Original Research

DIGITAL HEALTH

Digital Health
Volume 9: 1–15

DOI: 10.1177/20552076231165963
journals.sagepub.com/home/dhj
SAGE

iDHU-Ensem: Identification of dihydrouridine sites through ensemble learning models

Muhammad Taseer Suleman[1], Fahad Alturise[2], Tamim Alkhalifah[2] and Yaser Daanial Khan[1]

Abstract

Background: Dihydrouridine (D) is one of the most significant uridine modifications that have a prominent occurrence in eukaryotes. The folding and conformational flexibility of transfer RNA (tRNA) can be attained through this modification.

Objective: The modification also triggers lung cancer in humans. The identification of D sites was carried out through conventional laboratory methods; however, those were costly and time-consuming. The readiness of RNA sequences helps in the identification of D sites through computationally intelligent models. However, the most challenging part is turning these biological sequences into distinct vectors.

Methods: The current research proposed novel feature extraction mechanisms and the identification of D sites in tRNA sequences using ensemble models. The ensemble models were then subjected to evaluation using k-fold cross-validation and independent testing.

Results: The results revealed that the stacking ensemble model outperformed all the ensemble models by revealing 0.98 accuracy, 0.98 specificity, 0.97 sensitivity, and 0.92 Matthews Correlation Coefficient. The proposed model, iDHU-Ensem, was also compared with pre-existing predictors using an independent test. The accuracy scores have shown that the proposed model in this research study performed better than the available predictors.

Conclusion: The current research contributed towards the enhancement of D site identification capabilities through computationally intelligent methods. A web-based server, iDHU-Ensem, was also made available for the researchers at https://taseersuleman-idhu-ensem-idhu-ensem.streamlit.app/.

Keywords

Cancer, machine learning, dihydrouridine, ensemble, statistical moments, tRNA

Submission date: 29 December 2022; Acceptance date: 9 March 2023

Introduction

A bio-chemical process in which a primordial RNA is modified to develop functionally mature RNA is known as post-transcriptional modification (PTM). It is known that over 170 different RNA PTMs exist across all kingdoms of life. These modifications play critical roles in gene expression, metabolic responses, RNA folding, RNA localization, and many other diverse biological processes.[1,2] These modifications are also implicated in a wide variety of human diseases, including anemia, prostate cancer, tumorigenesis, respiratory chain defects, and intellectual disability.[3–7] The uridine modification in transfer RNA (tRNA) is the most prevalent of these PTMs, and it may undergo two of the most important alterations, known

[1]Department of Computer Science, School of systems and technology, University of Management and Technology, Lahore, Pakistan
[2]Department of Computer, College of Science and Arts in Ar Rass, Qassim University, Ar Rass, Qassim, Saudi Arabia

Corresponding author:
Fahad Alturise, Department of Computer, College of Science and Arts in Ar Rass, Qassim University, Ar Rass, Qassim, Saudi Arabia.
Email: falturise@qu.edu.sa

as the dihydrouridine (D) modification and the pseudouridine (ψ) modification. Dihydrouridine synthase (Dus), a member of the flavin enzyme family, catalyzes the production of D. The D-loop of tRNA is rich in the modified nucleoside dihydrouridine. D modifications have been linked to an increased risk of developing lung cancer in humans. Moreover, D alterations have been observed in several neurodegenerative diseases, such as Alzheimer's and Huntington's chorea.[8,9] D has been found to be involved in the regulation of gene expression in the heart, and its alterations have been linked to cardiovascular diseases.[10] D has been implicated in regulating gene expression during inflammation and its alterations have been linked to various inflammatory diseases. Consequently, the identification of D sites is crucial due to their importance in several biological processes. Traditional lab methods are used to find such modified locations, but these are time-consuming, expensive, and require a great deal of effort.[11] The sequence data helped to enhance the identification of D sites through *in silico* methods.

The most recent work on the identification of D sites was reported by Zhu et al.[12] In this work, the researcher derived significant features from the tRNA sequences using various feature extraction methods and developed five different machine learning models. The nucleotide chemical property along with the random forest model attained the highest accuracy (*Acc*), specificity (S_p), sensitivity (S_n), and Matthew's correlation coefficient (*MCC*), which were 92.73%, 0.98, 0.84, and 0.83, respectively. Similarly, Dou et al.[13] proposed a method, iRNAD_XGBoost, which was based on a feature selection strategy and an extreme gradient boost (XGBoost) model for the identification of D sites. The researchers reported high achievements of the proposed method in S_p and S_n accuracy metrics as compared to the existing predictors. Figure 1 represents a 3D structure of Dihydrouridine representing the double bonds.

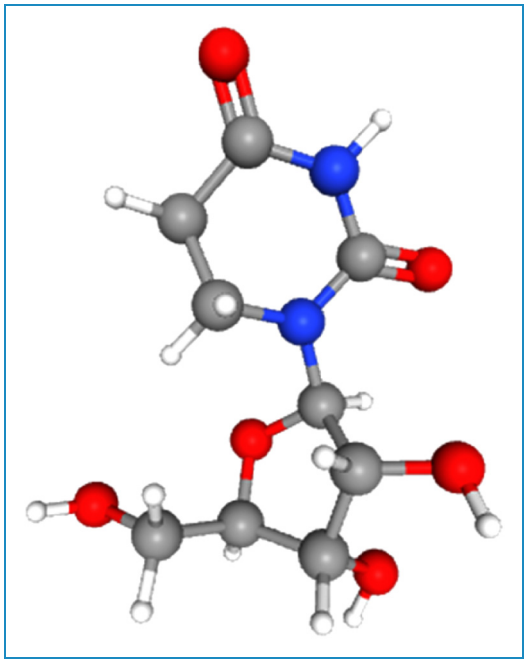

Figure 1. 3D structure of dihydrouridine.

Xu et al.[14] also built a predictor, iRNAD, to predict D modification from RNA samples. Nucleotide chemical property and nucleotide density were used to encode the samples. A classification model based on the support vector machine (SVM) was utilized, and its efficacy was evaluated using the jackknife test. The suggested model was found to have 96.18% *Acc*, with S_p and S_n values of 98.13% and 92%, respectively. In other research, Feng et al.[15] developed a SVM based ensemble method for the prediction of D sites in *Saccharomyces cerevisiae*. The features were developed through different feature extraction mechanisms, and a voting strategy was used to select the best features that were used as input to the SVM model for classification. Previously, Panwar and Raghava[16] worked on the prediction of uridine modifications through an SVM-based model. The model was then evaluated using a jackknife and an independent test set.

The present study aimed to identify D sites in tRNA sequences using ensemble methods. Several models were created, and they may be grouped into ensembles through the methods of stacking, bagging, and boosting. Models were trained, tested, and cross-validated using the benchmark dataset acquired from RMBase, which contained tRNA sequences of *Homosapiens, Mus musculus,* and *Saccharomyces cerevisiae*. By taking nucleotide location and formation into account, informative features from the sequences were derived. The dimensionality reduction of features was aided by using statistical moments.[17] Independent set testing and k-fold cross-validation were used to evaluate the efficacy of each ensemble model. The *Acc*, S_p, S_n, and *MCC* were used to observe the performance of each ensemble model. This research was comprised of many stages, such as selecting a group of representative benchmark datasets, extracting relevant features from tRNA sequence using positional and compositional information of nucleotides, developing a sample formulation, ensemble models training, and evaluation. Eventually, a web-based server was made available to the researchers for the enhancement in D site prediction. Figure 2 depicts the whole cycle of this research.

