DHU-Pred: accurate prediction of dihydrouridine sites using position and composition variant features on diverse classifiers

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

[1] Department of Computer Science, School of Systems and Technology, University of Management & Technology, Lahore, Pakistan

[2] Department of Computer, College of Science and Arts in Ar Rass Qassim University, Ar Rass, Qassim, Saudi Arabia

ABSTRACT

Background. Dihydrouridine (D) is a modified transfer RNA post-transcriptional modification (PTM) that occurs abundantly in bacteria, eukaryotes, and archaea. The D modification assists in the stability and conformational flexibility of tRNA. The D modification is also responsible for pulmonary carcinogenesis in humans.

Objective. For the detection of D sites, mass spectrometry and site-directed mutagenesis have been developed. However, both are labor-intensive and time-consuming methods. The availability of sequence data has provided the opportunity to build computational models for enhancing the identification of D sites. Based on the sequence data, the DHU-Pred model was proposed in this study to find possible D sites.

Methodology. The model was built by employing comprehensive machine learning and feature extraction approaches. It was then validated using in-demand evaluation metrics and rigorous experimentation and testing approaches.

Results. The DHU-Pred revealed an accuracy score of 96.9%, which was considerably higher compared to the existing D site predictors.

Availability and Implementation. A user-friendly web server for the proposed model was also developed and is freely available for the researchers.

Submitted 26 May 2022
Accepted 1 September 2022
Published 27 October 2022

Corresponding author
Tamim Alkhalifah,
tkhliefh@qu.edu.sa

Academic editor
Kenta Nakai

Additional Information and Declarations can be found on page 19

DOI **10.7717/peerj.14104**

Subjects Bioinformatics, Computational Biology, Genomics, Statistics, Data Mining and Machine Learning

Keywords Prediction, Dihydrouridine, Uridine modifications, Machine learning, Statistical moments, Classification, Random Forest, DHU-Pred, Post Transcriptional Modification, RNA

INTRODUCTION

Post-transcriptional modification (PTM) is the process of chemical alteration of primary ribonucleic acid (RNA) to produce a mature RNA that helps in performing different cell functions (*El Allali, Elhamraoui & Daoud, 2021*). So far, more than 150 PTMs have been identified in RNA (*Boccaletto et al., 2018*). Uridine is a primary nucleoside that is composed of uracil and ribose. Several enzymes play a pivotal role in uridine modification. Among these modifications, dihydrouridine (D) and pseudouridine (Y) are the most prevalent

modifications due to their roles in transfer RNA (tRNA) folding, gene expression, codon binding, and structural flexibility of tRNA. Eukaryotes, bacteria, and archaea all have high levels of this modification. Dihydrouridine base is formed at the uridine base by reducing the carbon–carbon double bond at positions 5 and 6. D formation is catalysed by an enzyme known as dihydrouridine synthase (Dus), from the flavin enzyme family, occurring in prokaryotes in three forms known as DusA, DusB, and DusC. It has been observed that D modification in human tRNA can be the cause of pulmonary carcinogenesis (*Tseng, Medina & Randerath, 1978*; *Kato et al., 2005*). Figure 1 shows the three-dimensional chemical structure of uridine and dihydrouridine.

The D modification is non-planar due to the lack of a double bond, which prevents base stacking. The structural flexibility, conformational folding, and stability of the tRNA structure are all strengthened by this modification (*Dyubankova et al., 2015*). The D site prediction is critical for fully comprehending its potential functions. Site-directed mutagenesis and mass spectrometry have been proposed as methods for detecting D modifications, although both are complex and time-consuming (*Madec et al., 2003*). The availability of sequence-based datasets has increased the possibility of applying computational intelligence methods for the prediction of PTM sites.

Researchers predicted the uridine modifications in the tRNA sequence through a support vector machine (SVM) (*Panwar & Raghava, 2014*). A three-stage approach was used in their research, including training and validation of new tRNA sequences on the previous model and specie-wise dataset training and validation. *Liu, Chen & Lin (2020)* proposed a predictor, XG-PseU, for the identification of pseudo uridine modification through an optimal feature selection method. *Feng et al. (2019)* proposed a method for the detection of D modification in *Saccharomyces cerevisiae* using an ensemble classifier. Three different feature extraction approaches were used by the authors, including nucleotide physicochemical property (NPCP), pseudo dinucleotide composition (PseDNC), and secondary structure component (SSC) with SVM for classification. For comparison, an SVM-based ensemble approach was adopted based on voting among these three extraction features. However, the metrics results were not optimal, revealing 83.08% accuracy (*Acc*), 89.71% specificity (S_p), 76.47% sensitivity (S_n), and a 0.62 Matthews correlation coefficient (*MCC*). Similarly, *Xu et al. (2019)* developed a predictor, iRNAD, for the prediction of D modification based on RNA samples of five species. The samples were encoded using nucleotide chemical property (NCP) and nucleotide density. The SVM was utilised as a classification model, and the jackknife test was used to assess the model's performance. The proposed model outlined a 96.18% *Acc* with S_p and S_n scores of 98.13% and 92.05%, respectively. *Dou et al. (2021)* published recently in which they proposed a model, iRNAD-XGBoost, in consideration of the imbalance problem using a hybrid sampling method and the feature selection method. However, an independent test of the model revealed the values of *Acc*, S_n, S_p, and *MCC* to be 93.75%, 91.67%, 94.74%, and 0.86, respectively.

The current study focused on the prediction of D sites in tRNA using a novel method for feature extraction from the RNA sequences obtained from three species, including *Homosapiens*, *Mus musculus*, and *Saccharomyces cerevisiae*. A novel methodology was adapted for the extraction and representation of feature vectors based on the position

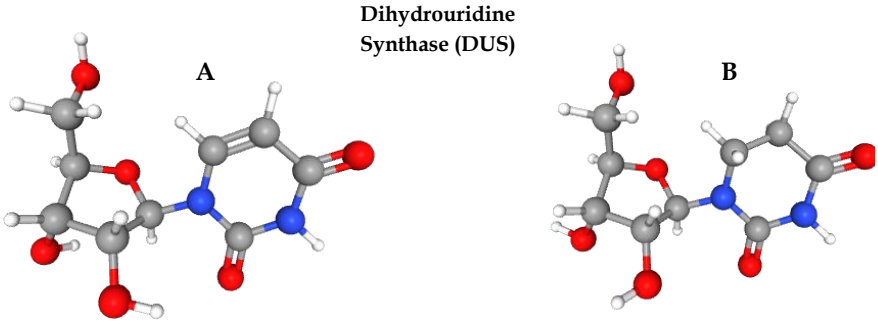

Figure 1 Formation of dihydrouridine from uridine. (A) 3D chemical structure of uridine. (B) Formation of Dihydrouridine (3D structure) through Dihydrouridine synthase (DUS) enzyme.
Full-size DOI: 10.7717/peerj.14104/fig-1

as well as the composition of nucleotide bases through the incorporation of statistical moments to increase the prediction capability of the model (*Amanat et al., 2019*; *Naseer et al., 2020*; *Barukab et al., 2019*). The development and training of computationally intelligent models was aided by these feature vectors. The performance of all models was assessed through various testing methods such as the independent set test, jackknife test, and k-fold cross-validation. The overall accuracy of the model was assessed through S_n, S_p, *MCC*, and *Acc.*

As shown in Fig. 2, the entire approach employed in this work included dataset collection, sample formulation, prediction model training, and model evaluation. Finally, DHU-Pred, a publicly accessible web server, was created to aid D modification research.

MATERIALS & METHODS

Dataset collection

The collection of the benchmark dataset was the initial phase of the research. The tRNA sequences were considered for the feature extraction and prediction model training in the current investigation. The sequences were obtained from RMBase (*Xuan et al., 2017*), also used by *Xu et al. (2019)*, *Feng et al. (2019)*, and *Dou et al. (2021)*. The benchmark dataset contained data from three species, including *Homosapiens* (Human), *Mus musculus* (Mouse), and *Saccharomyces cerevisiae* (Yeast) related to D modification.

