# Prediction of Splice Site Using AdaBoost with a new sequence encoding approach

Elham Pashaei[1], Alper Yilmaz[2], Mustafa Ozen[3], Nizamettin Aydin[1]
[1]Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey
[2]Department of Bioengineering, Yildiz Technical University, Istanbul, Turkey
[3]Department of Molecular Biology and Genetics, Biruni University, Istanbul, Turkey
e-mail: [1]elham.pashaei@std.yildiz.edu.tr; {[2]alyilmaz, [1]naydin}@yildiz.edu.tr; [3]mozen@biruni.edu.tr

***Abstract*—The Biological sequence data are increasing rapidly, so there is a vital need of effective method for gene detection. Predicting of splice site is an important part of gene finding. Therefore, attempts to improve the prediction accuracy of the computational methods for splice sites detection continue. In this paper we propose a hybrid algorithm for splice sites prediction by combining AdaBoost classifier with a novel nucleotide encoding method, namely FDDM. Our encoding method provides frequency difference between the true sites and false sites (FD) along with distance measure (DM). The proposed method produces an improvement in comparison with the result of current methods such as MM1-SVM, Reduced MM1-SVM, SVM-B, LVMM, DM-SVM, DM2-AdaBoost and MSC+Pos(+APR)-SVM, when applied to the HS3D dataset with repeated 10-fold cross validation. In addition, for demonstrating the stability of the method, we also applied it to NN269 dataset. The obtained results indicate that the new method is practicable and efficient.**

***Keywords—splice site prediction; nucleotide encoding method; AdaBoost classifier***

## I. INTRODUCTION

Eukaryotic genes consist of two parts, exons and introns. During DNA transcription genes are transcribed into mRNAs, but only some part of the genes carry codes for proteins. The coding portions of the genes are called exons [1]. Splice sites are boundaries between exon and intron. The intron-exon junction that is characterized by dinucleotide AG and exon-intron junction that is characterized by dinucleotide GT are known as acceptor splice site and donor splice site, respectively [2]. Splice site prediction in DNA sequence is a search problem for finding donor and acceptor boundaries.

Various computational methods such as hidden Markov model [3, 4] , decision trees [5], support vector machine [6-8], Bayesian network [9, 10] and artificial neural network [11, 12] have been produced for accurately predicting splice sites. However, existing complex dependencies between the bases around splice site [13] has caused that splice site prediction still remains a main bottleneck in genes detection. Hence, development of new methods is necessary for accurately predicting the splice sites [14].

The DNA sequence information is given as strings while the machine learning classifiers can only take numerical inputs. Hence the first step is to convert the DNA sequence into numbers [15]. Many encoding methods are available to model local sequence behavior such as first order Markov model (MM1), weight matrix model (WMM), maximal dependence decomposition (MDD) and so on.

Baten [6] has used MM1 encoding method to extract the features from splice sites sequences and has given them to support vector machine (SVM) for classifying splice sites. In order to focus only on more important features, Reduced MM1-SVM [16] has been developed using a subset of MM1 parameters as input for SVM to predict splice sites. Zhang [15] has combined linear SVM with a Bayes kernel (SVM-B) in order to improved accuracy and decreased time complexity. A length-variable Markov model (LVMM) [3] which is computationally expensive has been proposed [8]. The author has introduced 3 versions of the method, LVMM2, LVWMM2 and OLVWMM2 to make prediction on splice site. Although LVMM achieved good performance, it is difficult to determine the threshold parameters. Distribution of tri-nucleotides and Markov model-Support Vector Machine (DM-SVM) [8] is another method that has been proposed by Wei. The method has mapped candidate splice site sequences onto feature vectors with the distribution of tri-nucleotides and first order hidden Markov Model (MM1) parameters. Then, SVM classifier has been employed to predict splice sites. In [17] distribution of tri-nucleotides with a modified version of MM1 encoding method has been merged and has been fed to AdaBoost classifier (DM2-AdaBoost). Li [18] proposed a high accuracy splice site prediction method based on sequence component and position features. In this work, firstly the number and position of the consensus was determined by the Chi-square test, then the sequence multi-scale component (MSC), the position (Pos) and adjacent position relationship (APR) features were drawn out and finally a SVM classifier was constructed (MSC+Pos(+APR)-SVM).