Materials and methods

Typically, the samples in the benchmark dataset have been verified by experimentation and are thus free of uncertainty. These samples are then used for training prediction models and performance evaluation of those models. The goal is to compile a single, comprehensive, and useful benchmark dataset. Extensive experimental validation studies, such as

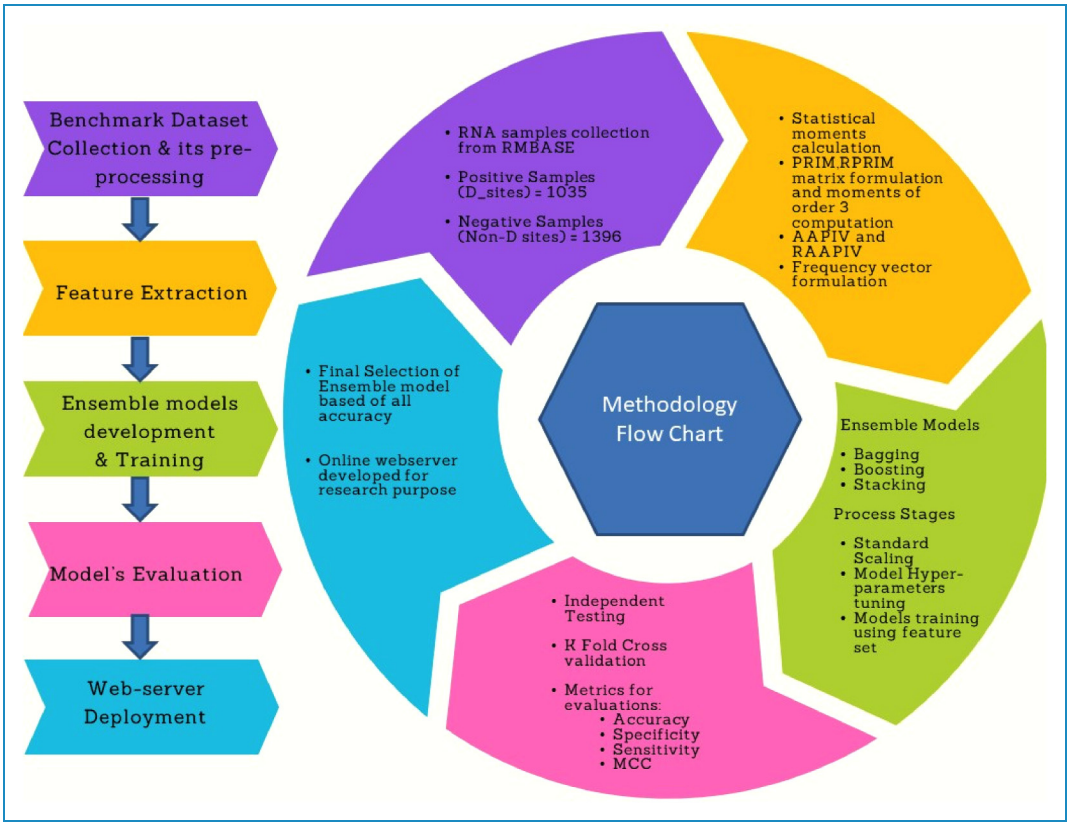

Figure 2. Complete flow diagram of current research methodology.

independent sets and validation tests (K-fold cross-validation and jackknife test), further corroborate the results of the experimentation. Since the conclusion is the result of a synthesis of several separate, unbiased dataset tests, it is crucial that the data are coherent and relevant. The dataset accumulation, model development, and results acquisition span almost 6 months. Directly, no humans or animals were involved in the study. However, experimentally verified tRNA sequences from three different species were used, including *Homosapiens*, *Mus musculus*, and *Saccharomyces cerevisiae*. This section includes the collection of benchmark datasets, feature vector generation, and the development of ensemble models.

Data samples

The data samples for the current study were obtained from RMBase. The acquired samples of tRNA belong to *Homosapiens*, *Mus musculus,* and *Saccharomyces cerevisiae.* It is important to mention here that similar data samples were used by Xu et al.,[14] Feng et al.,[15] and Duo et al.[18]

Positive and negative samples. The tRNA sample length was fixed to 41. This sample size was selected due to the optimal accuracy scores revealed during *in-silico* experiments. In each of the 41 nucleotide samples that made up the data sets, the "*U*" was in the middle, at position 21. A typical tRNA sample considered in this research study can be expressed as mentioned in equation (1):

$$B(U) = B_{-\Omega}B_{-(\Omega-1)}\ldots B_{-2}B_{-1}UB_{+1}B_{+2}\ldots B_{+(\Omega-1)}B_{+\Omega} \quad (1)$$

where $B_{-\Omega}$ represents the nucleotides from position 1–20, and the $B_{+\Omega}$ represents the nucleotides from position 22–41. Whereas the total length of a single tRNA sample can be represented as $2\Omega + 1$. There were a total of 1155 positive (D-sites) and 1669 negative (non-D sites) samples among the three species. While collecting negative samples, the position of uridine at the center and non-dihydrouridine modification was considered. However, once CD-HIT was set to 0.80 to get rid of duplicates, 1035 positive and 1396 negative samples were left.

Feature generation and representation from RNA samples

Because computer models cannot directly accept and analyze biological sequences, one of the most common procedures is to encode RNA sequences into fixed-length feature vectors.[19,20] These feature vectors are made up of numeric values that hold information about the RNA

sequences' attributes. The present work focused on the method for generating features based on the nucleotide's formation and position within a tRNA sequence. It was first suggested by Chou in their proposition of pseudo amino acid composition (PseAAC). That proposed technique was quickly being recognized as one of the most widely used and productive solutions to the issue of sequence pattern loss.[21] In the present work, pseudo-K-tuple nucleotide composition for feature vector generation was accomplished in a manner like PseAAC.[22,23] The nucleotide position and composition were used to create feature vectors for this study. Using the nucleotide formulation, $V_\phi(H)$, the samples in the dataset were characterized as described in equation (2):

$$V_\phi(H) = [V_1 V_2 V_3 \ldots V_U \ldots V_n]^T \tag{2}$$

In this study, a feature generation approach was used wherein a K-tuple nucleotide is represented by a vector whose components were all represented by the V. In this formula, "T" denotes the transposition of the collected feature set. Each site-specific nucleotide sample was 41 bp in length as represented in equation (3):

$$F = F_1F_2F_3 \cdots F_{18}F_{19}\boldsymbol{F}_{21} \cdots F_{39}F_{40}F_{41} \tag{3}$$

The central position, F_{21}, represents modified uridine, and the rest represents cytosine, guanine, adenine, and uridine within the nucleotide sequence.

Statistical moments computation. Moments are a statistical tool that statisticians and data analysts use to analyze a wide variety of data distributions.[24,25] The raw, Hahn and central moments were determined and then incorporated into features for dimensionality reduction. The reason for the selection of these moments was the unique properties associated with each moment class. The scale and location class were associated with the Hahn moment. However, central moments depend on the scale and vicinity.[26] The mean, variance, and asymmetry of the dataset were computed by employing raw and central moments. On the other hand, Hahn moments were determined using Hahn polynomials as a reference to preserve sequence order. According to equation (4), a matrix, J', is a m*n two-dimensional array with a single element, J_{pq}, representing the "qth" nucleotide in the "p" sequence.

$$J' = \begin{bmatrix} J_{11} & J_{12} & \ldots & J_{1q} \\ J_{21} & J_{22} & \ldots & J_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ J_{p1} & J_{p2} & \ldots & J_{pq} \end{bmatrix} \tag{4}$$

Raw moments were used to extract position-dependent features. For this purpose, the mean, variance, and non-symmetrical probability distribution were calculated.[27] Moments in their raw form were computed up to the third degree of polynomials (Z_{00}, Z_{01}, Z_{10}, Z_{11}, Z_{12}, Z_{21}, Z_{30}, Z_{03}), as shown in the expression (5), where $u + v$ denotes the sum of raw moments:

$$Z_{uv} = \sum_{a=1}^{m}\sum_{b=1}^{m} a^u b^v \beta_{ab} \tag{5}$$

The position has no bearing on the central moments. Instead, they are tied to the distribution's content and form. The central moments were determined using the random variable's aberrations from the mean.[28,29] The central moments were calculated for this investigation as indicated in equation (6):

$$n_{ij} = \sum_{b=1}^{n}\sum_{q=1}^{n} (b-x)^i (q-y)^j \beta_{bq} \tag{6}$$

Similarly, for the Hahn moments computation, Hahn polynomials were initially determined using the expression given in equation (7):

$$h_n^{u,v}(r,N) = (N+V-1)_n (N-1)_n \times \sum_{k=0}^{n} (-1)^k \frac{(-n)_k (-r)_k (2N+u+v-n-1)_k}{(N+v-1)_k (N-1)_k} \frac{1}{k!} \tag{7}$$

The orthogonal normalized Hahn of the two-dimensional data was calculated using the following expression (8):

$$H_{ij} = \sum_{q=0}^{N-1}\sum_{p=0}^{N-1} \beta_{ij} \tilde{h}_j^{u,v}(q,N) \tilde{h}_j^{u,v}(p,N), \; m,n = 0,1,\ldots N-1 \tag{8}$$

