Analytical Biochemistry 676 (2023) 115247

Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

PseU-Pred: An ensemble model for accurate identification of pseudouridine sites

Muhammad Taseer Suleman *, Yaser Daanial Khan

Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, 54770, Pakistan

ARTICLE INFO

Keywords:
Bioinformatics
Genomics
Proteomics
Cross-validation
Sequence analysis
Statistical moments

ABSTRACT

Pseudouridine (ψ) is reported to occur frequently in all types of RNA. This uridine modification has been shown to be essential for processes such as RNA stability and stress response. Also, it is linked to a few human diseases, such as prostate cancer, anemia, etc. A few laboratory techniques, such as Pseudo-seq and N3-CMC-enriched Pseudouridine sequencing (CeU-Seq) are used for detecting ψ sites. However, these are laborious and drawn-out methods. The convenience of sequencing data has enabled the development of computationally intelligent models for improving ψ site identification methods. The proposed work provides a prediction model for the identification of ψ sites through popular ensemble methods such as stacking, bagging, and boosting. Features were obtained through a novel feature extraction mechanism with the assimilation of statistical moments, which were used to train ensemble models. The cross-validation test and independent set test were used to evaluate the precision of the trained models. The proposed model outperformed the preexisting predictors and revealed 87% accuracy, 0.90 specificity, 0.85 sensitivity, and a 0.75 Matthews correlation coefficient. A web server has been built and is available publicly for the researchers at https://taseersuleman-y-test-pseu-pred-c2wmtj.streamlit.app/.

1. Introduction

Post-transcriptional modifications (PTMs) are a set of complex natural processes in which a primary transcript of RNA undergoes chemical alteration to produce mature RNA. The process of PTM is most common in eukaryotes. There are a variety of enzymes that play pivotal roles in such RNA modifications. So far, researchers have discovered more than 100 PTMs [1]. Among these PTMs, it is crucial to study uridine modifications. Orotidylate decarboxylase catalyzes the decarboxylation of orotidylate to generate uridine as uridine monophosphate (uridylate) in nature. Uridine is present as uridine monophosphate (UMP) in mother's milk [2]. Several uridine modifications have been reported by biologists. However, pseudouridine (ψ) and dihydrouridine (D) are the most frequently occurring modifications [3]. This research study deals with the identification of ψ sites in RNA samples. The ψ-synthase enzyme dissociates the sugar from the base of the uridine residue and spins it by 180° along the axis of the N3–C6 bond. Pseudouridine is formed because of the subsequent reattachment of the 5′-carbon of the base to the 1′-carbon of the sugar, which concludes the separation process and consequently results in the production of uridine [4]. A transformation of uridine to ψ-modification has been represented in Fig. 1. ψ-modification occurs in several RNA types, such as messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA). It has been observed that this modification is imperative for processes including RNA stabilization and the stress response [5]. Additionally, ψ-modifications in rRNA can impact the sensitivity of bacteria to antibiotics. ψ -modifications are responsible for a few human diseases such as dyskeratosis congenital [6], pituitary tumorigenesis [7], sideroblastic anemia [8], mitochondrial myopathy [9], lactic acidosis [10], and prostate cancer [11]. Several studies have found that ψ-modifications influence tRNA structure stabilization, gene regulation, and RNA splicing [12]. There exist conventional laboratory methods, including pseudo-seq and N3-CMC-enriched pseudouridine sequencing (CeU), for the detection of ψ sites [13,14]. Recently, fast, reliable, and inexpensive computational approaches for PTM prediction have emerged as an alternative to expensive and time-consuming laboratory studies. Ao et al. [15] have developed a website comprising pertinent information regarding recent research studies on DNA and RNA classification models. Based on the availability of sequence data, a few ψ site predictors were developed. Chen et al. [16] built a model, IRNA-PseU, for the identification of ψ

* Corresponding author.
E-mail addresses: s2018288002@umt.edu.pk (M.T. Suleman), yaser.khan@umt.edu.pk (Y.D. Khan).

https://doi.org/10.1016/j.ab.2023.115247
Received 9 March 2023; Received in revised form 25 June 2023; Accepted 8 July 2023
Available online 10 July 2023

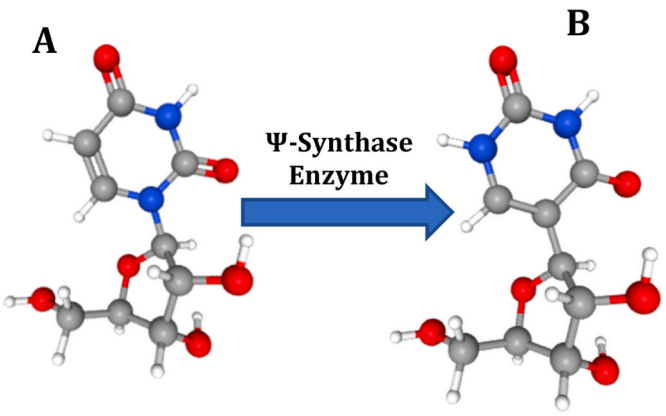

Fig. 1. (**A**). Formation of Uridine into (**B**) Pseudo uridine through ψ synthase enzyme.

sites in *Homosapiens*, *Saccharomyces cerevisiae*, and *Mus musculus*. Two independent datasets from *Homosapiens* and *Saccharomyces cerevisiae* were also made for independent set testing. The independent set test of iRNA-PseU revealed the accuracy of 65.0% in the *Homosapiens* and 73.0% for the *Saccharomyces cerevisiae* independent set respectively. In another study, researchers developed a support vector machine (SVM)-based model, PseUI, for predicting ψ sites in *Homosapiens*, *Saccharomyces cerevisiae*, and *Mus musculus*. The study employed a five-feature extraction strategy, including single nucleotide composition, position-specific nucleotide propensity, dinucleotide composition, pseudo dinucleotide composition, and position-specific dinucleotide propensity. The PseUI performance results revealed an accuracy of 65.50% and 68.50% in *Homosapiens* and *Saccharomyces cerevisiae*, respectively. In order to further optimize the ψ site prediction capability, Liu et al. [17] proposed RF-PseU for the prediction of ψ sites by training model on the same dataset of IRNA-PSeU. Deep learning methodologies have also been used in iPseU-CNN [18] and in the most recent study by Aziz et al. [19] for ψ site detection.

The current research focused on the identification of ψ sites through ensemble computational models. These models were categorized into stacking, bagging, and boosting. The benchmark dataset obtained from Chen et al. [16] was used for training as well as for testing and cross validation. It's also important to note that iRNA-PseU, PseUI, RF-PseU, and iPseU-CNN have used the same dataset for training and validation. The dataset is composed of RNA sequences belonging to three species: *Homosapiens*, *Saccharomyces cerevisiae*, and *Mus musculus*. The extraction of meaningful attributes from the sequences was carried out by considering the position and formation of nucleotide bases. Statistical moments were calculated that helped in feature dimensionality reduction [20]. The performance of these ensemble models was evaluated through k-fold cross validation and independent set testing. The accuracy metrics such as accuracy (Acc), sensitivity (S_n), specificity (S_p), and Matthew's correlation coefficient (MCC) were used to evaluate the models quantitatively. This research study was conducted in different phases, including benchmark dataset assortment, feature extraction and sample formulation, model development, training, and testing. Ultimately, a publicly accessible server was also made for facilitating in ψ sites detection. A complete flow of the current study has been represented in Fig. 2 which exhibits five steps of methodology.

2. Materials and methods

2.1. Dataset information

The training dataset was composed of data from three species. *HSP_990*, *SCV_628*, and *MMC_944* are training data sets for *Homosapiens*, *Saccharomyces cerevisiae*, and *Mus musculus*, respectively. The dataset was derived from RMBase v2.0, which contains data on different RNA modifications. This benchmark dataset was initially used by Chen et al. [16] for the identification of ψ sites. It should be noted that the same benchmark dataset has been employed in previous research studies including PseUI [21], RF-PseU [22], iRNAPseU [16], and iPseU-CNN [18].

