BioData Mining

RESEARCH

Open Access

m1A-Ensem: accurate identification of 1-methyladenosine sites through ensemble models

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

*Correspondence:
falturise@qu.edu.sa

[1] Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore 54770, Pakistan
[2] Department of Computer, College of Science and Arts in Ar Rass, Qassim University, Ar Rass, Qassim, Saudi Arabia

Abstract

Background: 1-methyladenosine (m1A) is a variant of methyladenosine that holds a methyl substituent in the 1st position having a prominent role in RNA stability and human metabolites.

Objective: Traditional approaches, such as mass spectrometry and site-directed mutagenesis, proved to be time-consuming and complicated.

Methodology: The present research focused on the identification of m1A sites within RNA sequences using novel feature development mechanisms. The obtained features were used to train the ensemble models, including blending, boosting, and bagging. Independent testing and k-fold cross validation were then performed on the trained ensemble models.

Results: The proposed model outperformed the preexisting predictors and revealed optimized scores based on major accuracy metrics.

Conclusion: For research purpose, a user-friendly webserver of the proposed model can be accessed through https://taseersuleman-m1a-ensem1.streamlit.app/.

Keywords: Respiratory Disease, Artificial Intelligence, Decision Trees, Statistical Model, Computational Model, Sequence Analysis, Genetics, Nucleotide Sequence, RNA, Computational Biology

Introduction

1-methyadenosine (m1A) sites are reported to be present in transfer RNA (tRNA), messenger RNA (mRNA), and ribosomal RNA (rRNA). In tRNA, these sites occurred in T¥C loop at position 58, as shown in Fig. 1. The identification of m1A sites is significant because of its prominent role in various human diseases such as Mitochondrial respiratory chain defects, Neurodevelopmental regression, X-linked intractable epilepsy, and Obesity [1–3]. Moreover, this PTM modification is actively involved in protein translation, reverse transcription, and reticence in tumors. The m1A site prediction is critical for fully comprehending its potential functions. Site-directed mutagenesis and mass spectrometry have been proposed as methods for detecting

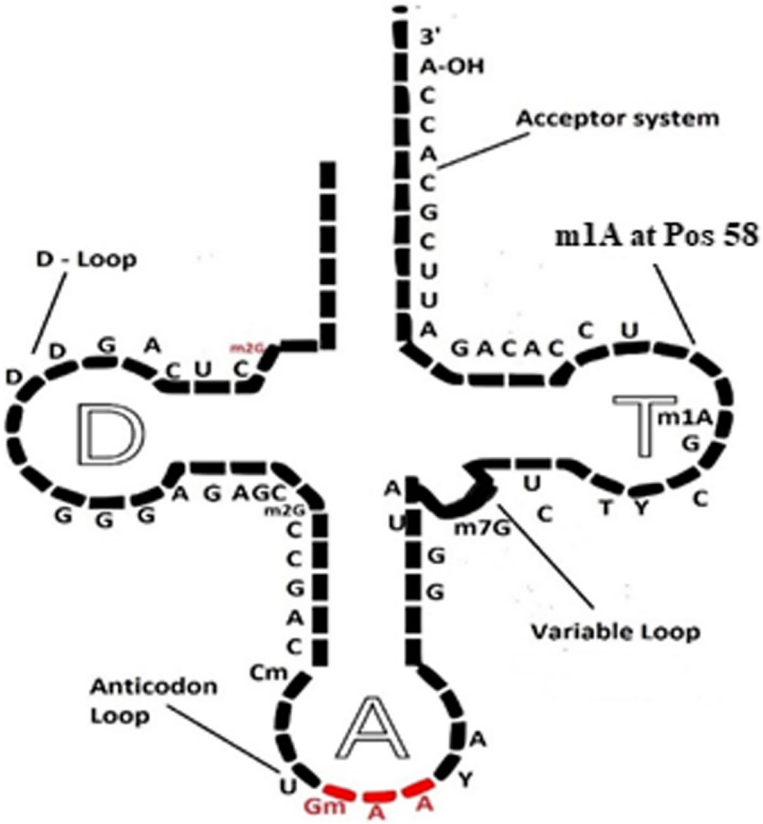

Fig. 1 Position 58 in tRNA loop contains 1-methyladenosine site

m1A sites, although both are complex and time-consuming [4]. The availability of sequence-based datasets has increased the possibility of applying computational intelligence methods for the prediction of PTM sites.