This paper proposes a hybrid algorithm which combines effective features with the AdaBoost classifier. To capture the features from DNA sequences, we have employed a novel encoding method namely Frequency Difference Distance Measure (FDDM). Our encoding method calculates mono-nucleotides and pair-wise nucleotides frequency difference for true sites and false sites. Then, an F-score [19] feature ranking method is used to select informative features.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

We combine selected features with the result of DM encoding method. Ultimately, the FDDM is fed into AdaBoost. We have investigated effect of our method FDDM-AdaBoost on HS3D dataset and have compared efficiency of proposed method with MM1-SVM [6], SVM-B [15], LVMM [3], DM-SVM [8], DM2-AdaBoost [17] and MSC+Pos(+APR)-SVM [18] methods. Also, for demonstrating the stability of our method, we also applied it to NN269 dataset.

The remainder of this paper is divided into 3 sections. Materials and methods are described in section 2. In Section 3, experimental results are explained. Section 4 provides the conclusion.

## II. METHODS AND MATERIALS

### A. Encoding Method

Machine learning based methods are used for solving the problem of splice site prediction. The input of the machine learning classifier is numeric feature vectors. For converting DNA sequences to feature vectors, various DNA encoding methods are employed. The DNA encoding methods try to extract as much information as the DNA sequences have in order to increase accuracy of classifiers.

*1) MN with FDTF encoding:* In this paper we use mono nucleotide (MN) with frequency difference between the true sites and false sites (FDTF) [20]. In this encoding method each of the DNA bases is given an integer number as A-1, T-2, G-3 and C-4. Then a position weight matrix is obtained from the true set by enumerating the frequency of each nucleotide occurs at each position, given by (1).

$$M_{ij} = \frac{1}{n}\sum_{t=1}^{n} O_i(N_{tj}) \ , i = A,T,C,G \tag{1}$$

$$O_i(x) = \begin{cases} 1, i = x \\ 0, else \end{cases} \quad ; j = 1,2,\dots l$$

where $n$ is the number of sequences in the true set, $l$ is the length of a sequence and $O_i(N_{tj}) \in \{A,C,G,T\}$. In the same way, a position weight matrix $M_{4*l}$ can be obtained for the false site. Each element in this matrix determines the number of times that a given nucleotide has been seen at a given position. A MN-FDTF encoding matrix is derived by subtracting the true coding matrix from the false coding matrix [20].

*2) PN with FDTF encoding:* In the pair wise nucleotides (PN) with frequency difference between the true sites and false sites (FDTF) encoding method, each di-nucleotide is given an integer number as, AA-1, AG-2, AC-3, AT-4, GA-5, GG-6, GC-7, CA-9, CG-10, CC-11, CT-12, TA-13, TG-14, TC-15 and TT-16.

Similar to MN-FDTF encoding method, a position weight matrix $M_{16*l}$ is obtained for true and false sites. Again by subtracting the true coding matrix from the false one, PN-FDTF encoding method is achieved [20].

*3) Distribution of tri-nucleotides:* DMk (Distance Measure of K-tuple) are proposed by Wei [21] as a novel sequence similarity measure for clustering gene sequences. Then in [8], DM (Distribution of tri-nucleotide and Markov model) has been employed as a portion of encoding method for improving accuracy of splice site prediction. Wei in [22] utilized the DM encoding for producing feature vector under another name, sequential information.

The sequential relationships enable the inclusion of biological knowledge to understand compositional differences of nucleotides in a splice site [22]. On the other hand, finding behavior of candidate splice sites in the period-3 (triplet of three base pairs) format can be helpful for their recognition [23].

For a DNA sequence, there are 64 distinct tri-nucleotides (3-tuple) to be considered. For a DNA sequence $S$, $p_r$ is the location of the $r^{th}$ occurrence of 3-tuple $w$, where $p_0 = 0$. Using them $\alpha_r$ is calculated by (2),

$$\alpha_r = 1/p_r - p_{r-1}, 1 \le r \le m \tag{2}$$

in which $m$ stands for the number of occurrences of $w$. $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$ allows us to find all subsequent repeats of $w$. $\beta_j$ is defined as a partial sum of $\alpha_r$, and is calculated by (3):

$$\beta_j = \sum_{i=1}^{j} \alpha_r \ , 1 \le j \le m \tag{3}$$

A discrete probability distribution $Q = (q_1, q_2, \dots, q_m)$ can be constructed by $q_i = \beta_i / \sum_{i=1}^{m} \beta_i$, and $\sum_{i=1}^{m} q_i = 1$. The Pseudo-Entropy (PE) of discrete probability distribution is computed by (4) [8]. Algorithm 1 shows the pseudo-code of DM algorithm.

