Multimedia Tools and Applications (2021) 80:30505–30522
https://doi.org/10.1007/s11042-021-10944-7

1155T: ADVANCED MACHINE LEARNING ALGORITHMS FOR BIO-MEDICAL DATA AND IMAGING

Recognition of splice-junction genetic sequences using random forest and Bayesian optimization

Abdel Karim Baareh[1] · Alaa Elsayad[2,3] · Mujahed Al-Dhaifallah[4]

Received: 3 February 2020 / Revised: 9 February 2021 / Accepted: 14 April 2021 /
Published online: 30 April 2021

Abstract
Recently, Bayesian Optimization (BO) provides an efficient technique for selecting the hyperparameters of machine learning models. The BO strategy maintains a surrogate model and an acquisition function to efficiently optimize the computation-intensive functions with a few iterations. In this paper, we demonstrate the utility of the BO to fine-tune the hyperparameters of a Random Forest (RF) model for a problem related to the recognition of splice-junction genetic sequences. Locating these splice-junctions prompts further understanding of the DNA splicing process. Specifically, the BO algorithm optimizes four RF hyperparameters: number of trees, number of splitting features, splitting criterion, and leaf size. The optimized RF model automatically selects the most predictive features of the training data. The dataset is obtained from the UCI machine learning repository where half of the records represent two different types of splice-junctions and the other half does not represent any splice-junction. Experimental results proved the advantage of the BO-RF with 99.96% and 97.34% training and test classification accuracies respectively. The results also demonstrated the ability of the RF model to select the most important features, ensuring the best possible results using Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and decision tree (DT) models. Some practical procedures in model development and evaluation such as out-of-bag error and cross-validation approaches are also referred to.

Keywords Splice junction recognition · Random forest · Bayesian optimization, feature selection, support vector machine · K-nearest neighbor · Decision tree

1 Introduction

The human eukaryotic deoxyribonucleic acid (DNA) represents the building block of life that holds the encoded genetic directives for living organisms. In the central dogma of molecular

✉ Alaa Elsayad
a.elsayyad@psau.edu.sa

Extended author information available on the last page of the article

biology, the transcription process converts the DNA information into a precursor mRNA. Then, the splicing process removes the non-coding regions (introns) and connects the coding regions (exons) to form the contiguous coding sequence (mRNA) which is in turn translated into protein as shown in Fig. 1. The proteins are built according to the instructions stored in the DNA sequence. Accordingly, the understanding and analysis of DNA and RNA genetic sequences play critical roles in the treatments of any genetic disorders [5, 30].

The main objective of this study is to build a recognition system to predict whether a particular sequence of DNA nucleotides includes an exon-intron border, intron-exon border, or neither of them. It is a challenging problem that requires the knowledge of characteristics, dependencies, and the relationship of nucleotides in the splice site surrounding regions. Usually, Genetic databases are imbalanced mixed with noise which weakens the learning process [19]. In the literature, this problem was treated as a classification task. Table 1 lists some recent studies that used different Artificial Intelligence (AI) models to automatically recognize these splice-junctions. The present study combined the high potential Random Forest (RF) model and the state-of-the-art Bayesian Optimization (BO) to build a splice-junction recognition system and to rank the input nucleotides according to their predictive power. RF model is an ensemble of decision trees with several successful applications in healthcare applications [15] and the BO algorithm has emerged as an efficient tool for optimizing computation-intensive functions. BO has proven highly effective in controlling machine learning and deep learning models [17]. Precisely, in this study, the BO algorithm optimizes four RF hyperparameters: number of trees, number of splitting features, splitting criterion, and leaf size. The optimized RF model automatically selects the most predictive nucleotides (features) of the training data. The use of the optimization algorithm gives reassurance to the merit of the resulting model because the more accurate the model, the more we can trust the resulting feature importance.

The BO algorithm is evaluated using the Gaussian process (GP), which is the standard surrogate model and three different acquisition functions: probability-of-improvement. Lower-confidence-bound, and expected improvement [9]. Our goal is to achieve the best optimization results in the fewest number of iterations. The proposed method is evaluated on a real-world dataset publicly available from the UCI repository which was obtained from the Genbank 64.1 primate data [31]. The optimized RF model automatically ranks the predictive features according to their importance and this arrangement is evaluated using the BO-optimized

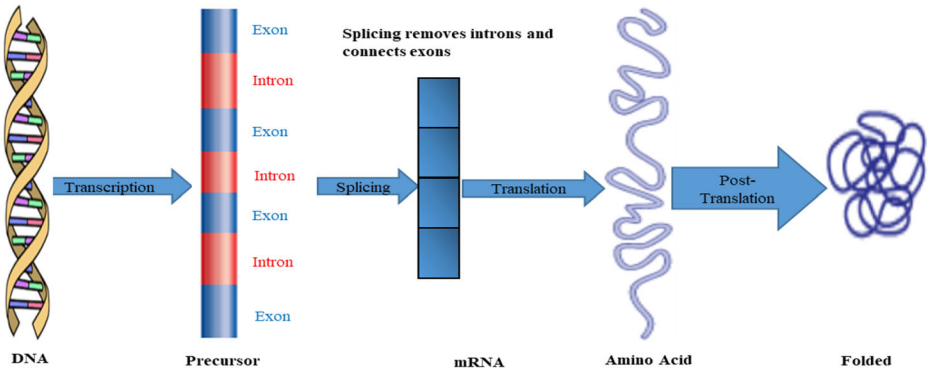


Fig. 1 Central dogma of molecular biology: RNA encoding and protein synthesis

Table 1 Related work on splice junction recognition

Dataset	Authors	AI Model	Feature Selection	Classification result
BG-570 dataset.	Zeng et al. [32] 2019	chi-square decision table (χ2-DT)		Test accuracy 92.40%.
Homo sapiens Splice Sites Database HS3D and GENCODE datasets	Zhang et al. [34] 2018	Deep learning		For HS3D, accuracy 0.95 for donor and for 0.92 for the acceptor. For GENCODE dataset, accuracy 0.91 for donor and 0.89 for acceptor
Homo sapiens Splice Site Data set (HS3D)	Pashaei et al. [25] 2017	Markovian encoding and RF	RF	Best results 96.62 AUC for balanced acceptors, 97.97 AUC for balanced donors
Homo Sapiens Splice Site Dataset (HS3D).	Meher et al. [22] 2016	RF, SVM, and Neural Network (NN)		For balanced dataset (P-1 encoding), RF achieved 0.940 F- measure (α=1)
Human, cattle, fish, and worm datasets.	Meher et al. [23] 2016	SVM	F-score	96.05%, 96.96%, 96.95%, 96.24% respectively.
Genbank 64.1 dataset	Htike et al. [12] 2013	Naïve Bayes	Genetic algorithm	96.68%. accuracy
Genbank 64.1 dataset	Cervantes et al. [4] 2011	Three SVM implementations: SMO, LibSVM, and Simple SVM		98.6%, 98:6%, and 98:2%, respectively
Genbank 64.1 dataset.	Damaševicius et al. [8] 2008	SVM with Power Series Kernel		90.59% for the F-measure (a combination of precision and recall metrics.)

versions of three popular AI models: SVM, KNN, and DT trained using the cross-validation approach. Overall, the contributions of this study are as follows:

- A hybrid machine learning model BO-RF for the recognition of the DNA splice junction sequence.
- Evaluation of three BO acquisition functions: expected improvement, lower-confidence-bound, and probability-of-improvement.
- Automatic feature ranking using the sensitivity analysis of the optimized RF model
- Performance evaluation of the optimized versions of RF, SVM, KNN, and DT models using the RF-based selected features.