Chen et al. [5] initially developed a predictor, RAMPred, for the identification of m1A sites using *Homosapiens*, *Mus musculus,* and *Saccharomyces cerevisiae* samples. The obtained RNA samples were encoded using nucleotide chemical property (NCP). The obtained features were used to train the support vector machine (SVM) based model. The results revealed 99.13% accuracy (*ACC*), 99.89% specificity (*Sp*), 98.38% sensitivity (*Sn*), and a 0.98 Matthews correlation coefficient (*MCC*). The researchers also developed an online webserver for RAMPred. In another study, Chen et al. [6] developed a predictor, iRNA-3typeA, for the identification of three types of RNA methylation sites, including 6-methyladenosine (m6A), m1A, and adenosine-to-inosine (A-to-I). The same data samples of *Homosapiens* and *Mus Musculus* were used previously in RAMPred. The results revealed an accuracy score of 99.13% in *Homosapiens* and 98.73% in *Mus musculus* species. A 41 nucleotides lengthy sample was used, and cross validation test was carried out for performance evaluation. In another study Liu et al. [7] suggested a prediction model, ISGm1A, that extract 75 genomic-based features from the RNA sequences. Five machine learning models were trained and validated through independent testing and cross validation. Sun et al. [8] developed

a deep learning framework, DeepMRMP, based on bidirectional gated recurrent unit (BGRU) for the identification of multiple RNA post transcriptional modified (PTM) sites in *Homosapiens*, *Mus Musculus* and *Saccharomyces Cerevisiae* species. One-hot encoding was used to encode the nucleotides within a sequence i.e. A = [1,0,0,0], C = [0,1,0,0], G = [0,0,1,0], U = [0,0,0,1]. The model revealed 70.5% *ACC,* 0.85 *Sn*, 0.95 *Sp* and 0.83 *mcc*.

Previous research studies dealt with the identification of m1A sites through traditional machine learning algorithms. However, such models are subjected to imbalanced data issue, overfitting and underfitting problems, and having limited context understanding. The current study proposed a novel framework for the prediction of m1A sites using ensemble models. These models were categorized into blending, bagging, and boosting which provides more rigorous training on dataset. It's worth mentioning here that RAMPred, iRNA3typeA, ISGm1A, and DeepMRMP have used the same dataset for training and validation. The dataset is composed of RNA sequences belonging to four species: *Homosapiens, Saccharomyces cerevisiae*, *Mus musculus* and *Schizosaccharomyces pombe*. 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 in few metrics developed for attributes extraction. The performance of these ensemble models was evaluated through k-fold cross validation and independent set testing. The accuracy metrics such as *ACC*, *Sp*, *Sn,* and *MCC* were used to evaluate the ensemble models quantitatively. The results revealed that the proposed model outperformed in all accuracy metrics comparable to the preexisting m1A sites predictors. 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 m1A sites detection. A methodology framework has been depicted in Fig. 2.

Materials and methods

Dataset collection

The dataset acquired from RMBase v2.0 [9] containing RNA samples from four species, including *Homosapiens*, *Saccharomyces cerevisiae*, *Mus musculus*, and *Schizosaccharomyces pombe* designated as *HS_17880*, *SC_3406*, *MM_4232* and *SP_958*. The