$$PE(q_1, q_2, \dots, q_m) = \sum_{i=1}^{m} q_i e^{1-q_i} \tag{4}$$

**Algoritm 1. DM method for encoding DNA**

**Input**: sequences $(S_1, S_2, \dots, S_N)$
**Output**: distribution of tri-nucleotides vector for each sequence consisting of 128 features, $(pe_1, pe_2, \dots, pe_{128})$

1) Split each sequence to upstream and downstream and for both of them search and locate each 64 distinct tri-nucleotides (3-tuple)
   a) For each 3-tuple, use (2) to calculate $\alpha_r, 1 \le r \le m$
   b) For each 3-tuple, use (3) to calculate $\beta_j, 1 \le j \le m$
   c) For each 3-tuple, use (4) to calculate $PE$
2) For each sequence, construct 64-component vector by $PE$ of all 3-tuples.

### B. AdaBoost

AdaBoost creates a strong classifier by iteratively considering weak classifiers (e.g. decision trees). It increases the weights of the mistakenly classified samples at each repetition. The AdaBoost algorithm of Freund and Schapire [24] is the first practical boosting algorithm, and remains one of the most widely used and studied in binary classification. AdaBoost.M1 classifier is one of the well-known and widely applicated version of boosting algorithm. In this paper we have employed AdaBoost.M1 classifier for splice site prediction, using "adabag" R package. Pseudo code for AdaBoost.M1 is given in Algorithm 2 [25]. We have experimentally adjusted the number of iteration 350.

**Algoritm 2. AdaBoost.M1 Methods**

**Input**: sequence of $m$ examples $< (x_1, y_1), \dots, (x_m, y_m) >$ with labels $y_i \in Y = \{1, \dots, k\}$
**Initialize**: $D_1(i) = 1/m$ for $i = 1, \dots, m$.
**Do For** $t = 1, \dots, T$:

- Call **weaklearn** using distribution $D_t$.
- Get back a hypothesis $h_t: X \longrightarrow Y$
- Calculate the error of $h_t$: $\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$.
- If $\epsilon_t > 1/2$ , then set $T = t - 1$ and abort loop.
- Set $\beta_t = \epsilon_t/(1 - \epsilon_t)$.
- Update distribution $D_t$: $D_{t+1}(i) \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i)=y_i \\ 1 & \text{otherwise} \end{cases}$
  where $Z_t$ is a normalization constant.

**Output** the final hypothesis:

$$h_{fin}(x) = argmax_{y\in Y} \sum_{t:h_t(x)=y} log \frac{1}{\beta_t}$$

### C. Data set

The experiments have been performed on the Homo Sapiens Splice Site Data set (HS3D [26]. The data set is composed of 2796 confirmed true donor sites, 2880 true confirmed acceptor sites, 271937 confirmed false donor sites and 329374 confirmed false acceptor sites. The length of each sequence is 140 nucleotides. The consensus nucleotides AG has located at position 69 and 70 for acceptor sites and consensus nucleotide GC (GT) has located at position 71 and 72 for donor splice sites. We have made balanced dataset (1:1) and unbalanced dataset (1:10) by selecting all the true splice sites for both of them. The ratio between number of true splice site and randomly selected false splice site in the balanced dataset is the same, while number of randomly selected false splice sites is 10 times more than true splice sites in unbalanced dataset.

We have performed an extra evaluation on the unbalanced NN269 dataset (1:4) [27] to estimate the reproducibility and consistency of our method. The dataset has gathered from 269 human genes that are composed of 1324 confirmed true acceptor sites, 5552 confirmed false acceptor sites, 1324 confirmed true donor sites and 4922 confirmed false donor sites. The training dataset for acceptor (donor) site are made up of 1116 true acceptor (donor) sites and 4672 false acceptor (4140 false donor) sites. The test dataset contains 208 true acceptor (donor) sites and 881 false acceptor (782 false donor) sites. The length of the sequences in acceptor splice site is 90 nucleotides whereas donor splice sites have the length of 15 nucleotides. The consensus dinucleotide AG in acceptor splice site is at positions 69 and 70 and the consensus nucleotides GT in donor splice site is at positions 7 and 8.

