Improving Genomic Prediction with Machine Learning Incorporating TPE for Hyperparameters Optimization

Mang Liang †, Bingxing An †, Keanning Li, Lili Du, Tianyu Deng, Sheng Cao, Yueying Du, Lingyang Xu, Xue Gao, Lupei Zhang ©, Junya Li and Huijiang Gao *

Institute of Animal Sciences, Chinese Academy of Agricultural Sciences, Beijing 100193, China
* Correspondence: gaohuijiang@caas.cn
† These authors contributed equally to this work.

Simple Summary: Machine learning has been a crucial implement for genomic prediction. However, the complicated process of tuning hyperparameters tremendously hindered its application in actual breeding programs, especially for people without experience tuning hyperparameters. In this study, we applied a tree-structured Parzen estimator (TPE) to tune the hyperparameters of machine learning methods. Overall, incorporating kernel ridge regression (KRR) with TPE achieved the highest prediction accuracy in simulation and real datasets.

Abstract: Depending on excellent prediction ability, machine learning has been considered the most powerful implement to analyze high-throughput sequencing genome data. However, the sophisticated process of tuning hyperparameters tremendously impedes the wider application of machine learning in animal and plant breeding programs. Therefore, we integrated an automatic tuning hyperparameters algorithm, tree-structured Parzen estimator (TPE), with machine learning to simplify the process of using machine learning for genomic prediction. In this study, we applied TPE to optimize the hyperparameters of Kernel ridge regression (KRR) and support vector regression (SVR). To evaluate the performance of TPE, we compared the prediction accuracy of KRR-TPE and SVR-TPE with the genomic best linear unbiased prediction (GBLUP) and KRR-RS, KRR-Grid, SVR-RS, and SVR-Grid, which tuned the hyperparameters of KRR and SVR by using random search (RS) and grid search (Grid) in a simulation dataset and the real datasets. The results indicated that KRR-TPE achieved the most powerful prediction ability considering all populations and was the most convenient. Especially for the Chinese Simmental beef cattle and Loblolly pine populations, the prediction accuracy of KRR-TPE had an 8.73% and 6.08% average improvement compared with GBLUP, respectively. Our study will greatly promote the application of machine learning in GP and further accelerate breeding progress.

Keywords: hyperparameters optimization; tree-structured Parzen estimator; genomic prediction; machine learning

1. Introduction

Genomic selection (GS) has been widely used in actual animal and plant breeding processes, which predicts the genomic estimated breeding value (GEBV) with the whole genome markers information [1]. Compared with the traditional selection methods, which relied on progeny testing, GS tremendously accelerated the breeding process by predicting GEBV with genotypes [2]. The Holstein dairy cattle of America was the first breed that applied GS to select high-fitness individuals and has continued up until the present [3]. After more than ten years of development, GS has been gradually applied to practical genetic improvement of pigs, horses, chickens, rice, and wheat, and it significantly improved breeding [4–10]. Genomic best linear unbiased regression (GBLUP) and BayesB are the representative methods of GS. The former directly estimated GEBV by constructing the
genetic relationship matrix (GRM) using genotypes, and the latter estimated the effect value of each single nucleotide polymorphism (SNP) using the Bayesian strategy to calculate the GEBVs [3,11,12]. However, GBLUP and BayesB predicted GEBV by building a linear model with naively ignored interaction effects and epistasis, and their prediction accuracy did not meet the desired accuracy in some populations [13]. The more robust and better compatibility genomic prediction (GP) model was what the breeder had been looking for, and they had attempted to improve the prediction accuracy by constructing a nonlinear GP model by using machine learning (ML) [14–16].

ML is a computer program learning from data to build a highly accurate prediction model without human intervention or assistance. Plentiful studies have proven the dominant position of ML in building prediction models for complex projects, such as disease diagnosis, driverless vehicles, weather forecasts, and quantitative investments [17–21]. In addition, ML has been considered to be the most effective implementation to parse high-throughput genome data [22]. ML has been applied to reconstruct the 3D structure of the protein, GP, and the genome-wide association study, among which the application of ML to GP was the major topic of the breeders [23–26]. González-Camacho et al. [27] compared the prediction accuracy of reproducing kernel Hilbert space (RKHS), support vector regression (SVR), deep learning (DL), and random forests (RF) with Bayesian lasso (BL) in sixteen wheat datasets. The results indicated that the prediction accuracy of SVR was higher than BL, DL, and RF. Okut et al. [28] used an artificial neural network (ANN) with Bayesian regularization to predict the GEBVs of the marbling score in Angus. The results showed that the ANN performed as well as BayesCπ, and the author believed that ANN was as useful as other GP methods for animal breeding. Montesinos-López et al. [29] evaluated the prediction accuracy of a multilayer perceptron and support vector machine (SVM) with seven real datasets; the predictions of the ML methods were very competitive, and SVM was the most efficient in terms of the computational time required. Although ML’s performance might not be preeminent in the portion dataset, the overwhelming majority of breeders were still fully confident that ML was the future of GP [27,29].

Unfortunately, the sophisticated process of tuning the hyperparameters of the ML and lacking a brilliant tuning strategy obstructed the further application of ML in actual animal and plant breeding. Generally, grid search is a highly recommended trick for parameter optimization, but it relies on rich hyperparameters tuning experience and consumes a lot of time because of its nonintelligence. Another commonly used hyperparameters tuning strategy was random search (RS), which automates hyperparameters optimization. But it seeks the optimized parameters through simple multiple random attempts, which is not the ideal parameter optimization strategy for GP.