Position Relative Incidence Matrix (PRIM) determination. The purpose of this research was to improve the model's prognostic accuracy. This meant that a complete model for feature extraction was required to achieve the goal. Nucleotide base indexing within an RNA sequence can be easily formulated mathematically which helps in extracting statistical information. Keeping in view of this property, position relative incidence matrix (PRIM) was designed to obtain the count of each nucleotide's position with respect to others, that is, What would be the relative position of "A," "G," "U," and "C" with "A" which occurred at position 2? The matrix, SN_{PRIM}, in equation (9) was designed to obtain single nucleotide relative position information:

$$SN_{PRIM} = \begin{bmatrix} S'_{A\to A} & S'_{A\to G} & S'_{A\to U} & S'_{A\to C} \\ S'_{G\to A} & S'_{G\to G} & S'_{G\to U} & S'_{G\to C} \\ S'_{U\to A} & S'_{U\to G} & S'_{U\to U} & S'_{U\to C} \\ S'_{C\to A} & S'_{C\to G} & S'_{C\to U} & S'_{C\to C} \end{bmatrix} \tag{9}$$

Besides single nucleotide position, relativity paired (dual or tripartite) nucleotide combinations were also considered. A matrix, DU_{PRIM}, was also designed for the representation of dual nucleotide combinations within sequence, that is, CA, AG, UU, GU etc. The matrix is expressed in equation (10):

$$DU_{PRIM} = \begin{bmatrix} D_{AA\to AA} & D_{AA\to AG} & D_{AA\to AU} & \cdots & D_{AA\to j} & \cdots & D_{AA\to CC} \\ D_{AG\to AA} & D_{AG\to AG} & D_{AG\to AU} & \cdots & D_{AG\to j} & \cdots & D_{AG\to CC} \\ D_{AU\to AA} & D_{AU\to AG} & D_{AU\to AU} & \cdots & D_{AU\to j} & \cdots & D_{AU\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{GA\to AA} & D_{GA\to AG} & D_{GA\to AU} & \cdots & D_{GA\to j} & \cdots & D_{GA\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{N\to AA} & D_{N\to AG} & D_{N\to AU} & \cdots & D_{N\to j} & \cdots & D_{N\to CC} \end{bmatrix} \tag{10}$$

The matrix, DU_{PRIM}, yielded 256 coefficients, and these were reduced through statistical moments, which helped in feature reduction. For a detailed attributes withdrawal, a matrix, TR_{PRIM}, as expressed in equation (11) was designed to attain statistics about tri-nucleotide combination, that is, UUA, GCA, and CCC 4096 unique coefficients were generated from this matrix, and the statistical moments were calculated for reducing this number:

$$TR_{PRIM} = \begin{bmatrix} T_{AAA\to AAA} & T_{AAA\to AAG} & T_{AAA\to AAU} & \cdots & T_{AAA\to j} & \cdots & T_{AAA\to CCC} \\ T_{AAG\to AAA} & T_{AAG\to AAG} & T_{AAG\to AAU} & \cdots & T_{AAG\to j} & \cdots & T_{AAG\to CCC} \\ T_{AAU\to AAA} & T_{AAU\to AAG} & T_{AAU\to AAU} & \cdots & T_{AAU\to j} & \cdots & T_{AAU\to CCC} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ T_{AAC\to AAA} & T_{AAC\to AAG} & T_{AAC\to AAU} & \cdots & T_{AAC\to j} & \cdots & T_{AAC\to CCC} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ T_{N\to AAA} & T_{N\to AAG} & T_{N\to AAU} & \cdots & T_{N\to j} & \cdots & T_{N\to CCC} \end{bmatrix} \tag{11}$$

Reverse Position Relative Indices Matrix (RPRIM) formation. The primary objective of determining feature vectors is to collect as much valuable information as possible for the development of a robust prediction model. The availability of biological sequences has opened infinite possibilities for applying various mathematical and statistical techniques, which have helped in getting discrete information. The extracted information is fed into the artificial intelligence models, which help in the quick analysis and prediction of critical sites within sequences. It was observed that reversing the sequence can help obtain more obscured information from the sequences.[30,31] Three reverse position relative incidence matrices (RPRIM) were created on a similar pattern of PRIM. A general form of these matrices is shown in equation (12) in which, $R_{N\to j}$, represents any nucleotide, N, located at a specific index within a sequence with respect to, j_{th} base:

$$RV_{RPRIM} = \begin{bmatrix} R_{1\to 1} & R_{1\to 2} & R_{1\to 3} & \cdots & R_{1\to \mathrm{y}} & \cdots & R_{1\to \mathrm{j}} \\ R_{2\to 1} & R_{2\to 2} & R_{2\to 3} & \cdots & R_{2\to \mathrm{y}} & \cdots & R_{2\to \mathrm{j}} \\ R_{3\to 1} & R_{3\to 2} & R_{3\to 3} & \cdots & R_{3\to \mathrm{y}} & \cdots & R_{3\to \mathrm{j}} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ R_{\mathrm{x}\to 1} & R_{\mathrm{x}\to 2} & R_{\mathrm{x}\to 3} & \cdots & R_{\mathrm{x}\to \mathrm{y}} & \cdots & R_{4\to \mathrm{j}} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ R_{\mathrm{N}\to 1} & R_{\mathrm{N}\to 2} & R_{\mathrm{N}\to 3} & \cdots & R_{\mathrm{N}\to \mathrm{y}} & \cdots & R_{\mathrm{N}\to \mathrm{j}} \end{bmatrix} \tag{12}$$

For dimensionality reduction, moments were calculated for RPRIM matrices, which helped in obtaining more useful features to be used in the ensemble model's training.

Accumulative Absolute Position Incidence Vector (AAPIV). The accumulative absolute position incidence vector (AAPIV)[32] was formulated to deliver the accumulated information that is related to the location in which each individual nucleotide base is found. These vectors returned the total count of each nucleotide (either single or in combination) within each sequence. Three AAPIV are named as S_{AAPIV4} (13), $D_{AAPIV16}$ (14), and $T_{AAPIV64}$ (15), which hold count of single, paired, and tripartite nucleotides, respectively:

$$S_{AAPIV4} = \{\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4,\} \quad (13)$$

$$D_{AAPIV16} = \{\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3,\ldots,\ \mathcal{E}_{15},\mathcal{E}_{16},\} \quad (14)$$

$$T_{AAPIV64} = \{\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3,\ldots,\ \mathcal{E}_{63},\mathcal{E}_{64}\} \quad (15)$$

whereas $\mathcal{E}_N$(N = 1, 2, 3....) represents the statistical count of nucleotides as a single or combination.

Frequency Vector (FV) determination. Attribute generation relies heavily on being able to extract positional and compositional information from the sequence. The frequency count of each nucleotide inside a sequence may be used to infer information about the sequence's pattern.[33,34] To keep track of how many times each nucleotide or nucleotide pair appears in the sequence, we computed a frequency vector, denoted by, F, in which each frequency count of the n_{th} nucleotide is stored in f_n. The formula for calculating such a vector is expressed in equation (16):

$$F = \{f_1,\ f_2,\ \ldots..,\ f_n\} \quad (16)$$

Formulation of feature vector. Feature vectors for the AAPIV, FV, $V_\phi(H)$, PRIM, and RPRIM were generated from the benchmark dataset through the methods. Every feature in the dataset is represented by the feature input vector (FIV). The FIV rows correspond to the dataset's individual samples, with each FIV containing 522 unique values. In a similar vein, the Expected Output Vector was constructed using illustrative resources that were classified according to their expected function. This FIV is used to train, assess, and test ensemble methods.

Ensemble models development

For the current research study, D sites and non-D sites were classified using ensemble methods. Increased accuracy in predictions led to the adoption of ensemble approaches over more traditional machine learning models.[35,36] The ensemble procedures were divided into two major categories, that is, parallel and sequential. Classifiers that employ bagging, like random forest, use bootstrap aggregation to minimize variance by repeating the same procedure with random subsamples of the dataset. On the other hand, sequential ensemble methods like boosting allow the model to be improved upon by using large weights in comparison to earlier models. For predicting D sites, the current research made use of stacking, bagging, and boosting ensemble techniques. The ensemble models deployed in the research study were flexible and worked perfectly for identification purposes. Moreover, the model has been fine-tuned to alleviate aleatoric uncertainties as the meticulously collected dataset only contains verified biological sequences with tolerable homology. The overfitting and underfitting uncertainties were also removed through hyperparameter optimization of the models. Experiment and design specifics for these ensemble approaches were provided in the following sections.