Positive and negative samples

Each data set sample was composed of 41 nucleotides with U at its center, *i.e.,* at position 21. The experimental results revealed the optimal accuracy scores were achieved using a sequence length of 41 nucleotides. In addition, an RNA sample containing the D site was expressed as mentioned in Eq. (1).

$$P(U) = P_{-\epsilon}P_{-(\epsilon-1)}....P_{-2}P_{-1}UP_{+1}P_{+2}....P_{+(\epsilon-1)}P_{+\epsilon}. \quad (1)$$

In Eq. (1), the symbol U represents uridine (U), the center of nucleotide sequences, and the subscript value ϵ is set as 20. Thus, the total length of the nucleotide sequence is ($2\ \epsilon$+1). $P_{-\epsilon}$ represents the ϵ-th upstream nucleotide from the central uridine and $P_{+\epsilon}$

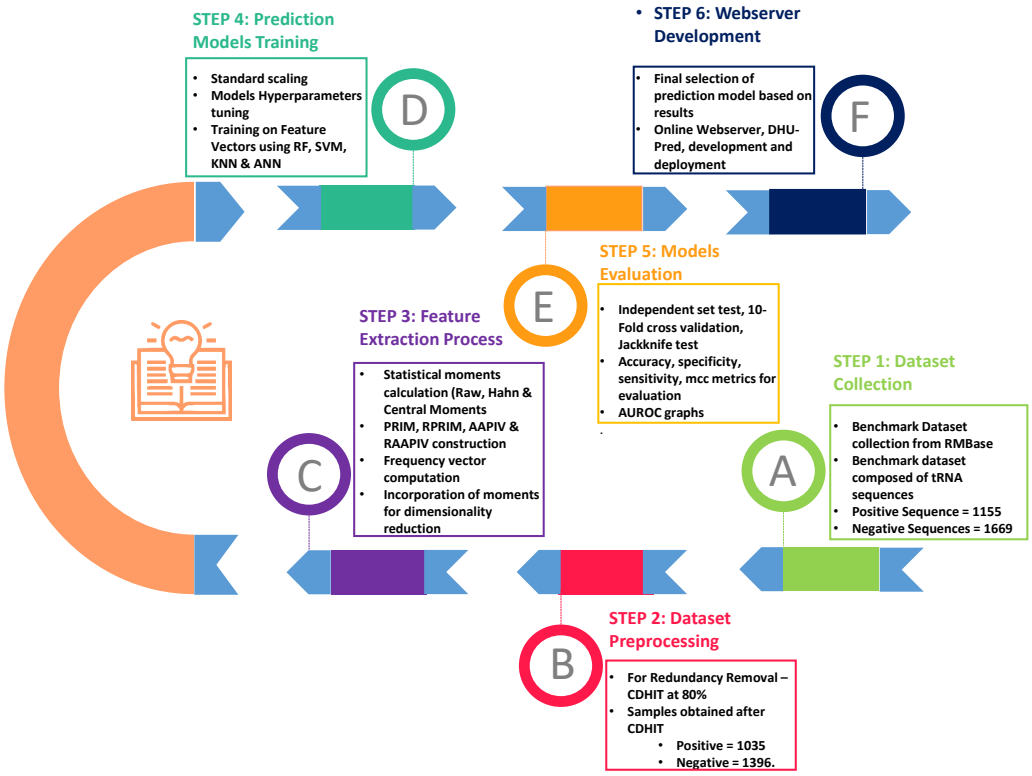

Figure 2 **Flow chart of the methodology.**

Full-size DOI: 10.7717/peerj.14104/fig-2

represents the ϵ-th downstream nucleotide. The positive samples signify the sequence with D modification, whereas the negative samples express the sequences without D modification. The total positive and negative sites of all three previously mentioned species were 1,155 and 1,669, respectively. However, removing redundant samples with CD-HIT (16) at 0.80 reduced the sample size to 1,035 positive and 1,396 negative samples.

Sequence logo

The distribution of nucleotide bases in the obtained sequences can be illustrated with the help of the sequence logo. An online Two Sample Logo tool (*Vacic, Iakoucheva & Radivojac, 2006*) was used for the said purpose. The sequence logo shown in Fig. 3 expressed the distribution of cytosine (C), guanine (G), adenine (A), and uracil (U) in the dataset. The nucleotide base distribution from the centre nucleotide base (*i.e.*, uracil) is different between positive sites (from position 22 to 41) and negative sites (from position 1 to 20). It can be observed from Fig. 3 that G and C are enriched in the region located from position 19 (negative site) to position 31 (positive site). However, the base A is symmetrically distributed along the whole region within all nucleotides. The nucleotide base U is mostly concentrated in and around the centre of all RNA samples.

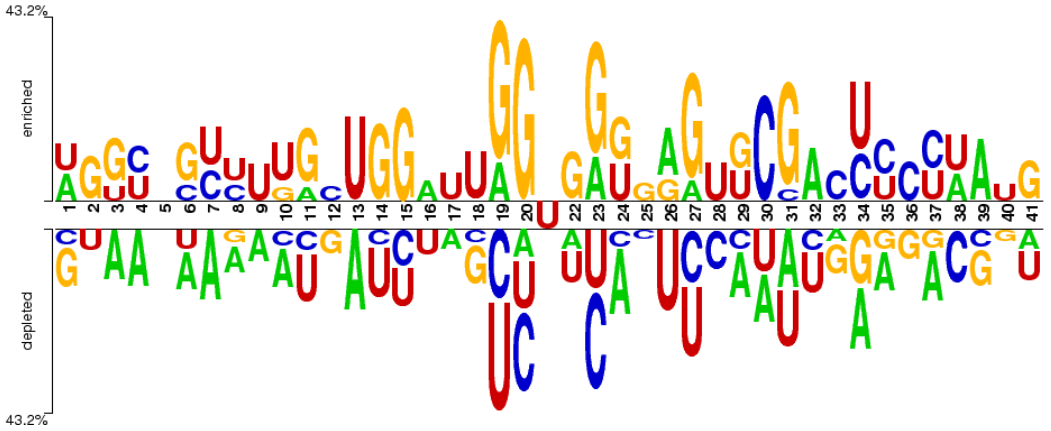

Figure 3 **Distribution of nucleotides in the sample dataset with U in the middle.**
Full-size DOI: 10.7717/peerj.14104/fig-3

Feature generation and representation from RNA samples

Encoding RNA sequences into feature vectors is one of the most prevalent steps because computational models cannot handle and process biological sequences directly. As a result, statistical analysis of the acquired samples can better retrieve the obscured information within the sequences. The current study dealt with the feature generation mechanism based on the position and composition of nucleotides within a given sequence. Chou suggested the pseudo amino acid composition (PseAAC) as one of the most popular and effective ways of dealing with the problem of sequence pattern loss. The current study implemented a similar approach to PseAAC in pseudo K-tuple nucleotide composition for feature vector generation (*Chen, Chou & Chen, 2015*; *Xiao et al., 2019*; *Awazu, 2017*). These vectors served as input for model training, as mentioned in these comprehensive research works *Mahmood et al. (2020)* and *Shah & Khan (2020)*. For the current research, the feature vectors were developed based on the position and composition of nucleotides in each sequence. The samples in the dataset were characterised as follows using the nucleotide formulation, $P_{\epsilon}(K)$,described in Eq. (2).

$$P_{\epsilon}(K) = [£_1 £_2 £_3 \ldots £_U \ldots £_{\Omega}]^T. \tag{2}$$

Where, at K-tuple nucleotide, £ represents each component in a vector based on the feature generation mechanism adopted in this research. Where T represents the transpose of the accumulated feature formulation. Each nucleotide sample of a specific site was 41 base pairs (bp) in length and is expressed in Eq. (3).

$$P = R_1 R_2 R_3 \cdots R_{18} R_{19} \boldsymbol{R}_{21} \cdots R_{39} R_{40} R_{41}. \tag{3}$$

In Eq. (3), $\boldsymbol{R}_{21} = \boldsymbol{U}$ and R_1 $(n = 1, 2\ldots\ldots\ldots, 41;\ \mathrm{n} \neq 21)$ represent any nucleotide such as cytosine, guanine, adenine, and uracil.

Statistical moment calculation

The quantification of the collected nucleotide sequences is based on their composition and position. Moments have been applied to various data distributions by statisticians

and data analysts (*Malebary & Khan, 2021*). For this purpose, central, raw, and Hahn moments were used in the feature extraction process. Raw and Hahn moments are scale and location variant, whereas central moments are scale and vicinity variant. Therefore, the dataset's mean, asymmetry, and variance were calculated using raw and central moments. On the contrary, Hahn moments were calculated by the reference of Hahn polynomials to maintain the sequence order information (*Khan et al., 2020*). *Butt et al. (2016)* used these moments as a means for feature extraction, which was used to identify membrane proteins. A matrix K' in Eq. (4) is a $m^{\star}n$ two-dimensional matrix in which a single element, k_{mn}, represents the nth nucleotide base in mth sequence.

$$K' = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mn} \end{bmatrix}. \tag{4}$$

Raw moments were used to extract location variant features by calculating the dataset's mean, variance, and unequal probability distribution. Raw moments expressed in Eq. (5) where, $u+v$ is the sum of raw moments and R_{00}, R_{01}, R_{10}, R_{11}, R_{12}, R_{21}, R_{30}, R_{03}, were calculated up to 3rd-degree polynomial.