### D. Proposed Method

After calculating MN-FDTF, PN-FDTF and DM for whole sequences, we apply F-score to MN-FDTF and determine the position of the features whose F-score are more than average value of all F-score. By using these positions, we construct a new contiguous position vector (means that it contained consensus sites AG for acceptor and GT for Donor sites besides the selected position by F-score). The features from MN-FDTF and PN-FDTF are chosen using this new position vector. Ultimately, we merge MN-FDTF, PN-FDTF and DM vector as the input of classifier. We have called this encoding method as FDDM. Besides, by applying F-score feature ranking to DM encoding method we can make another version of proposed encoding method. Then we fed them to AdaBoost classifier. Algorithm 3 shows the pseudo-code of the FDDM algorithm.

**Algorithm 3. Proposed FDDM-AdaBoost Method**

**Input**: the candidate splice site sequences, $(S_1, S_2, \dots, S_N)$, length of sequence , $l$
**Output**: labels of unknown sequences
**Steps:**

1. For $n = 1$ to $N$ do
   Compute vector values DM= $(pe_1, pe_2, \dots, pe_{128})$ using Distance Measure encoding
   End for
2. Compute feature vectors MN-FDTF, $MN_{N*l}$
3. Compute feature vectors PN-FDTF, $PN_{N*l-1}$.
4. Calculate F-score of each feature in $MN_{N*l}$ and calculate the average value of all F-scores as the threshold, $p_t$ .Determine position of features from $MN_{N*l}$ whose F-score are more than the threshold $p_t$.
5. Construct a new contiguous vector of position (add position of AG for acceptor and position of GT for donor site among the selected position from previous step).
6. By employing this new position vector, the final features, $MN_{N*i}$, are chosen.
7. Employ the same vector of position and choose the contiguous features from the $PN_{N*l-1}$, $PN_{N*i}$.
8. (Optionally) Calculate F-score of each feature in $DM$ and calculate the average value of all F-scores as the threshold, $d_t$ . Choose the features whose F-score are more than the threshold $d_t$.
9. Merge ($MN_{N*i}$, $PN_{N*i}$, $DM_{N*128}$) respectively.
10. Apply classifier on the training set to obtain the model and use the model to predict the splice sites of testing sequences.

### E. Evaluation measures

This paper has employed 4 evaluation criteria namely a global accuracy ($Q^9$), Matthew's correlation coefficients ($Mcc$), sensitivity ($S_n$) and specificity ($S_p$). The advantages of $Q^9$ is that it is independent from distribution of the classes in the dataset and is computed by (5).

$$Q^9 = (1 + q^9)/2 \tag{5}$$

where

$$q^9 = \begin{cases} \dfrac{(TN - FP)}{(TN + FP)} & if \;\; TP + FN = 0 \\ \dfrac{(TP - FN)}{(TP + FN)} & if \;\; TN + FP = 0 \\ 1 - \sqrt{2\left[\left(\dfrac{FN}{TP + FN}\right)^2 + \left(\dfrac{FP}{TN + FP}\right)^2\right]} & if\ TP + FN \neq 0 \\ & and\ TN + FP \neq 0 \end{cases}$$

The TP, TN, FP and FN show the number of true positives, true negatives, false positives and false negatives, respectively. The better classification accuracy is shown by higher $Q^9$ and $Mcc$. The $Mcc$ is defined as follow, using (6).

$$Mcc = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP*FN)*(TN*FP)*(TP*FP)*(TN*FN)}} \tag{6}$$

The repeated 10-fold cross-validation is used for performance estimation in this study. Both of the datasets are divided into 10 equal parts (folds). 9 out of 10 folds are used as training set and the remaining fold is used as test set. Average performance estimation is calculated. We have repeated this process 5 times and final average has been reported.

In unbalanced dataset (1:10), the same proportion of the number of true sites versus false sites is considered for each fold. Besides, under-sampling technique [28-30] is applied on training set (9 out of 10 parts) to decrease number of false samples equal with number of true sample without modifying testing set.