The remainder of this paper is organized as follows: Section 2 describes the proposed methodology including the dataset description, the cross-validation approach, and the performance metric. Section 3 briefly presents the RF modeling, the out-of-bag error, and feature importance ranking. Section 4 reviews the BO algorithm including the definition of the Gaussian process model and three acquisition functions. Finally, all experimental results and discussions are presented in Section 5.

2 Methodology

For this study, we suggested a two stages method based on the RF classification model. In the first stage, we fine-tuned the RF hyperparameters using three different BO acquisition

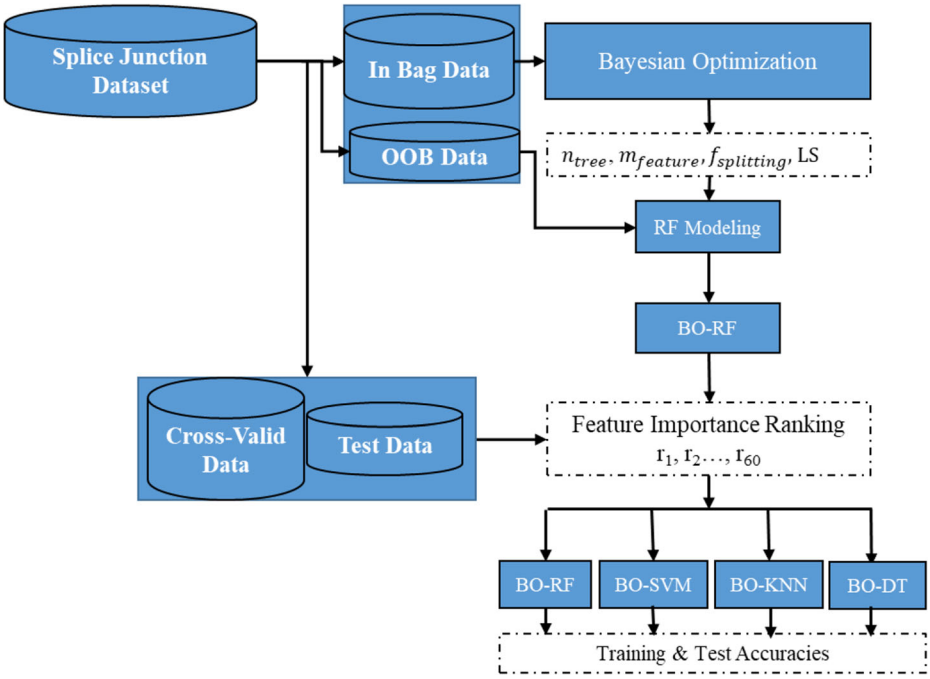


Fig. 2 Workflow for BO-RF modeling, automatic feature ranking, and model evaluation

functions: expected improvement, lower-confidence-bound, and probability-of-improvement based on the splice-junction dataset. The optimization process controlled four RF hyperparameters: number of trees, number of splitting features, splitting criterion, and leaf size. The objective was to minimize the out-of-bag (OOB) error function. The resulting optimal RF model ranked all predictive features according to their contribution to the classification process. In the second stage, the ranked features were evaluated using BO optimized versions of SVM, KNN, and DT models. The ranked features were included in the modeling process one by one according to their importance. These models were trained to minimize the cross-validation error. Fig. 2 shows the workflow.

2.1 Dataset description

The proposed method is evaluated on a real-world dataset collected from Genbank 64.1(FTP site: genbank.bio.net) and is available in the UCI machine learning repository [31].

The dataset consists of 3190 records with 60 features (DNA nucleotides) categorically classified with no missing values. They represent splice junctions where RNA splicing occurs. The 60 features represent the DNA nucleotides, starting at position −30 and ending at position +30. Each one is filled by one of {A, G, T, and C}. Other characters indicate ambiguity among the standard characters according to Table 2. Fields with ambiguities are processed as they are without being replaced. The dataset contains 767 exon-intron EI records (donor site), 768 intron-exon IE records (acceptor site), and 1655 not splice site N records with percentages of about 25% EI, 25% IE, and 50% N. The objectives of the proposed BO-RF classification model are to determine whether or not a particular sequence has a splice site, to define its type: EI (donor site) IE (acceptor site), or N (not a splice site), and to identify the most predictive features in the training data.

2.2 Model training and evaluation

The entire dataset is initially divided into training and test subsets. Then, in the first stage, the RF training process benefits from the OOB capability of the RF modeling methodology [2]. The OOB approach is described in Sec. 3. But, in the second stage, the objective is to compare the performances of the optimized versions of RF, SVM, KNN, and DT models using the ranked features. This time, models are trained to minimize the cross-validation error. K-fold cross-validation guarantees the participation of all training records in model learning and validation [27]. The algorithm partitions the whole training records into k separate subsets conducts k rounds of model training and validation, chooses one partition for validation, trains the model on the remaining k-1 partitions, and then uses the one that left-out to validate the model. For each fold, the algorithm calculates the

Table 2 Dataset feature description (ambiguity)