2.1.1. Positive and negative samples

The *HSP_990* training dataset contained 495 ψ-sites (positive) and 495 non-ψ-sites (negative) sequences. The *SCV_628* training dataset contained 314 ψ-sites and 314 non-ψ sites sequences. Similarly, the *MMC_944* training dataset contained 944 sequences, half of which were positive samples. Whereas the independent testing data sets covered only two species, *Homosapiens* (*HSP_200*) and *Saccharomyces cerevisiae* (*SCV_200*), both of which contained 100 positive samples and 100 negative samples, respectively. There were no independent samples available for *Mus musculus*. For all training and testing data samples, the window size was set at 41 because optimal results were obtained by selecting this size. For positive (ψ sites) sequences, pseudouridine modification of the center uridine has been identified. Moreover, negative (non ψ sites) sequences were chosen based on the experimentally proved non modified uridine at central position 21.

An RNA sample containing the ψ site expressed as mentioned in (1).

$$Q(U) = Q_{-\tau}Q_{-(\tau-1)}\cdots Q_{-2}Q_{-1}UQ_{+1}Q_{+2}\cdots Q_{+(\tau-1)}Q_{+\tau} \quad (1)$$

The total length of a single RNA sample is formulated as $2\tau + 1$, where the subscripted value τ is 20. The symbol "*U*" represents uridine

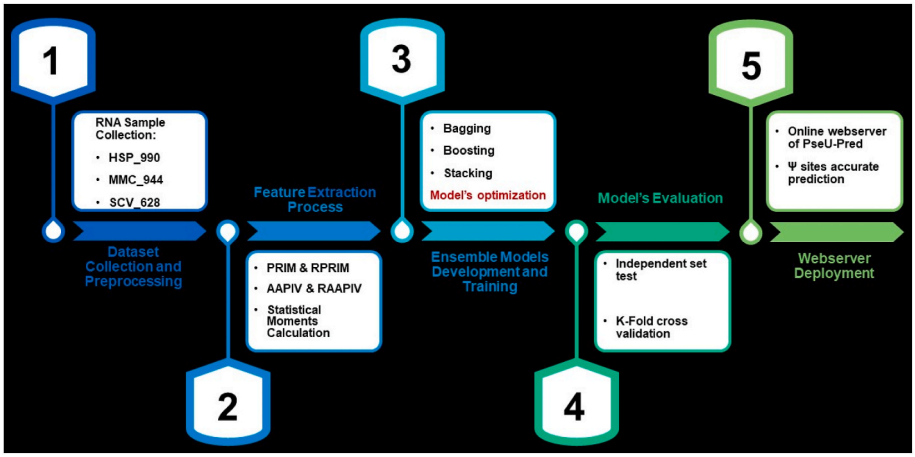

Fig. 2. Complete Flow diagram of current research methodology.

(either modified to ψ site or non-modified). $Q_{-\tau}$ represents the τ-th upstream nucleotide from the central uridine and $Q_{+\tau}$ represents the τ-th downstream nucleotide.

2.2. Feature extraction and development phase

Feature extraction is the most prevalent step in computational processes. This phase entails the feature extraction that highlights the dataset's characteristics [23]. Biotechnology has made significant progress due to recent developments in information and data sciences. However, the development of similar computationally intelligent models that transmute raw biological data into enumerated, quantifiable vectors is the most challenging. Also, it must be avoided that a single sequence or its related attributes are lost. This is because machine learning algorithms only require vectors as input [21,22]. Chou proposed a pseudo-amino acid composition for proteins (PseAAC) to prevent the total loss of the sequence-pattern information. The success of PseAAC led to the development of pseudo-K-tuple nucleotide composition (PseKNC) [24]. Furthermore, an RNA sequence, *C*, can be illustrated as shown in (2).

$$C = C_1, C_2, C_3, \cdots, C_i, \cdots, C_n \tag{2}$$

Whereas,

$$C_i \in \{A(adenine), C(cytosine), U(uracil), G(guanine)\}$$

denotes a random nitrogenous base at any arbitrary position within a nucleotide sequence. The current study used PseKNC's feature extraction technique with the addition of statistical moments for feature reduction [25]. Following that, the genomic data was translated into $Y^{'}$, a generalized stable numerical encoding, as expressed in (3).

$$Y^{'} = [Y_1 Y_2 Y_3 Y_4 \ldots Y_u \ldots Y_\Omega]^T \tag{3}$$

Where, Y_u, represents a random numerical coefficient representing a single feature. The transpose was applied on *C* of (2) to yield discrete coefficients, Y_i, where i = 1, 2, 3, …, ω represents the linear length of the sequence. The components in (3) were useful for deriving information from the gene sequence. A factual method was used to describe the components and get a numerical representation of the positive samples in the benchmark datasets.

2.2.1. Statistical moments calculation

Statistical moments were incorporated to obtain a fixed-size feature vector from the genomic data. Each moment furnished a unique piece of information about the type of data. Moments of different distributions have been studied by analysts and mathematicians [26,27]. The central, Hahn, and raw moments were computed to form a succinct feature set and subsequently reduce the enormous size of the input vector. The scale and area of relevant moments were included in the feature set to serve as a tool that may be used to distinguish between sequences that have functionally distinct roles [28]. These moments were embellished into the feature vector that subsequently formed a key component. As per scientific studies, genomic and proteomic sequence characteristics have been shown to vary with composition as well as the relative positioning of their bases. Therefore, only mathematical, and computational models that are receptive to the relative placement of component bases within genomic sequences are most suited for providing the feature vector [29]. In this work, raw, central, and Hahn moments were used for transmuting the features into succinct coefficients and subsequently reflect the tendency and irregularity of the data. Raw and Hahn moments are scale and location variants, rendering them more useful to decode the information within the sequence [30]. A two-dimensional matrix, $B^{'}$, was constructed from the sequences, with each entry, B_{mn}, representing the n_{th} nucleotide base in the, m_{th}, sequence as expressed in (4).

$$\mathrm{B}' = \begin{bmatrix} B_{11} & B_{12} & \ldots & B_{1n} \\ B_{21} & B_{22} & \ldots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \ldots & B_{mn} \end{bmatrix} \tag{4}$$

In order to extract location variant characteristics from extracted features, raw moments are used [31]. Raw moments are described in (5), where the total number of raw moments is denoted by the value of u + v. The coefficients E_{00}, E_{01}, E_{10}, E_{11}, E_{12}, E_{21}, E_{30}, and E_{03} were determined up to the third-degree polynomial.

$$E_{jk} = \sum_{c=1}^{m} \sum_{d=1}^{m} c^j d^k \beta_{cd} \tag{5}$$

The significance of the central moments is unrelated to the nucleotide's location. These, on the other hand, are associated with the composition and form of the distribution [32]. Furthermore, the centroid, $\mathscr{x}\mathscr{y}$, must be determined in order to get the central moments. For the current study, the central moments were computed and expressed as (6).

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

Orthogonal moments are often preferred because they can represent data with the least amount of redundant information [33]. Hahn moments perform relatively better than Chebyshev and Krawtchouk moments [34]. The reversible property of these moments guarantees that the predictor receives the effect of the whole sequence of data within the succinct feature vector, even though the original sequences have been greatly transformed into a fixed length. Hahn polynomials can be expressed as in (7).

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

Where, (u,v), are adjustable parameters that control polynomial shapes. Given a sequence in the form of a two-dimensional matrix, ***M*×*M***, the Hahn moment can be described as mentioned in (8).

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

2.2.2. Position Relative Incidence Matrix (PRIM)

This study aimed to enhance the model's predictive capabilities. Therefore, a comprehensive model for feature extraction was necessary for achieving this objective. The position relative incidence matrix (PRIM) is presented as a means of giving an account of the relative placement of nucleotide bases relative to one another [35]. The matrix, $\boldsymbol{F_{PRIM}}$ (9), is a **4×4** matrix that represents any single nucleotide, $\mathsf{K}_{\boldsymbol{m}}$, at position "***m***", with respect to other nucleotides within a sequence. The matrix generated 16 unique coefficients.

$$F_{PRIM} = \begin{bmatrix} \mathsf{K}_{A\to A} & \mathsf{K}_{A\to G} & \mathsf{K}_{A\to U} & \mathsf{K}_{A\to C} \\ \mathsf{K}_{G\to A} & \mathsf{K}_{G\to G} & \mathsf{K}_{G\to U} & \mathsf{K}_{G\to C} \\ \mathsf{K}_{U\to A} & \mathsf{K}_{U\to G} & \mathsf{K}_{U\to U} & \mathsf{K}_{U\to C} \\ \mathsf{K}_{C\to A} & \mathsf{K}_{C\to G} & \mathsf{K}_{C\to U} & \mathsf{K}_{C\to C} \end{bmatrix} \tag{9}$$

Where, $\mathsf{K}_{i\to j}$, represents the relative positioning of an arbitrary nucleotide base with respect to any other arbitrary base within a sequence. The occurrence of nucleotide base pairs (i.e., AA, AG, AU, …, CG, CU, CC) is significant in the feature extraction process. The formation of a 16×16 matrix known as G_{PRIM} (10), which results in 256 coefficients, was used to consider the frequency with which these base pairings occur in comparison to one another.