Bagging ensemble. Bagging is a technique wherein the trained samples are split into smaller subsamples that are then used to update the base models. Subsampling was carried out by using a model based on row sampling with replacement. Test data were checked using the trained models, and a consensus was reached on a prediction method. The four bagging models that were developed and trained for this investigation were: bagging classifier, random forest, extra tree classifier, and decision tree classifier. Figure 3 is a schematic of a typical bagging ensemble model. The model's hyperparameter optimization was

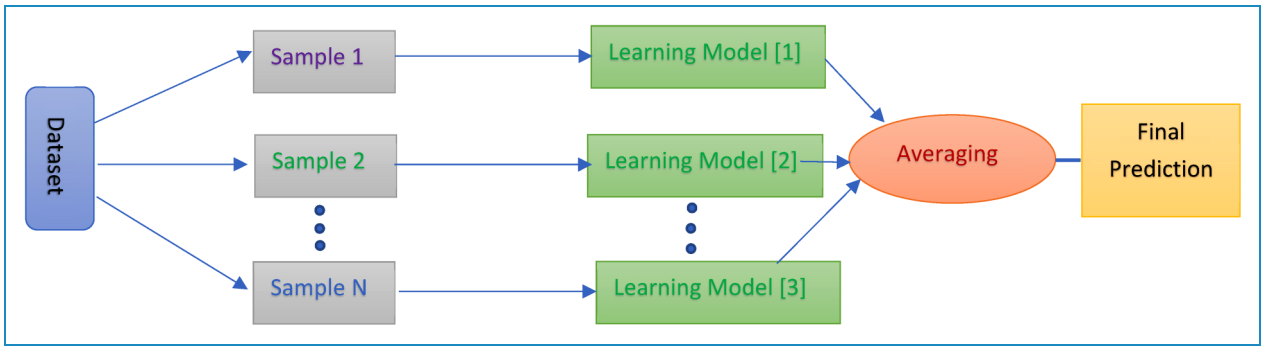

Figure 3. Bagging ensemble model.

considered for bagging ensemble models' development. The random forest was tuned by considering the number of estimators (*n_estimators* = 100), tree depth (*max_depth* = 80), *max_features* as "Auto," the minimum split of data samples (*min_samples_split* = 50), and *min_samples_leaf* to "10." Similarly, several parameters were considered for tuning the extra tree classifier, such as the number of estimators and tree depth. The number of estimators (*n_estimators*) for the extra tree classifier was set to 100. Similarly, the *max_depth* was set to "40." Features selection was made "*Auto*" in this classifier. Similarly, the bootstrap value was set to "*Bool*." For parameter optimization of the decision tree classifier, the splitter, *max_depth*, *min_samples_leaf*, *min_weight_fraction_leaf*, was set to "50," "10," "None," and "0.2," respectively. The bagging classifier was hyper tuned by setting the *base_estimator* as "*Decision tree classifier*" and the number of estimators (*n_estimators*) was set to "100." The *oob_score* was set to "True," and the random state was set as "0."

Boosting ensemble methods development. The boosting ensemble technique uses an optimization strategy that considers the results of previously run models. It does the differentiable loss in a sequential fashion. Several boosting ensemble techniques, including gradient boosting, histogram-based gradient boosting (HGB), Adaboost, and XGB, were employed for training in the present investigation. An example of the boosting ensemble model used in this research is shown in Figure 4. The XGB model was hyper-tuned by considering parameters such as maximum iteration (*max_iter*), depth of this boosting algorithm (*max_depth*), and the random_state, which was set to "0." Similarly, the Adaboost was tuned by considering *n_estimators*, *random_state,* and *min_weight_fraction_leaf,* which were set as "100," "None," and "0.2," respectively. The "Gradient Boost Classifier" was selected as a base estimator. Moreover, *max_iter*, *max_depth*, and *warm_start* were set to "300," "50," and "True" for tuning the HGB classifier, respectively. For optimizing the gradient boost algorithm, the learning rate, number of estimators, and criterion were selected. The learning rate was set to "0.2," while the number of estimators (*n_estimators*) and criterion was set to "150" and "*mse*," respectively.

Stacking ensemble model. Stacking typically considers several diverse weak classifiers known as base models. The whole dataset is divided into sub-samples and then base models are being trained in parallel. The meta-learner is then trained to provide a prediction based on the individual weak learners' predictions.[37] As can be seen in Figure 5, the current research made use of a variety of different classification methods, including a grading boost classifier as a meta classifier on top of artificial neural network, k-nearest neighbor, SVM, and multi-layer perceptron that were used as base classifiers. Each of the basic models was trained using the training data and then used to provide a prediction probability, P. Predictions were made on the test data using the output from the meta classifier, which had been trained using the outputs of the individual base models. Table 1 lists all classifiers along with their hyperparameter settings for optimal performance.

Evaluation metrics

In this research, four metrics, S_n, S_p, *Acc*, and *MCC* were used to evaluate the prediction models.[38–41] The TP denotes the D sites, whereas the TN denotes the non-D sites. A similar notation, FN, represents the total number of modified sites that were indeed actual D sites but were misidentified as false D sites. Furthermore, FP stands for the total number of false D sites that were misidentified. However, it is important to note that the measurements only apply to systems with a single class. The accuracy metrics equations have been mentioned in equation (17):

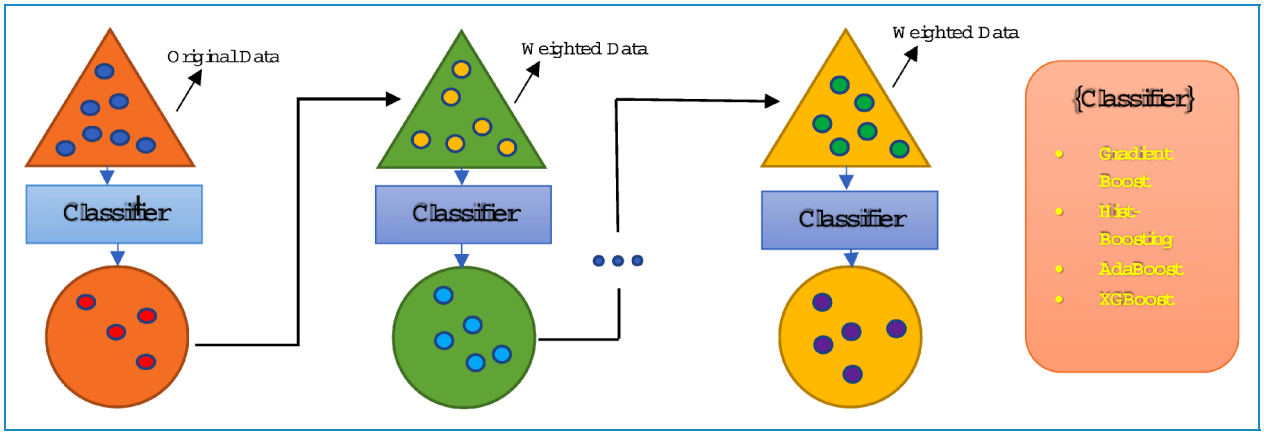

Figure 4. Boosting ensemble model.

$$\begin{cases} S_n = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} & 0 \le S_n \le 1 \\ S_p = \dfrac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} & 0 \le S_p \le 1 \\ Acc = TP + TN \ / \ (TP + FP + FN + TN) & 0 \le Acc \le 1 \\ MCC = (\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}) \ / \ \sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})} & -1 \le MCC \le 1 \end{cases} \quad (17)$$

These accuracy metrics were used to assess the prediction capabilities of each ensemble model deployed for this research. The true positive rate and true negative rate were used to exhibit the model's classification accuracy for actual D sites and non-D sites. These metrics also helped when the proposed model's performance was compared with other available models in identifying D modifications through sequence data.

Test methods

The ensemble models were assessed through well-known testing methods, that is, independent testing and 10-fold cross-validation.[42,43] It is worth mentioning that separate test samples were used from training samples in the independent set test. While cross-validation involved splitting the data into subsets, also called folds. Only a single fold was reserved for validation purposes, while the others were utilized in model training. To ensure that each fold had its chance to serve as a validation fold exactly once, the fold rotation was performed.