Dataset Character	DNA Meaning
D	A or G or T
N	A or G or C or T
S	C or G
R	A or G

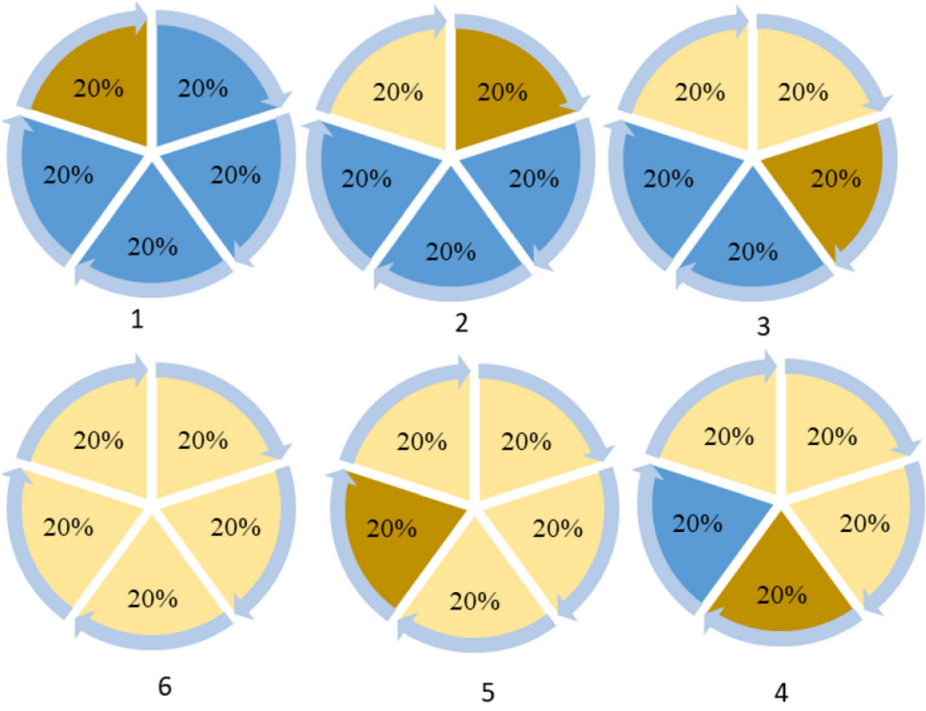


Fig. 3 Graphical depiction of the five-fold cross-validation procedure: the complete dataset is randomly split into five equal partitions. Four partitions are used to train the model and one for validation. The process is repeated five times such that all training records participate in model training and validation

classification error for the in-fold data by a model, which is trained on the records of the other k-1 partitions. The Graphical procedure of five-fold cross-validation is depicted in Fig. 3. The trained models are evaluated and compared based on the overall accuracy of training and test records. For a particular class y, the true positive (TP) equals the number of actual y records that are correctly classified and the (TN) is the number of non-y records that are properly classified to any $non-y$ class. The overall accuracy is computed as follows:

- *General Accuracy of y class*

$$\frac{TP_y + TN_y}{N} \tag{1}$$

where N is the total number of records. Then the overall accuracy in this multi-class classification problem is defined by

- *Overall Accuracy*

$$\frac{\sum_y TP_y}{N} \tag{2}$$

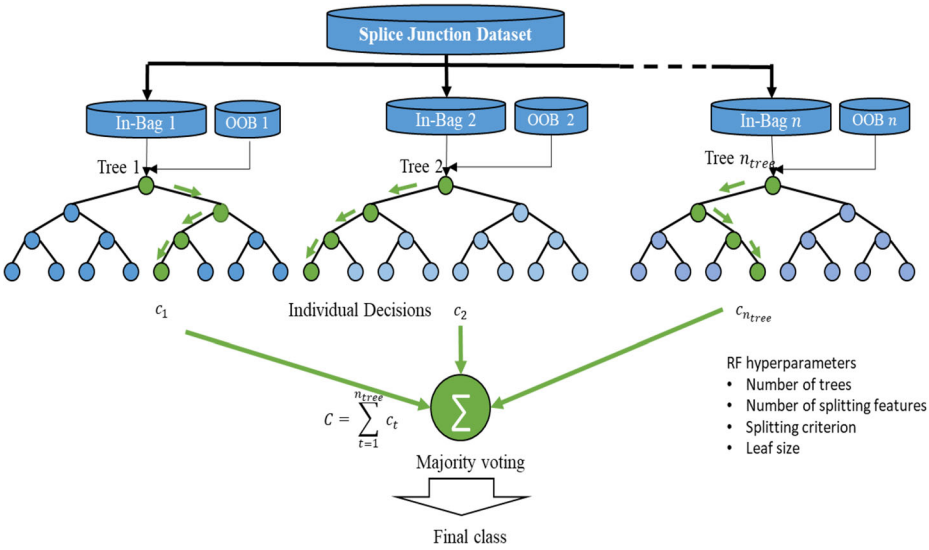


Fig. 4 Structure of random forest

3 Random forest (RF)

In the last years, the RF model has emerged as a practical tool for classification, regression, and visualization in particularly in bioinformatics [1, 13, 35]. It is an ensemble of high-performance decision trees, where predictions are produced by the majority of votes. To keep low bias and low dependence between trees, Breiman in [2] proposed two sources of diversity. First, each tree is trained on different subsets from the training data (bootstrap). Second, at every splitting node, the algorithm uses only a random subset of the available features. Individual trees are built without pruning, that is, they are left to grow to their fullest depths. Assuming $\mathfrak{D} = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$, represents the N training data, where $\mathbf{x}$ represents the input feature vector $\mathbf{x_i} = (x_{i,1}, \ldots, x_{i,p})^T$ with p features, and y_i is the output class. Let the number of decision trees is n_{tree}, number of splitting features is $m_{feature} < p$, splitting criteria is $f_{splitting}$, and trees are allowed to grow up such that it has $LS_{records}$ minimum number of records in terminal leaves (leaf size). The RF learning procedure works as illustrated by Algorithm 1. Fig. 4 shows the structure of the RF algorithm. All trees can be grown in parallel to reduce the bias and variance of the model at the same time. RF provides a reliable feature importance estimate and offers effective approximations of the test error without suffering the cost of repetitive model training associated with cross-validation as described in the subsequent sections.

Algorithm 1 Random Forest: Learning procedure

Input:

Training data $\mathfrak{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, number of decision trees n_{tree}, number of splitting features $m_{feature}$ to, splitting criterion $f_{splitting}$, and leaf size $LS_{records}$

Process

For j=1 to n_{tree}

1. Take a bootstrap subset $\mathfrak{D}_j$ of size n from $\mathfrak{D}$, typically n is about two thirds (0.66) of N.
2. Train the j decision tree on $\mathfrak{D}_j$:
   a. At each decision node, start with all features
   b. Select randomly $m_{feature}$ features out of the p features.
   c. Find the best binary partition based on $f_{splitting}(x_{i,1}, \ldots, x_{i,m_{feature}})$.
   d. Split the node into two descend nodes until $LS_{records}$ leaf size has achieved

End and **Output** n_{tree} trained decision trees (RF model)

3.1 Out-of-bag error

Out-of-bag error (OOB) is an estimation technique for measuring the prediction error of the RF model. Typically, the RF algorithm trains each tree using two-thirds of the training records (called in-bag data), and validate the tree with the remaining third (called out-of-bag OOB data) (Fig. 4). Eventually, each record contributes to the training of two-thirds of the trees and validating the other third. These OOB predictions are compared to their known classes to compute the OOB error to estimate the RF generalization. For a particular training record, the OOB error is the mean prediction error using only the trees that did not have this record in their in-bag data. Algorithm 2 describes the OOB prediction procedure.

Algorithm 2 OOB prediction procedure

Consider the bootstrap version $\mathfrak{D}_j = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and let $\hat{h}_j(\mathbf{x})$ refers to the predicted class label for record $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p})^T$, of the j^{th} tree.