There is an interest in using the tree-structured Parzen estimator (TPE) algorithm to optimize hyperparameters, which tunes the hyperparameters automatically with a Bayesian strategy and performs well in the reported studies [30–33]. A distinctive feature of this method is that it utilizes tree-structured adaptive Parzen estimators as a surrogate, whereas the standard Bayesian optimization algorithm utilizes Kriging (i.e., Gaussian process regression [34]). This surrogate naturally handles not only continuous variables but also discrete, categorical, and conditional variables that are difficult to handle using Kriging [30]. Nguyen et al. [31] tuned the hyperparameters of long short-term memory (LSTM) with TPE. The performance of LSTM-TPE outperformed LSTM-RS in all of the datasets. Shen et al. [33] applied natural gradient boosting with TPE to predict the runoff on the monthly, weekly, and daily scales at the Yichang and Pingshan stations in the upper Yangtze River. The results showed the proposed model improved in all indicators compared to the benchmark model. The root mean square error of the monthly runoff prediction was reduced by 9% on average and 7% on the daily scale. Their studies inspired us to apply TPE to optimize the hyperparameters of ML in GP.
Therefore, we integrated KRR and SVR with TPE to simplify the progress of optimizing hyperparameters and broaden the application of ML in GP. To evaluate the performance of TPE, we compared the prediction accuracy of KRR and SVR, which optimized hyperparameters by TPE, RS, and Grid, with GBLUP in the simulation and real datasets. The Pearson correlation coefficients between the predicted GEBV and phenotypes in the validation dataset were calculated. The mean values of fifty replicates (repeat five-fold cross-validation ten times) were used to quantify the prediction accuracy.

2. Materials and Methods

2.1. Materials

Simulation dataset: The simulation dataset was downloaded from the public dataset, including 4000 individuals [35]. It contained 3000 reference individuals in generation 1–3 and 1000 validation individuals in generation 4. This dataset’s phenotype consisted of three simulated traits (T1, T2, and T3), and the genotypes consisted of five chromosomes with 10,000 SNPs. This dataset has been used widely to evaluate the performance of genomic prediction methods [36,37].

Chinese Simmental beef cattle dataset: The data on the population was provided by the Institute of Animal Science of the Chinese Academy of Agricultural Sciences. In total, 1301 Chinese Simmental beef cattle individuals in this study were born between 2008 and 2020 from Ulgai, Xilingol League, and Inner Mongolia in China. The phenotypes of this dataset consisted of seven quantitative traits, which can be divided into two types: (1) growth and development traits: live weight (LW, kg) and average daily gain (ADG, kg/day) and (2) slaughter traits: net meat weight (NMW, kg), the thickness of thigh meat (TT, cm), tenderloin weight (TW, kg), and eye muscle weight (EMW, kg). Animals were genotyped with the Illumina BovineHD BeadChip, which contained 770,000 SNPs. PLINK v1.09 [38] was taken for quality control to remove animals with missing genotypes for more than 10% of SNPs, filtering out the SNPs with a minor allele frequency (MAF) lower than 5% and a call rate (CR) lower than 95, a significant deviation from the Hardy–Weinberg equilibrium \( p < 10^{-6} \). After quality control, 1287 individuals with 671,990 SNPs have been retained.

Loblolly pine dataset: The original genotypes of the 951 individuals in the loblolly pine dataset contained 7216 SNPs [39]. After QC, 4853 SNPs were retained for the subsequent analyses in this study. Eight traits were selected to reconstruct the phenotypes of this dataset, which refers to growth, development, and disease resistance, three types of traits: crown width along the planting beds (CWAL, cm), total height to the base of the live crown (HTLC, cm), branch angle average (BA, degree), crown width across the planting beds (CWAC, cm), average branch diameter (BD, cm), gall volume (GALL), the presence or absence of rust (Rust_bin), and the basal height of the live crown (BHLC, cm). The heritability of these traits was from 0.12 to 0.45.

Pig dataset: All of the individuals in this dataset were genotyped with the Illumina PorcineSNP60 chip, and the SNPs were filtered with the following criteria: MAF < 0.03, genotype call rate <95%, and HWE test \( p \)-value < \( 10^{-4} \). Referring to the previously reported studies [40], we selected four traits (T2, T3, T4, T5) to assess the performance of each method in this study because the number of records of trait T1 was limited.

German Holstein population: This dataset consisted of 5024 bulls, in which the individuals were genotyped with the Illumina Bovine SNP50 Beadchip, and three phenotypic types of information were recorded. In the QC procedure, the SNPs with a call rate less than 95%, MAF less than 0.01, and HWE \( (p < 1 \times 10^{-4}) \) were removed. The genotype data in the following analyses contained 42,551 SNPs. According to Hu et al. and Zhe et al. [41,42], three traits—somatic cell score (SCS), milk fat percentage (MFP, %), and milk yield (MY, kg)—were selected to represent different genetic architectures.

The statistical description of each dataset is shown in Table 1.
Table 1. Summary of five datasets.