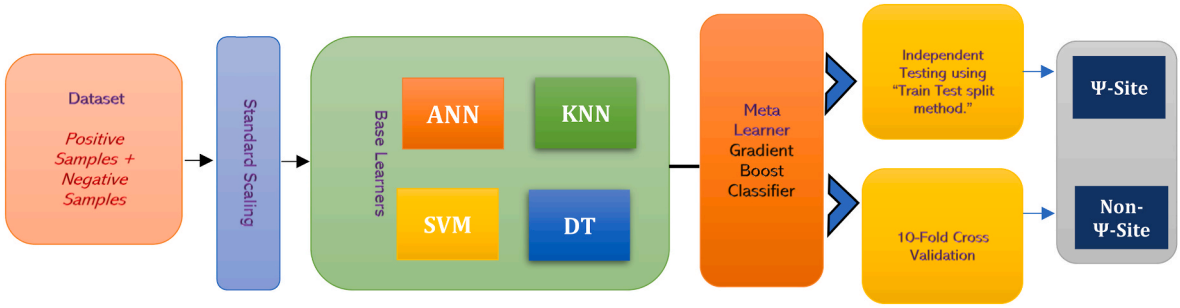

Fig. 3. Stacking ensemble model.

$$G_{PRIM} = \begin{bmatrix} \dot{G}_{AA\to AA} & \dot{G}_{AA\to AG} & \dot{G}_{AA\to AU} & \cdots & \dot{G}_{AA\to j} & \cdots & \dot{G}_{AA\to CC} \\ \dot{G}_{AG\to AA} & \dot{G}_{AG\to AG} & \dot{G}_{AG\to AU} & \cdots & \dot{G}_{AG\to j} & \cdots & \dot{G}_{AG\to CC} \\ \dot{G}_{AU\to AA} & \dot{G}_{AU\to AG} & \dot{G}_{AU\to AU} & \cdots & \dot{G}_{AU\to j} & \cdots & \dot{G}_{AU\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dot{G}_{GA\to AA} & \dot{G}_{GA\to AG} & \dot{G}_{GA\to AU} & \cdots & \dot{G}_{GA\to j} & \cdots & \dot{G}_{GA\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dot{G}_{N\to AA} & \dot{G}_{N\to AG} & \dot{G}_{N\to AU} & \cdots & \dot{G}_{N\to j} & \cdots & \dot{G}_{N\to CC} \end{bmatrix} \quad (10)$$

Similarly, another matrix, H_{PRIM} (11), was formed for the tri-nucleotide base combination (i.e., AAA, AAG, AAU, …. CCG, CCU, CCC). A total of 4096 coefficients were yielded by this matrix. The central, Hahn and raw moments were computed for F_{PRIM}, G_{PRIM} and H_{PRIM}, that resulted in forming coefficients up to order 3.

2.2.3. Reverse Position Relative Incidence Matrix (RPRIM)

The main goal of feature vector determination is to collect as much pertinent data as possible in order to build a reliable prediction model. A reverse position relative indices matrix (RPRIM) was produced by reversing the sequence order in an effort to obtain more information that was embedded within the sequences [36]. As with PRIM matrices, mononucleotide, dinucleotide, and trinucleotide combinations were used to calculate RPRIM. For this purpose, I_{RPRIM}, was calculated as stated in (12).

$$H_{PRIM} = \begin{bmatrix} \hat{H}_{AAA\to AAA} & \hat{H}_{AAA\to AAG} & \hat{H}_{AAA\to AAU} & \cdots & \hat{H}_{AAA\to j} & \cdots & \hat{H}_{AAA\to CCC} \\ \hat{H}_{AAG\to AAA} & \hat{H}_{AAG\to AAG} & \hat{H}_{AAG\to AAU} & \cdots & \hat{H}_{AAG\to j} & \cdots & \hat{H}_{AAG\to CCC} \\ \hat{H}_{AAU\to AAA} & \hat{H}_{AAU\to AAG} & \hat{H}_{AAU\to AAU} & \cdots & \hat{H}_{AAU\to j} & \cdots & \hat{H}_{AAU\to CCC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{H}_{AAC\to AAA} & \hat{H}_{AAC\to AAG} & \hat{H}_{AAC\to AAU} & \cdots & \hat{H}_{AAC\to j} & \cdots & \hat{H}_{AAC\to CCC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{H}_{N\to AAA} & \hat{H}_{N\to AAG} & \hat{H}_{N\to AAU} & \cdots & \hat{H}_{N\to j} & \cdots & \hat{H}_{N\to CCC} \end{bmatrix} \quad (11)$$

$$I_{RPRIM} = \begin{bmatrix} V_{1\to 1} & V_{1\to 2} & V_{1\to 3} & \cdots & V_{1\to y} & \cdots & V_{1\to j} \\ V_{2\to 1} & V_{2\to 2} & V_{2\to 3} & \cdots & V_{2\to y} & \cdots & V_{2\to j} \\ V_{3\to 1} & V_{3\to 2} & V_{3\to 3} & \cdots & V_{3\to y} & \cdots & V_{3\to j} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ V_{x\to 1} & V_{x\to 2} & V_{x\to 3} & \cdots & V_{x\to y} & \cdots & V_{4\to j} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ V_{N\to 1} & V_{N\to 2} & V_{N\to 3} & \cdots & V_{N\to y} & \cdots & V_{N\to j} \end{bmatrix} \quad (12)$$

The moments were calculated for the RPRIM matrix to obtain the refined feature values.

2.2.4. Frequency vector determination

The sequence's positional and compositional information is vital for generating attributes. The compositional information about a particular sequence is derived from the frequency count of each nucleotide within the sequence. For the current study, a frequency vector, ȣ, was calculated that stored the count for each nucleotide or nucleotide pair within the sequence [37]. The calculation of such a vector has been expressed in (13).

$$ȣ = \{ȵ_1, ȵ_2, \ldots.., ȵ_n\} \quad (13)$$

Where, $ȵ_i$, is the count of the i_{th} nucleotide in a sequence.

2.2.5. Generation of Accumulative Absolute Position Incidence Vector (AAPIV)

The AAPIV is responsible for providing the accumulated information pertaining to the occurrence of each individual nucleotide base [38]. Keeping in view of the single and paired nucleotide bases, three AAPIV were generated and designated as V_{AAPIV4} (14), $V_{AAPIV16}$ (15), and $V_{AAPIV64}$ (16).

$$V_{AAPIV4} = \{\delta_1, \delta_2, \delta_3, \delta_4,\} \quad (14)$$

$$V_{AAPIV16} = \{\delta_1, \delta_2, \delta_3, \ldots., \delta_{15}, \delta_{16},\} \quad (15)$$

$$V_{AAPIV64} = \{\delta_1, \delta_2, \delta_3, \ldots., \delta_{63}, \delta_{64}\} \quad (16)$$

Where, δ_i, can be calculated as provided in (17).

$$\delta_i = \sum_{k=1}^{n} p_k \quad (17)$$

2.2.6. Reverse Accumulative Absolute Position Incidence Vector (RAAPIV) generation

The reverse sequence provided a more in-depth perspective on the hidden patterns that were contained inside the gene sequence. In this research study, the process of computing AAPIV based on reversing the sequence is referred to as reverse accumulative absolute position incidence vector (RAAPIV) [39,40]. Three vectors, $V_{RAAPIV4}$, $V_{RAAPIV16}$, and $V_{RAAPIV64}$ were computed as expressed in (18), (19), and (20).

$$V_{RAAPIV4} = \{\tau_1, \tau_2, \tau_3, \tau_4\} \quad (18)$$

$$V_{RAAPIV16} = \{\tau_1, \tau_2, \tau_3, \ldots., \tau_{16}\} \quad (19)$$

$$V_{RAAPIV64} = \{\tau_1, \tau_2, \tau_3, \ldots., \tau_{64}\} \quad (20)$$

2.2.7. Feature vector formulation

The construction of a single feature vector was the final phase of the feature extraction procedure. This feature vector was then used as an input into the prediction model. The resultant feature vector contained 522 unique values, which were obtained through PRIM, RPRIM, FV, AAPIV, and RAAPIV. A single sample is represented by each feature vector in the dataset. Positive samples were identified as "1" and negative samples as "0" for binary classification.