Results

The independent testing result is mentioned in Table 2. It can be observed from the Table 2 that the stacking ensemble model revealed the highest score in *Acc*, S_n, S_p, and *MCC* as compared to rest of the ensemble models. All boosting ensemble models had good *Acc* score with Gradient boost, HGB, Adaboost, and XGB ensemble achieving 0.92, 0.93, 0.92, and 0.91 values, respectively. Figure 6 represents all the bagging ensemble model's area under the curve revealed during independent set testing. However, all the bagging ensemble models failed to exhibit excellent performance relative to stacking ensemble and boosting ensemble models. However, the bagging ensemble model succeeded in achieving a better S_n value with the random forest, extra tree classifier, decision tree classifier, and bagging classifier achieving 0.89, 0.90, 0.85, and 0.91, respectively. The whole dataset is put through its paces using a cross-validation test, with the data set being divided into "k" separate folds.[44,45] This more stringent test proves that the model is stable. The dataset was divided into "*k*" folds. Model was trained using "*k-1*" folds with the remaining fold used as testing. In the research, the 10 folds were used for cross validation repeatedly. The results of cross validation are mentioned in Table 3. It can be observed from Table 3 that almost every ensemble model outperformed in the k-fold cross-validation. The random forest, gradient boost, HGB, AdaBoost, and stacking ensemble model performed well by revealing 0.98 *Acc*. However, for S_p accuracy metrics, the HGB and stacking model achieved the highest value of 0.99. Similarly, for S_n, only the extra tree classifier and the Adaboost ensemble model did not achieve good scores relatively. Similarly, for *MCC* metric, only HGB had shown maximum value of 0.98. Figure 7 shows the receiver operating characteristics graphs revealed by all ensemble models in 10-fold cross validation. Few statistical tests were also performed to

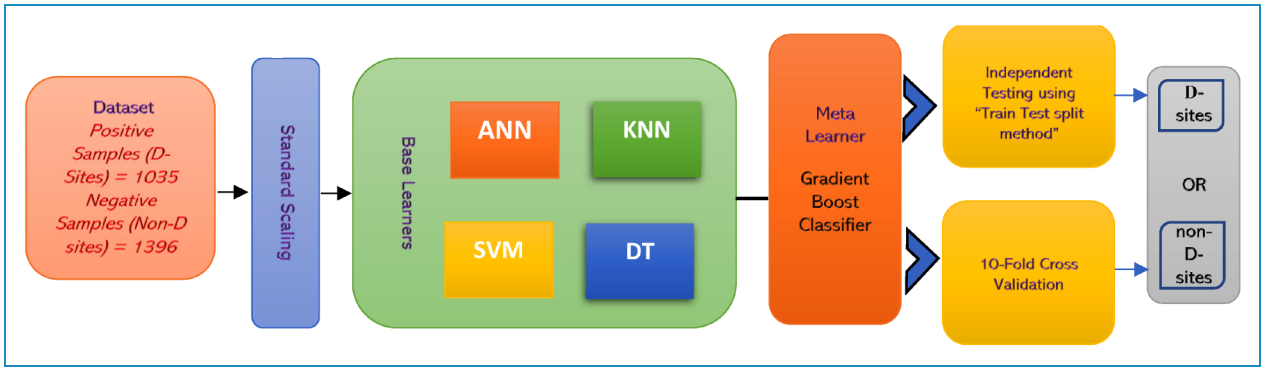

Figure 5. Stacking ensemble model deployed for this study.

Table 1. Hyper-parameters tuning of the stacking model's classifiers.

Base models	ANN	KNN	SVM	DT
Hyper-Parameters value(s)	*Hidden_layer_sizes = 5,2* *Random_state = 1* *Activation = relu* *Solver = lbfgs* *Learning rate = adaptive* *Alpha = 0.0001*	*k = 3*	*C = 10* *Gamma = 0.0001* *Kernel = rbf* *Coefficient = 0.0* *Probability = true* *Verbose = false* *Random_state = none*	*Splitter = 'random'* *Max_depth = 80* *min_samples_leaf = 4* random_state = None
Meta Classifier (Hyperparameters)	Gradient Boost classifier *n_estimators = 100, criterion =* 'mse'			

ANN: artificial neural network; KNN: k-nearest neighbor; SVM: support vector machine.

Table 2. Independent testing result of bagging, boosting, and stacking ensemble models.

Model		Acc	S_p	S_n	MCC	*p* values
Bagging	*Random Forest*	0.98	0.97	0.98	0.97	0.00167
	Extra Tree Classifier	0.97	0.97	0.96	0.96	0.00153
	Decision Tree	0.94	0.91	0.98	0.89	0.00141
	Bagging classifier	0.97	0.96	0.97	0.94	0.00120
Boosting	*Gradient Boost*	0.98	0.97	0.98	0.96	0.00136
	HGB	0.98	0.99	0.98	0.98	0.00166
	AdaBoost	0.98	0.98	0.96	0.97	0.00145
	XGBoost	0.97	0.96	0.98	0.94	0.00134
Stacking		0.98	0.99	0.98	0.97	0.00167
Standard deviation (σ)		0.012	0.022	0.008	0.025	0.00153

validate the performance of ensemble models applied in this study. The main objective of conducting these tests is to compare the learning algorithms for the accurate results which these provided for classification. A two-proportion test, also known as the Z test, was performed on the ensemble models. The Z test was used to validate whether the two samples were different or not. The critical value (p) should be less than 0.05 in order to reject the null hypothesis for exhibiting the difference between two samples. A resampled paired t-test used a predefined set of trials for the measurement of algorithm accuracy. McNemar's test is a statistical significance test used to determine whether the difference between two proportions in a 2×2 contingency table is statistically significant. The "p" value obtained through these aforementioned tests is represented in Table 4.

Discussion

The rapid and precise prediction of PTM sites has been made feasible by the availability of sequencing data and sophisticated computer methods. The accurate identification of such modified sites helps in the diagnosis of various PTM-linked diseases such as breast cancer,[46] acute infantile liver failure,[47] asthma,[48] diabetes,[49] and leukemia.[50] For the current research, independent testing and cross-validation were used to assess the performance of the model in making predictions. It is worth noting that the reliability and validity of the tests were independently evaluated using a data set that was kept completely distinct from the rest. But the whole dataset was utilized for the cross-validation. The model's effectiveness was evaluated using a variety of accuracy metrics. A violin plot[51] uses

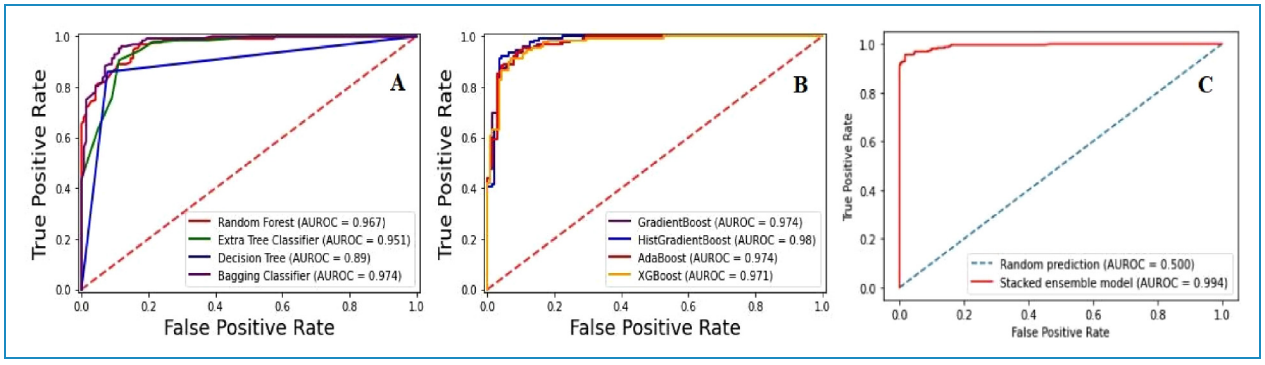

Figure 6. Independent testing receiver operating characteristics (ROC) curves (A). Bagging ensemble models ROC (B). Boosting ensemble models ROC (C). Stacking model ROC.

Table 3. K-fold cross-validation scores of *Acc*, S_p, S_p, and *MCC* for ensemble models.

Model		Acc	S_p	S_n	MCC	*p* value
Bagging	*Random Forest*	0.87	0.86	0.89	0.75	0.00195
	Extra Tree Classifier	0.89	0.88	0.90	0.79	0.00184
	Decision Tree	0.89	0.92	0.85	0.78	0.00176
	Bagging classifier	0.91	0.90	0.91	0.82	0.00163
Boosting	*Gradient Boost*	0.92	0.91	0.92	0.83	0.00191
	HGB	0.93	0.94	0.92	0.86	0.00121
	AdaBoost	0.92	0.92	0.92	0.84	0.00299
	XGBoost	0.91	0.93	0.89	0.83	0.00183
Standard deviation (σ)		0.018	0.024	0.022	0.033	
Stacking	*Stacked*	0.96	0.96	0.95	0.92	0.00163
	Baseline (KNN)	0.87	0.78	0.98	0.77	0.00192
	Baseline (DT)	0.93	0.92	0.95	0.87	0.00122
	Baseline (ANN)	0.90	0.84	0.97	0.81	0.00199
	Baseline (SVM)	0.95	0.95	0.96	0.91	0.00186

ANN: artificial neural network; KNN: k-nearest neighbor; SVM: support vector machine.

density curves to visually illustrate how numerical data is distributed over one or more groups. White dots may be used to indicate the median, black bars can show the interquartile range, and dark lines can show the ranges of the lowest and highest values around the bar. Figure 8 is a violin plot that shows the accuracy values found in each fold for the best stacking, bagging, and boosting ensemble models. The output of computationally intelligent models can be observed in detail through powerful visualization. Methods of boundary visualization have been used to show how well each ensemble model can predict and analyze.