| Dataset          | Trait | N     | h²   | Mean | SD   |
|------------------|-------|-------|------|------|------|
| **Simulation**   | T1    | 3000  | 0.36 | 0.00 | 176.52 |
|                  | T2    | 3000  | 0.35 | 0.00 | 9.51  |
|                  | T3    | 3000  | 0.52 | 0.00 | 0.02  |
| **Beef cattle**  | LW    | 1285  | 0.30 | 510.37 | 73.50 |
|                  | ADG   | 1282  | 0.28 | 0.97 | 0.21  |
|                  | NMW   | 1273  | 0.32 | 233.02 | 41.75 |
|                  | TT    | 1275  | 0.30 | 17.89 | 2.12  |
|                  | TW    | 1284  | 0.29 | 8.75 | 1.98  |
|                  | EMW   | 1281  | 0.38 | 10.67 | 2.22  |
| **Loblolly pine**| HTLC  | 861   | 0.31 | 20.30 | 73.31 |
|                  | BA    | 861   | 0.45 | 2.28 | 42.03 |
|                  | BD    | 910   | 0.11 | 0.05 | 1.20  |
|                  | BHLC  | 861   | 0.35 | 0.092 | 0.51 |
|                  | CWAC  | 861   | 0.45 | 2.28 | 42.03 |
|                  | CWAL  | 861   | 0.27 | 2.44 | 27.33 |
|                  | GALL  | 807   | 0.12 | −0.022 | 1.31 |
|                  | Rust_bin | 807 | 0.21 | −0.014 | 0.40 |
| **Pig**          | T2    | 2715  | 0.16 | 0.00 | 1.12  |
|                  | T3    | 3141  | 0.22 | 0.71 | 0.96  |
|                  | T4    | 3152  | 0.32 | −1.07 | 2.33 |
|                  | T5    | 3184  | 0.38 | 37.99 | 60.45 |
| **German Holstein** | MY   | 5024  | 0.95 | 370.79 | 641.60 |
|                  | MFP   | 5024  | 0.94 | −0.06 | 0.28  |
|                  | SCS   | 5024  | 0.88 | 102.32 | 11.73 |

Note: N: Number of individuals with phenotypes. h²: heritability. SD: standard deviation.

2.2. Methods

**TPE**: TPE was a Bayesian optimization algorithm that optimized the hyperparameters automatically. Hyperparameters optimization of TPE can be represented as:

\[ x^* = \arg \min_{x \in \mathbb{R}} (-f_M(x)) \]

where \( f_M(x) \) represents an objective score to maximize model \( M \), \( x^* \) is the set of hyperparameters that yields the highest score, and \( x \) can take on any values in the space \( \mathbb{R} \).

In the optimized progress of Equation (1), excepted improvement (EI) was used as the criterion. \( EI \) is the expectation under some model \( M \) of \( f_M(x) \) that will exceed some threshold \( y^* \):

\[ EI_{y^*}(x) = \int_{-\infty}^{y^*} \max(y^* - y, 0) p_M(y|x) dy \]

In direct contrast to the Gaussian and random research model \( p_M(y|x) \) are TPE models \( p(x|y) \) and \( p(y) \), where \( p(x|y) \) can be written as:

\[ p(x|y) = \begin{cases} l(x), & \text{if } y < y^* \\ g(x), & \text{if } y \geq y^* \end{cases} \]

where \( l(x) \) is the density formed by using the observations \( \{x^{(i)}\} \) such that the corresponding objective score was lower than \( y^* \) and \( g(x) \) is the density formed by using the remaining observations.

With Bayes’ rule, the EI equation becomes:

\[ EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p_M(y|x) dy = \int_{-\infty}^{y^*} (y^* - y) \frac{p_M(x|y)p_M(y)}{p_M(x)} dy \]
Finally, the EI can be represented as:

\[
EI_y(x) = \frac{y^* \cdot l(x) - l(x) \cdot \int_{-\infty}^{y^*} p_M(y) dy}{\gamma(x) + (1 - \gamma) \cdot g(x)} \alpha \left( \gamma + g(x) \right)^{-1}
\]

Equation (5) indicated that EI is proportional to the ratio of \(l(x)/g(x)\), and therefore, to maximize EI, we should draw scores of the hyperparameters, which are more likely to be under \(l(x)\) than under \(g(x)\). The TPE works by drawing sample hyperparameters, evaluating them in terms of \(l(x)/g(x)\), and returning the candidate \(x^*\) greatest EI for each iteration. In this study, we achieved TPE with the help of a hyperopt Python package, which can be obtained at https://github.com/hyperopt/hyperopt, accessed on 29 September 2022. The process of using TPE to optimize the hyperparameters of KRR or SVR is demonstrated in Figure 1. In this study, the TPE was performed with Python software with the hyperopt package.

![Figure 1](image-url)

**Figure 1.** Optimization of the hyperparameters of KRR and SVR to construct the GP model using TPE. The working process can be divided into three steps: (1) select the ML algorithm and the dimension reduction methods. In this study, the ML we used was KRR or SVR, and the method to reduce the dimension was principal component analysis; (2) determine the hyperparameters that need to be optimized, which include the number of the principal component to construct the prediction model and the hyperparameters of KRR (e.g., kernel, alpha, etc.) and SVR (e.g., kernel, gamma, C, etc.); and (3) optimize the hyperparameters by using TPE and training the prediction model for GP.

**PCA:** Principal component analysis (PCA) is a popular technique for analyzing large datasets containing a high number of dimensions per observation, increasing the interpretability of data while preserving the maximum amount of information and enabling the visualization of multidimensional data. In this study, we applied PCA to reduce the dimensionality of the input feature, and the number of (k) components incorporated in SVR and KRR was optimized by TPE. The PCA procedure was performed using the Python software with the sklearn package.
**KRR**: KRR utilizes the kernel trick \((\phi(x_i))\) to construct a newly latent feature space and then build the ridge regression model in the latent space. KRR is represented as \(y_i = \beta^T \phi(x_i)\), where \(\beta\) is the vector of weights. Regularized least squares is applied to optimize \(\beta\) as follows:

\[
\min_{\beta} L_{KRR} = \frac{1}{2} \| \beta \|^2 + \frac{1}{2} C \sum_{i=1}^{n} (\hat{y}_i - \beta^T \phi(x_i))^2
\]

where \(C\) is the regularization constant. By taking the derivative of \(L_{KRR}\) concerning \(\beta\) equating the resulting equations to zero, the output weight vector \(\beta\) is obtained as:

\[
\beta = \left( \phi^T \phi + \frac{I}{C} \right)^{-1} \phi^T \hat{y}
\]

where \(\phi(x_i)\) is the instance in the latent feature space. \(I\) is an identity matrix. According to the representer theorem, \(\beta\) can be represented with \(\alpha\) as:

\[
\beta = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \phi^T \alpha
\]

thus:

\[
\alpha = \left( \phi^T \phi + \frac{I}{C} \right)^{-1} \hat{y} = \left( K + \frac{I}{C} \right)^{-1} \hat{y}
\]

where \(K\) is the kernel matrix whose entries are obtained as:

\[
K(x_i, x_j) = \phi(x_i) \phi(x_j)^T
\]