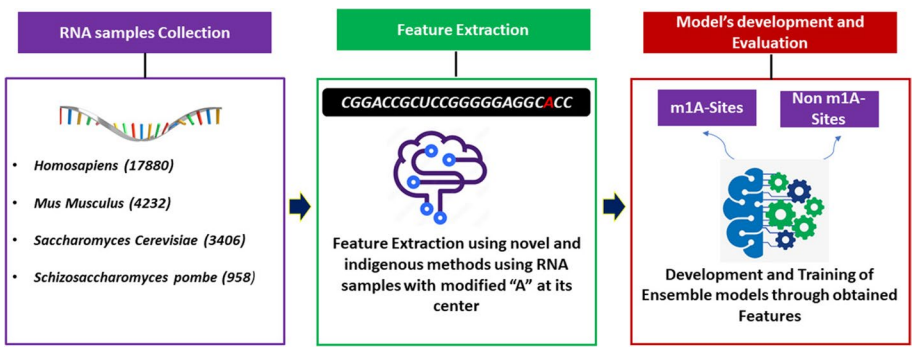

Fig. 2 Current research methodology

Table 1 Details of RNA samples used in this study

Dataset	M1A_sites (positive)	Non-m1A sites (Negative)
HS_17880	8940	8940
SC_3406	1703	1703
MM_4232	2116	2116
SP_958	479	479

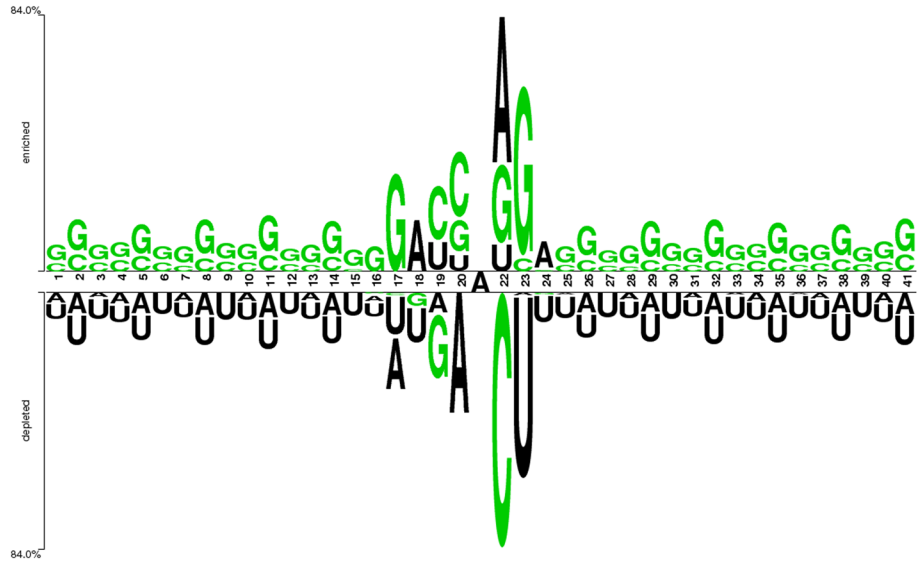

Fig. 3 Two sample logos of the data samples representing nucleotide distributions

dataset details have been mentioned in Table 1. After CD-Hit at 80%, the positive samples obtained were 11,978 and the negative samples obtained were 12,716. The cutoff was selected at 80% because of large number of samples. There might be a possibility of homology existing within samples. The window size for each RNA sample was chosen at 41 since this yielded the best overall performance. The window size was selected due the availability of 41nt verified samples and the optimized results revealed by this specific length. The m1A site-expressing RNA sample described in [1].

$$B(A) = B_{-\tau}B_{-(\tau-1)}....B_{-2}B_{-1}AB_{+1}B_{+2}....B_{+(\tau-1)}B_{+\tau} \quad (1)$$

whereas "*A*" represents modified adenine of RNA sequences with methylated m1A sites.

The arrangement of nucleotide bases within the acquired sequences can be visualized using a sequence logo. To achieve this, an online tool known as the "Two Sample Logo" was utilized. Figure 3 displays the sequence logo, which effectively represents the presence of cytosine (C), guanine (G), adenine (A), and uracil (U) within the dataset.

The nucleotides sample logo illustrates the concentration of "U" and "A" nucleotides throughout the sequence. However, the central position at "21" includes the "A".

Moreover, the nucleotide "G" is symmetrically distributed along the whole samples. It can be observed that "C" is only located from position 19 to 23 within nucleotide sequence.