Figure 9 displays the decision surface plots of the classification algorithms used in this study. The decision surface

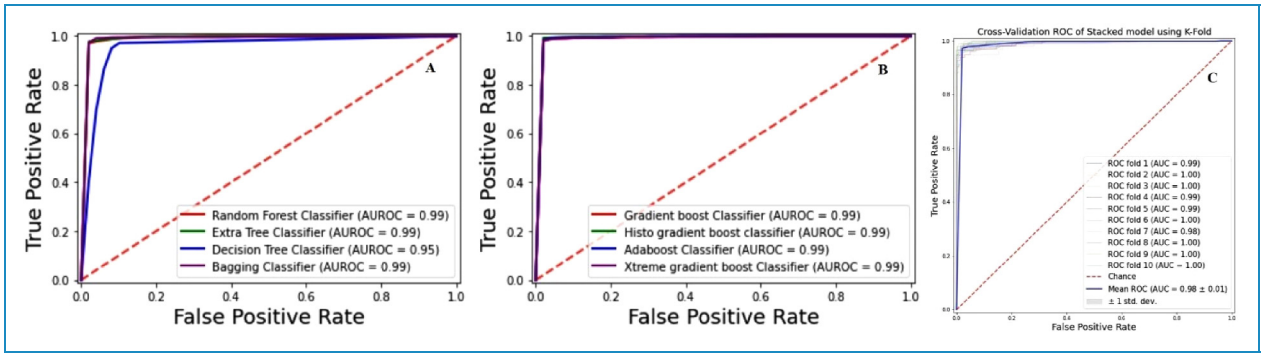

Figure 7. A 10-fold cross-validation receiver operating characteristics (ROC) curves (A). Bagging ensemble models (B). Boosting ensemble models ROC (C). Stacking model ROC.

Table 4. Critical values (p values) exhibiting the significance of ensemble models in Z-test, resampled paired t-test, and McNemar's test.

Model		Z-test	Resampled paired t-test	McNemar's test
Bagging	*Random Forest*	0.00167	0.00092	0.0015
	Extra Tree Classifier	0.00153	0.00051	0.0022
	Decision Tree	0.00141	0.00064	0.0031
	Bagging classifier	0.00120	0.00098	0.0097
Boosting	*Gradient Boost*	0.00136	0.00088	0.0076
	HGB	0.00166	0.00076	0.0065
	AdaBoost	0.00145	0.00065	0.0054
	XGBoost	0.00134	0.00043	0.0044
Stacking		0.00167	0.00050	0.0033

plots helped in visualizing the prediction performance of an ensemble model across the input feature space. When a specific classifier has been trained using a dataset, it defines a set of hyperplanes that are used to separate data points of one class from other. These can be also called as decision boundaries as it helps in depicting data points to be separately placed across boundaries of different classes. The first step includes the training of data being used to fine-tune the model. The trained model was then used to make predictions for a grid of values over the input domain. It can be observed that extra-tree classifier outperformed in exhibiting true classification boundaries.

Comparison with pre-existing models

The proposed model, iDHU-Ensem, was built on the best performing stacking ensemble model and compared with pre-existing predictors to assess the model's efficacy on the independent datasets. iDHU-Ensem was compared with the available predictors such as D-pred,[15] iRNAD,[14] and the RF-based model developed by Zhu et al.[12] The D-pred performs prediction and analysis of RNA sequences derived from *Saccharomyces cerevisiae* through an SVM-based ensemble model. Moreover, iRNAD also utilized SVM for classification of D sites and non-D sites using data samples of *Homosapiens*, *Mus musculus*, *Escherichia coli, Saccharomyces cerevisiae,* and *Drosophila melanogaster*. Similarly, Zhu et al. employed RF and SVM for classifying D sites. The independent testing results revealed that the proposed model outperformed the existing predictors in accuracy. Separate 207 positive and 280 negative samples were used for independent testing. Table 5 presents the result of independent testing and 10-fold cross validation which shows that iDHU-Ensem outperformed the other available predictors. The iRNAD and D-Pred and revealed 92.86% and 83.09% accuracy, respectively, while the Zu et al. model revealed a 96.97% accuracy score. However, the proposed model outperformed in accuracy metrics, revealing an accuracy score of 98%. The proposed model also achieved optimal scores in Sp and Sn revealing 0.98 and 0.97 scores, respectively. Similarly, the iDHU-Ensem revealed a 98% accuracy score followed by D-pred, IRNAD and Zhu et al. predictor in 10-fold cross-validation. Similarly, the MCC value exhibited by D-pred was very low as compared to all predicting models. However, the predictor proposed by Zhu et al. exhibits better MCC. The reason can be the selection of samples in the independent set. However, with the most rigorous test, 10-fold cross-validation, the results revealed that iDHU-Ensem achieved optimal scores in *Acc*, S_n, S_p, and MCC. The results can be better shared through visualization tools. For this purpose, Figure 10 displays the comparison results of pre-existing predictors with the proposed model, iDHU-Ensem, through bar plots and radar maps. It can be seen that the iDHU-Ensem outperformed, revealing high

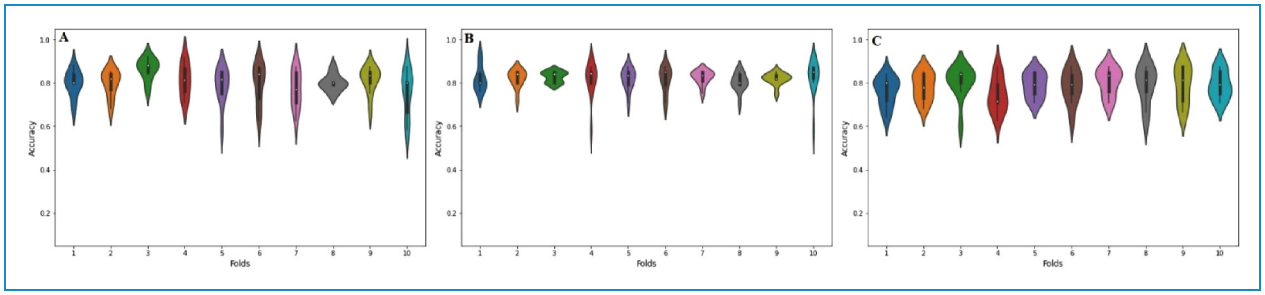

Figure 8. Violin plots of 10-fold cross-validation accuracy (Acc) metric results for (A) bagging ensemble (B) boosting ensemble and (C) stacking ensemble.

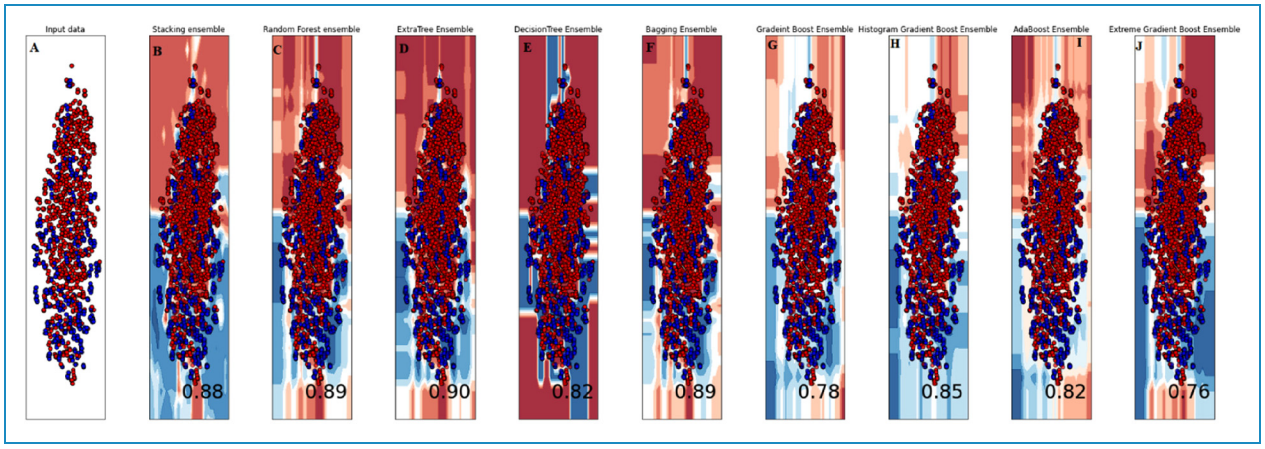

Figure 9. Boundary visualization of ensemble models used in this study as follows: (A) Input data, (B) Stacking, (C) Random Forest, (D) ExtraTree, (E) Decision Tree, (F) Bagging, (G) Gradient Boost, (I) Histo Gradient Boost, (H) Adaboost, and (I) XGBoost.