Finally, with a new test instance \(x_i\), the predicted value is obtained using dual weights and similarity between the test sample \(x_i\) and all the training samples used for prediction.

\[
y(x_i) = k'(K + \frac{I}{C})^{-1} \hat{y}
\]

where \(k' = K(x_i, x_j), j = 1, 2, \ldots, n\). In this study, the KRR was performed using the Python software with `sklearn` package, and the hyperparameters that need to be tuned are demonstrated in Table 2.

**Table 2.** The hyperparameters need to be tuned.

| KRR               | SVR               |
|-------------------|-------------------|
| **Kernel**        | **Kernel**        |
| Cosine, RBF, Linear | RBF, Linear, Ploy |
| **Gamma**         | **Degree**        |
| 0.000001–0.001    | 1, 2, 3, 4        |
| **Alpha**         | **Gamma**         |
| 0–10              | 0.000001–0.001    |
| **K**             | **C**             |
| 1–n               | 0.1–100           |
| **n**             | **1–n**           |

\(n\): the number of the principal component of the results of PCA. Cosine: Cosine kernel, RBF: Radial basis function kernel, Linear: Linear kernel, Ploy: Polynomial kernel

**SVR**: For regression with a continuous response, SVR is fitted by the following formula:

\[
f(x) = \beta_0 + h(x)^T \beta
\]

where \(h(x)^T\) is the kernel function. As for the ‘\(\epsilon\)-insensitive’ SVM regression, the SVR problem is represented as:

\[
\min_{\beta, \beta_0} \frac{1}{2} \| \beta \|^2 + \sum_{i=1}^{n} V(\gamma_i - f(x_i))
\]
where
\[ V_\varepsilon(r) = \begin{cases} 0 & \text{if } |r| < \varepsilon \\ |r| - \varepsilon & \text{otherwise} \end{cases} \] (14)

\( V_\varepsilon(r) \) is an ‘\( \varepsilon \)-insensitive’ error measure. If the absolute error between \( f(x_i) \) and \( y_i \) of the \( i \)th instance is bigger than \( \varepsilon \), the loss is calculated. \( \lambda \) is a positive constant. \( \| \cdot \|_2^2 \) denotes the norm under a Hilbert space.

The SVR can be written as follows:
\[
\hat{\beta} = \sum_{i=1}^{n} (\hat{\alpha}_i^* - \hat{\alpha}_i)x_i, \quad (15)
\]
and,
\[
f(x) = \sum_{i=1}^{n} (\hat{\alpha}_i^* - \hat{\alpha}_i)x_i \quad (16)
\]
where \( \hat{\alpha}_i^*, \hat{\alpha}_i \) are positive weights given to each observation and estimated from the data, and the inner product kernel \( K(x_i, x_i) \) is an \( n \times n \) symmetric and positive definite matrix. All of the SVR procedure was performed using the Python software with sklearn package, and the hyperparameters needed to be optimized, as shown in Table 2.

**GBLUP**: GBLUP estimates GEBVs using phenotypes and genomic relationships that are calculated with the whole genome markers’ information [43]. In GBLUP, the effect of each SNP was assumed to follow the identical normal distribution [44]. The formula of GBLUP is as follows:
\[
y^* = Z\gamma + e \quad (17)
\]
where \( y^* \) is the vector of the corrected phenotype, and \( Z \) is an incidence matrix for individual effects. \( \gamma \sim N(0, G\sigma^2_g) \) is a vector of breeding values. The \( \sigma^2_g \) is genetic variance. \( e \sim N(0, I\sigma^2_e) \) is a vector of residuals, where \( I \) is an identity matrix and \( \sigma^2_e \) is the residual variance. \( G \) matrix was calculated with the formula:
\[
G = \frac{ZZ'}{2\sum_p(1-p^2)}, \quad p_i \text{ is the MAF of the } i\text{-th marker.}
\]
This study performed GBLUP using the software R with the rrBLUP package.

**2.3. Assessing Prediction Performance**

This study quantified the prediction accuracies with the Pearson correlation coefficients between predicted GEBV and the phenotypes. The prediction accuracy demonstrated in this paper was the mean of fifty replicates of ten-times repeat five-fold cross-validation for each trait. The Pearson correlation coefficient was calculated as follows:
\[
r(y^*, \text{GEBV}) = \frac{\text{cov}(y^*, \text{GEBV})}{\sqrt{\text{var}(y^*)\text{var}(\text{GEBV})}}, \quad \text{where } y^* \text{ was the phenotype.}
\]

To test the statistical significance of the observed differences between the method, Friedman’s and Nemenyi post hoc tests were performed with the Python software with the Orange and Scipy package. Friedman’s test operated on the average rank of the methods and checked the validity of the hypothesis (null hypothesis) and verified that all methods were equivalent. The Nemenyi post hoc test was used to find which methods, in particular, differed from GBLUP.