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

Central moments do not depend upon the location. Instead, these are related to the composition and shape of the distribution (*TLo & Don, 1989*). The central moments were calculated based on the deviations of the random variable from the mean. For this study, the central moments were computed as expressed in Eq. (6).

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

Hahn moments were computed using Hahn polynomials. Hahn moments calculation mentioned in the Eq. (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 following expression Eq. (8) was used to determine the orthogonal normalized Hahn of the two-dimensional data.

$$H_{ij} = \sum_{q=0}^{N-1}\sum_{p=0}^{N-1} \beta_{ij} h_j^{\widetilde{u,v}}(q,N) h_j^{\widetilde{u,v}}(p,N), \quad m,n = 0,1,\dots,N-1. \tag{8}$$

Construction of Position Relative Incidence Matrix (PRIM)

The current study focused on improving the model's prediction abilities. Therefore, a complete feature generation model was required for the said purpose. The relative positions

of nucleotides within an RNA sequence are helpful and become the basis for mathematical formulation. For this purpose, three types of position relative incidence matrix (PRIM) were constructed by considering single nucleotide composition (SNC), di-nucleotide composition (DNC), and tri-nucleotide composition (TNC). These matrices were created to reveal the relative positions of nucleotide bases, which helped in comprehensively quantizing the relative positions of nucleotides. The matrix, A_{Prim}, is a 4*4 matrix Eq. (9) that produced a total of 16 coefficients.

$$A_{prim} = \begin{bmatrix} \mathfrak{T}_{A\to A} & \mathfrak{T}_{A\to G} & \mathfrak{T}_{A\to U} & \mathfrak{T}_{A\to C} \\ \mathfrak{T}_{G\to A} & \mathfrak{T}_{G\to G} & \mathfrak{T}_{G\to U} & \mathfrak{T}_{G\to C} \\ \mathfrak{T}_{U\to A} & \mathfrak{T}_{U\to G} & \mathfrak{T}_{U\to U} & \mathfrak{T}_{U\to C} \\ \mathfrak{T}_{C\to A} & \mathfrak{T}_{C\to G} & \mathfrak{T}_{C\to U} & \mathfrak{T}_{C\to C} \end{bmatrix}. \tag{9}$$

Where, $\mathfrak{T}_{i\to j}$, represents the relative position of any nucleotide (*i.e.,* A, C, U, or G) to other nucleotides. The matrix, B_{Prim}, is a 16*16 matrix Eq. (10) that denotes the DNC producing 16 unique combinations of nucleotides (*i.e.,* AA, AG, AU, …, CG, *CU, CC*). This matrix yielded a total of 256 coefficients. However, with the fusion of statistical moments, only 30 coefficients were derived.

$$B_{prim} = \begin{bmatrix} \mathfrak{d}_{AA\to AA} & \mathfrak{d}_{AA\to AG} & \mathfrak{d}_{AA\to AU} & \cdots & \mathfrak{d}_{AA\to j} & \cdots & \mathfrak{d}_{AA\to CC} \\ \mathfrak{d}_{AG\to AA} & \mathfrak{d}_{AG\to AG} & \mathfrak{d}_{AG\to AU} & \cdots & \mathfrak{d}_{AG\to j} & \cdots & \mathfrak{d}_{AG\to CC} \\ \mathfrak{d}_{AU\to AA} & \mathfrak{d}_{AU\to AG} & \mathfrak{d}_{AU\to AU} & \cdots & \mathfrak{d}_{AU\to j} & \cdots & \mathfrak{d}_{AU\to CC} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \mathfrak{d}_{GA\to AA} & \mathfrak{d}_{GA\to AG} & \mathfrak{d}_{GA\to AU} & \cdots & \mathfrak{d}_{GA\to j} & \cdots & \mathfrak{d}_{GA\to CC} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \mathfrak{d}_{N\to AA} & \mathfrak{d}_{N\to AG} & \mathfrak{d}_{N\to AU} & \cdots & \mathfrak{d}_{N\to j} & \cdots & \mathfrak{d}_{N\to CC} \end{bmatrix}. \tag{10}$$

The matrix, C_{Prim}, is a 64*64 matrix Eq. (11) representing 64 unique tri-nucleotide combinations (*i.e.,* AAA, AAG, AAU, …., CCG, CCU, CCC). C_{Prim} yielded 4,096 coefficients. However, with the incorporation of central, raw, and Hahn moments, 30 coefficients were computed.

$$C_{prim} = \begin{bmatrix} \Psi_{AAA\to AAA} & \Psi_{AAA\to AAG} & \Psi_{AAA\to AAU} & \cdots & \Psi_{AAA\to j} & \cdots & \Psi_{AAA\to CCC} \\ \Psi_{AAG\to AAA} & \Psi_{AAG\to AAG} & \Psi_{AAG\to AAU} & \cdots & \Psi_{AAG\to j} & \cdots & \Psi_{AAG\to CCC} \\ \Psi_{AAU\to AAA} & \Psi_{AAU\to AAG} & \Psi_{AAU\to AAU} & \cdots & \Psi_{AAU\to j} & \cdots & \Psi_{AAU\to CCC} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ \Psi_{AAC\to AAA} & \Psi_{AAC\to AAG} & \Psi_{AAC\to AAU} & \cdots & \Psi_{AAC\to j} & \cdots & \Psi_{AAC\to CCC} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ \Psi_{N\to AAA} & \Psi_{N\to AAG} & \Psi_{N\to AAU} & \cdots & \Psi_{N\to j} & \cdots & \Psi_{N\to CCC} \end{bmatrix}. \tag{11}$$

Reverse Position Relative Indices Matrix (RPRIM)

The main objective in feature vector determination is to extract as much information as possible to develop a reliable predictive model. Reversing the sequence order to get more embedded information within the sequences yielded a reverse position relative indices matrix (RPRIM). Eq. (12) states, V_{RPRIM}, in which any arbitrary element, $R_{i\to j}$, represents the relative position value of the i th nucleotide base to the j th nucleotide. The calculation of RPRIM was carried out using mononucleotide, di-nucleotide, and tri-nucleotide combinations like PRIM matrices.

$$V_{RPRIM} = \begin{bmatrix} V_{1\to1} & V_{1\to2} & V_{1\to3} & \cdots & V_{1\to y} & \cdots & V_{1\to j} \\ V_{2\to1} & V_{2\to2} & V_{2\to3} & \cdots & V_{2\to y} & \cdots & V_{2\to j} \\ V_{3\to1} & V_{3\to2} & V_{3\to3} & \cdots & V_{3\to y} & \cdots & V_{3\to j} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ V_{x\to1} & V_{x\to2} & V_{x\to3} & \cdots & V_{x\to y} & \cdots & V_{4\to j} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ V_{N\to1} & V_{N\to2} & V_{N\to3} & \cdots & V_{N\to y} & \cdots & V_{N\to j} \end{bmatrix}. \tag{12}$$

Frequency Matrices (FMs) generation

It is necessary to extract information about the location as well as the composition of the sequence for generating attributes. The frequency vector $\dot{\mathrm{G}}$ in Eq. (13), on the other hand, provided the count for each nucleotide in the sequence.

$$\dot{\mathrm{G}} = \{\delta_1, \delta_2, \ldots.., \delta_n\}. \tag{13}$$

Where δ_i represents the count of each ith nucleotide within a sequence. The frequency vector $\dot{\mathrm{G}}$ was computed for single as well as paired nucleotides.