Feature extraction and development phase

The most important phase of computational procedures is feature extraction. During this stage, features are extracted to emphasize the dataset's unique characteristics [10]. Due to recent advances in information and data sciences, biotechnology has made major strides forward. Yet, the most difficult aspect is the development of computationally sophisticated models that transform raw biological input into counted, quantified vectors. Moreover, the loss of a single sequence or its associated properties must be prevented. This is due to the fact that all inputs to machine learning algorithms are vectors. The current research adopted a novel feature extraction method which includes various matrices and vectors for attaining the useful attributes from the sequences. These specialized vectors and matrices were indigenously developed for extracting divulged as well as concealed features within the sequences. This would be helping in developing more robust computational models that would assist in identification of m1A sites in an optimized way. To prevent the complete loss of the sequence-pattern information, Chou developed a pseudo-amino acid composition for proteins (PseAAC) [11]. Then pseudo-K-tuple nucleotide composition (PseKNC) was formulated as a result of the PseAAC success [12, 13]. Additionally, an RNA sequence, X, can be illustrated, as shown in [2].

$$X = X_1, X_2, X_3, \ldots, X_i, \ldots, X_n \tag{2}$$

whereas,

$$X_i \epsilon \{C\left(cytosine\right), A(Adenine), G\left(guanine\right), U(uracil)\}$$

represents a nitrogenous base at a random position within an RNA sample. The genomic data used in this study was transformed into a matrix, $f\prime$, as shown in [3].

$$f\prime = [f_1 f_2 f_3 f_4 \ldots f_u \ldots f_\Omega]^\mathsf{T} \tag{3}$$

A single feature, f_u, depicts an arbitrary numerical coefficient which characterize a single feature. The transpose was taken for yielding discrete coefficients.

Statistical moments calculation

A fixed-length feature vector was computed from the genomic data using statistical moments [14]. Statistical moments are essential tools in statistics and probability theory that provide valuable information about the distribution of data. They are used to describe the shape, central tendency, spread, and other characteristics of a dataset. The significance of statistical moments lies in their ability to summarize and quantify various aspects of data distributions, making them useful in a wide range of applications, including data analysis, modeling, and decision-making. Moments of various distributions have been studied by analysts and mathematicians [15]. By computing the central, Hahn, and raw moments, a compact feature set was generated, which was then utilized to reduce the colossal input vector. Therefore, moments were computed

for dimensionality reduction. The feature set was expanded to incorporate the scale and area of important moments to help differentiate between functionally distinct sequences. According to scientific investigations, genomic and proteomic sequence-based characteristics alter with the content and relative location of their bases [16]. Hence, the feature vector is best generated using mathematical and computational models that are sensitive to the relative location of component bases within genomic sequences. The features were transformed into compact coefficients that accurately reflect the data's mean and standard deviation using raw, central, and Hahn moments. While attempting to decipher a sequence, scale and position variations like the Raw and Hahn moments are preferable. $Atwo-dimensionalmatrix$, $Ʉ'$, was built from the sequences, with each entry, $Ʉ_{mn}$, representing the n_{th} nucleotide base in the, m_{th}, sequence as expressed in [4].

$$Ʉ' = \begin{bmatrix} Ʉ_{11} & Ʉ_{12} & \dots & Ʉ_{1j} \\ Ʉ_{21} & Ʉ_{22} & \dots & Ʉ_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ Ʉ_{i1} & Ʉ_{i2} & \dots & Ʉ_{ij} \end{bmatrix} \tag{4}$$

Raw moments are used to derive location variant characteristics from extracted features [17]. Raw moments are described in [5], where the total number of raw moments is denoted by the value of u + v. The coefficients $ℕ_{00}$, $ℕ_{01}$, $ℕ_{10}$, $ℕ_{11}$, $ℕ_{12}$, $ℕ_{21}$, $ℕ_{30}$, and $ℕ_{03}$ were determined up to the third-degree polynomial [18, 19].

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

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 [20]. Moreover, the central moments are associated with the nucleotides' composition and distribution. For the current study, the central moments were computed and expressed in [6] as follows.