**3. Results**

**Simulation dataset**: Firstly, we demonstrated the performance of KRR and SVR, tuning the hyperparameters with TPE, RS, and Grid, compared with GBLUP in the simulated dataset (Table 3). The simulated dataset, QTLMAS 16th, includes three traits, and the heritability was 0.45, 0.38, and 0.48, respectively. Table 2 summarizes the prediction accuracy performances of each strategy. The results showed that the average prediction accuracy of GBLUP is similar to KRR-TPE, and the prediction accuracy of SVR is slightly lower. For KRR, using TPE to optimize the hyperparameters achieved the highest accuracy, but KRR-RS also achieved good performance and was on average just 0.24% lower than KRR-TPE. The performance of SVR was unsatisfactory in this dataset, except for the lower prediction accuracy in T1 and T2. It can also not be applied to the GEBV prediction of T3.
Table 3. The average prediction accuracy of each method in the simulation dataset.

| Trait | KRR | SVR | GBLUP |
|-------|-----|-----|-------|
|       | TPE | RS  | Grid  | TPE | RS  | Grid  |       |
| T1    | 0.402 | 0.400 | 0.394 | 0.39 | 0.386 | 0.393 | 0.406 |
| T2    | 0.398 | 0.395 | 0.395 | 0.387 | 0.382 | 0.384 | 0.400 |
| T3    | 0.546 | 0.541 | 0.544 |       |       |       | 0.545 |

Note: the prediction accuracy of each method was assessed by the Pearson correlation coefficient between the predicted GEBV and the phenotypes of each trait. Five-fold cross-validation repeated ten times was used to ensure the high credibility of the results.

**Chinese Simmental beef cattle**: In addition to the simulation data, we also evaluated the prediction ability of each strategy in the real animal and plant dataset from cattle, cows, pigs, and pine. We compared the average prediction accuracy of each method in Table 4. The results indicated that the KRR-TPE predicted more accurate GEBV compared with KRR-RS, KRR-Grid, and GBLUP, improving prediction accuracy by 1.67% (−0.47–6.12%), 10.36% (8.73–13.08%), and 8.73% (3.40–17.88%) respectively. Although the performance of SVR did not meet the expectations compared with the reported studies, the prediction ability of SVR-TPE and SVR-RS was comparable with GBLUP for the Chinese Simmental beef cattle population.

Table 4. Comparison of prediction accuracy of each method using seven quantitative traits in the Chinese Simmental beef cattle.

| Trait | KRR | SVR | GBLUP |
|-------|-----|-----|-------|
|       | TPE | RS  | Grid  | TPE | RS  | Grid  |       |
| LW    | 0.298 | 0.288 | 0.264 | 0.278 | 0.271 | 0.286 | 0.276 |
| ADG   | 0.274 | 0.275 | 0.252 | 0.215 | 0.202 | 0.222 | 0.265 |
| MW    | 0.298 | 0.287 | 0.272 | 0.268 | 0.271 | 0.262 | 0.270 |
| TT    | 0.294 | 0.295 | 0.267 | 0.249 | 0.288 | 0.247 | 0.274 |
| TW    | 0.208 | 0.196 | 0.197 | 0.193 | 0.201 | 0.185 | 0.193 |
| EMW   | 0.314 | 0.316 | 0.288 | 0.291 | 0.286 | 0.27  | 0.295 |

Note: prediction accuracy performance was measured by the Pearson correlation coefficient. The prediction accuracy was calculated by five-fold cross-validation repeated ten times, which means that the prediction accuracy in Table 3 is the mean of the fifty Pearson correlation coefficients between the predicted GEBV by each method and the phenotypes.

**Loblolly Pine**: The Loblolly Pine dataset concluded eight quantitative traits and the heritability of the traits was from 0.12 to 0.43. Figure 2 demonstrates the prediction ability of each method in the Loblolly Pine dataset. The KRR-TPE achieved the highest predictive ability only in four traits (HTLC, BD, BHLC, GALL). For the other traits (BA, CWAC, CWAL, Rust_Bin), the prediction ability of KRR-TPE was almost the same as SVR-TPE, which had the highest prediction accuracy. Considering all of the traits of the Loblolly Pine dataset, KRR-TPE possessed the most powerful prediction ability, the prediction accuracy of which was 0.70% (−0.21–2.23%), 5.98% (−0.42–13.85%), 1.76% (−0.51–9.13%), 4.83% (−1.26–12.15%), 1.35% (−0.42–4.96%) and 6.08% (0.45–20.33%) higher than KRR-RS, KRR-Grid, SVR-TPE, SVR-RS, and SVR-Grid, respectively.

**German Holstein population and pig population**: Moreover, we calculated the average prediction accuracy of each method in the German Holstein population and pig population, both of which were genotyped using the chips with a marker density of about 50 k. Figure 3 compares the predictive ability of various methods to predict GEBV in the German Holstein and pig populations. For the pig population (Figure 3A), except that KRR-Grid performed worse than GBLUP, an average of 5.63%, the rest of the methods were comparable with GBLUP, and there was no method that outperformed other methods. Figure 3B reports the predictive performance of the methods in the German Holstein population, which was similar to the performance in the pig population; there was no obvious difference in the average of each method.
the Pearson correlation between predicted GEBV and phenotypic values with five-fold cross-validation and was repeated ten times.

![Figure 2](image2.png)

**Figure 2.** Comparison of prediction accuracy of KRR, SVR, with different hyperparameters-optimized strategies, and GBLUP for Loblolly Pine datasets. The prediction accuracy was assessed by the Pearson correlation between predicted GEBV and phenotypic values and was repeated ten times.

![Figure 3](image3.png)

**Figure 3.** The comparison of the prediction accuracy of each method in the actual animal dataset: pig population (A) and German Holstein population (B). The prediction accuracy was assessed by the Pearson correlation coefficient between predicted GEBV and phenotypic values with five-fold cross-validation and was repeated ten times.