1. For j=1 to n_{tree}
   For i=1 to N
   a. Consider $\mathfrak{R}_i = \{j : (\mathbf{x}_i, y_i) \notin \mathfrak{D}_j\}$ as the OOB records for the bootstrap $\mathfrak{D}_j$ and J_i as the cardinality of $\mathfrak{R}_i$
   b. Calculate the prediction of $\mathbf{x}_i$

$$\hat{f}_{OOB}(\mathbf{x}) = \arg max_y \sum_{j \in J_i} I(\hat{h}_j(\mathbf{x}_i) = y)$$

   End
End

For classification with zero-one error function, the generalization error rate is computed as follows:

$$E_{OOB} = \frac{1}{N} \sum_{i=1}^{N} I\left(y_i \neq \widehat{f}_{OOB}(\mathbf{x}_i)\right) \tag{3}$$

The Oob predictions $\widehat{f}_{OOB}(\mathbf{x})$ allow to compute the class-wise error rate for each class, and compute the OOB "confusion matrix" by cross-tabulating y_i and $\widehat{f}_{OOB}(\mathbf{x}_i)$.

3.2 Feature importance

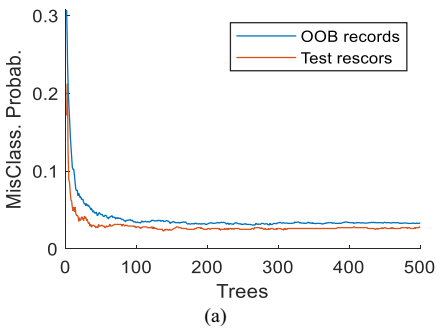
The RF algorithm can measure the importance of all features according to their contributions to the prediction of the output class [35]. This measure is calculated by directing all the OOB records down the RF trees and assessing the predicted output. Then, for every feature k, its values are randomly permuted in the OOB records, while preserving all other features fixed and once again, the algorithm generates the predicted outputs (for permuted data),i.e., we have two sets of OOB predictions: one set obtained for real data and one for feature-k-permuted data. Let $errOOB_t$ refers to the error of a single tree t for the real OOB data and $errOOB_{tk}$as error of the permuted OOB data. Then, the importance of the feature x_k is defined as follows:

$$Importance\ (x_k) = \frac{1}{n_{tree}} \sum_{t}^{n_{tree}} (errOOB_{tk} - errOOB_t) \tag{4}$$

The large value of this measure means great importance to the feature and vice versa. However, the importance values can be high even for features that are not relevant (predictive) of the class label, as long as the RF model can use them to overfit. It depends on how the RF model handles training data.

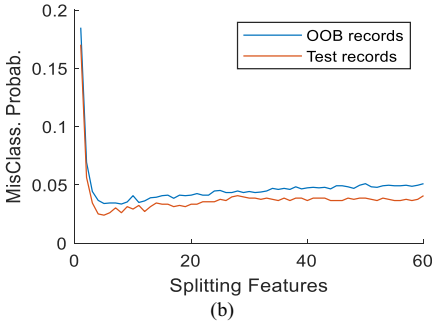
3.3 RF hyperparameter estimation

The RF algorithm has four hyperparameters to be prespecified: the number of trees n_{tree}, number of splitting features $m_{feature}$, splitting criterion $f_{splitting}$, and leaf size $LS_{records}$. Small values of n_{tree} lead to overfitting, while large values increase the model complexity. Typically, n_{tree} is set between 100 and 500. Regarding, $m_{feature}$ small values increase the RF diversity but, they may upsurge the error rate. Usually, it is set to the square root of the total number of features. There are three common splitting criteria: Gini's diversity index (GDI), Twoing rule, and maximum deviance reduction (simply Deviance) [18]. The minimum number of records per terminal leaf $LS_{records}$ controls the depth of individual trees. Figs. 5 shows the influences of these hyperparameters on the performance of the RF model for the recognition of splice-junction sites. Experiments were done using MATLAB package [21]. The dataset was divided into 70% training and 30% test subsets. The criteria are the misclassification probabilities for the OOB and test records. We set the default values to be 300 for n_{tree}, 8 for m_{attrib}, GDI for $f_{splitting}$, and 1 for minimum leaf size. Figure 5a shows the effect of the number of trees. The OOB error is 0.0305 with 263 trees. However, the minimum test error is 0.0230 with 137 trees. In general, the two curves have similar behaviors. Figure 5b shows the effect of the number of splitting features that are randomly selected at each splitting node where the minimum OOB



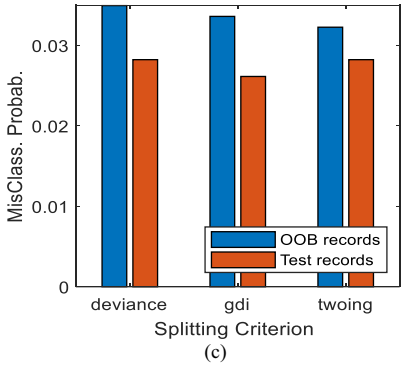
(a)

The minimum OOB error is 0.0305 with 263 trees and the minimum test error is 0.0230 with 137 trees.



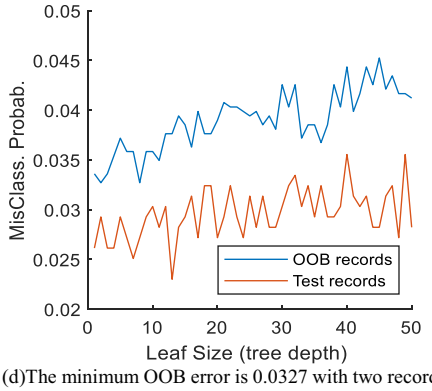
(b)

The minimum OOB error is 0.0336 with 8 features and the minimum test error is 0.0240 with 5 features.



(c)

The minimum OOB error is 0.0322 with Twoing criterion and the minimum test error is 0.0261 with GDI



(d)The minimum OOB error is 0.0327 with two records per leaf and the minimum test error is 0.0230 with 13 records per leaf.

Fig. 5 The misclassification probabilities for OOB and test records for (**a**) number of trees, (**b**) number of splitting features (**c**) splitting criterion (**d**) minimum leaf size

error is achieved with 8 features and the minimum test error requires 12 features. Similarly, Fig. 5c and d show the effects of splitting criterion and leaf size.

The optimal values of these hyperparameters ensure the construction of diverse and correct RF trees. Literature shows that different optimization algorithms were used to optimize all or some of these RF hyperparameters [10, 11, 26].

4 Bayesian optimization (BO)

BO algorithm is applied to select the hyperparameters of different AI models at hand based on the accuracy of the subsequent model classifications (objective functions). The algorithm runs iterative evaluations of a specified objective function exploring the solution space. The main advantage of BO is its success in finding good solutions with only a few iterations [3]. Its strategy maintains a surrogate model to represent the relationship between the hyperparameters and the objective function to guide the movement in the solution space. This surrogate model is progressively improved in a closed-loop method. Initially, the surrogate is prototyped based on some seed points and then this prototype selects the next point to evaluate the objective function. The resulting values improve the prototype itself, and so on until enough information about the objective function is available and the global minimum is generated. BO algorithm employs an acquisition function that uses the surrogate model to determine the next optimization. The Gaussian process is the most popular surrogate model while common acquisition functions include the probability of improvement, lower confidence bound and expected improvement.