2.3. Ensemble models development and training

For the current study, ensemble methods were used for the classification of ψ sites and non-ψ sites. Ensemble methods were preferred over conventional machine learning models because of their enhanced prediction capabilities [41]. Ensemble methods can be broadly divided into parallel and sequential methods. The bootstrap aggregation often used as bagging classifiers, such as random forest, uses the same process with

Table 1
Parameter tuning of the stacking model.

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 & its Hyper-parameter value(s)	**Gradient Boost classifier** *n_estimators* = 100, *criterion* = 'mse'			

random subsamples of the dataset using bootstrap sampling to reduce variance [42]. However, in sequential ensemble techniques such as boosting, the model can be boosted by using higher weights as compared to the previous model. Stacking, bagging, and boosting ensemble models were applied in this investigation.

2.3.1. Stacking ensemble

Stacking combines classification or regression models using a meta-classifier or meta-regressor [43]. The base-level models are trained using a complete training set, and then the meta-model uses their outputs as features. For the current study, the artificial neural network (ANN), k-nearest neighbor (KNN), support vector machine (SVM), and decision tree (DT) were used as base models, whereas the grading boost classifier was used as a meta classifier, as shown in Fig. 3. The base models were trained on training samples, and then each base model generated a prediction probability, P. The output of each base model was then used to train the meta classifier and finally make predictions on the test data. The hyperparameter optimization of all classifiers has been mentioned in Table 1.

2.3.2. Bagging ensemble

In bagging, the trained samples were divided into subsamples for the base models, such that the subsample must be less than the trained samples. The subsampling technique adopted row sampling with a replacement model. The test data was then validated through these trained models, and the final prediction was based on voting [44]. For the current study, the four bagging models were designed and trained, including the bagging classifier, random forest, extra tree, and decision tree classifier. A general representation of the bagging ensemble model has been shown in Fig. 4.

As an ensemble classifier, random forest uses many decision trees to analyze data from random samples and provide a single score. Considering optimizing a random forest model, the number of trees (n_estimators), depth of each tree (max_depth), maximum features (max_features), and a few other important parameters such as min_samples_split, bootstrap, and min_samples_leaf, the model was trained. The extra tree classifier employs a meta estimator that averages the results from fitting several randomized decision trees (also known as "extra-trees") to different subsamples of the dataset. In this research, the extra tree classifier's parameters are tuned by considering n_estimators, max_depth, max_features, and bootstrap. After learning basic decision rules from historical data, a decision tree classifier is used to anticipate the target variable's class or value (from the training data). For the current study, the decision classifier was optimized by considering splitter, max_depth, min_weight_fraction_leaf, and max_features. Bagging is an ensemble approach to machine learning that aggregates the predictions of several decision trees. The Bagging classifier works well and is the basis for several ensemble decision tree algorithms, including random forest, additional trees, pasting, random subspaces, and random patches. For the current study, the bagging classifier model was trained by considering parameters such as base_estimator, n_estimators, oob_score, and random_state. Table 2 contains the hyper-parameter optimization information of the aforementioned bagging models.

2.3.3. Boosting ensemble

The boosting ensemble method adopts the mechanism of optimizing the model based on the output of the previous model [45]. It works in a sequential way by reducing the differentiable loss. For the current study, a few boosting ensemble methods were used for training, which include gradient boosting, histogram-based gradient boosting (HGB), adaboost, and extreme gradient boosting (XGB). Fig. 5 depicts the boosting ensemble model applied in the current study.

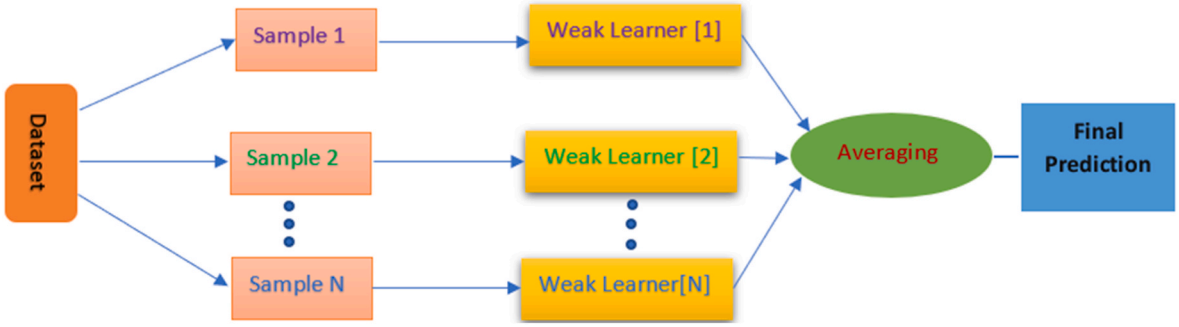

Fig. 4. Bagging ensemble method representation.

Table 2
Parameters tuning of the bagging ensemble models.

Bagging models	Random Forest	Extra tree classifier	Decision Tree classifier	Bagging classifier
Hyper-Parameter value(s)	*n_estimators* = 200 *max_depth* = 50 *max_features* = 'Auto' *min_samples_split* = 10 *min_samples_leaf* = 5	*n_estimators* = 100 *max_depth* = 40 *max_features* = 'Auto' *Bootstrap* = bool	*Splitter* = 'random' *Max_depth* = 80 *min_samples_leaf* = 4 *random_state* = 'None' *min_weight_fraction_leaf* = 0.1	*Base_estimator* = 'DecisionTreeClassifier' *N_estimators* = 100 *Oob_score* = 'True' *Random_state* = 0

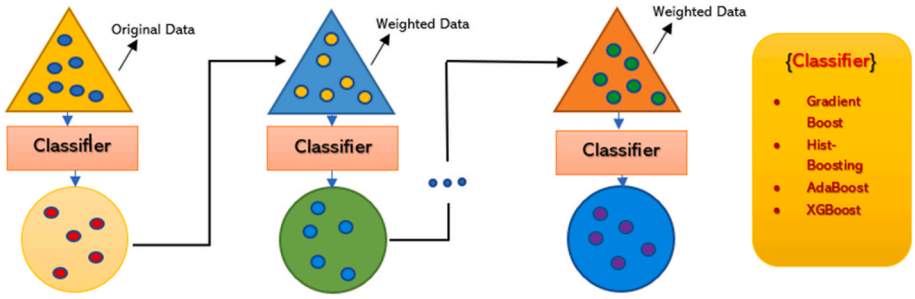

Fig. 5. Boosting Ensemble model.

Table 3
Hyper-parameters optimization of the boosting ensemble models.

Boosting ensemble models	Gradient Boost	Hist-Boost	Adaboost	XGB
Hyper-Parameter value(s)	*learning_rate* = 0.1 *n_estimators* = 100 *criterion* = 'mse'	*max_iter* = 200 max_depth = 40 *warm_start* = 'True'	*Base_estimator* = 'Gradientboostclassifier' *n_estimators* = 50 *random_state* = 'None' *min_weight_fraction_leaf* = 0.1	*max_iter* = 100 *max_depth* = 40 *random_state* = 0

Gradient boosting is sometimes referred to as "gradient tree boosting" or "gradient boosting machines" (GBM). Decision tree models are used to build ensembles. Trees are introduced to the ensemble one at a time and fitted to correct previous model prediction errors. Boosting is a form of ensemble machine learning model. Similarly, HGB is an optimized boosting ensemble technique that also performs well on a large dataset. The Adaboost ensemble method is based on a few decision trees that are added to the model one at a time as "week learners." Another open source boosting ensemble method, XGB, was deployed for optimizing the prediction capabilities of the model. It is intended to be both extremely effective and computationally efficient, maybe even more so than existing open-source versions. As shown in Table 3, the boosting ensemble models were fine-tuned by adjusting various hyper-parameters.

3. Results and discussion

Because of the availability of sequencing data and smart computational algorithms, it is now possible to predict the PTM sites accurately and quickly. The precise identification of these sites helps in the diagnosis of various PTM-linked diseases such as Bowen-Conradi syndrome [46], mitochondrial infantile liver disease [47], mitochondrial respiratory chain defects [48], bladder cancer [49], Rolandic epilepsy [50], and neuropathy [51]. To enhance the prediction capabilities of the model, it was evaluated through independent testing and cross validation. It is important to mention that a separate dataset was used for independent test evaluation. However, the whole dataset was used for cross validation. Different accuracy metrics were used to score the performance of the model.