For evaluation of NN269 dataset area under ROC curve and area under precision recall curve has been employed. The Receiver Operator Characteristic (ROC) curve is obtained by plotting sensitivity against 1-specificity and is used for visualizing the performance of the binary classifier. The area under ROC curve (AUC) is utilized for summarizing the performance in a single number. On the other hand, plotting True Positive Rate versus the False Positive Rate gives precision recall curve (PRC) and the area under PRC curve (auPRC) presents the performance in a single number. The increment in the value of AUC and auPRC lead to a more accurate model performance.

## III. Results and Discussion

### A. Prediction result for 1:1 data set

We have run repeated 10-fold cross validation on both balanced and unbalanced datasets and the results are shown in Table I, II, III and Table IV for donor and acceptor splice sites. The accuracy of our classification method is compared with these of MM1-SVM, Reduced MM1-SVM, SVM-B, LVMM, DM-SVM, DM2-AdaBoost and MSC+Pos(+APR)-SVM methods. For SVM-based method, we have used grid search method to find optimal parameters. The results of the LVMM , DM-SVM and MSC+Pos(+APR)-SVM methods are taken from [3], [8] and [18] respectively.

There are some differences between our results for MM1-SVM, Reduced MM1-SVM, SVM-B and DM2-AdaBoost and the results reported in Ref. [3] and [17]. This is due to several factors such as different parameter value used in SVM, different length of sequences used for extracting the information, different number of iteration used by AdaBoost classifier.

In particular, for clarifying that whether any performance improvements in our proposed method are achieved by using AdaBoost instead of SVM or whether this comes from the encoding method (or a combination of both) we have performed FDDM-SVM method additionally.

Result show that FDDM-SVM method improves the performance in comparison to the other methods, except in DM2-AdaBoost and MSC+Pos(+APR)-SVM (See Table I and II). Besides, using AdaBoost instead of SVM classifier causes our proposed method performs better.

The results of 1:1 dataset on both acceptor and donor dataset are shown in the Table I and II respectively. According to the results, our proposed method, i.e. FDDM-AdaBoost, outperforms all other methods clearly for both acceptor and donor splice site, except the MSC+Pos-SVM method for acceptor sites and MSC+Pos+APR-SVM method for donor sites. For acceptor and donor sites, the $Q^9$ score and $Mcc$ of FDDM-AdaBoost is obviously better than those of other methods. In addition, FDDM-AdaBoost also performs better than MM1-SVM, Reduced MM1-SVM, SVM-B, DM-SVM and DM2-AdaBoost in terms of sensitivity and specificity.

TABLE I. PERFORMANCE EVALUATION ON 1:1 ACCEPTOR SITES

| Methods | Acceptor Splice Site 1:1 | | | |
|---|---|---|---|---|
| | $S_n$ | $S_p$ | $Q^9$ | $Mcc$ |
| MM1-SVM | 90.51 | 86.89 | 88.48 | 77.48 |
| Reduced MM1-SVM | 90.84 | 87.12 | 88.76 | 78.03 |
| SVM-B | 91.78 | 87.21 | 89.17 | 79.10 |
| DM-SVM | 92.36 | 90.47 | 91.29 | - |
| DM2-AdaBoost | 93.68 | 91.11 | 92.19 | 84.84 |
| MSC+Pos-SVM | 95.38 | 93.26 | - | 88.70 |
| FDDM-SVM | 92.95 | 91.04 | 91.88 | 84.03 |
| FDDM-AdaBoost | 94.78 | 91.86 | 93.08 | 86.69 |

TABLE II. PERFORMANCE EVALUATION ON 1:1: DONOR SITES

| Methods | Donor Splice Site 1:1 | | | |
|---|---|---|---|---|
| | $S_n$ | $S_p$ | $Q^9$ | $Mcc$ |
| MM1-SVM | 93.40 | 91.20 | 92.16 | 84.64 |
| Reduced MM1-SVM | 93.70 | 91.51 | 92.42 | 85.26 |
| SVM-B | 95.01 | 90.26 | 92.22 | 85.37 |
| DM-SVM | 94.28 | 93.61 | 93.84 | - |
| DM2-AdaBoost | 96.17 | 93.82 | 94.80 | 90.03 |
| MSC+Pos+APR-SVM | 97.21 | 94.99 | - | 92.20 |
| FDDM-SVM | 96.15 | 93.70 | 94.68 | 89.90 |
| FDDM-AdaBoost | 96.98 | 94.20 | 95.32 | 91.23 |