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

Orthogonal moments are often preferred because they can represent data with the least amount of redundant information. Yet, even if the original sequences have been drastically shortened to a fixed length, the predictor still gets the effect of the whole sequence of data within the reduced feature vector due to the reversible quality of these moments. Hahn polynomials can be written as follows:

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

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

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

Position Relative Incidence Matrix (PRIM)

The position relative incidence matrix (PRIM) was used to represent the relative positioning of nucleotide bases within an RNA sample [21]. The matrix, $\boldsymbol{E_{PRIM}}$ [9], is a 4X4 matrix that represents any single nucleotide, $V_{\boldsymbol{m}}$, at position "$\boldsymbol{m}$", with respect to other nucleotides within a sequence. The matrix generated 16 unique coefficients.

$$E_{PRIM} = \begin{bmatrix} V_{A\to A} & V_{A\to G} & V_{A\to U} & V_{A\to C} \\ V_{G\to A} & V_{G\to G} & V_{G\to U} & V_{G\to C} \\ V_{U\to A} & V_{U\to G} & V_{U\to U} & V_{U\to C} \\ V_{C\to A} & V_{C\to G} & V_{C\to U} & V_{C\to C} \end{bmatrix} \tag{9}$$

where, $V_{i\to j}$, represents the relative positioning of an arbitrary nucleotide base with respect to any other random 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 16X16 matrix known as $\check{U}_{PRIM}$ [10], which results in 256 coefficients, was used to consider the frequency with which these base pairings occur in comparison to one another.

$$\check{U}_{PRIM} = \begin{bmatrix} \check{U}_{AA\to AA} & \check{U}_{AA\to AG} & \check{U}_{AA\to AU} & \cdots & \check{U}_{AA\to j} & \cdots & \check{U}_{AA\to CC} \\ \check{U}_{AG\to AA} & \check{U}_{AG\to AG} & \check{U}_{AG\to AU} & \cdots & \check{U}_{AG\to j} & \cdots & \check{U}_{AG\to CC} \\ \check{U}_{AU\to AA} & \check{U}_{AU\to AG} & \check{U}_{AU\to AU} & \cdots & \check{U}_{AU\to j} & \cdots & \check{U}_{AU\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \check{U}_{GU\to AA} & \check{U}_{GA\to AG} & \check{U}_{GU\to AU} & \cdots & \check{U}_{GA\to j} & \cdots & \check{U}_{GA\to CC} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \check{U}_{N\to AA} & \check{U}_{N\to AG} & \check{U}_{N\to AU} & \cdots & \check{U}_{N\to j} & \cdots & \check{U}_{N\to CC} \end{bmatrix} \tag{10}$$

Similarly, another matrix, $Ł_{PRIM}$ [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 E_{PRIM}, $\check{U}_{PRIM}$ and $Ł_{PRIM}$, that resulted in forming coefficients up to order 3.

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

Reverse Position Relative Incidence Matrix (RPRIM)

The primary objective of determining feature vectors is to collect as much relevant information as possible to develop an accurate prediction model. Reversing the sequence order yielded a reverse position relative indices matrix (RPRIM) in an effort to extract

more information contained within the sequences [22]. Similarly with PRIM matrices, RPRIM was calculated using mononucleotide, dinucleotide, and trinucleotide combinations. For this reason, R_{RPRIM} was computed according to [12].

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

Frequency vector determination

The sequence's positional and compositional information is crucial in developing a feature set [23, 24]. The composition of the sequence can be determined by counting the frequency of each nucleotide. A frequency vector (δ) is used to store the count for each nucleotide or nucleotide pair in the sequence, and the method for calculating this vector has been described in [13].

$$\delta = \{\eta_1, \eta_2, \ldots\ldots, \eta_n\} \tag{13}$$

where, η_i, is the count of the i_{th} nucleotide in a sequence.