4.1 Gaussian process model

Gaussian process (*GP*) is a probabilistic regression model that can represent a black-box objective function $f(\mathbf{x})$ using mean $m(\mathbf{x})$ and kernel $k(\mathbf{x}, \mathbf{x}')$ functions. It is assumed that f and its parameters $\mathbf{x}$ are assumed to have a common Gaussian distribution [28].

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right) \tag{5}$$

For simplicity, the mean function is assumed to be $m(\mathbf{x}) = 0$, i.e., the model is completely defined by its kernel function k. The ARD 5/2 Matérn function is a common kernel which is a twice differentiable function and depends only on the distance between points $\mathbf{x}$ and $\mathbf{x}'$ [24]:

$$K_{M52}\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2\left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right), \tag{6}$$

where $r = \sqrt{(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}+\mathbf{x}')}$ the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$, σ_f is the function standard deviation, σ_l is the characteristic length scale. Their values are found by maximizing the marginal log-likelihood of the available data $D_{1:t}=\{(\mathbf{x_i}, y_i)\}_{i=1}^{t}$ where t is the iteration index. Once the kernel is determined, the distribution at any new location $\mathbf{x}_{t+1}$can be predicted as follows:

$$P\left(y_{t+1}|\mathcal{D}_{1:t}, \mathbf{x}_{t+1}\right) = \mathcal{N}\left(\mu_t(\mathbf{x}_{t+1}), \sigma_t^2(\mathbf{x}_{t+1}) + \sigma_{noise}^2\right) \tag{7}$$

$$\mu_t(\mathbf{x}_{t+1}) = \mathbf{k}^T \left[\boldsymbol{K}+\sigma^2_{noise}\mathbf{I}\right]^{-1} y^T_{1:t} \tag{8}$$

$$\sigma^2_t(\mathbf{x}_{t+1}) = k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}^T \left[\boldsymbol{K}+\sigma^2_{noise}\boldsymbol{I}\right]^{-1} \mathbf{k} \tag{9}$$

where $\boldsymbol{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}$

$\mathbf{k} = [k(\mathbf{x}_{t+1}, \mathbf{x}_1) k(\mathbf{x}_{t+1}, \mathbf{x}_2) \cdots k(\mathbf{x}_{t+1}, \mathbf{x}_t)]$.

Where σ^2_{noise}represents the noise variance.

4.2 Acquisition function

BO algorithm employs a certain acquisition function $\alpha(\mathbf{x})$ to derive the optimization process in the solution space. It is computed using the Gaussian μ_t and σ^2_t (as in Eqs. (8), (9)). Specifically, the algorithm works in a closed iterative approach. At each iteration, the point that maximizes $\alpha(\mathbf{x})$ represents the best guess to sample the objective function and then, the sampling results update the GP model then, the algorithm returns to maximize $\alpha(\mathbf{x})$ again but with the updated GP to guess a new candidate and so on. The iteration is continued until sufficient information about the objective function is available and then the global minimum is reached. Algorithm 3 and Fig. 6 illustrate the concept of the BO process.

Algorithm 3: Bayesian optimization algorithm [29].

1. Initially evaluate the objective function *n* times (seed points)
2. Prototype the GP model using the *n* seed points from step 1
3. Select the acquisition function α(·)
4. for *t* = 1, 2, . . ., *max_iteration* do
   a. Find the point $\mathbf{x_{t+1}}$ that maximizes the function α(·) over the current GP :
      $\mathbf{x_{t+1}} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}|\mathcal{D}_{1:t})$.
   b. Sample the objective function at $\mathbf{x_{t+1}}$: $y_{t+1} = f(\mathbf{x_{t+1}}) + \epsilon_{t+1}$
   c. Append the newly generated data $(\mathbf{x_{t+1}}, y_{t+1})$ to $\mathcal{D}_{1:t}$
      $\mathcal{D}_{1:t+1} = \mathcal{D}_{1:t} \cup (\mathbf{x_{t+1}}, y_{t+1})$
   d. Update the GP using $\mathcal{D}_{1:t+1}$
5. End

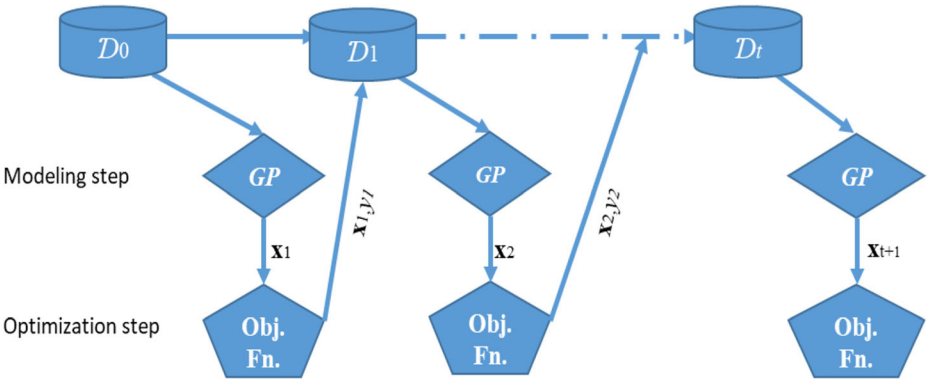


Fig. 6 A graph depicts the BO concept. The optimization refines progressively the GP model, which in turn is used to generate the best guess to sample the objective function. Guess and sample iteration is continued until the global minimum is reached

Several acquisition functions have been introduced in the literature. They have different approaches to balance between exploring areas with high variance σ_t^2 and exploiting those with low mean μ_t. In this paper, we evaluate the performance of three popular acquisition functions as follows:

- **Probability of improvement (PI)**

The PI acquisition function α_{PI} is the simple one that requires less computation [16]. It selects the candidate point $\mathbf{x}$ that most likely maximizes α_{PI} over the current best point $\mathbf{x}_{best}$as follows:

$$\begin{aligned} \alpha_{PI}(\mathbf{x}) &\triangleq P\Big(f(\mathbf{x}) < f(\mathbf{x}_{best})\Big) \\ \alpha_{PI}(\mathbf{x}) &= \Phi\left(\frac{f(\mathbf{x}_{best}) - \mu_t(\mathbf{x})}{\sigma_t(\mathbf{x})}\right) \end{aligned} \tag{10}$$

where $\Phi(\cdot)$ is the unit normal cumulative distribution function (CDF). The point that maximizes α_{PI} is the best guess to minimize the objective function f.