### B. *Prediction results for 1:10 data set*

Due to existing more false splice sites than true sites in real genome sequences, we examined our method on the unbalanced (1:10) dataset. The results of 1:10 dataset on both acceptor and donor dataset are shown in Table III and IV respectively. From Table III, it can be seen that FDDM-AdaBoost outperforms all the methods significantly in acceptor site. The obtained result in Table IV shows that FDDM-AdaBoost outperforms the MM1-SVM, Reduced MM1-SVM, SVM-B, OLVWMM2, DM-SVM and DM2-AdaBoost method in donor sites, except MSC+Pos+APR-SVM method.

LVMM2 (OLVWMM2) have some parameters that have to been specified by grid based search before using but our method has less parameter to tune in comparison to LVMM2 (OLVWMM2) and our results are better.

TABLE III. PERFORMANCE EVALUATION ON 1:10 ACCEPTOR SITES

| Methods | Acceptor Splice Site 1:10 | | | |
|---|---|---|---|---|
| | $S_n$ | $S_p$ | $Q^9$ | $AUC$ |
| MM1-SVM | 90.28 | 87.13 | 88.54 | 95.45 |
| Reduced MM1-SVM | 90.76 | 87.25 | 88.80 | 95.46 |
| SVM-B | 92.26 | 87.80 | 89.74 | 96.09 |
| LVMM2 | 91.22 | 89.70 | 90.39 | - |
| DM-SVM | 92.15 | 90.73 | 91.36 | - |
| DM2-AdaBoost | 93.57 | 90.85 | 92.05 | 97.32 |
| MSC+Pos-SVM | 93.54 | 89.70 | - | 96.43 |
| FDDM-SVM | 93.11 | 90.83 | 91.86 | 97.40 |
| FDDM-AdaBoost | 94.94 | 92.22 | 93.40 | 98.09 |

TABLE IV. PERFORMANCE EVALUATION ON 1:10 DONOR SITES

| Methods | Acceptor Splice Site 1:10 | | | |
|---|---|---|---|---|
| | $S_n$ | $S_p$ | $Q^9$ | $AUC$ |
| MM1-SVM | 93.78 | 89.40 | 91.15 | 96.74 |
| Reduced MM1-SVM | 93.60 | 89.14 | 91.00 | 96.66 |
| SVM-B | 94.88 | 90.68 | 92.40 | 97.43 |
| OLVWMM2 | 94.24 | 92.42 | 93.23 | - |
| DM-SVM | 94.69 | 93.39 | 93.99 | - |
| DM2-AdaBoost | 96.16 | 93.75 | 94.77 | 98.50 |
| MSC+Pos+APR-SVM | 98.28 | 92.91 | - | 99.03 |
| FDDM-SVM | 95.50 | 93.63 | 94.46 | 98.55 |
| FDDM-AdaBoost | 96.83 | 94.18 | 95.28 | 98.81 |

In comparison to DM-SVM and DM2-AdaBoost methods, the FDDM-AdaBoost performs better than DM-SVM and DM2-AdaBoost methods for both donor and acceptor sites in term of specificity, sensitivity, global accuracy and AUC.

Also, in comparison of the our method's result with Li's work [18], i.e. MSC+Pos(+APR)-SVM method in 1:10 dataset, the AUC of our method for acceptor site and donor site are 98.09% and 98.81%, while the AUC of the MSC+Pos(+APR)-SVM for acceptor and donor site are 96.43% and 99.03% , respectively. It indicates that our method has 1.66% increase in acceptor and 0.22% decrease in donor sites compared to MSC+Pos(+APR)-SVM method. Although our method could not generate better accuracy than this method in 1:1 dataset, but our method has lower time complexity due to too large number of features that are produced by the MSC+Pos(+APR)-SVM method. Besides, the main advantage of our proposed method is that it is simple and only one parameter should be tuned, that is number of iteration in AdaBoost classifier. But MSC+Pos(+APR)-SVM method has more than 3 parameters for tuning without considering the parameters that should be tune for SVM classifier.

It can be concluded that overall FDDM-AdaBoost exhibits good prediction performance in both balanced and unbalanced data sets.