Generation of Accumulative Absolute Position Incidence Vector (AAPIV)

The extraction of compositional information did not provide enough information regarding the position-specific calculation of each nucleotide. For this purpose, accumulative absolute position incidence vectors (AAPIVs) of lengths 4, 16, and 64 were computed, which are represented as K_{AAPIV4}, $K_{AAPIV16}$, and $K_{AAPIV64}$ in Eqs. (14), (15) and (16) respectively.

$$K_{AAPIV4} = \{\rho_1, \rho_2, \rho_3, \rho_4, \} \tag{14}$$

$$K_{AAPIV16} = \{\rho_1, \rho_2, \rho_3, ..., \rho_{15}, \rho_{16}, \} \tag{15}$$

$$K_{AAPIV64} = \{\rho_1, \rho_2, \rho_3, ..., \rho_{63}, \rho_{64}\}. \tag{16}$$

Any element ρ_i is computed as follows:

$$\rho_i = \sum_{k=1}^{n} p_k. \tag{17}$$

Reverse Accumulative Absolute Position Incidence Vector (RAAPIV) generation

The reverse accumulative absolute position incidence vector (RAAPIV) helped explore hidden information related to the relative positions of nucleotides in the sequence. The length of RAAPIV was 4, 16, and 64, expressed as $K_{RAAPIV\,4}$ Eq. (18), $K_{RAAPIV\,16}$ Eq. (19), and $K_{RAAPIV\,64}$ Eq. (20), respectively, but with reverse sequence order.

$$K_{RAAPIV\,4} = \{\tau_1, \tau_2, \tau_3, \tau_4\} \tag{18}$$

$$K_{RAAPIV\,16} = \{\tau_1, \tau_2, \tau_3, ..., \tau_{16}\} \tag{19}$$

$$K_{RAAPIV\,64} = \{\tau_1, \tau_2, \tau_3, ..., \tau_{64}\}. \tag{20}$$

Feature vector formulation

In the concluding step of extracting features, a single feature vector was created and fed into the prediction model. Therefore, the following steps were taken in developing the final feature set: (1) Statistical moments were computed initially for PRIM and RPRIM for feature dimensionality reduction. (2) The resultant features were then assimilated into FV, AAPIV, and RAAPIV. Ultimately, a feature vector with 522 attributes was obtained. Each feature vector represents a single sample within the dataset. For binary classification, the positive samples were designated as "1", and the negative samples were designated as "0".

Feature scaling technique

A standard scalar technique (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) within the Python framework was used in this research study to standardise feature values obtained through the methods mentioned above. It is a common method to preprocess data before putting it into a computational model.

Prediction models incorporation

Transforming raw biological sequences into discrete quantifiable vectors is a challenging task in artificial intelligence. The vectors serve as input to the machine learning algorithms such as random forest (RF), support vector machine (SVM), gradient boost (GB), *etc.* In this section, the development and training of prediction models are discussed in detail.

Random forest (RF)

A RF is an ensemble technique that combines various decision trees to get a more appropriate and accurate prediction result. Many decision trees participated in the classification, but, the majority voting by these decision trees won. The margin function $mg\,(X, Y)$ in Eq. (21) describes the ensemble of classifiers with the training set drawn at random from the distribution of the random vector Y, X.

$$mg\,(X, Y) = av_k I\,(h_k\,(X) = Y) - max_{j \neq y} av_k I\left(h_k\,(X) = j\right). \tag{21}$$

Wenric & Shemirani (2018) utilised RF classifiers to rank genes by their expression values with the RNA-sequence sample set. This research implemented the algorithm using

scikit-learn (version 0.24.2; https://scikit-learn.org/stable/). Optimal results were achieved by tuning hyperparameters such as maximum depth, maximum features, minimum samples, and the number of estimators. Nevertheless, tuning hyperparameters had a profound effect on the performance of the RF-based model (*Probst, Wright & Boulesteix, 2019*). In the current study, the *max_depth* was set to 100. Similarly, *max_features* was configured to Auto and *min_sample_leaf*, defining the number of samples required to be a leaf node, was set to 6. Following numerous experiments, the subsequent parameters for model training were determined, as shown in Table 1.

Support Vector Machine (SVM)

In supervised machine learning approaches, SVM is used for classification, regression, and outlier identification. In bioinformatics, SVM is well applied to the prediction problems of proteins and DNA/RNA sequences as well (*Han, Wang & Zhou, 2019*; *Manavalan et al., 2019*; *Meng et al., 2019*). Researchers (*Feng et al., 2019*; *Xu et al., 2019*) utilised SVM for the classification of D sites and non-D sites. The SVM-based model was deployed in the current study using the Python Scikit-learn library. Considering the hyper-parameters optimization through experiments, the following parameters were tuned to get the best results, as shown in Table 2.

K Nearest Neighbor (KNN)

KNN is a supervised machine learning method that uses training data to perform classification. It forecasts the values of new outcomes based on the closely matched training data points. Dongardive and Abraham (*Dongardive & Abraham*) conducted experiments using KNN with different neighbour values. They achieved the highest accuracy of 84%, with a neighbour value of 15 on the dataset containing 717 protein sequences. For experimental purposes, the research also employed the KNN model with the neighbour value (K) set to 3.

Artificial Neural Network (ANN)

A network of artificial neurons, often referred to as nodes, is what constitutes an ANN. These nodes are brought together to form a network that performs functions analogous to those of a biological neuron found in the brain. These nodes form different layers. There can be various hidden layers, with input and output layers. Each input signal is routed into a single input layer neuron before being passed on to the hidden layer. The final level of processing is completed by the output layer, which sends out output signals. ANN models have been extensively used in many research areas especially computational biology (*Hussain et al., 2019a*; *Hussain et al., 2019b*). In the present study, the ANN model was trained by modifying parameters such as hidden layer sizes, activation, solver, alpha, and learning rate, as indicated in Table 3.

RESULTS AND DISCUSSION

This research study was carried out to predict D sites located in tRNA using samples from three species through popular machine learning algorithms. Prediction models were developed and trained using a benchmark dataset. Models were evaluated through

Table 1 RF model tuning parameters.

Parameter	Value
N_estimators	1,000
max_depth	100
Max_features	Auto
Min_samples_leaf	6
Min_samples_split	10

Table 2 Hyperparameter optimization of support vector machine.

Parameter	Value
C	5
Probability	True
Gamma	'auto'
Kernel	'rbf'
Random_state	'None'

Table 3 ANN hyperparameters tuning.

Parameter	Value
Random_state	1
Activation	'relu'
Solver	'adam'
Learning rate	0.001
Hidden_layer_sizes	5,2
Alpha	0.0002

well-known metrics used in many research studies. For example, the current research study used four metrics for the evaluation of prediction models, including sensitivity (S_n), specificity (S_p), accuracy (Acc), and Mathew's correlation coefficient (MCC).

Metrics formulation

Four different metrics were used to evaluate the computational models including S_n, S_p, Acc, and the MCC as expressed in Eq. (22). N^+represents the true D sites, whereas N^-represents the rogue D sites. Similarly, the symbol N_-^+ shows the number of modified sites that were true D sites but incorrectly predicted as rogue D sites. Similarly, N_+^- represents the number of rogue D sites incorrectly predicted as true D sites. However, it is necessary to mention here that metrics in Eq. (22) are only valid for single-class systems. For multiple classes of systems that are more prominent in biomedicine (*Cao et al., 2018*; *Qiu et al., 2016*), system medicine (*Cheng et al., 2017*), and system biology (*Jain & Kihara,*

Table 4 Evaluation metrics result of jackknife test for RF, SVM, KNN, and ANN.

Computational model	*Acc*	*MCC*	S_n	S_p
RF	95%	0.90	0.92	0.97
SVM	94.6	0.89	0.97	0.92
KNN	92%	0.85	0.99	0.85
ANN	96.7%	0.93	0.98	0.95

2019), a completely different set of metrics are required as discussed in *Chou (2013)*.

$$\begin{cases} S_n = 1 - \dfrac{N_-^+}{N^+} \quad 0 \le S_n \le 1 \\ S_p = 1 - \dfrac{N_+^-}{N^-} \quad 0 \le S_p \le 1 \\ Acc = 1 - \dfrac{N_-^+ + N_+^-}{N^+ + N^-} \quad 0 \le Acc \le 1 \\ MCC = \dfrac{1 - \left(\frac{N_-^+}{N^+} + \frac{N_+^-}{N^-}\right)}{\sqrt{\left(1 + \frac{N_+^- - N_-^+}{N^+}\right)\left(1 + \frac{N_-^{\pm N_+^-}}{N^-}\right)}} \quad -1 \le MCC \le 1. \end{cases} \tag{22}$$

Test methods

The prediction models used in this research study were evaluated through independent set tests, jackknife testing, and 10-fold cross-validation. The jackknife test usually imparts unique value to a similar dataset. Thus, in jackknife, the learning algorithm is applied once for each sample, using the selected sample as a single test set and all other samples in the dataset as the training set. The ANN revealed maximum *Acc*, *MCC*, and S_n scores. The jackknife test results are mentioned in Table 4. In Fig. 4, it is observed that the area under the curve of the RF-based predictor is at its maximum.