Table 5. Comparative analysis of iDHU-Ensem with other D site predictors.

	Independent set test				10-fold Cross-validation			
Model	Acc (%)	S_p	S_n	MCC	Acc (%)	S_p	S_n	MCC
D-Pred	83.09	0.89	0.76	0.62	85	0.91	0.77	0.65
iRNAD	92.86	0.96	0.86	0.83	91	0.94	0.86	0.80
Zhu et al.	96.97	0.97	0.96	0.94	97.31	0.98	0.97	0.95
iDHU-Ensem	98	0.98	0.97	0.92	98	0.99	0.98	0.97

scores in all accuracy metrics. The design and development of a novel feature extraction method helped in achieving the optimal scores in the classification of D and non-D sites. The feature development method helped in obtaining obscured information from the sequences, which assisted in providing all the required input values to the ensemble computationally intelligent models. Also, the parameter tuning of ensemble models helped exhibit these high accuracy values. The D site identification is substantial due to its involvement in different biological activities and human lung cancer.

Limitations of current research

The limitation of the current research study is the limited availability of experimentally proven tRNA sequences. Since no hypothetical samples were created, the available

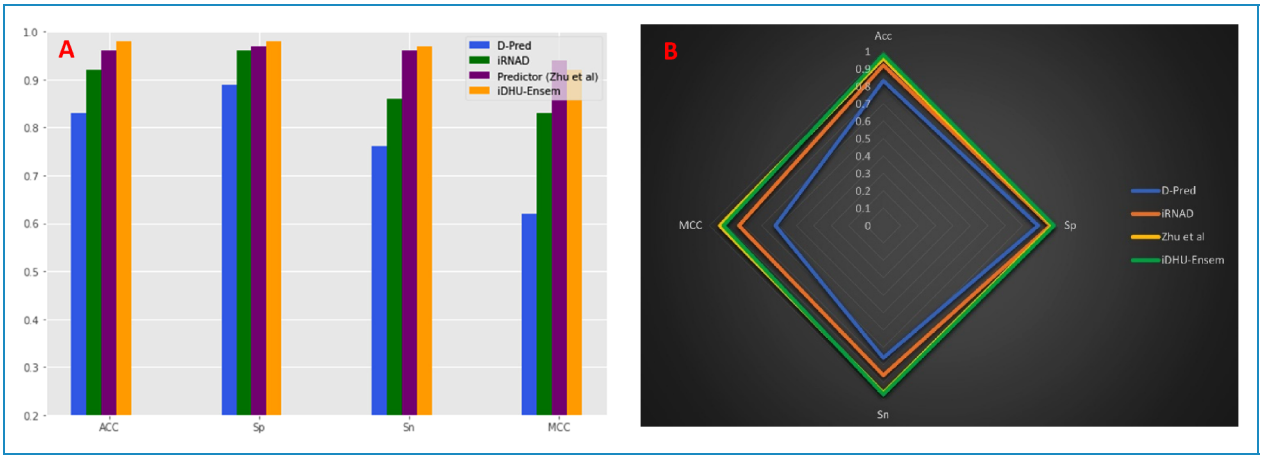

Figure 10. Visualization of comparative analysis of iDHU-Ensem with D-Pred, iRNAD, and Predictor by Zhu et al. in terms of *ACC*, S_p, S_n, and *MCC* (A) Bar plot (B) Radar map.

concrete samples were used in the processes of feature extraction, computational model development, training, and testing of models. Moreover, the tRNA sequences belonging to three species, such as *Homosapiens*, *Mus musculus*, and *Saccharomyces cerevisiae*, were only available from the verified databases. Therefore, only these species were considered for the identification of D sites in tRNA sequences.

Webserver availability

The predictor was made available by deploying a web-based server, iDHU-Ensem. The webserver is available at https://taseersuleman-idhu-ensem-idhu-ensem.streamlit.app/. The free availability of the predictor facilitates rapid and straightforward computational analysis for the identification of dihydrouridine sites.

Conclusion

This work employed an ensemble approach to identify dihydrouridine (D) sites, one of the most common PTMs, in RNA sequences. The D modification plays a pivotal part in the stability of RNA. The researchers also discovered its abundance in cancerous cells. The current research employed an innovative method for extracting features from RNA sequences that make use of the positional and compositional characteristics of individual nucleotides. The dimensionality reduction of the obtained features was carried out using statistical moments. Multiple ensemble models using stacking, bagging, and boosting were trained using the final feature set. Independent testing and cross-validation were then used to assess the efficacy of the trained models. Accuracy, specificity, sensitivity, and Matthew's correlation coefficient were used to assess the models. The best-performing ensemble model was then used to construct the final proposed model, iDHU-Ensem. The proposed model's performance in classifying D sites with non-D sites was also compared with the pre-existing models. However, iDHU-Ensem was found to have the greatest score across the board for accuracy measures. Therefore, it can be stated that iDHU-Ensem optimized the identification of D sites.

Contributorship: The manuscript was prepared by Muhammad Taseer Suleman and Fahad Alturise. The implementation of this study is done by Muhammad Taseer Suleman and Yaser Daanial Khan. The manuscript was reviewed and supervised by Yaser Daanial Khan and Tamim Alkhalifah. All authors contributed to the text in the manuscript and reviewed and approved the final version of the manuscript.

Declaration of Conflicting Interests: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding: The authors received no financial support for the research, authorship, and/or publication of this article.

Guarantor: MTS

Supplemental material: Supplemental material for this article is available online.

ORCID iDs: Fahad Alturise https://orcid.org/0000-0001-9176-7984
Tamim Alkhalifah https://orcid.org/0000-0001-8407-2068

