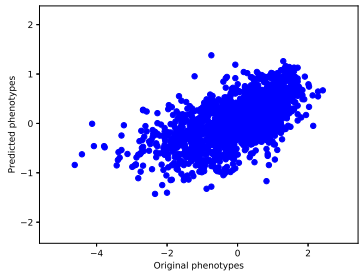
(a) Original vs predicted phenotypes of grain length with ensemble CNN.



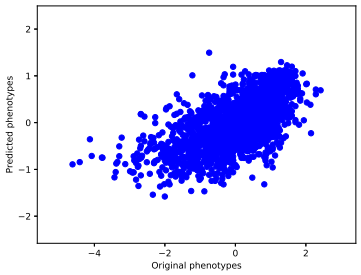
(b) Original vs predicted phenotypes of grain length with RF.



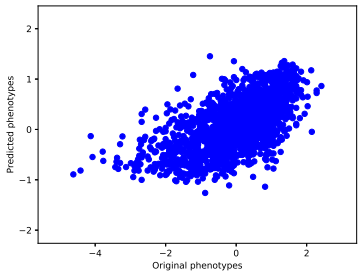
(c) Original vs predicted phenotypes of grain length with ensemble SVR.



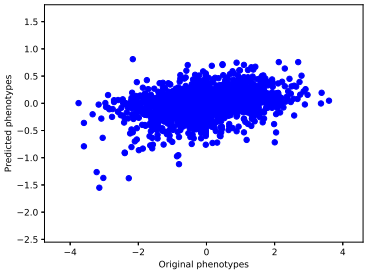
(d) Original vs predicted phenotypes of test weight with ensemble CNN.



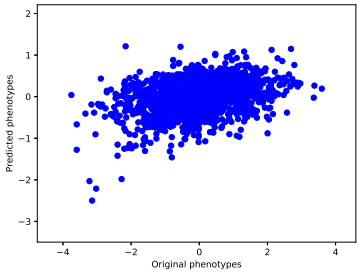
(e) Original vs predicted phenotypes of test weight with RF.



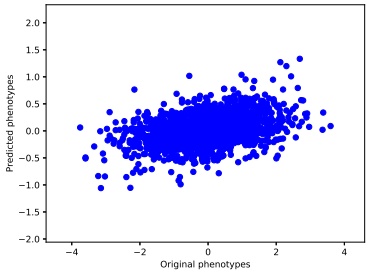
(f) Original vs predicted phenotypes of test weight with ensemble SVR.



(g) Original vs predicted phenotypes of plant height with ensemble CNN.



(h) Original vs predicted phenotypes of plant height with RF.



(i) Original vs predicted phenotypes of plant height with ensemble SVR.

Fig. 5: Original vs. predicted phenotypes for length, test weight, and plant height.

ensemble models are very similar to each other; thus, there is no definitive answer to which model is the best. In deep learning, the percentage of dropout neurons and the number of neurons in the fully connected layer plays a crucial part in tuning the model for better prediction. Cost and gamma are the two hyper-parameters that plays a similar role in SVR. In single models, we observe that the optimization of these parameters improves the accuracy of prediction dramatically. Thus, in the future, we will explore how the optimization of the hyper-parameters influences the performance of ensemble models. Currently, our models predict a single trait. Deep learners are known for their ability to predict multiple outputs at the same time. We will investigate if the ensemble deep learning model can predict multiple traits with the same or better accuracy. We will also integrate environmental information with our models so that the models can predict phenotypic differences based on the environment.

REFERENCES

[1] G. Acquaah. *Principles of plant genetics and breeding*. John Wiley & Sons, 2009.

[2] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between ir effectiveness measures and user satisfaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM, 2007.

[3] J. A. Bhat, S. Ali, R. K. Salgotra, Z. A. Mir, S. Dutta, V. Jadon, A. Tyagi, M. Mushtaq, N. Jain, P. K. Singh, et al. Genomic selection in the era of next generation sequencing for complex traits in plant breeding. *Frontiers in genetics*, 7:221, 2016.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

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

[6] CIMMYT. http://genomics.cimmyt.org/mexican_iranian/traverse/iranian/, accessed July, 2019.

[7] J. Crossa, D. Jarquín, J. Franco, P. Pérez-Rodríguez, J. Burgueño, C. Saint-Pierre, P. Vikram, C. Sansaloni, C. Petroli, D. Akdemir, et al. Genomic prediction of gene bank wheat landraces. *G3: Genes, Genomes, Genetics*, 6(7):1819–1834, 2016.

[8] T. G. Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002.

[9] G. Doquire and M. Verleysen. Mutual information-based feature selection for multilabel classification. *Neurocomputing*, 122:148–155, 2013.
[10] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.
[11] J. A. Holliday, T. Wang, and S. Aitken. Predicting adaptive phenotypes from multilocus genotypes in sitka spruce (picea sitchensis) using random forest. *G3: Genes, Genomes, Genetics*, 2(9):1085–1093, 2012.
[12] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.
[13] S. Khaki and L. Wang. Crop yield prediction using deep neural networks. *Frontiers in plant science*, 10, 2019.
[14] Y. Liu and D. Wang. Application of deep learning in genomic selection. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2280–2280. IEEE, 2017.
[15] W. Ma, Z. Qiu, J. Song, J. Li, Q. Cheng, J. Zhai, and C. Ma. A deep convolutional neural network approach for predicting phenotypes from genotypes. *Planta*, 248(5):1307–1318, Nov 2018.
[16] R. Mcdowell. Genomic selection with deep neural networks. Master's thesis, 2016.
[17] P. J. Milton. Breeding field crops. 1979.
[18] H. Oakey, B. Cullis, R. Thompson, J. Comadran, C. Halpin, and R. Waugh. Genomic selection in multi-environment crop trials. *G3: Genes, Genomes, Genetics*, 6(5):1313–1326, 2016.
[19] J. O. Ogutu, H.-P. Piepho, and T. Schulz-Streeck. A comparison of random forests, boosting and support vector machines for genomic selection. *BMC Proceedings*, 5(3):S11, May 2011.
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
[21] M. Prez-Enciso and L. M. Zingaretti. A guide on deep learning for complex trait genomic prediction. *Genes*, 10(7), 2019.
[22] J. R. Quinlan et al. Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996.
[23] H. Rachmatia, W. A. Kusuma, and L. S. Hasibuan. Prediction of maize phenotype based on whole-genome single nucleotide polymorphisms using deep belief networks. *Journal of Physics: Conference Series*, 835:012003, may 2017.
[24] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.
[25] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.
[26] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330, 2008.