For the evaluation of models, an independent set test was used (*Bui et al., 2016*; *Wójcikowski et al., 2019*). The dataset was separated into two groups in this study, *i.e.*, the training dataset and the testing dataset. The dataset was split into an 80% training dataset and a 20% testing dataset for evaluation using the train-test split method of the python sci-kit learn library. It is essential to mention that the test samples were separate from the training samples during independent testing. The RF-based model revealed a maximum accuracy score of 96.9% in the independent set test. Similarly, the S_n, S_p, and *MCC* scores achieved by the RF-based model were the highest among the other three models as mentioned in Table 5. The results revealed that the RF-based model had shown a high AUC value compared to other models used in the study, as shown in Fig. 5. When the separate dataset is unavailable for validation, the cross-validation technique is adapted for model evaluation. The k parameter in K-fold cross-validation refers to the number of groups into which a sample of a given dataset should be divided. This test is widely used for evaluation due to limited data samples for validation. A 10-fold cross-validation was

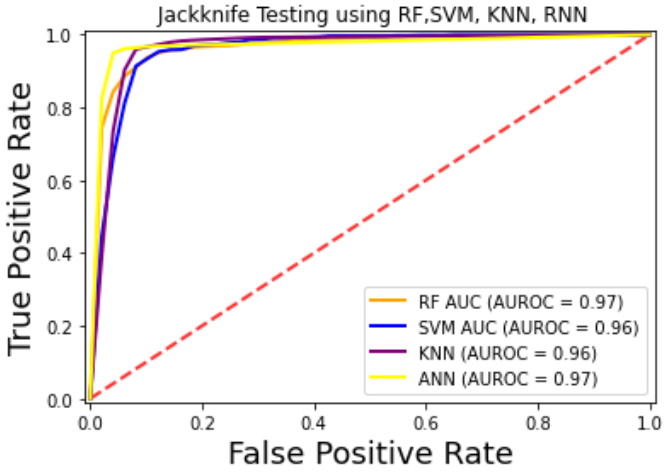

Figure 4 **ROC-curve of jackknife test.**

Full-size DOI: 10.7717/peerj.14104/fig-4

Table 5 **Evaluation metrics result of the independent set test for RF, SVM, KNN, and ANN.**

Computational model	*Acc*	*MCC*	S_n	S_p
RF	96.9%	0.93	0.98	0.97
SVM	91.5%	0.83	0.94	0.89
KNN	85%	0.74	0.97	0.75
ANN	96.3%	0.92	0.97	0.94

adopted in the current study. Through 10-fold cross-validation, RF divulged the maximum *Acc*, *MCC*, and S_p scores, among other models. as presented in Table 6. Cross-validation results have also been shown in the receiver operating characteristic (ROC) curve in Fig. 6, which depicts the area under the curve (AUC) of all the four prediction models used in this research. Violin plots and heat maps were used for visualizing cross-validation results. A violin plot uses density curves to represent numeric data distributions for one or more groups. For example, the median, interquartile range, and lower and upper adjacent values can be depicted through a white dot in the plot, a black bar in the center, and dark black lines stretched from the bar, respectively. Figure 7 shows violin plots representing accuracy values calculated in each fold for all prediction models. Moreover, heat maps can represent data graphically in the form of a matrix. Because they synthesize data and present it pictorially, heat maps provide an excellent visual summary of information. Its main advantage over other visualization tools is that it allows a large amount of information to be delivered fast. A heatmap is shown in Fig. 8, which depicts the cross-validation scores of all folds.

Decision boundary visualization

Through supervised machine learning models, numerical prediction is sometimes not enough in many classification problems. It is critical to visualize the real decision boundary

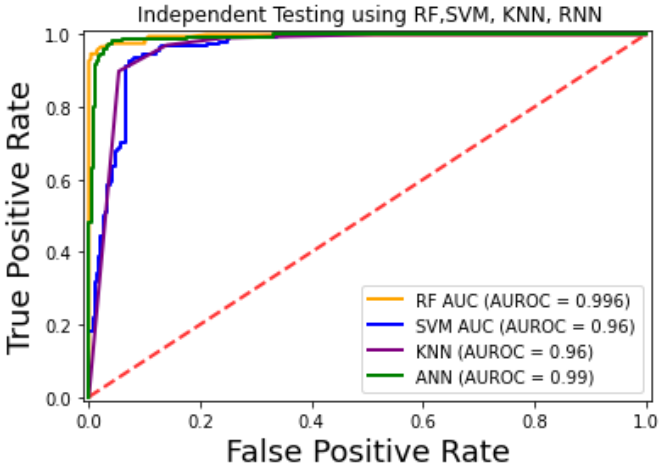

Figure 5 **ROC-Curve of the independent set test.**

Full-size DOI: 10.7717/peerj.14104/fig-5

Table 6 **Evaluation metrics result of 10-fold Cross-validation for RF, SVM, KNN, and ANN.**

Computational model	*Acc*	*MCC*	S_n	S_p
RF	93.4%	0.86	0.92	0.94
SVM	93%	0.85	0.91	0.93
KNN	90%	0.81	0.97	0.85
ANN	92%	0.83	0.90	0.93

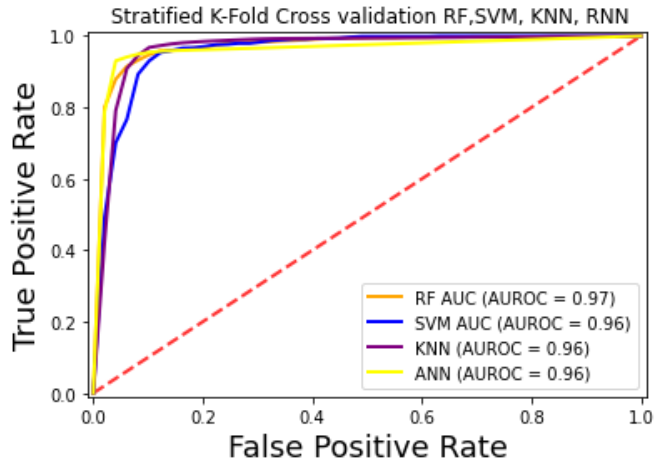

Figure 6 **ROC-Curve of k-fold Cross-validation.**

Full-size DOI: 10.7717/peerj.14104/fig-6

between the classes. Therefore, a decision surface was applied to the classification algorithms used in this work. A trained machine learning system predicts a coarse grid across the input feature space in this decision surface plot. First, the model was fitted onto