**General evaluation:** Firstly, we used Friedman’s test and the Nemenyi post hoc test to test the statistical significance of the observed difference in all datasets, and the testing results are shown in Figure 4. The *p*-value of Friedman’s test was 0.003, which means the average rank of each method was not equivalent. According to the Nemenyi post hoc test, the average rank of KRR-TPE differed significantly from GBLUP, and the average rank of other methods was not significantly different. Subsequently, we compared the time consumed by GBLUP, KRR-TPE, and SVR-TPE (Figure 5), and the results indicated that KRR-TPE was slower than GBLUP and faster than SVR-TPE.

![Figure 4](image4.png)

**Figure 4.** Comparison of the average rank of each method. A group of methods that do not significantly differ from GBLUP (*p* = 0.05) are connected by the red line.
As ML has entered the twenty-first century, the theory and application of ML have been greatly developed. Breeders desired to predict more accurate GEBV with ML, further accelerating the genetic gain in animal and plant breeding. Before constructing the prediction model, the hyperparameters should be appointed, which determine whether machine learning algorithms can effectively mine information in the high-throughput genome data. The most-reported studies that used ML to predict GEBV tuned hyperparameters manually, which was cumbersome and relied on rich experience and was thus not friendly for novices. Therefore, there is an urgent need to develop an uncomplicated hyperparameters optimization strategy to promote ML application in GP. Thus, we simplified the process of predicting GEBV by using the automatic parameters tuning strategy, TPE, which optimized the hyperparameters automatically and performed better in reported studies [30–33]. The performance of KRR and SVR, tuned by TPE, RS, and Grid separately, were assessed by the simulation dataset and four real datasets. Overall, the performance of KRR-TPE outperformed the other methods.

Typically, the grid search sets a series of initial values for each hyperparameter and then evaluates the prediction ability of the estimator with every combination to select the optimum parameters with the most robust hyperparameters configuration. Although grid search works well in most cases, it can only ensure that the parameters selected are relatively optimal because it did not search from the complete parameter space. RS is essentially similar to grid search, randomly selecting the combination from the hyperparameters space and then repeating this process. If we repeat this process enough times, we can get the best configuration of the hyperparameters. To a certain extent, RS was capable of tuning hyperparameters automatically, but it seems impractical for genomic prediction due to being time-consuming. Although grid search and RS might find the super parameters relatively suitable for the prediction model, these have an obvious disadvantage and are still not intelligent enough.

The TPE algorithm is designed to optimize quantization hyperparameters with a more brilliant strategy to find a quantization configuration that achieves an expected accuracy target. TPE is an iterative process that uses the history of evaluated hyperparameters to create a probabilistic model using the Bayesian algorithm, which is used to advise on the next configuration of hyperparameters to evaluate. Therefore, it is highly performant and very time-efficient compared with RS and grid search [45]. Our results again proved that TPE is an excellent automatic tuning hyperparameters strategy. According to the results of the general evaluation section, the KRR-TPE was the only method significantly superior to GBLUP.

As for why the performance of SVR-TPE did not outperform SVR-Grid, we analyzed the process of tuning the hyperparameters of SVR and believe there were the following two main determinants. Firstly, to ensure the consistency of experimental results and the acceptability of calculation time for the actual breeding program, the iteration times to optimize the hyperparameters of SVR-TPE and KRR-TPE were set to only 200. However, SVR
had more hyperparameters compared with KRR, and insufficient iterations might result in SVR-TPE not achieving higher prediction accuracy. Secondly, our team had previously accumulated a lot of experience in SVR hyperparameters optimization in GP. Consequently, we can easily achieve the high prediction accuracy of SVR by manually tuning hyperparameters, but it is not realistic for most breeders and experimenters. Therefore, we have reason to believe that TPE has great potential to promote the wide application of ML in GP.

Although the computing speed of computers has been significantly improved, computing speed is still a major problem to be solved when widely applying ML to GP. To further tap the potential of automatic hyperparameters tuning and promote the application of ML in actual animal breeding, running the GP program in the graphics processing unit (GPU) might be a practical tactic. From my perspective, only by combining computer science, mathematics, and biology can we fully phase high-throughput genome information and achieve a breakthrough in animal breeding.

5. Conclusions

In conclusion, we first integrated KRR with tree-structured Parzen estimator optimized hyperparameters automatically to simplify the process of using machine learning for genomic prediction. The results indicate that incorporating KRR and TPE can greatly and more conveniently improve the prediction accuracy of GEBV. Especially for the Chinese Simmental beef cattle and Loblolly pine populations, the prediction accuracy of KRR-TPE had an 8.73% and 6.08% average improvement compared with GBLUP, respectively. Our study also promotes machine learning’s wider application in actual animal and plant breeding.

Author Contributions: M.L. wrote, and J.L. and H.G. revised the paper. K.L., M.L. and B.A. performed experiments. L.X., L.Z., Y.D. and T.D. collected the data. L.D., S.C., X.G. and T.D. participated in the design of the study and contributed to the acquisition of data. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundations of China (32172693), the National Natural Science Foundations of China (31872975), the Program of National Beef Cattle and Yak Industrial Technology System (CARS-37), the Technology Project of Inner Mongolia Autonomous Region (2020GG0210).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Chinese Simmental Beef Cattle dataset: Data are available from the dryad digital repository: http://DOI:10.5061/dryad.4qc06. German Holstein dataset: Data can be obtained at https://www.g3journal.org/content/5/4/615.supplemental. Pig dataset: Data are available from https://academic.oup.com/g3journal/article/2/4/429/6026060?login=true#supplementary-data. Loblolly pine dataset: The quality-controlled genotypes can be found at https://www.genetics.org/highwire/filestream/412827/field_highwire_adjunct_files/1/FileS1.zip and the complete phenotypes at https://www.genetics.org/highwire/filestream/412827/field_highwire_adjunct_files/4/FileS4.xlsx.