- **Lower confidence bound (LCB)**

LCB acquisition function α_{LCB} represents the statistical lower bound on the minimum lower confidence envelope $G_{LCB}(\mathbf{x})$ of the objective function. It is computed by subtracting the weighted value of the standard deviation σ_t from the value of the GP predictive mean μ_t as follows:

$$G_{LCB}(\mathbf{x}) = \mu_t(\mathbf{x}) - \kappa\sigma_t(\mathbf{x}) \tag{11}$$

Where the parameter κ manages the balance between exploitation and exploration, two is a common value of κ [7]. Then, α_{LCB} maximizes the negative of G_{LCB} as follows:

$$\alpha_{LCB}(\mathbf{x}) = \kappa\sigma_t(\mathbf{x}) - \mu_t(\mathbf{x}) \tag{12}$$

- **Expected Improvement (EI)**

EI acquisition function is considered the most common one due to the work of Jones et al. [14]. It considers the amount of expected improvement when selecting the next candidate point as follows:

$$\alpha_{EI}(x) = E[\max(0, f(\mathbf{x}_{best}) - \mu_t(\mathbf{x}))] \tag{13}$$

$$\alpha_{EI}(\mathbf{x}) = \begin{cases} (f(\mathbf{x}_{best}) - \mu_t(\mathbf{x})) \cdot \Phi(Z) + \sigma_t(\mathbf{x}) \cdot \Phi(Z) & (\sigma_t(\mathbf{x}) > 0) \\ 0 & (\sigma_t(\mathbf{x}) = 0) \end{cases} \tag{14}$$

where $Z = \frac{f(\mathbf{x}_{best}) - \mu_t(\mathbf{x})}{\sigma_t(\mathbf{x})}$ and $\Phi(\cdot)$ is the probability density function (PDF) for the normal distribution.

5 Experimental results and discussion

This section presents some experiments that demonstrate the utility of BO for finding the optimal hyperparameters for the RF prediction model. BO controls for RF hyperparameters: number of trees, number of splitting features, splitting criterion, and leaf size. The available data is divided into 70% training and 30 test subsets. Training data is used to construct the optimized RF model that minimized the OOB misclassification probability. Then, the optimized model is evaluated based on the test data. We compare the performances of three different BO acquisition functions: the probability of improvement, lower confidence bound, and expected improvement. Figure 7 shows the results of OOB misclassification probabilities for the three BO acquisition functions against the evaluation number, where the number of seed points is set to 10, and the objective function evaluation limit is set to 60 with no deterministic conditions on the optimizable variables or coupled constraints on the resulting models. Every hyperparameter has a prespecified range, number of trees from 100 to 500, number of splitting features from 2 to 10, the splitting criterion options are (GDI, Towing, and deviance) and finally, the leaf size ranges from 2 to 10. Fig. 7 shows that the probability of improvement acquisition function achieves the minimum OOB error. The resulting optimizable hyperparameters are the number of trees 375, splitting features 7, splitting criterion 'Twoing', and leaf size 1. Table 3 shows the values of OOB and the values of optimizable

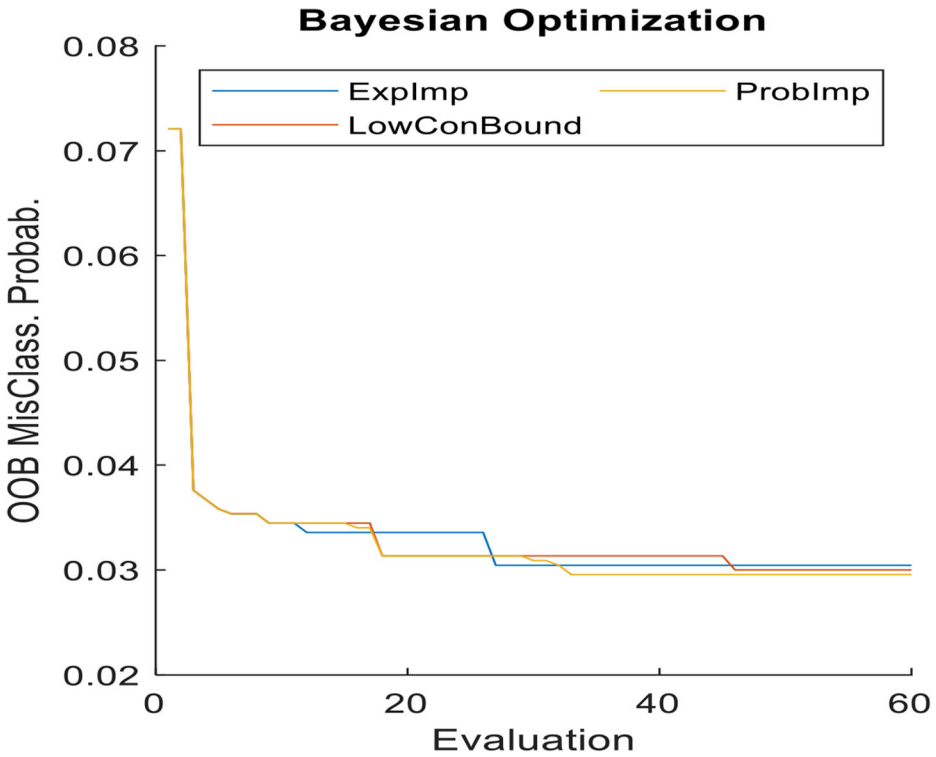


Fig. 7 The OOB misclassification probability of the RF model optimized using three different BO acquisition functions. Expected improvement achieves 0.030452 with 27 evaluations, lower confidence bound achieves 0.030004 with 46 evaluations, and the probability of improvement achieves 0.029557 with 33 evaluations

Table 3 Optimization results, the training and test accuracies for the three BO acquisition functions

Acquisition Function	OOB error	Evaluations	Trees	Splitting Features	Criterion	Min. Leaf Size	Training Accuracy	Test Accuracy
probability-of-improvement	0.029557	33	375	7	Twoing	1	99.96%	97.34%
lower-confidence-bound	0.030004	46	422	6	Twoing	1	99.96%	97.18%
expected-improvement	0.030452	27	500	6	Deviance	4	99.37%	97.49%

hyperparameters as well as the training and test classification accuracies for the three acquisition functions.

The probability of improvement function achieves 99.96% and 97.34% training and test accuracies respectively. Lower confidence bound achieves 99.96% and 97.18% and expected improvement achieves 99.37% and 97.49% respectively. In general, the results of the optimization of the three functions, as well as the results of the classification are very close to each other. The optimal RF model generates the relative feature importance shown in Fig. 8. This figure shows the 30 most important features in the training data based on the BO-RF model. The model selects automatically features that decrease the OOB prediction error. In addition to the RF model, the validity of this feature arrangement is examined using BO optimized versions of SVM, KNN, and DT models as follows:

(1) *SVM* model builds a decision hyperplane with the maximal margin width to divide the variable space into two regions (binary classifier) [6]. In the non-linear situations, the model allows a misclassification slake variable ξ around the margin with a regularization constraint C. In our experiments, the radial basis function (RBF) is selected as the kernel in the SVM model [20]. It is a common kernel with only one adjustable parameter (σ). The BO algorithm is applied to find the optimal values of C and σ that minimize the 10-fold cross-validation error.