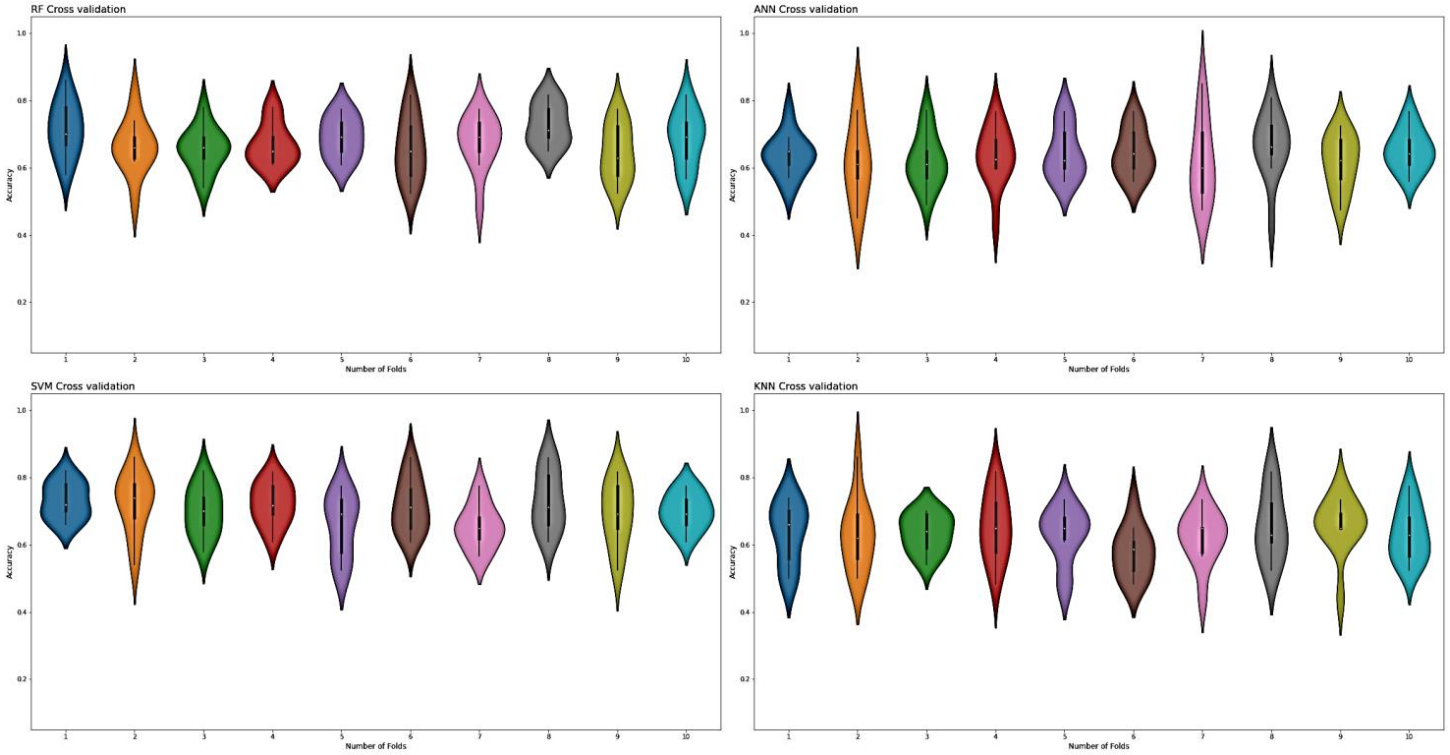

Figure 7 **Violin charts RF, SVM, KNN and ANN cross validation.**

Full-size DOI: 10.7717/peerj.14104/fig-7

the training dataset. Afterwards, the trained model was utilized to make predictions for a grid of values across the input domain. The contourf() function from matplotlib (https://matplotlib.org/3.5.0/api/_as_gen/matplotlib.pyplot.contourf.html) and scatterplot (https://seaborn.pydata.org/generated/seaborn.scatterplot.html) had been used for plotting. Figure 9 represents the decision surface plots of the classification algorithms employed in this study.

Comparative analysis

For comparative analysis, two models, *i.e.,* iRNAD (*Xu et al., 2019*) and D-pred (*Feng et al., 2019*), were observed in this study. The benchmark dataset details of DHU-Pred, iRNAD, and D-pred have been shown in Table 7. After a thorough investigation of iRNAD and D-Pred, it was observed that only SVM was used for categorization. The current research study dealt with the development of four different prediction models, their evaluation through standard testing methods, and their performance comparison through standardised metrics. Moreover, the novel feature extraction method and the development of refined feature vectors helped achieve optimized results in predicting D sites. Therefore, the DHU-Pred outperformed the comparative models.

Independent set testing was carried out on proposed and comparative models. It is essential to mention that the test samples differed from the training samples. Independent testing was carried out using 207 positive and 280 negative samples as mentioned in

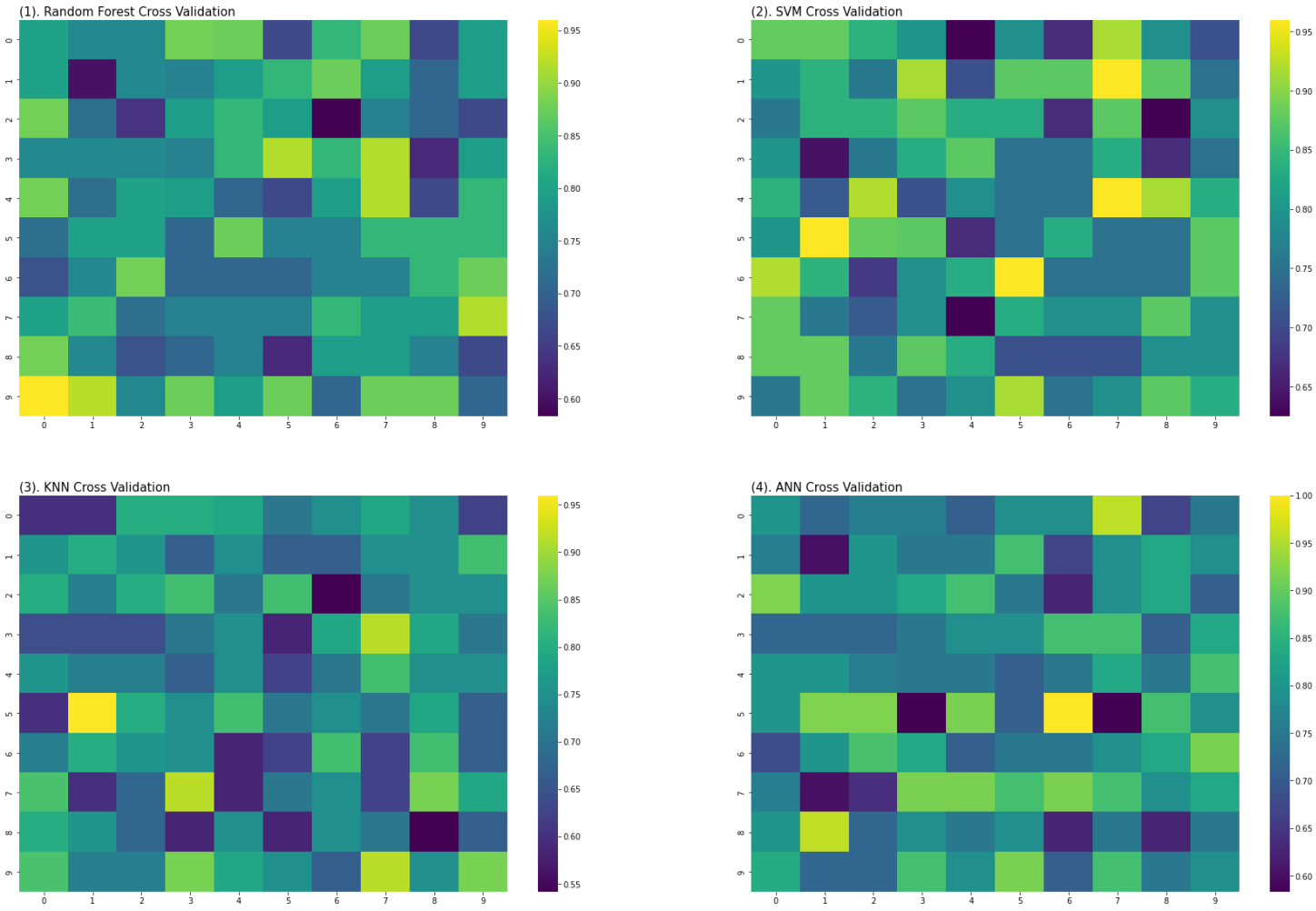

Figure 8 **Heat maps of RF, SVM, KNN and ANN (cross validation results).**

Full-size DOI: 10.7717/peerj.14104/fig-8

Table 7 **Dataset information of DHU-Pred, IRNAD and D-Pred.**

Predictor	Database used for tRNA sequence retrieval	Species	Samples count
DHU-Pred	RMBase	*Homosapiens, Mus musculus, & Saccharomyces Cerevisiae*	Positive = 1,035 Negative = 1,396
iRNAD	RMBase, Modomics	*Homosapiens, Mus musculus, Saccharomyces Cerevisiae, Escherichia coli, & Drosophila melanogaster*	Positive = 176 Negative = 374
D-Pred	RMBase	*Saccharomyces Cerevisiae*	Positive = 68 Negative = 68

Table 8. However, k-fold cross-validation on the whole dataset was applied, in which the dataset was divided into 10 folds (for $k = 10$), such that in each of the 10 iterations, the model was trained using k-1 folds and then validated on the remaining fold. Therefore, the