3.1. Metrics for evaluation

The prediction models in this study were investigated using four different metrics: sensitivity (S_n), specificity (S_p), accuracy (Acc), and Mathew's correlation coefficient (MCC). The K^+ symbols denote the true ψ sites, whereas the K^- symbols denote the renegade ψ sites. A similar notation, K_-^+, represents the total number of changed sites that were indeed legitimate ψ sites but were misidentified as rogue ψ sites. Furthermore, K_+^- stands for the total number of fake ψ sites that were misidentified. However, it's important to note that the measurements only apply to systems with a single class. The accuracy metrics equations have been mentioned in (21).

$$\begin{cases} S_n = 1 - \dfrac{K_-^+}{K^+} & 0 \le S_n \le 1 \\ S_p = 1 - \dfrac{K_+^-}{K^-} & 0 \le S_p \le 1 \\ Acc = 1 - \dfrac{K_-^+ + K_+^-}{K^+ + K^-} & 0 \le Acc \le 1 \\ MCC = \dfrac{1 - \left(\dfrac{K_-^+}{K^+} + \dfrac{K_+^-}{K^-}\right)}{\sqrt{\left(1 + \dfrac{K_+^- - K_-^+}{K^+}\right)\left(1 + \dfrac{K_-^{\pm K_+^-}}{K^-}\right)}} & -1 \le MCC \le 1 \end{cases} \tag{21}$$

The S_n, often termed the "true positive rate" (TPR), is the proportion of genuine ψ sites that provide an accurate result. All positive panel

Table 4
Independent set test results comparison of stacking, bagging, and boosting ensemble models.

Model	Classifier(s)	Independent Data samples							
		HS_200				SC_200			
Stacking	**Stacked**	***ACC***	***Sp***	***Sn***	***MCC***	***ACC***	***Sp***	***Sn***	***MCC***
		0.82	0.83	0.82	0.65	080	0.81	0.79	0.63
Bagging	**Random Forest**	0.87	0.90	0.85	0.75	0.87	0.89	0.84	0.73
	Extra Tree Classifier	0.86	0.85	0.87	0.72	0.86	0.85	0.88	0.75
	Decision Tree	0.71	0.87	0.55	0.44	0.68	0.84	0.52	0.41
	Bagging classifier	0.85	0.90	0.81	0.71	0.86	0.91	0.82	0.72
Boosting	**Gradient Boost**	0.85	0.91	0.79	0.70	0.80	0.85	0.74	0.75
	HistGradient Boost	0.83	0.92	0.75	0.67	0.85	0.94	0.80	0.72
	AdaBoost	0.86	0.91	0.80	0.71	0.82	0.90	0.81	0.69
	XGBoost	0.85	0.90	0.80	0.70	0.86	0.91	0.81	0.72

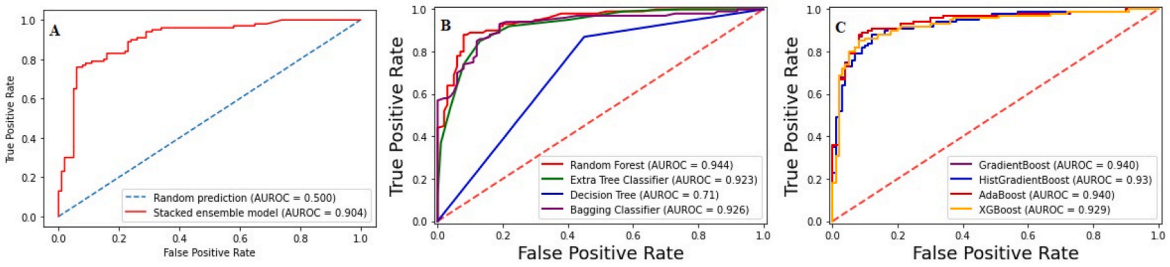

Fig. 6. Independent Testing ROC curves (**A**). Stacked ensemble ROC (**B**). Boosting ensemble models ROC (**C**). Bagging ensemble techniques ROC.

Table 5
K-Fold cross validation results of stacking, bagging, and boosting ensemble models.

Model	Classifier(s)	K-Fold Cross Validation results											
		HSP_990				SCV_628				MMC_944			
Stacking	**Stacked**	***ACC***	***Sp***	***Sn***	***MCC***	***ACC***	***Sp***	***Sn***	***MCC***	***ACC***	***Sp***	***Sn***	***MCC***
		0.94	0.94	0.97	0.88	0.91	0.90	0.95	0.85	0.90	0.89	0.93	0.81
Bagging	**Random Forest**	0.92	0.90	0.94	0.84	0.93	0.90	0.95	0.86	0.88	0.85	0.88	0.80
	Extra Tree Classifier	0.90	0.87	0.94	0.80	0.88	0.85	0.92	0.78	0.80	0.72	0.74	0.60
	Decision Tree	0.88	0.85	0.94	0.78	0.86	0.83	0.93	0.75	0.89	0.86	0.94	0.77
	Bagging classifier	0.93	0.90	0.97	0.86	0.95	0.92	0.98	0.87	0.79	0.82	0.69	0.70
Boosting	**Gradient Boost**	0.95	0.94	0.97	0.90	0.91	0.93	0.95	0.86	0.90	0.92	0.94	0.83
	HistGradient Boost	0.94	0.94	0.90	0.88	0.88	0.90	0.85	0.86	0.86	0.87	0.86	0.82
	AdaBoost	0.95	0.94	0.97	0.90	0.92	0.93	0.95	0.88	0.91	0.93	0.96	0.89
	XGBoost	0.93	0.90	0.97	0.86	0.89	0.88	0.93	0.85	0.79	0.78	0.78	0.65

samples (or true ψ sites) are measured using the sensitivity. The S_p, or true negative rate (TNR), is the fraction of non-ψ sites within RNA samples that were predicted to be class negative. The *MCC* is a more reliable statistical rate that only yields a high score if prediction performed well in obtaining an optimized confusion matrix according to the size of the dataset's positive and negative components. The *Acc* measures the overall prediction accuracy of a model.

3.2. Test methods

For the current research study, independent testing and cross validation methods were used for the model's performance evaluation. Separate independent samples were provided in the benchmark dataset that belong to the *HSP_200* and *SCV_200* species. Each independent set contained 100 positive samples (ψ sites) and 100 negative samples (non-ψ sites). On the other hand, k-fold cross-validation was performed on the whole dataset. This included dividing the dataset into 10 folds (because k was set to 10) in such a way that, for each of the 10 iterations, the model was trained using k-1 folds and then verified on the fold that was left over. So, cross-validation was different from an independent test, which used a separate sample for validation.

Table 6
Comparative results of accuracy metrics using different feature matrices and vectors.

Features	Independent Data samples							
	HS_200				SC_200			
	ACC	*Sp*	*Sn*	*MCC*	*ACC*	*Sp*	*Sn*	*MCC*
PRIM	0.86	0.83	0.83	0.66	0.60	0.50	0.70	0.20
RPRIM	0.83	0.82	0.83	0.66	0.65	0.55	0.74	0.23
FV	0.77	0.75	0.74	0.61	0.70	0.73	0.72	0.59
PRIM + AAPIV	0.81	0.86	0.76	0.63	0.61	0.56	0.66	0.23
PRIM + RAAPIV	0.75	0.79	0.70	0.58	0.64	0.59	0.49	0.27
AAPIV + RAAPIV	0.76	0.80	0.73	0.60	0.80	0.81	0.80	0.60

3.2.1. Independent set testing

An independent set test was carried out using separate data samples of *HSP_200* and *SCV_200* using the stacking, bagging, and boosting models. All the accuracy metrics were best served by the proposed model, as shown in Table 4.

From the results provided in Table 4, it can be observed that the random forest ensemble model outperformed in *HSP_200* independent testing. The random forest ensemble model achieved the highest score in *ACC* and *MCC*. However, in terms of S_n, the extra tree classifier revealed a high score among all types of ensemble models implemented in this research. As a result, for *HSP_200* independent testing, a random forest ensemble proved to be the best model for ψ site and non-ψ site classification. For *SCV_200*, the highest scores in *ACC*, S_n, and *MCC* revealed by the extra tree classifier ensemble model are 0.87, 0.88, and 0.75, respectively. The HGB ensemble succeeded in achieving highest S_p which is 0.94. Therefore, for the *SCV_200* independent testing, the extra tree classifier outperformed in achieving relatively high scores. All the bagging ensemble model's area under the curve (AUC) in independent set testing is depicted in Fig. 6.