(2) *KNN* model labels a record based on its similarity to the training ones. For a new record, KNN computes its distance from each of the training records and defines its neighbors. Then, the model places this new record within the class that contains the greatest number of K-nearest neighbors [33]. In this study, BO finds the optimal values for the number K, the type of distance measure, the distance weight, and the exponent that minimizes the 10-fold cross-validation error.

(3) *DT* model creates a classification tree by specifying the optimal partition features with the ability to handle both continuous and categorical variables. The model recursively separates records into branches to construct a tree to improve the classification accuracy [18]. In this study, the BO regulates the values of the maximum number of splitting, the minimum leaf size, and the splitting criterion to minimize the 10-fold cross-validation error.

A set of experiments was conducted to check the validity of the RF-based feature ranking using SVM, KNN, and DT. In the first experiment, the training and test data contained only the two most important features, and then BO optimized versions of the four models were trained and tested. Then, the features are added one by one according to their ranking and each time the training and testing processes are repeated. The accuracies of training and test subsets determine the quality of the classification model that appropriate to the required splice

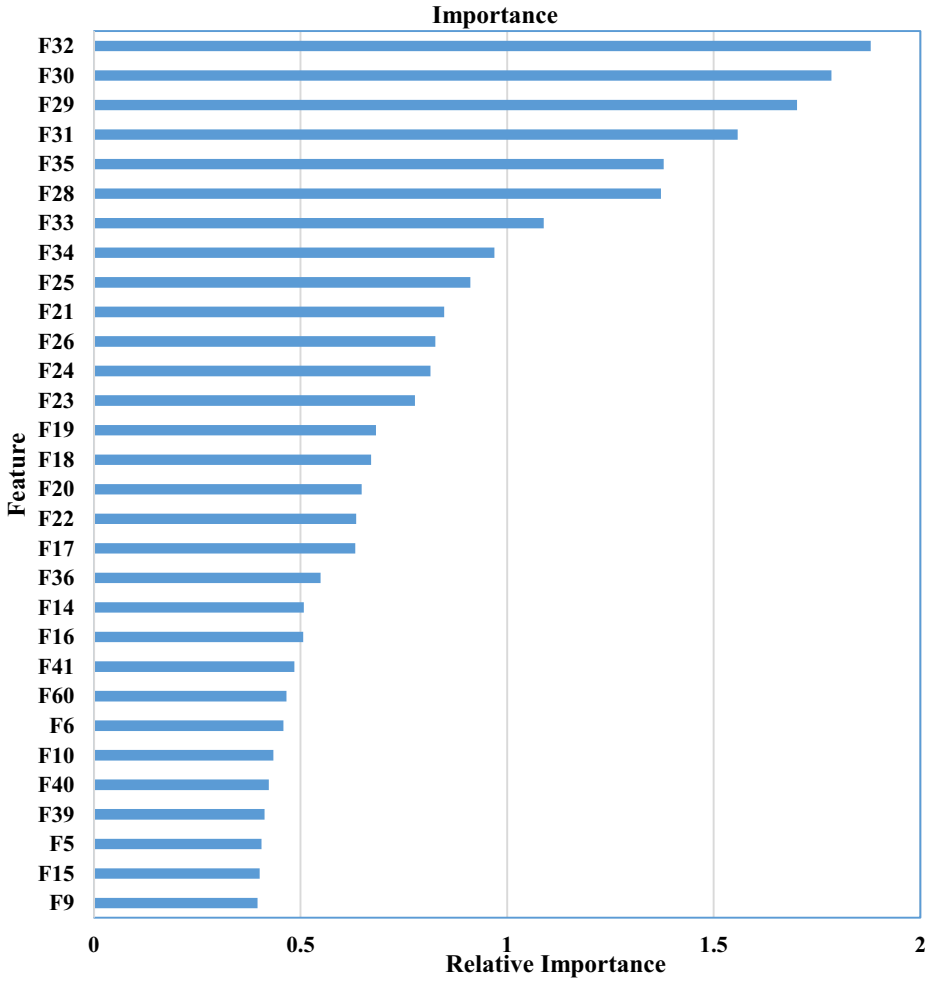
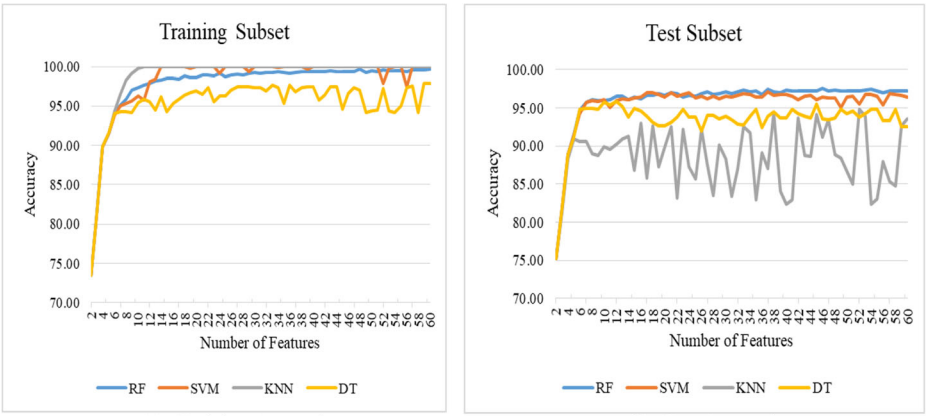


Fig. 8 Relative importance of the splice junction features using the BO-RF model

recognition. The resulting training and test accuracies for all these experiments are depicted in Fig. 9a and b respectively.

These curves demonstrate the high performance of all models using a few features arranged by the BO-RF model. All models achieve training and test accuracies greater than 70% using only the top two features. Using the top five features, they achieve training and test accuracies by more than 90%. With the addition of more ranked features, the performance of both RF and SVM improves, but the performance of DT and KNN fluctuates. Decision trees' performance changes by increasing the number of features, but it is not as bad as the KNN. If the number of features exceeds 7, the KNN begins to suffer from overfitting. On the contrary, Both RF and SVM RF and SVM provide precious results with very few features and their performances continue to improve and stabilize as the number of features increases. Using the top seven features, RF and SVM models achieve training and test accuracies of more than 95%. Using the



(a) Training Accuracies (b) Test Accuracies

Fig. 9 Training and test accuracies vs. Number of features (ordered by their relative importance generated from the BO-RF model) using BO optimized versions of RF, SVM, KNN, and DT classification models. **a** Training Accuracies, **b** Test Accuracies

full set of features, RF achieves 99.96% and 97.34% for training and test accuracies respectively, while SVM achieves 99.95% and 96.44% training and test accuracies.