### C. *Evaluation on NN269*

In order to estimate the reproducibility and consistency of our method, we performed an additional evaluation on the NN269 dataset. Table V has summarized the predictive accuracy of proposed model and other methods in the term of AUC and auPRC for the NN296 dataset.

TABLE V. COMPARISON OF DIFFERENT MODELS ON NN269 DATASET

| Methods | Acceptor site | | Donor site | |
|---|---|---|---|---|
| | AUC | auPRC | AUC | auPRC |
| IC-S-SVM | 96.28 | - | 96.66 | - |
| MC-SVM | 96.74 | 88.33 | 97.64 | 89.57 |
| MM1-SVM | 97.41 | - | 97.90 | - |
| LIK | 98.19 | 92.48 | 98.04 | 92.65 |
| WD | 98.16 | 92.53 | 98.50 | 92.86 |
| WDS | 98.65 | 94.36 | 98.13 | 92.47 |
| FDDM-SVM | 97.93 | 92.28 | 98.31 | 92.77 |
| FDDM-AdaBoost | 98.51 | 93.69 | 98.20 | 93.02 |

IC-S-SVM= IC Shapiro SVM [16]; MC-SVM=Markov Chain-SVM [6] ; LIK= SVM using the locality improved kernel [31]; WD= weighted degree kernel[32] ; WDS= weighted degree kernel with shifts [33].

From Table V, the proposed method FDDM-AdaBoost demonstrates high performance in comparison to other methods except the WD [32] methods in term of AUC for donor site and WDS [33] method in both term of AUC and auPRC for acceptor sites. The FDDM-AdaBoost approximately shows good performance in the dataset NN269 despite of not demonstrating dominant win.

## IV. CONCLUSION

In this paper an AdaBoost based splice site detection method is proposed. The method uses a novel encoding method, FDDM, to preprocess the input sequence. FDDM is combination of frequency difference between true and false splice sites with the distribution of tri-nucleotides. The

experimental results demonstrated that the prediction accuracy of proposed algorithm is better than others. Moreover, the proposed method may be extended for identifying other specific sites in the sequences.

## REFERENCES


[1] A. Y. Salekdeh and K. C. Wiese, "Improving Splice-Junctions Classification employing a Novel Encoding Schema and Decision-Tree," *Evolutionary Computation (CEC),* pp. 1302 - 1307, 2011.

[2] T. Nassa, S. Singh, and N. Goel, "Splice site detection in DNA sequences using probabilistic neural network," *International Journal of Computer Applications,* vol. 76, 2013.

[3] Q. Zhang, Q. Peng, Q. Zhang, Y. Yan, K. Li, and J. Li, "Splice site prediction of human genome using Length-variable Markov model and feature selection," *Expert Systems with Applications,* vol. 37, pp. 2771-2782, 2010.

[4] M. Yin and J. Wang, "Effective hidden Markov models for detecting splicing junction sites in DNA sequences," *Information Sciences,* vol. 139, pp. 139-163, 2001.

[5] H. Lopes, C. Lima, and N. Murata, "A configware approach for high-speed parallel analysis of genomic data," *Jornal of Circuits Systems and Computers* vol. 16, pp. 527-540, 2007.

[6] A. Baten, B. Chang, S. Halgamuge, and J. Li, "Splice site identification using probabilistic parameters and SVM classification," *BMC Bioinformatics,* vol. 7(Suppl 5), 2006.

[7] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Ratsch, "Accurate splice site prediction using support vector machines," *BMC Bioinformatics,* vol. 8(Suppl 10), 2007.

[8] D. Wei, H. Zhang, Y. Wei, and Q. Jiang, "A Novel Splice Site Prediction Method using Support Vector Machine," *Journal of Computational Information Systems,* vol. 9, pp. 8053-8060, 2013.

[9] D. Cai, A. Delcher, B. Kao, and S. Ksif, "Modeling splice sites with Bayes networks," *Bioinformatics,* vol. 16, pp. 152-158, 2000.

[10] T. Chen, C. Lu, and W. Li, "Prediction of splice sites with dependency graphs and their expanded bayesian networks," *Bioinformatics,* vol. 21, pp. 471-482, 2005.

[11] J. Rajapakse and L. Ho, "Markov encoding for detecting signals in genomic sequences," *IEEE-Acm Transactions on Computational Biology and Bioinformations,* vol. 2, pp. 131-142, 2005.