3.2.2. Cross validation

The cross-validation approach is used to test all the samples while splitting the dataset into "k" disjoint folds. The robustness of a model is demonstrated by this more stringent test. In this test, k-1 folds (partitions) were trained on the model, while testing was performed on the left-over fold. The test was repeated 10 times due to the number of folds used in this study, i.e., k = 10. Cross-validation results have been listed in Table 5.

For the *Homosapiens* dataset, *HSP_990*, the stacking ensemble model revealed a 0.94 score in *ACC* and S_p, 0.97 S_n and 0.88 *MCC*. For the *SCV_628* set, the accuracy metric scores were relatively low, such as 0.91 *ACC*, 0.90 S_p, 0.95 S_n and 0.85 *MCC*. For *MMC_944*, the scores were too close to the *SCV_628* validation results, revealing 0.90 *ACC*, 0.89 S_p, 0.93 S_n and 0.81 *MCC*. In the bagging ensemble model, the bagging classifier revealed maximum values in all accuracy metrics using both the *HSP_990* and *SCV_628* datasets. However, the scoring revealed by the bagging classifier in *MMC_944* was not very impressive. The decision tree classifier revealed maximum accuracy scores while cross validating

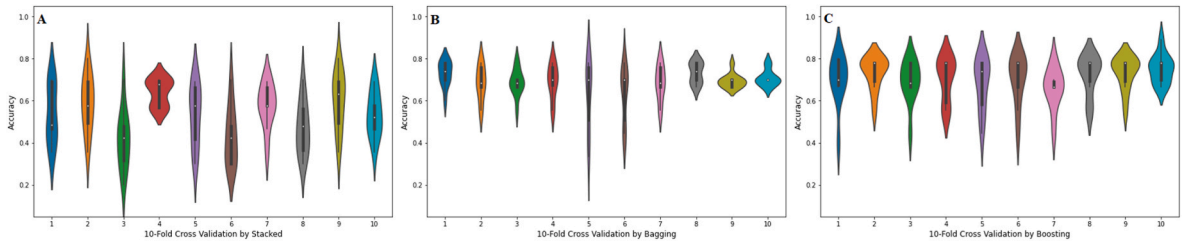

Fig. 7. Violin plots of 10-Fold cross validation accuracy (Acc) metric results for (**A**) Stacked ensemble (**B**) Bagging ensemble and (**C**) Boosting ensemble.

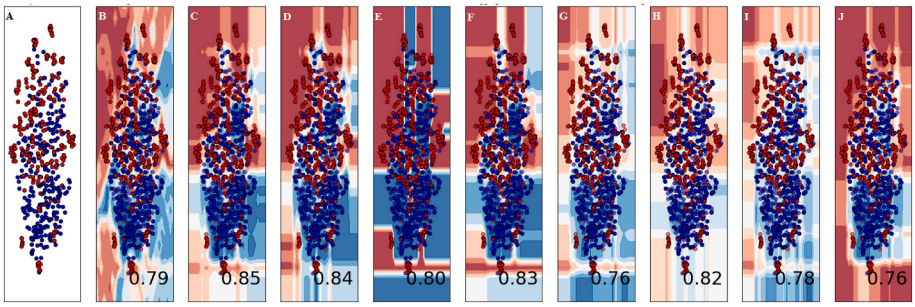

Fig. 8. Boundary visualization of ensemble models used in this study as follows: (**A**). Input data (**B**). Stacking (**C**). Random Forest (**D**) ExtraTree (**E**) Decision Tree (**F**) Bagging (**G**) Gradient Boost (**I**) Histo Gradient Boost (**H**) Adaboost (**I**) XGBoost.

MMC_944 as shown in Table 5.

Moreover, in the boosting ensemble model, the gradient boost and Adaboost revealed similar scores in *ACC*, S_p, S_n and *MCC* using the *Homosapiens* dataset. Using the *SCV_628* dataset, the Adaboost showed 0.92 *ACC*, 0.93 S_p, 0.95 S_n and 0.88 *MCC*.

For k-fold cross validation, each species data set was then subjected to 10-fold cross validation, deploying its own samples. The results in Table 5 also indicated that each dataset, i.e., *HSP_990*, *SCV_628* and *MMC_944*, was cross-validated separately using 10-fold cross-validation over all the ensemble models. In order to gain deeper insights, the models were trained and tested using a set of obtained features. The accuracy values of independent testing based on these features are presented in Table 6. The table reveals that individual features from different matrices and vectors do not demonstrate optimized scores across all metrics comparable to the combination of those features.

The violin plot is a graphical representation of the distribution of numerical data for one or more groups using density curves [52]. The median may be represented by a white dot on the plot, the interquartile range by a black bar in the middle, and the lower and higher neighbouring values by dark black lines extending from the bar. Fig. 7 shows violin plots for the accuracy values found in each fold for the best ensemble models for stacking, bagging, and boosting.

The use of supervised machine learning models may be helpful in many categorization tasks. However, numerical prediction alone is not always sufficient. It is essential to have a clear image of the actual decision border that separates the groups. As a result, the classification algorithms utilized in this study were subjected to a decision surface in order to improve accuracy. In this decision surface map, a trained machine learning system predicts a coarse grid over the input feature space. First, the model was calibrated using the dataset used for training. After that, the trained model was applied to the problem of making predictions for a grid of values across the input domain. For the purpose of charting, the contourf() function that is available in matplotlib [53] and scatterplot [54] was used. The decision surface plots of the classification algorithms used in this work are shown in Fig. 8, which may be seen below.

Table 7
Comparison with existing predictors.

Predictor	Independent Data samples							
	HS_200				SC_200			
	ACC	*Sp*	*Sn*	*MCC*	*ACC*	*Sp*	*Sn*	*MCC*
PseUI [21]	0.65	0.68	0.63	0.31	0.68	0.65	0.72	0.37
RF-PseU [22]	0.75	0.72	0.78	0.50	0.77	0.79	0.73	0.54
iRNA-PseU [16]	0.65	0.70	0.60	0.30	0.73	0.65	0.81	0.46
iPseU-CNN [18]	0.69	0.60	0.77	0.40	0.73	0.77	0.68	0.47
Predictor by Aziz et al [19]	0.74	0.75	0.73	0.48	0.76	0.73	0.80	0.53
PseU-Pred	**0.87**	**0.90**	**0.85**	**0.75**	**0.87**	**0.85**	**0.88**	**0.72**

3.3. Comparative analysis

The proposed model, PseU-Pred, was built on the best performing random forest ensemble model and compared with preexisting predictors to assess the model's efficacy on the independent datasets. The predictors were PseUI, iRNA-PseU, RF-PseU, iPseU-CNN, and a predictor developed by Aziz et al. It was observed that the PseU-Pred outperformed independent set testing in both *HSP_200* and *SCV_200*. The accuracy metrics score of each predictor is listed in Table 7.

From the results mentioned in Table 7, it was observed that the proposed model, PseU-Pred, outperformed in both independent set tests. It is important to mention here that the current analysis employed the identical *Saccharomyces cerevisiae* and *Homosapiens* samples as previous prediction studies. The proposed model achieved 0.87 *ACC* in both independent tests, which is relatively high compared to other predictors. Moreover, the S_p, S_n and *MCC* were quite higher than other predictors. Fig. 9 depicts a bar chart comparison of the proposed model with the predictors in all accuracy metrics. The limitation of current research study is the availability of limited experimentally proved RNA sequences. Since no hypothetical samples were created, the available concrete samples were used in the process of feature extraction, computational models' development, training and testing of models.

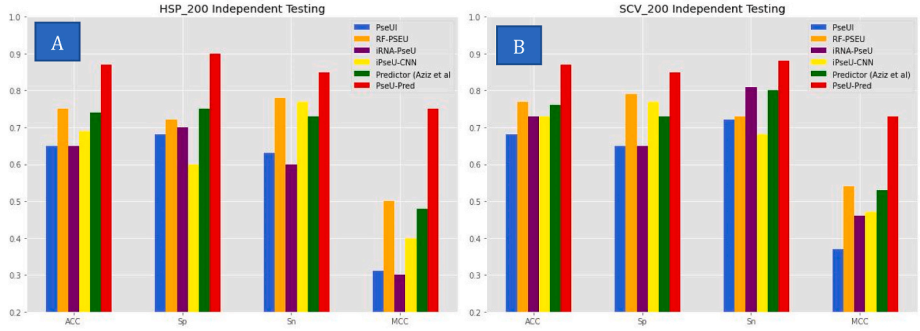

Fig. 9. Bar chart comparison of proposed model with comparative models in ***ACC***, S_p, S_n and ***MCC*** (**A**) For *HSP_200* independent set (**B**) For *SCV_200* independent set.