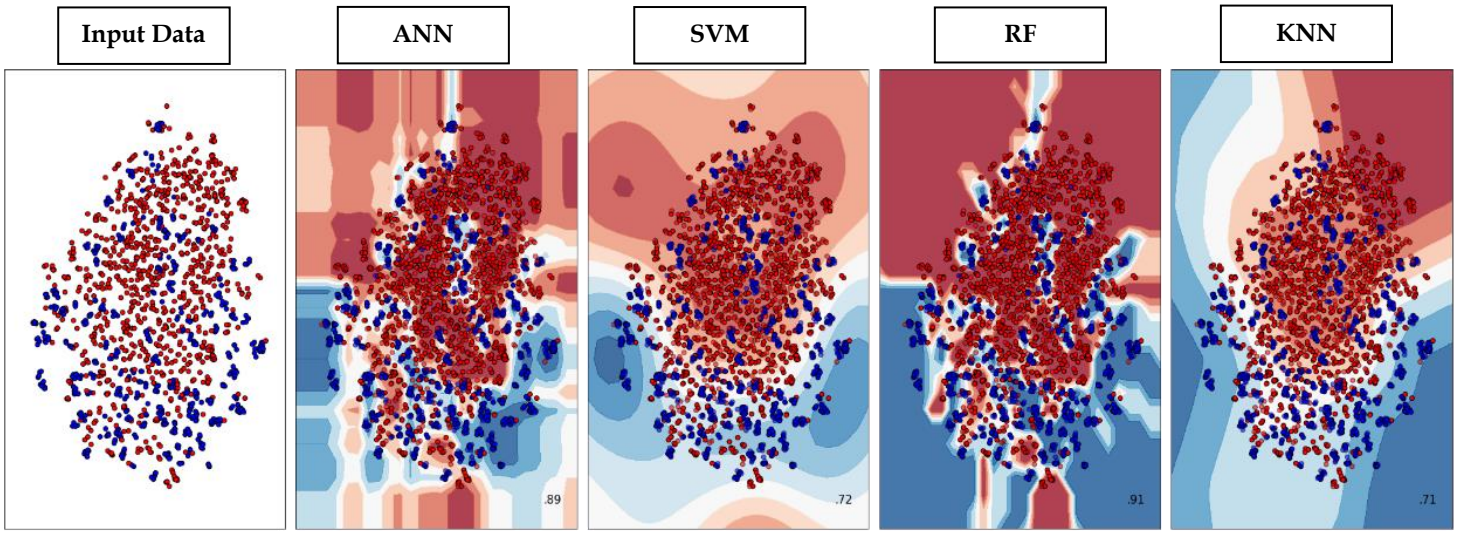

Figure 9 **Decision Boundary plots of different classification algorithms used in this study.**

Full-size DOI: 10.7717/peerj.14104/fig-9

Table 8 **Performance results of DHU-Pred, iRNAD, and D-Pred.**

Model	*Test samples*	*Acc*	S_n	S_p	*MCC*	*F1-score*
DHU-Pred	Positive = 207 Negative = 280	96.9%	98%	99%	0.97	0.96
iRNAD (*Xu et al., 2019*)	Positive = 207 Negative = 280	91.6%	92.05%	98.13%	0.91	0.89
D-Pred (*Feng et al., 2019*)	Positive = 207 Negative = 280	85.2%	73.1%	97.2%	0.72	0.74

cross-validation approach adopted in this study was meticulous and different from that of independent set testing, where a specific separate sample was used for testing.

The iRNAD and D-Pred revealed 91.6% and 85.2% accuracy, respectively, while DHU-Pred revealed a 96.9% accuracy score through independent testing, as in Table 8. The results in Table 8 show that the *Sn* and *Sp* scores achieved by iRNAD were 92.05% and 98.13%, while D-Pred revealed 73.1% and 97.2%, respectively. On the contrary, DHU-Pred revealed the *Sn* and *Sp* scores were 98% and 99%, respectively. The AUC-ROC graph in Fig. 10 also reveals that DHU-Pred outperformed both models, showing the high AUC value. This achievement was the comprehensive feature extraction method from the tRNA sequences.

Moreover, the inclusion of statistical moments into the obtained feature set helped build a more robust model for predicting D sites. The prediction of D sites is vital due to their role in the conformational flexibility of RNA and their significant presence in cancerous tissues. Therefore, the formulation of the benchmark dataset, the comprehensive method for feature generation and representation, the incorporation of different computational models, and evaluation through various testing methods helped us make a better model

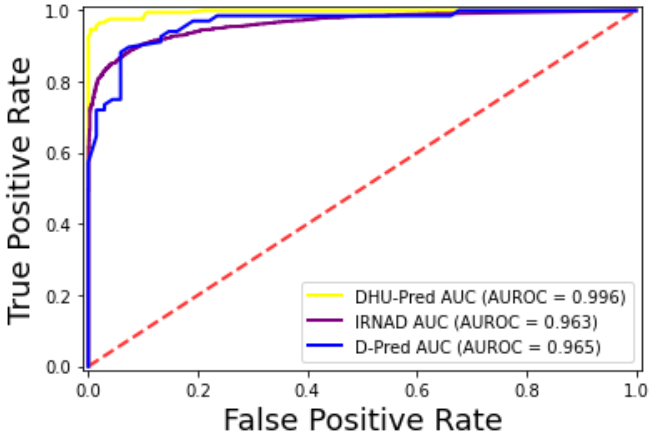

Figure 10 **ROC-Curve of DHU-Pred, iRNAD and D-Pred.**
Full-size DOI: 10.7717/peerj.14104/fig-10

for D sites prediction than other available models. Therefore, based on the detailed experiments, it can be concluded that DHU-Pred, represents high accuracy, robustness, and expansibility for identifying the D modification sites.

WEBSERVER

The availability of a web server is essential because it provides a quick and easy means of computational analysis. Furthermore, the accessibility to such online tools helps researchers in any future developments. For this purpose, an online web server for the proposed model, DHU-Pred, was developed and is freely available at https://dhu-prediction-app.herokuapp.com/.

CONCLUSION

Eukaryotes, bacteria, and even certain archaea all have high concentrations of D, a modified pyrimidine nucleoside. It aids nucleotide base conformational flexibility. Human pulmonary carcinogenesis is heavily influenced by this modification. In this research, computationally intelligent techniques were used to anticipate where D sites located in tRNA sequences. Features were computed for the stated goal using a convoluted approach based on statistical moments and position relative indices. The feature vectors were then incorporated into computational models for training. Cross-validation, jackknife testing, and independent set testing were used to assess these models. Using an independent set test, it was shown that the suggested RF-based model, DHU-Pred, revealed the highest results in all measures. DHU-Pred was compared extensively to popular academic models. Results from a comparison revealed that DHU-Pred performed far above the competition. As a result, the suggested model improved the identification capabilities of modified sites using the approaches described in the current study.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This project was supported by the Deanship of Scientific Research, Qassim University. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Deanship of Scientific Research, Qassim University.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Muhammad Taseer Suleman performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Tamim Alkhalifah conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Fahad Alturise performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Yaser Daanial Khan conceived and designed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

DNA Deposition

The following information was supplied regarding the deposition of DNA sequences:

The RNA sequence data used in this research study can be found in RMBase through the following URL: https://rna.sysu.edu.cn/rmbase/index.php.

The raw data is also available in the Supplemental Files.

Data Availability

The following information was supplied regarding data availability:

The code and data are available at Github: https://github.com/taseersuleman/DHU-Prediction-app.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj.14104#supplemental-information.

REFERENCES

Amanat S, Ashraf A, Hussain W, Rasool N, Khan YD. 2019. Identification of lysine carboxylation sites in proteins by integrating statistical moments and position relative features via general PseAAC. *Current Bioinformatics* **15**:396–407 DOI 10.2174/1574893614666190723114923.

Awazu A. 2017. Prediction of nucleosome positioning by the incorporation of frequencies and distributions of three different nucleotide segment lengths into a general pseudo k-tuple nucleotide composition. *Bioinformatics* **33**:42–48 DOI 10.1093/bioinformatics/btw562.

Barukab O, Khan YD, Khan SA, Chou K-C. 2019. iSulfoTyr-PseAAC: identify tyrosine sulfation sites by incorporating statistical moments via Chou's 5-steps rule and pseudo components. *Current Genomics* **20**:306–320 DOI 10.2174/1389202920666190819091609.

Boccaletto P, Machnicka MA, Purta E, Wirecki TK, Ross R, Limbach A, Kotter A, Helm M, Bujnicki JM. 2018. MODOMICS: a database of RNA modification pathways. 2017 update. *Nucleic Acids Research* **46**:303–307 DOI 10.1093/nar/gkx1030.

Bui VM, Lu CT, Ho TT, Lee TY. 2016. MDD-SOH: exploiting maximal dependence decomposition to identify S-sulfenylation sites with substrate motifs. *Bioinformatics* **32**:165–172 DOI 10.1093/bioinformatics/btv558.