[12] S. Marashi, H. Goodarzi, M. Sadeghi, C. Eslahchi, and H. Pezeshk, "Importance of RNA secondary structure information for yeast donor and acceptor splice site prediction by neural networks," *Comput Biol Chem,* vol. 30, pp. 50-57, 2006.

[13] K. Tsai, S. Lin, S. Shih, J. Lai, and C. Chenn, "Genomic splice sirte prediction algorithm based on nucleotide sequence pattern for RNA viruses," *Comput Biol Chem,* vol. 33, pp. 171-175, 2009.

[14] W. Bin and Z. Jing, "A Novel Artificial Neural Network and an Improved Particle Swarm Optimization used in Splice Site Prediction," *J Appl Computat Math,* vol. 3: 166, 2014.

[15] Y. Zhang, C.-H. Chu, Y. Chen, H. Zha, and X. Ji, "Splice site prediction using support vector machines with a Bayes kernel," *Expert Systems with Applications,* vol. 30, pp. 73-81, 2006.

[16] A. Baten, S. Halgamuge, and B. Chang, "Fast splice site detection using information content and feature reduction," *BMC Bioinformatics,* vol. 9(Suppl 12), 2008.

[17] E. Pashaei, M. Ozen, and N. Aydin, "Splice sites prediction of human genome using AdaBoost," presented at the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI): IEEE, Las Vegas, USA, 2016, p.^pp. 300-303.

[18] J. Li, L. Wang, H. Wang, L. Bai, and Z. Yuan, "High-accuracy splice sites prediction based on sequence component and position features," *Genetics and Molecular Research,* vol. 11, pp. 3432-3451, 2012.

[19] Y. W. Chen and C. J. Lin, "Combining SVMs with Various Feature Selection Strategies," in *Feature Extraction Studies in Fuzziness and Soft Computing,* . vol. 207, S. G. I. Guyon, M. Nikrevesh, L. Zadeh, Ed., ed New York: Springer, 2006, pp. 315-324.

[20] J. Huang, T. Li, K. Chen, and J. Wu, "An approach of encoding for prediction of splice sites using SVM," *Biochimie,* vol. 88, pp. 923-929, 2006.

[21] D. Wei, Q. Jiang, Y. Wei, and S. Wang, "A novel hierarchical clustering algorithm for gene sequences," *BMC Bioinformatics,* vol. 13: 174, 2012.

[22] D. Wei, W. Zhuang, Q. Jiang, and Y. Wei, "A new classification method for human gene splice site prediction," *Health Information Science Springer,* pp. 121-130, 2012.

[23] M. Akhtar, "Comparison of gene and exon prediction techniques for detection of short coding regions," *International journal of Information Technology,* vol. 11, pp. 26-35, 2005.

[24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* vol. 55, pp. 119-139, 1997.

[25] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence,* vol. 14, pp. 771-780, 1999.

[26] P. Pollastro and S. Rampone, "HS3D, a dataset of Homo sapies splice site regions, and its extraction procedure from a major public database," *International journal of Modern Physics,* vol. C13, pp. 1105-1117, 2002.

[27] M. Reese, F. Eeckman, D. Kupl, and D. Haussler, "Improved splice site detection in Genie," *Journal of Computational Biology,* vol. 4, pp. 311-324, 1997.

[28] R. Longadge, S. S. Dongre, and L. Malik, "Class Imbalance Problem in Data Mining: Review," *International Journal of Computer Science and Network (IJCSN),* vol. 2, 2013.

[29] W. J. Lin and J. J. Che, "Class-imbalanced classifiers for high-dimensional data," *Briefings in Bioinformatics,* vol. 14, pp. 13-26, 2012.

[30] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering(IJETAE),* vol. 2, pp. 42-47, 2012.

[31] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K. Müller, "Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites," *BioInformatics,* vol. 16, pp. 799-807, 2000.

[32] G. Rätsch and S. Sonnenburg, *Accurate Splice Site Detection for Caenorhabditis elegans*. London, England: MIT Press, 2004.

[33] G. Rätsch, S. Sonnenburg, and B. Schölkopf, "RASE: Recognition of Alternatively Spliced Exons in C. elegans," *Bioinformatics,* vol. 21, pp. 369-377, 2005.