Moreover, there exists a constraint of the RNA sequences belonging to only three species such as *Homosapiens*, *Mus musculus*, and *Saccharomyces cerevisiae*. Therefore, only these species were considered for identification of ψ sites in RNA sequences. The enhancement in the identification of ψ sites is important since this RNA modification is responsible for a few diseases such as Dyskeratosis congenita, pituitary tumorigenesis, and prostate cancer. Predicting the presence of pseudouridine residues in RNA sequences can provide valuable biological and clinical insights including RNA stability, RNA regulatory mechanisms, identification of ψ sites-associated diseases, and also RNA-based therapeutics. Therefore, the comprehensive strategy for feature development and representation, the merging of several computational models, and the assessment using a variety of testing methodologies made it feasible for the creation of a model for predicting ψ sites that is superior to other models now available. Based on the extensive trials, it can be stated that the proposed model has a high degree of precision, resilience, and scalability for finding modified ψ sites.

4. Webserver availability

A web server offers a quick and simple way to do computational analysis. Additionally, the availability of such internet resources aids scholars in any upcoming breakthroughs. PseU-Pred, a free online web server for the suggested model, was created with this objective in mind and is accessible at https://taseersuleman-y-test-pseu-pred-c2wmtj.streamlit.app/.

5. Conclusion

The proposed work fixated on the identification of one of the most prevalent post-transcriptional modifications, pseudouridine (ψ), within RNA sequences through ensemble methods. The prediction of ψ sites is important because the said modification is involved in various human diseases such as Dyskeratosis congenital, Pituitary tumorigenesis, Sideroblastic Anemia, Mitochondrial Myopathy, Lactic acidosis, and prostate cancer. A novel feature extraction mechanism based on positional as well as compositional attributes of nucleotides within RNA sequences was adopted. Statistical moments were determined that helped in feature dimensionality reduction. The subsequent feature set was further employed to train various ensemble models based on stacking, bagging, and boosting. The trained models were then evaluated through cross validation. An independent test was also carried out using a separate sample of *Homosapiens* and *Saccharomyces cerevisiae*. All the models were evaluated against popular accuracy metrics such as accuracy, sensitivity, specificity, and Matthew's correlation coefficient. The proposed model, PseU-Pred, was then built based on the best performing ensemble model. A comparative analysis of PseU-Pred was carried out with existing predictors. It was observed that PseU-Pred revealed the highest score in all accuracy metrics as compared to other predictors. Therefore, it has been concluded that the proposed model enhanced the ability to identify modified ψ sites using the methods discussed above.

Author contributions

The manuscript was prepared by Muhammad Taseer Suleman. The implementation of the current research study is done by Muhammad Taseer Suleman and Yaser Daanial Khan. The manuscript was reviewed and supervised by Yaser Daanial Khan.

Funding

There is no funding for the production and publication of this research.

Data and code availability

The code and data of the current research is available at https://github.com/taseersuleman/Y_test.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data and code of the current research is available at https://github.com/taseersuleman/Y_test.

References

[1] P. Boccaletto, et al., MODOMICS : a Database of RNA Modification Pathways . 2017 Update, vol. 46, November 2017, pp. 303–307, https://doi.org/10.1093/nar/gkx1030, 2018.
[2] R. Wurtman, A nutrient combination that can affect synapse formation, Nutrients 6 (4) (Apr. 2014) 1701–1710, https://doi.org/10.3390/nu6041701.
[3] B. Panwar, G.P.S. Raghava, Prediction of uridine modifications in tRNA sequences, BMC Bioinf. 15 (2014) 326, https://doi.org/10.1186/1471-2105-15-326.
[4] M. Charette, M.W. Gray, Pseudouridine in RNA: what, where, how, and why, IUBMB Life 49 (5) (2000) 341–351, https://doi.org/10.1080/152165400410182.
[5] B.S. Zhao, C. He, Pseudouridine in a new era of RNA modifications, Cell Res. 25 (2) (2015) 153–154, https://doi.org/10.1038/cr.2014.143.
[6] J. Carrillo, et al., High resolution melting analysis for the identification of novel mutations in DKC1 and TERT genes in patients with dyskeratosis congenita, Blood Cells Mol. Dis. 49 (3–4) (2012) 140–146, https://doi.org/10.1016/j.bcmd.2012.05.008.
[7] C. Bellodi, et al., Loss of function of the tumor suppressor DKC1 perturbs p27 translation control and contributes to pituitary tumorigenesis, Cancer Res. 70 (14) (2010) 6026–6035, https://doi.org/10.1158/0008-5472.CAN-09-4730.
[8] A. Zeharia, 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. 20 (5) (2005) 449–452, https://doi.org/10.1177/08830738050200051301.