Both RF and SVM achieve very high performance with few features (ranked by the RF model). In general, their performances are very close and outperforming the other two models (KNN and DT). These results demonstrate that the RF-based model facilitates the selection of the most predictive features ensuring the best possible predictive results using other prediction models. That is, the RF model can be used both as a predictor and as a feature selector with high efficiency. During all previous experiments, the BO methodology provided a very practical tool that enabled improving the performance of different prediction systems.

6 Conclusion

Understanding the genetic sequence and processes in the central dogma of molecular biology is an essential stage toward handling many genetic disorders. In this paper, we presented several experiments towards applying a set of artificial intelligence models to understand the sequences of nuclides and locating splicing sites. Experiments demonstrated the predictive power of an RF model in the recognition of the DNA splicing sites as well as its high ability to identify the most predictive features. That is, RF was used to achieve two objectives, firstly predicting the splicing sites, and secondly, to determine the most important features. To ensure the best results, the BO method has been used to adjust the model hyperparameters to achieve the minimum out-of-bag-error. The more accurate the model, the more we trust the resulting feature ranking. The optimization adjusted four hyperparameters; the number of trees, the number of splitting features, splitting criterion, and leaf size. The optimization worked using the Gaussian process surrogate with three different acquisition functions namely; the probability of improvement, lower confidence bound and expected Improvement. The probability of improvement yielded the best results, and the resulting RF model achieved classification accuracy for training and test data 99.96% and 97.34% respectively. The validity of RF-based feature ranking was tested using BO optimized

versions of SVM, KNN, and DT models, where they all achieved training and test accuracy greater than 70% using only the top two features and achieved more than 90% using the top five ones. In general, RF and SVM provided high and steady performance. BO provides an efficient and smart manner to fine-tune prediction models at hand.

References

1. Boulesteix A-L, Janitza S, Kruppa J, König IR (2012) Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Rev Data Min Knowl Discov 2(6):493–507
2. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
3. Brochu E, Cora VM, De Freitas N (2010) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv: 1012.2599
4. Cervantes J, Chau AL, Espinoza A T, Castilla JSR (2011) Fast Splice Site Classification Using Support Vector Machines in Imbalanced Data-sets. In Proceedings of the International Conference on Bioinformatics & Computational Biology (BIOCOMP), p. 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)
5. Cooper TA, Wan L, Dreyfuss G (2009) RNA and disease. Cell 136(4):777–793
6. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
7. Cox DD, John S (1997) SDO: A statistical method for global optimization. In: Alexandrov NM, Hussaini MY (eds) Multidisciplinary Design Optimization: State of the Art, pp. 315–329
8. Damaševicius R (2008) Splice site recognition in DNA sequences using k-mer frequency based mapping for support vector machine with power series kernel. In 2008 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 687–692. IEEE
9. Dewancker I, McCourt M, Clark S (2016) Bayesian optimization for machine learning: A practical guidebook. arXiv preprint arXiv:1612.04858
10. Elyan E, Gaber MM (2017) A genetic algorithm approach to optimising random forests applied to class engineered data. Inf Sci 384:220–234
11. Faris H, Aljarah I, Al-Shboul B (2016) A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering. In International Conference on Computational Collective Intelligence, pp. 498–508. Springer, Cham
12. Htike ZZ, Win SL (2013) Classification of eukaryotic splice-junction genetic sequences using averaged one-dependence estimators with subsumption resolution. Procedia Comput Sci 23:36–43
13. Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, Wang Y, Dong Q, Shen H, Wang Y (2017) Artificial intelligence in healthcare: past, present and future. Stroke Vasc Neurol 2(4):230–243
14. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492
15. Kaur P, Kumar R, Kumar M (2019) A healthcare monitoring system using random forest and internet of things (IoT). Multimed Tools Appl 78(14):19905–19916
16. Kushner HJ (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. J Basic Eng 86(1):97–106
17. Lévesque J-C (2018) Bayesian hyperparameter optimization: overfitting, ensembles and conditional spaces
18. Lin N, Noe D, He X, Phoam H (2006) Tree-based methods and their applications. Springer Handb Eng Stat London: Springer-Verlag:551–570
19. Lorena A C, Batista GEAPA, de Leon Ferreira ACP, Monard MC (2002) Splice Junction Recognition using Machine Learning Techniques. In WOB, pp. 32–39
20. Luts J, Ojeda F, Van de Plas R, De Moor B, Van Huffel S, Suykens JAK (2010) A tutorial on support vector machine-based methods for classification problems in chemometrics. Anal Chim Acta 665(2):129–145
21. Mathworks C (2018) MATLAB documentation
22. Meher PK, Sahu TK, Rao AR (2016) Prediction of donor splice sites using random forest with a new sequence encoding approach. BioData Min 9(1):4
23. Meher PK, Sahu TK, Rao AR, Wahi SD (2016) Identification of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features. Algorithms Mol Biol 11(1):16

24. Minasny B, McBratney AB (2005) The Matérn function as a general model for soil variograms. Geoderma 128(3–4):192–207
25. Pashaei E, Ozen M, Aydin N (2017) Splice site identification in human genome using random forest. Heal Technol 7(1):141–152
26. Probst P (2019) Hyperparameters, tuning and meta-learning for random forest and other machine learning algorithms. PhD diss, lmu
27. Rácz A, Bajusz D, Héberger K (2018) Modelling methods and cross-validation variants in QSAR: a multi-level analysis$. SAR QSAR Environ Res 29(9):661–674
28. Rasmussen CE (2006) CKI Williams Gaussian processes for machine learning
29. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems, pp. 2951–2959
30. Stranger BE, Dermitzakis ET (2006) From DNA to RNA to disease and back: the'central dogma'of regulatory disease variation. Hum Genomics 2(6):1–8
31. The Machine Learning Database Repository (n.d.) https://archive.ics.uci.edu/ml/datasets/ Molecular+Biology+(Splice-junction+Gene+Sequences)
32. Zeng Y, Yuan H, Yuan Z, Chen Y (2019) A high-performance approach for predicting donor splice sites based on short window size and imbalanced large samples. Biol Direct 14(1):6
33. Zhang S (2020) Cost-sensitive KNN classification. Neurocomputing 391:234–242
34. Zhang Y, Liu X, MacLeod J, Liu J (2018) Discerning novel splice junctions derived from RNA-seq alignment: a deep learning approach. BMC Genomics 19(1):971
35. Ziegler A, König IR (2014) Mining data with random forests: current options for real-world applications. Wiley Interdisciplinary Rev Data Min Knowl Discov 4(1):55–63

Affiliations

Abdel Karim Baareh[1] · **Alaa Elsayad**[2,3] · **Mujahed Al-Dhaifallah**[4]

[1] Computers Science Department, Al-Balqa Applied University, Ajloun College, Ajloun, Jordan

[2] College of Engineering, Prince Sattam Bin Abdulaziz University, Wadi Eldawasir, Kingdom of Saudi Arabia

[3] Computers and Systems Department, Electronics Research Institute, Giza 12622, Egypt

[4] Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Kingdom of Saudi Arabia