Acknowledgments: This work was supported by funds from the National Natural Science Foundations of China (32172693), the National Natural Science Foundations of China (31872975) and the Program of National Beef Cattle and Yak Industrial Technology System (CARS-37). China Agriculture Research System of MOF and MARA supported statistical analysis and writing of the paper. Technology Project of Inner Mongolia Autonomous Region (2020GG0210) also supported this work.

Conflicts of Interest: The authors declare no conflict of interest.

Code Availability: The code in this study is available in https://github.com/ML-GS/KRR-TPE.
References

1. Meuwissen, T.H.E.; Hayes, B.J.; Goddard, M.E. Prediction of total genetic value using genome-wide dense marker maps. *Genetics* **2001**, 157, 1819–1829. [CrossRef]

2. Hayes, B.J.; Bowman, P.J.; Chamberlain, A.J.; Goddard, M.E. Invited review: Genomic selection in dairy cattle: Progress and challenges. *J. Dairy Sci.* **2009**, 92, 433–443. [CrossRef] [PubMed]

3. Hayes, B.J.; Visscher, P.M.; Goddard, M.E. Increased accuracy of artificial selection by using the realized relationship matrix. *Genet. Res.* **2009**, 91, 47–60. [CrossRef] [PubMed]

4. Rupp, R.; Mucha, S.; Larroque, H.; Mcewan, J.; Conington, J. Genomic application in sheep and goat breeding. *Anim. Front.* **2016**, 6, 39–44. [CrossRef]

5. Tusell, L.; Pérez-Rodriguez, P.; Forni, S.; Wu, X.L.; Gianola, D. Genome-enabled methods for predicting litter size in pigs: A comparison. *Animal* **2013**, 7, 1739–1749. [CrossRef]

6. Stock, K.F.; Jönsson, L.; Ricard, A.; Mark, T. Genomic applications in horse breeding. *Anim. Front.* **2016**, 6, 45–52. [CrossRef]

7. Kranis, A.; Gheyas, A.A.; Boschiero, C.; Turner, F.; Le, Y.; Smith, S.; Talbot, R.; Pirani, A.; Brew, F.; Kaiser, P. Development of a high density 600K SNP genotyping array for chicken. *BMC Genom.* **2013**, 14, 59. [CrossRef]

8. Zhao, Y.; Gowda, M.; Liu, W.; Würschum, T.; Reif, J.C. Accuracy of genomic selection in European maize elite breeding Populations. *Theor. Appl. Genet.* **2011**, 124, 769–776. [CrossRef]

9. Poland, J.; Endelman, J.; Da Wson, J.; Rutkoski, J.; Wu, S.; Manes, Y.; Dreisigacker, S.; Crossa, J.; Sanchez-Villeda, H.; Sorrells, M. Genomic Selection in Wheat Breeding using Genotyping-by-Sequencing. *Plant Genome* **2012**, 5, 103–113. [CrossRef]

10. Grenier, C.; Cao, T.V.; Ospina, Y.; Quintero, C.; Châtel, M.H.; Tohme, J.; Ahmadi, N. Accuracy of Genomic Selection in a Rice Synthetic Population Developed for Recurrent Selection Breeding. *PloS ONE* **2015**, 10, e0154976. [CrossRef]

11. Vanraden, P.M. Efficient methods to compute genomic predictions. *J. Dairy Sci.* **2008**, 91, 4414–4423. [CrossRef] [PubMed]

12. Meuwissen, T.H.; Solberg, T.R.; Shepherd, R.; Woolliams, J.A. A fast algorithm for BayesB type of prediction of genome-wide estimates of genetic value. *Genet. Sel. Evol.* **2009**, 41, 2. [CrossRef] [PubMed]

13. Gianola, D.; Okut, H.; Weigel, K.A.; Rosa, G.J. Predicting complex quantitative traits with Bayesian neural networks: A case study with Jersey cows and wheat. *BMC Genet.* **2011**, 12, 87. [CrossRef] [PubMed]

14. Carvalho, A.; Da, C.; Tiago, B.; Alves, F.; Rafael, E.; Frossard, R.; Roberto, C.; Galvo, D. Genome-wide prediction for complex traits under the presence of dominance effects in simulated populations using GBLUP and machine learning methods. *J. Anim. Sci.* **2020**, 98, skaa179.

15. Ornella, L.; Gonzalez-Camacho, J.M.; Dreisigacker, S.; Crossa, J. Applications of genomic selection in breeding wheat for rust resistance. In *Wheat Rust Diseases*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 173–182.

16. Ghafouri, F.; Alipour, S.; Mohamadian Jeshaghi, S. Application of machine learning approach and its subset algorithms in estimating genomic breeding values. *Prof. J. Domest.* **2020**, 20, 19–29.

17. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 2003.

18. Ansari, M.F.; Alankarkaur, B.; Kaur, H. A Prediction of Heart Disease Using Machine Learning Algorithms. In *Proceedings of the ICIPCN: International Conference on Image Processing and Capsule Networks*, Bangkok, Thailand, 27–28 May 2021.

19. Austin, P.C.; Tu, J.V.; Ho, J.E.; Levy, D.; Lee, D.S. Using methods from the data-mining and machine-learning literature for disease classification and prediction: A case study examining classification of heart failure subtypes. *J. Clin. Epidemiol.* **2013**, 66, 398–407. [CrossRef]

20. Shammut, M. Driverless Cars: A Historical Overview. 2020.

21. Mohammed, A.A.; Minhias, R.; Wu, Q.; Sid-Ahmed, M.A. Human face recognition based on multidimensional PCA and extreme learning machine. *Pattern Recognit.* **2012**, 44, 2588–2597. [CrossRef]

22. Libbrecht, M.W.; Noble, W.S. Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* **2015**, 16, 321–332. [CrossRef] [PubMed]

23. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, 596, 583–589. [CrossRef]

24. Kolosov, N.; Daly, M.J.; Artomov, M. Prioritization of disease genes from GWAS using ensemble-based positive-unlabeled learning. *Eur. J. Hum.* **2021**, 29, 1527–1535. [CrossRef]

25. Hao, T.; Nikoloski, Z. Machine learning approaches for crop improvement: Leveraging phenotypic and genotypic big data. *J. Plant Physiol.* **2021**, 257, 153354.