[9] E. Fernandez-Vizarra, A. Berardinelli, L. Valente, V. Tiranti, M. Zeviani, Nonsense mutation in pseudouridylate synthase 1 (PUS1) in two brothers affected by myopathy, lactic acidosis and sideroblastic anaemia (MLASA), BMJ Case Rep. (2009), https://doi.org/10.1136/bcr.05.2009.1889.
[10] J.R. Patton, Y. Bykhovskaya, E. Mengesha, C. Bertolotto, N. Fischel-Ghodsian, 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. 280 (20) (2005) 19823–19828, https://doi.org/10.1074/jbc.M500216200.
[11] P. S, et al., DKC1 overexpression associated with prostate cancer progression, Br. J. Cancer 101 (8) (2009) 1410–1416, https://doi.org/10.1038/sj.bjc.6605299 [Online], https://www.embase.com/search/results?subaction=viewrecord&from=export&id=L50639197%0A.
[12] Y. Ge, RNA pseudouridylation: new insights into an old modification, Trends Biochem. Sci. 38 (4) (2014) 210–218.
[13] A. Basak, C.C. Query, A pseudouridine residue in the spliceosome core is part of the filamentous growth program in yeast, Cell Rep. 8 (4) (2014) 966–973, https://doi.org/10.1016/j.celrep.2014.07.004.
[14] X. Li, et al., Chemical pulldown reveals dynamic pseudouridylation of the mammalian transcriptome, Nat. Chem. Biol. 11 (8) (2015) 592–597, https://doi.org/10.1038/nchembio.1836.
[15] C. Ao, S. Jiao, Y. Wang, L. Yu, Q. Zou, Biological sequence classification: a review on data and general methods, Research (Jan. 2022), https://doi.org/10.34133/research.0011, 2022.
[16] W. Chen, H. Tang, J. Ye, H. Lin, K.C. Chou, iRNA-PseU: identifying RNA pseudouridine sites, Mol. Ther. Nucleic Acids 5 (2016) e332, https://doi.org/10.1038/mtna.2016.37.
[17] K. Liu, W. Chen, H. Lin, XG-PseU: an eXtreme Gradient Boosting based method for identifying pseudouridine sites, Mol. Genet. Genom. 295 (1) (2020) 13–21, https://doi.org/10.1007/s00438-019-01600-9.
[18] M. Tahir, H. Tayara, K.T. Chong, iPseU-CNN: identifying RNA pseudouridine sites using convolutional neural networks, Mol. Ther. Nucleic Acids 16 (2019) 463–470, https://doi.org/10.1016/j.omtn.2019.03.010.
[19] A.Z. Bin Aziz, M. Al Mehedi Hasan, J. Shin, Identification of RNA pseudouridine sites using deep learning approaches, PLoS One 16 (2 February) (2021), https://doi.org/10.1371/journal.pone.0247511.
[20] O. Barukab, Y.D. Khan, S.A. Khan, K.-C. Chou, iSulfoTyr-PseAAC: identify tyrosine sulfation sites by incorporating statistical moments via chou's 5-steps rule and pseudo components, Curr. Genom. 20 (4) (2019) 306–320, https://doi.org/10.2174/1389202920666190819091609.
[21] J. He, T. Fang, Z. Zhang, B. Huang, X. Zhu, Y. Xiong, PseUI: pseudouridine sites identification based on RNA sequence information, BMC Bioinf. 19 (1) (2018) 1–11, https://doi.org/10.1186/s12859-018-2321-0.
[22] Z. Lv, J. Zhang, H. Ding, Q. Zou, RF-PseU: a random forest predictor for RNA pseudouridine sites, Front. Bioeng. Biotechnol. 8 (2020), https://doi.org/10.3389/fbioe.2020.00134.
[23] S. Naseer, W. Hussain, Y.D. Khan, N. Rasool, 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. Bioinf (2020), https://doi.org/10.1109/tcbb.2020.3040747, 1–1.
[24] S. Naseer, W. Hussain, Y.D. Khan, N. Rasool, Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations, Anal. Biochem. 615 (2021), https://doi.org/10.1016/j.ab.2020.114069.
[25] W. Hussain, N. Rasool, Y.D. Khan, A sequence-based predictor of zika virus proteins developed by integration of PseAAC and statistical moments, Comb. Chem. High Throughput Screen. 23 (8) (2020) 797–804, https://doi.org/10.2174/1386207323666200428115449.
[26] Y.D. Khan, N. Amin, W. Hussain, N. Rasool, S.A. Khan, K.C. Chou, iProtease-PseAAC(2L): a two-layer predictor for identifying proteases and their types using Chou's 5-step-rule and general PseAAC, Anal. Biochem. 588 (2020), https://doi.org/10.1016/j.ab.2019.113477.
[27] A.H. Butt, Y.D. Khan, CanLect-Pred: a cancer therapeutics tool for prediction of target cancerlectins using experiential annotated proteomic sequences, IEEE Access 8 (2020) 9520–9531, https://doi.org/10.1109/ACCESS.2019.2962002.
[28] S.J. Malebary, Y.D. Khan, Evaluating machine learning methodologies for identification of cancer driver genes, Sci. Rep. 11 (1) (2021), https://doi.org/10.1038/s41598-021-91656-8.
[29] W. Hussain, N. Rasool, Y.D. Khan, Insights into machine learning-based approaches for virtual screening in drug discovery: existing strategies and streamlining through FP-CADD, Curr. Drug Discov. Technol. 17 (2020), https://doi.org/10.2174/1570163817666200806165934.
[30] M.K. Mahmood, A. Ehsan, Y.D. Khan, K.-C. Chou, iHyd-LysSite (EPSV): identifying hydroxylysine sites in protein using statistical formulation by extracting enhanced position and sequence variant feature technique, Curr. Genom. 21 (7) (2020) 536–545, https://doi.org/10.2174/1389202921999200831142629.
[31] Y.D. Khan, F. Ahmed, S.A. Khan, Situation recognition using image moments and recurrent neural networks, Neural Comput. Appl. 24 (7–8) (2014) 1519–1529, https://doi.org/10.1007/s00521-013-1372-4.
[32] I. Transactions, O.N. Pattern, M. Intelligence, 3-D Moment Forms : Their Construction and Application to Object Identification and Positioning, vol. I, 1989, pp. 1053–1064. October.
[33] J. Zhou, H. Shu, H. Zhu, C. Toumoulin, L. Luo, Image analysis by discrete orthogonal Hahn moments (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Lect. Notes Comput. Sci. 3656 (2005) 524–531, https://doi.org/10.1007/11559573_65. LNCS.
[34] P.T. Yap, R. Paramesran, S.H. Ong, Image analysis by Krawtchouk moments, IEEE Trans. Image Process. 12 (11) (2003) 1367–1377, https://doi.org/10.1109/TIP.2003.818019.
[35] Y.D. Khan, E. Alzahrani, W. Alghamdi, M.Z. Ullah, Sequence-based identification of allergen proteins developed by integration of PseAAC and statistical moments via 5-step rule, Curr. Bioinf. 15 (9) (2020) 1046–1055, https://doi.org/10.2174/1574893615999200424085947.
[36] M. Awais, W. Hussain, N. Rasool, Y.D. Khan, iTSP-PseAAC: identifying tumor suppressor proteins by using fully connected neural network and PseAAC, Curr. Bioinf. 16 (5) (2021) 700–709, https://doi.org/10.2174/1574893615666210108094431.
[37] S. Naseer, R.F. Ali, Y.D. Khan, P.D.D. Dominic, iGluK-Deep: computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions, J. Biomol. Struct. Dyn. (2021), https://doi.org/10.1080/07391102.2021.1962738.
[38] Y.D. Khan, N.S. Khan, S. Naseer, A.H. Butt, iSUMOK-PseAAC: prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC, PeerJ 9 (2021), https://doi.org/10.7717/peerj.11581.
[39] S.J. Malebary, Y.D. Khan, Identification of antimicrobial peptides using chou's 5 step rule, Comput. Mater. Continua (CMC) 67 (3) (2021) 2863–2881, https://doi.org/10.32604/cmc.2021.015041.
[40] S.A. Khan, Y.D. Khan, S. Ahmad, K.H. Allehaibi, N-MyristoylG-PseAAC: sequence-based prediction of N-myristoyl Glycine sites in proteins by integration of PseAAC and statistical moments, Lett. Org. Chem. 16 (3) (2018) 226–234, https://doi.org/10.2174/1570178616666181217153958.
[41] D. Che, Q. Liu, K. Rasheed, X. Tao, Decision tree and ensemble learning algorithms with their applications in bioinformatics, Adv. Exp. Med. Biol. 696 (2011) 191–199, https://doi.org/10.1007/978-1-4419-7046-6_19.
[42] F. Huang, G. Xie, R. Xiao, Research on ensemble learning, 2009 Int. Conf. Artif. Intell. Comput. Intell. AICI 3 (2009) 249–252, https://doi.org/10.1109/AICI.2009.235, 2009.
[43] Y. Zhang, J. Ma, S. Liang, X. Li, J. Liu, A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets, GIsci. Rem. Sens. 59 (1) (2022) 234–249, https://doi.org/10.1080/15481603.2021.2023842.
[44] A. Mosavi, F. Sajedi Hosseini, B. Choubin, M. Goodarzi, A.A. Dineva, E. Rafiei Sardooi, Ensemble boosting and bagging based machine learning models for groundwater potential prediction, Water Resour. Manag. 35 (1) (2021) 23–37, https://doi.org/10.1007/s11269-020-02704-3.
[45] K. Mamudur, M.R. Kattamuri, Application of boosting-based ensemble learning method for the prediction of compression index, J. Inst. Eng.: Series A 101 (3) (2020) 409–419, https://doi.org/10.1007/s40030-020-00443-7.
[46] J. Armistead, et al., Mutation of a gene essential for ribosome biogenesis, EMG1, causes Bowen-Conradi syndrome, Am. J. Hum. Genet. 84 (6) (2009) 728–739, https://doi.org/10.1016/j.ajhg.2009.04.017.
[47] P. Gaignard, et al., Mitochondrial infantile liver disease due to trmu gene mutations: three new cases, JIMD Rep. 11 (2013) 117–123, https://doi.org/10.1007/8904_2013_230.
[48] J. Uusimaa, et al., Reversible infantile respiratory chain deficiency is a unique, genetically heterogenous mitochondrial disease, J. Med. Genet. 48 (10) (2011) 660–668, https://doi.org/10.1136/jmg.2011.089995.
[49] K. Shimada, et al., A novel human AlkB homologue, ALKBH8, contributes to human bladder cancer progression, Cancer Res. 69 (7) (2009) 3157–3164, https://doi.org/10.1158/0008-5472.CAN-08-3530.
[50] E.M. Reinthaler, et al., Analysis of ELP4, SRPX2, and interacting genes in typical and atypical rolandic epilepsy, Epilepsia 55 (8) (2014), https://doi.org/10.1111/epi.12712.
[51] S.A. Slaugenhaupt, et al., Tissue-specific expression of a splicing mutation in the IKBKAP gene causes familial dysautonomia, Am. J. Hum. Genet. 68 (3) (2001) 598–605, https://doi.org/10.1086/318810.
[52] M.C. Thrun, T. Gehlert, A. Ultsch, Analyzing the fine structure of distributions, PLoS One 15 (10 October) (2020), https://doi.org/10.1371/journal.pone.0238835.
[53] matplotlib.pyplot.contour. https://matplotlib.org/3.5.0/api/_as_gen/matplotlib.pyplot.contour.html. (Accessed 20 March 2022).
[54] seaborn.scatterplot. https://seaborn.pydata.org/generated/seaborn.scatterplot.html. (Accessed 20 April 2022).