References

1. Arif M, Ahmed S, Ge F, et al. StackACPred: Prediction of anticancer peptides by integrating optimized multiple

feature descriptors with stacked ensemble approach. *Chemom Intell Lab Syst* Epub ahead of print 2022; 220. DOI: 10.1016/j.chemolab.2021.104458
2. Nour S, Salem SA and Habashy SM. ILipo-PseAAC: identification of lipoylation sites using statistical moments and general PseAAC. *Comput Mater Continua* 2022; 71: 215–230.
3. Ramser J, Winnepenninckx B, Lenski C, et al. A splice site mutation in the methyltransferase gene FTSJ1 in Xp11.23 is associated with non-syndromic mental retardation in a large Belgian family (MRX9). *J Med Genet* 2004; 41: 679–683.
4. Zeharia A, Fischel-Ghodsian N, Casas K, et al. Mitochondrial myopathy, sideroblastic anemia, and lactic acidosis: an automosal recessive syndrome in Persian Jews caused by a mutation in the PUS1 gene. *J Child Neurol* 2005; 20: 449–452.
5. Bellodi C, Krasnykh O, Haynes N, et al. Loss of function of the tumor suppressor DKC1 perturbs p27 translation control and contributes to pituitary tumorigenesis. *Cancer Res* 2010; 70: 6026–6035.
6. Metodiev MD, Thompson K, Alston CL, et al. Recessive mutations in TRMT10C cause defects in mitochondrial RNA processing and multiple respiratory chain deficiencies. *Am J Hum Genet* 2016; 98: 993–1000.
7. Nakano S, Suzuki T, Kawarada L, et al. NSUN3 Methylase initiates 5-formylcytidine biogenesis in human mitochondrial tRNA met. *Nat Chem Biol* 2016; 12: 546–551.
8. Durr A, Gargiulo M and Feingold J. The presymptomatic phase of huntington disease. *Rev Neurol (Paris)* 2012; 168: 806–808.
9. Mendez MF. Early-onset Alzheimer's disease: nonamnestic subtypes and type 2 AD. *Arch Med Res* 2012; 43: 677–685.
10. Kathiresan S and Srivastava D. Genetics of human cardiovascular disease. *Cell* 2012; 148: 1242–1257.
11. Suleman MT and Khan YD. m1A-pred: Prediction of modified 1-methyladenosine sites in RNA sequences through artificial intelligence. *Comb Chem High Throughput Screen* Epub ahead of print 2022; 25. DOI: 10.2174/1386207325666220617152743
12. Zhu H, Ao CY, Ding YJ, et al. Identification of D modification sites using a random forest model based on nucleotide chemical properties. *Int J Mol Sci* Epub ahead of print 2022; 23. DOI: 10.3390/ijms23063044
13. Dou L, Zhou W, Zhang L, et al. Accurate identification of RNA D modification using multiple features. *RNA Biol* Epub ahead of print 2021. DOI: 10.1080/15476286.2021.1898160
14. Xu ZC, Feng PM, Yang H, et al. IRNAD: a computational tool for identifying D modification sites in RNA sequence. *Bioinformatics* 2019; 35: 4922–4929.
15. Feng P, Xu Z, Yang H, et al. Identification of D modification sites by integrating heterogeneous features in saccharomyces cerevisiae. *Molecules* Epub ahead of print 2019; 24. DOI: 10.3390/molecules24030380
16. Panwar B and Raghava GPS. Prediction of uridine modifications in tRNA sequences. *BMC Bioinform* 2014; 15: 326.
17. Barukab O, Khan YD, Khan SA, et al. iSulfoTyr-PseAAC: identify tyrosine sulfation sites by incorporating statistical moments via chou's 5-steps rule and pseudo components. *Curr Genomics* 2019; 20: 306–320.
18. Dou L, Zhou W, Zhang L, et al. Accurate identification of RNA D modification using multiple features. *RNA Biol* 2021; 18: 2236–2246.
19. Rasool N, Husssain W and Khan YD. Revelation of enzyme activity of mutant pyrazinamidases from mycobacterium tuberculosis upon binding with various metals using quantum mechanical approach. *Comput Biol Chem* 2019; 83: 107108.
20. Alghamdi W, Alzahrani E, Ullah MZ, et al. 4mC-RF: improving the prediction of 4mC sites using composition and position relative features and statistical moment. *Anal Biochem* 2021; 633: 114385.
21. Malebary SJ, Alzahrani E and Khan YD. A comprehensive tool for accurate identification of methyl-glutamine sites. *J Mol Graph Model* 2022; 110: 108074.
22. Naseer S, Hussain W, Khan YD, et al. Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations. *Anal Biochem* Epub ahead of print 2021; 615. DOI: 10.1016/j.ab.2020.114069
23. Naseer S, Hussain W, Khan YD, et al. Iphoss(deep)-PseAAC: identify phosphoserine sites in proteins using deep learning on general pseudo amino acid compositions via modified 5-steps rule. *IEEE/ACM Trans Comput Biol Bioinform* 2020; 19: 1–1.
24. Butt AH and Khan YD. CanLect-Pred: a cancer therapeutics tool for prediction of target cancerlectins using experiential annotated proteomic sequences. *IEEE Access* 2020; 8: 9520–9531.
25. Shahid M, Ilyas M, Hussain W, et al. ORI-Deep: improving the accuracy for predicting origin of replication sites by using a blend of features and long short-term memory network. *Brief Bioinform* Epub ahead of print 2022; 23. DOI: 10.1093/bib/bbac001
26. Malebary SJ and Khan YD. Evaluating machine learning methodologies for identification of cancer driver genes. *Sci Rep* Epub ahead of print 2021; 11. DOI: 10.1038/s41598-021-91656-8
27. Hussain W, Rasool N and Khan YD. Insights into machine learning-based approaches for Virtual Screening in Drug Discovery: Existing strategies and streamlining through FP-CADD. *Curr Drug Discov Technol* Epub ahead of print 2020; 17. DOI: 10.2174/1570163817666200806165934
28. Mahmood MK, Ehsan A, Khan YD, et al. iHyd-LysSite (EPSV): identifying hydroxylysine sites in protein using statistical formulation by extracting enhanced position and sequence variant feature technique. *Curr Genomics* 2020; 21: 536–545.
29. Barukab O, Khan YD, Khan SA, et al. DNAPred_prot: identification of DNA-binding proteins using composition- and position-based features. *Appl Bionics Biomech* 2022; 2022: 1–17.
30. Khan YD, Alzahrani E, Alghamdi W, et al. Sequence-based identification of allergen proteins developed by integration of PseAAC and statistical moments via 5-step rule. *Curr Bioinform* 2020; 15: 1046–1055.
31. Alghamdi W, Attique M, Alzahrani E, et al. LBCEPred: a machine learning model to predict linear B-cell epitopes. *Brief Bioinform* Epub ahead of print 2022; 23. DOI: 10.1093/bib/bbac035
32. Hussain W, Rasool N and Khan YD. A sequence-based predictor of Zika virus proteins developed by integration of PseAAC and statistical moments. *Comb Chem High Throughput Screen* 2020; 23: 797–804.
33. Awais M, Hussain W, Rasool N, et al. iTSP-PseAAC: identifying tumor suppressor proteins by using fully connected neural network and PseAAC. *Curr Bioinform* 2021; 16: 700–709.

34. Shah AA, Malik HAM, Mohammad A, et al. Machine learning techniques for identification of carcinogenic mutations, which cause breast adenocarcinoma. *Sci Rep* 2022; 12: 11738.
35. Hung TNK, Le NQK, Le NH, et al. An AI-based prediction model for drug-drug interactions in osteoporosis and Paget's diseases from SMILES. *Mol Inform* 2022; 41: 2100264.
36. Le N-Q-K, Nguyen T-T-D and Ou Y-Y. Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties. *J Mol Graph Model* 2017; 73: 166–178.
37. Khandelwal Y. Ensemble stacking for machine learning and deep learning. *Anal Vidhya* 2021; 9.
38. Naseer S, Ali RF, Khan YD, et al. iGluK-Deep: computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions. *J Biomol Struct Dyn* Epub ahead of print 2021: 11691–11704. DOI: 10.1080/07391102.2021.1962738
39. Malebary SJ and Khan YD. Identification of antimicrobial peptides using Chou's 5 step rule. *Comput Mater Continua* 2021; 67: 2863–2881.
40. Khan SA, Khan YD, Ahmad S, et al. N-MyristoylG-PseAAC: sequence-based prediction of N-myristoyl glycine sites in proteins by integration of PseAAC and statistical moments. *Lett Org Chem* 2018; 16: 226–234.
41. Butt AH, Alkhalifah T, Alturise F, et al. A machine learning technique for identifying DNA enhancer regions utilizing CIS-regulatory element patterns. *Sci Rep* 2022; 12: 15183.
42. Khan YD, Khan NS, Naseer S, et al. iSUMOK-PseAAC: Prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC. *PeerJ* Epub ahead of print 2021; 9. DOI: 10.7717/peerj.11581
43. Malebary SJ, Khan R and Khan YD. Protopred: advancing oncological research through identification of proto-oncogene proteins. *IEEE Access* 2021; 9: 68788–68797.
44. Hassan A, Alkhalifah T, Alturise F, et al. RCCC_Pred: a novel method for sequence-based identification of renal clear cell carcinoma genes through DNA mutations and a blend of features. *Diagnostics* 2022; 12: 3036.
45. Shah AA, Alturise F, Alkhalifah T, et al. Evaluation of deep learning techniques for identification of sarcoma-causing carcinogenic mutations. *Digit Health* 2022; 8: 205520762211337.
46. Kopajtich R, Nicholls TJ, Rorbach J, et al. Mutations in GTPBP3 cause a mitochondrial translation defect associated with hypertrophic cardiomyopathy, lactic acidosis, and encephalopathy. *Am J Hum Genet* 2014; 95: 708–720.
47. Zeharia A, Shaag A, Pappo O, et al. Acute infantile liver failure due to mutations in the TRMU gene. *Am J Hum Genet* 2009; 85: 401–407.
48. Patton JR, Bykhovskaya Y, Mengesha E, et al. Mitochondrial myopathy and sideroblastic anemia (MLASA): missense mutation in the pseudouridine synthase 1 (PUS1) gene is associated with the loss of tRNA pseudouridylation. *J Biol Chem* 2005; 280: 19823–19828.
49. Larrieu D, Britton S, Demir M, et al. Chemical inhibition of NAT10 corrects defects of laminopathic cells. *Science (1979)* 2014; 344: 527–532.
50. Haverty PM, Fridlyand J, Li L, et al. High-resolution genomic and expression analyses of copy number alterations in breast tumors. *Genes Chromosomes Cancer* 2008; 47: 530–542.
51. Thrun MC, Gehlert T and Ultsch A. Analyzing the fine structure of distributions. *PLoS One* Epub ahead of print 2020; 15. DOI: 10.1371/journal.pone.0238835