Butt AH, Khan SA, Jamil H, Rasool N, Khan YD. 2016. A prediction model for membrane proteins using moments based features. *BioMed Research International* **2016**:8370132 DOI 10.1155/2016/8370132.

Cao C, Liu F, Tan H, Song D, Shu W, Li W, Zhou Y, Bo X, Xie Z. 2018. Deep learning and its applications in biomedicine. *Genomics, Proteomics and Bioinformatics* **16**:17–32 DOI 10.1016/j.gpb.2017.07.003.

Chen W, Chou K, Chen W. 2015. Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. *Molecular BioSystems* **11**:2620–2634 DOI 10.1039/C5MB00155B.

Cheng X, Zhao S-G, Xiao X, Chou K-C. 2017. iATC-mISF: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals. *Bioinformatics* **33**:2610–2610 DOI 10.1093/bioinformatics/btx387.

Chou KC. 2013. Some remarks on predicting multi-label attributes in molecular biosystems. *Molecular BioSystems* **9**:1092–1100 DOI 10.1039/c3mb25555g.

Dongardive J, Abraham S. Protein sequence classification based on N-gram and K-nearest neighbor algorithm. In: Behera GS, Durga PM, eds. *Computational Intelligence in Data Mining Volume 2*. 163–171 DOI 10.1007/978-81-322-2731-1.

Dou L, Zhou W, Zhang L, Xu L, Han K. 2021. Accurate identification of RNA D modification using multiple features. *RNA Biology* **18(12)**:2236–2246 DOI 10.1080/15476286.2021.1898160.

Dyubankova N, Sochacka E, Kraszewska K, Nawrot B, Herdewijn P, Lescrinier E. 2015. Contribution of dihydrouridine in folding of the D-arm in tRNA. *Organic and Biomolecular Chemistry* **13**:4960–4966 DOI 10.1039/c5ob00164a.

El Allali A, Elhamraoui Z, Daoud R. 2021. Machine learning applications in RNA modification sites prediction. *Computational and Structural Biotechnology Journal* **19**:5510–5524 DOI 10.1016/j.csbj.2021.09.025.

Feng P, Xu Z, Yang H, Lv H, Ding H, Liu L. 2019. Identification of D modification sites by integrating heterogeneous features in *Saccharomyces cerevisiae*. *Molecules* **24(3)**:24030380 DOI 10.3390/molecules24030380.

Han X, Wang X, Zhou K. 2019. Develop machine learning-based regression predictive models for engineering protein solubility. *Bioinformatics* **35**:4640–4646 DOI 10.1093/bioinformatics/btz294.

Hussain W, Khan YD, Rasool N, Khan SA, Chou KC. 2019a. SPrenylC-PseAAC: a sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-prenylation sites in proteins. *Journal of Theoretical Biology* **468**:1–11 DOI 10.1016/j.jtbi.2019.02.007.

Hussain W, Khan YD, Rasool N, Khan SA, Chou K-C. 2019b. SPalmitoylC-PseAAC: a sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying S-palmitoylation sites in proteins. *Analytical Biochemistry* **568**:14–23 DOI 10.1016/j.ab.2018.12.019.

Jain A, Kihara D. 2019. Phylo-PFP: improved automated protein function prediction using phylogenetic distance of distantly related sequences. *Bioinformatics* **35**:753–759 DOI 10.1093/bioinformatics/bty704.

Kato T, Daigo Y, Hayama S, Ishikawa N, Yamabuki T, Ito T, Miyamoto M, Kondo S, Nakamura Y. 2005. A novel human tRNA-dihydrouridine synthase involved in pulmonary carcinogenesis. *Cancer Research* **65**:5638–5646 DOI 10.1158/0008-5472.CAN-05-0600.

Khan YD, Alzahrani E, Alghamdi W, Ullah MZ. 2020. Sequence-based identification of allergen proteins developed by integration of PseAAC and statistical moments via 5-step rule. *Current Bioinformatics* **15**:1046–1055 DOI 10.2174/1574893615999200424085947.

Liu K, Chen W, Lin H. 2020. XG-PseU: an eXtreme gradient boosting based method for identifying pseudouridine sites. *Molecular Genetics and Genomics* **295**:13–21 DOI 10.1007/s00438-019-01600-9.

Lo C-H, Don H-S. 1989. 3-D moment forms: their construction and application to object identification and positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**:1053–1064 DOI 10.1109/34.42836.

Madec E, Stensballe A, Kjellstro S, Obuchowski M, Jensen ON, Cladie L, Se SJ. 2003. Mass spectrometry and site-directed mutagenesis identify several autophosphorylated residues required for the activity of PrkC, a Ser/Thr kinase from *Bacillus subtilis*. *Journal of Molecular Biology* **2836**:459–472 DOI 10.1016/S0022-2836(03)00579-5.

Mahmood MK, Ehsan A, Khan YD, Chou K-C. 2020. iHyd-LysSite (EPSV): identifying hydroxylysine sites in protein using statistical formulation by extracting enhanced position and sequence variant feature technique. *Current Genomics* **21**:536–545 DOI 10.2174/1389202921999200831142629.

Malebary SJ, Khan YD. 2021. Evaluating machine learning methodologies for identification of cancer driver genes. *Scientific Reports* **11**:12281 DOI 10.1038/s41598-021-91656-8.

Manavalan B, Basith S, Shin TH, Wei L, Lee G. 2019. mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* **35**:2757–2765 DOI 10.1093/bioinformatics/bty1047.

Meng C, Jin S, Wang L, Guo F, Zou Q. 2019. AOPs-SVM: a sequence-based classifier of antioxidant proteins using a support vector machine. *Frontiers in Bioengineering and Biotechnology* **7**:224 DOI 10.3389/fbioe.2019.00224.

Naseer S, Hussain W, Khan YD, Rasool N. 2020. Sequence-based identification of arginine amidation sites in proteins using deep representations of proteins and PseAAC. *Current Bioinformatics* **15**:937–948 DOI 10.2174/1574893615666200129110450.

Panwar B, Raghava GPS. 2014. Prediction of uridine modifications in tRNA sequences. *BMC Bioinformatics* **15**:326 DOI 10.1186/1471-2105-15-326.

Probst P, Wright MN, Boulesteix AL. 2019. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9**(**3**):e1301 DOI 10.1002/widm.1301.

Qiu WR, Sun BQ, Xiao X, Xu ZC, Chou KC. 2016. iPTM-mLys: identifying multiple lysine PTM sites and their different types. *Bioinformatics* **32**:3116–3123 DOI 10.1093/bioinformatics/btw380.

Shah AA, Khan YD. 2020. Identification of 4-carboxyglutamate residue sites based on position based statistical feature and multiple classification. *Scientific Reports* **10**:2–11 DOI 10.1038/s41598-020-73107-y.

Tseng WC, Medina D, Randerath K. 1978. Specific inhibition of transfer RNA methylation and modification in tissues of mice treated with 5-fluorouracil. *Cancer Research* **38**:1250–1257.

Vacic V, Iakoucheva LM, Radivojac P. 2006. Two sample logo: a graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* **22**:1536–1537 DOI 10.1093/bioinformatics/btl151.

Wenric S, Shemirani R. 2018. Using supervised learning methods for gene selection in RNA-Seq case-control studies. *Frontiers in Genetics* **9**:1–9 DOI 10.3389/fgene.2018.00297.

Wójcikowski M, Kukiełka M, Stepniewska-Dziubinska MM, Siedlecki P. 2019. Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. *Bioinformatics* **35**:1334–1341 DOI 10.1093/bioinformatics/bty757.

Xiao X, Xu ZC, Qiu WR, Wang P, Ge HT, Chou KC. 2019. iPSW(2L)-PseKNC: a two-layer predictor for identifying promoters and their strength by hybrid features via pseudo K-tuple nucleotide composition. *Genomics* **111**:1785–1793 DOI 10.1016/j.ygeno.2018.12.001.

Xu ZC, Feng PM, Yang H, Qiu WR, Chen W, Lin H. 2019. IRNAD: a computational tool for identifying D modification sites in RNA sequence. *Bioinformatics* **35**:4922–4929 DOI 10.1093/bioinformatics/btz358.

Xuan J, Sun W, Lin P, Zhou K, Liu S, Zheng L, Qu L, Yang J. 2017. RMBase v2. 0: deciphering the map of RNA modifications from epitranscriptome sequencing data. *Nucleic Acids Research* **46(D)**:D327–D334 DOI 10.1093/nar/gkx934.