26. Hao, T.; Kük, A.; Nikoloski, Z. Integrating molecular markers into metabolic models improves genomic selection for Arabidopsis growth. *Nat. Commun.* **2020**, 11, 2410.

27. Manuel, G.; Leonardo, O.; Paulino, P.R.; Daniel, G.; Susanne, D.; José, C. Applications of Machine Learning Methods to Genomic Selection in Breeding for Rust Resistance. *Plant Genome* **2018**, 11, 170104.

28. Okut, H.; Wu, X.L.; Rosa, G.J.; Bauck, S.; Woodward, B.W.; Schnabel, R.D.; Taylor, J.F.; Gianola, D. Predicting expected progeny difference for marbling score in Angus cattle using artificial neural networks and Bayesian regression models. *Genet. Sel. Evol.* **2013**, 45, 34. [CrossRef]

29. Montesinos-López, O.A.; Martín-Vallejo, J.; Crossa, J.; Gianola, D.; Hernández-Suárez, C.M.; Montesinos-López, A.; Juliana, P.; Singh, R. A benchmarking between deep learning, support vector machine and Bayesian threshold best linear unbiased prediction for predicting ordinal traits in plant breeding. *G3 Genomes Genet.* **2019**, 9, 601–618. [CrossRef]
30. Ozaki, Y.; Tanigaki, Y.; Watanabe, S.; Onishi, M. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Cancun, Mexico, 8–12 July 2020; pp. 533–541.

31. Nguyen, H.-P.; Liu, J.; Zio, E. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. *Appl. Soft Comput.* **2020**, *89*, 106116. [CrossRef]

32. Erwianda, M.S.F.; Kusumawardani, S.S.; Santosa, P.I.; Rimadana, M.R. Improving confusion-state classifier model using xgboost and tree-structured parzen estimator. In Proceedings of the 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 5–6 December 2019; pp. 309–313.

33. Shen, K.; Qin, H.; Zhou, J.; Liu, G. Runoff Probability Prediction Model Based on Natural Gradient Boosting with Tree-Structured Parzen Estimator Optimization. *Water* **2022**, *14*, 545. [CrossRef]

34. Rasmussen, C.E.; Williams, C. Gaussian processes for machine learning the mit press. *Camb. MA* **2006**, *32*, 68.

35. Usai, M.G.; Gaspa, G.; Macciotta, N.P.; Carta, A.; Casu, S. XVI th QTLMAS: Simulated dataset and comparative analysis of submitted results for QTL mapping and genomic evaluation. *BMC Proc.* **2014**, *8*, S1. [CrossRef]

36. Zhe, Z.; Erbe, M.; He, J.; Ober, U.; Li, J. Accuracy of Whole-Genome Prediction Using a Genetic Architecture-Enhanced Variance-Covariance Matrix. *G3 Genes Genomes Genet.* **2015**, *5*, 615–627.

37. Li, H.; Su, G.; Jiang, L.; Bao, Z. An efficient unified model for genome-wide association studies and genomic selection. *Genet. Sel. Evol.* **2017**, *49*, 64. [CrossRef] [PubMed]

38. Purcell, S.; Neale, B.; Todd-Brown, K.; Thomas, L.; Ferreira, M.; Bender, D.; Maller, J.; Sklar, P.; Bakker, P.; Daly, M.J. PLINK: A tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **2007**, *81*, 559–575. [CrossRef]

39. Eckert, A.J.; Van Heerwaarden, J.; Wegrzyn, J.L.; Nelson, C.D.; Ross-Ibarra, J.; Gonzalez-Martinez, S.C.; Neale, D.B. Patterns of population structure and environmental associations to aridity across the range of loblolly pine (*Pinus taeda* L., Pinaceae). *Genetics* **2010**, *185*, 969. [CrossRef]

40. Cleveland, M.A.; Hickey, J.M.; Forni, S. A common dataset for genomic analysis of livestock populations. *G3 Genes Genomes Genet.* **2012**, *2*, 429–435. [CrossRef] [PubMed]

41. Hu, Z.L.; Park, C.A.; Wu, X.L.; Reecy, J.M. Animal QTLdb: An improved database tool for livestock animal QTL/association data dissemination in the post-genome era. *Nucleic Acids Res.* **2013**, *41*, D871. [CrossRef] [PubMed]

42. Zhe, Z.; Ober, U.; Erbe, M.; Hao, Z.; Gao, N. Improving the Accuracy of Whole Genome Prediction for Complex Traits Using the Results of Genome Wide Association Studies. *PLoS ONE* **2014**, *9*, e93017. [CrossRef]

43. Henderson, C.R. Best linear unbiased estimation and prediction under a selection model. *Biometrics* **1975**, *31*, 423–447. [CrossRef]

44. Habier, D.; Fernando, R.L.; Dekkers, J.C.M. The Impact of Genetic Relationship Information on Genome-Assisted Breeding Values. *Genetics* **2007**, *177*, 2389–2397. [CrossRef]

45. Rong, G.; Li, K.; Su, Y.; Tong, Z.; Liu, X.; Zhang, J.; Zhang, Y.; Li, T. Comparison of Tree-Structured Parzen Estimator Optimization in Three Typical Neural Network Models for Landslide Susceptibility Assessment. *Remote Sens.* **2021**, *13*, 4694. [CrossRef]