Generation of Accumulative Absolute Position Incidence Vector (AAPIV)

The AAPIV (accumulated information of individual nucleotide bases) is a method used to provide information on the frequency of each individual nucleotide base in a sequence [25]. This method is responsible for collecting and accumulating data related to the occurrence of each nucleotide base, including single and paired nucleotide bases [26, 27]. To achieve this, three different AAPIV vectors were generated, each representing a different level of granularity. These vectors were given the names S_{AAPIV4} [14], $S_{AAPIV16}$ [15] and $S_{AAPIV64}$ [16]. Each vector represents a different level of granularity, with S_{AAPIV4} containing information on four nucleotides, $S_{AAPIV16}$ containing information on sixteen nucleotides, and $S_{AAPIV64}$ containing information on sixty-four nucleotides. These vectors provide a useful tool for analyzing the composition of nucleotide sequences and can be used in a variety of biological applications.

$$S_{AAPIV4} = \{b_1, b_2, b_3, b_4,\} \tag{14}$$

$$S_{AAPIV16} = \{b_1, b_2, b_3, \ldots, b_{15}, b_{16},\} \tag{15}$$

$$S_{AAPIV64} = \{p_1, p_2, p_3, \ldots, p_{63}, p_{64}\} \tag{16}$$

where, p_{i} can be calculated as provided in [17].

$$\delta_i = \sum_{k=1}^{n} \mathrm{p}_k \tag{17}$$

Reverse Accumulative Absolute Position Incidence Vector (RAAPIV) Generation

To analyze the reversed sequences, a reverse accumulative absolute position incidence vector (RAAPIV) had been devised in the research. Specifically, it involves reversing the order of the nucleotide sequences in order to gain a different perspective on the underlying data. There are three types of nucleotide combinations that were examined using the RAAPIV: single nucleotide combinations, di-nucleotide combinations, and tri-nucleotide combinations. The vector length for each of these combinations differs, with a length of 4 for single nucleotides, 16 for di-nucleotides, and 64 for tri-nucleotides. The expression (18), (19) and (20) referred to the combination of single nucleotide, dinucleotides and trinucleotides respectively. Overall, this technique provides a way to gain new insights into genetic sequences by analyzing them from a different perspective.

$$J_{RAAPIV4} = \{j_1, j_2, j_3, j_4\} \tag{18}$$

$$J_{RAAPIV16} = \{j_1, j_2, j_3, \ldots, j_{16}\} \tag{19}$$

$$J_{RAAPIV64} = \{j_1, j_2, j_3, \ldots, j_{64}\} \tag{20}$$

Feature vector formulation

The outcome of the feature extraction operation was the creation of a single feature vector. This feature vector was then utilized as a prediction model input with 522 distinct values collected by PRIM, RPRIM, FV, AAPIV, and RAAPIV. Each feature vector in the dataset represents an individual sample. For binary classification, positive samples were labelled as "1" and negative samples as "0″ [28, 29]. Table 2 contains the detail of the number of features obtained from each vector or matrix individually.

Table 2 Number of features obtained from each vector and matrix

Vector/Matrix	Features obtained (Dimensions)
PRIM (E_{PRIM}, $\check{U}_{PRIM}$, $Ł_{PRIM}$)	90
RPRIM (R_{RPRIM})	90
Frequency Vector	84
AAPIV (S_{AAPIV4}, $S_{AAPIV16}$, $S_{AAPIV64}$)	84
RAAPIV ($J_{RAAPIV4}$, $J_{RAAPIV16}$, $J_{RAAPIV64}$)	84
two-dimensional matrix, ʜ′	90
Total	**522**

Ensemble models development and training

Ensemble methods have gained popularity in the field of machine learning due to their enhanced prediction capabilities as compared to conventional single-model approaches [30, 31]. These methods combine the strengths of multiple models to achieve better overall performance, and they can be broadly classified into parallel and sequential methods. To address real world challenges, ensemble models help in building trust, model aggregation, prediction on different patterns based on diverse classifiers and features-based analysis. Parallel ensemble methods, such as bootstrap aggregation (or bagging), involve training multiple models concurrently on different subsets of the data. Sequential ensemble methods, on the other hand, involve training models sequentially, with each subsequent model learning from the errors of the previous one. Ensemble-based classification has been reported in various research studies. Akbar et al. [20] devised a novel method for the identification of anticancer peptides based on the genetic algorithms-based ensemble models which achieved optimized accuracy scores. Moreover, in another research study, authors devised an ensemble-based model for the identification of antitubercular peptides and the accuracy scores reported to be more than 90% [32]. Ahmed et al. [33] proposed, iAFPs-EnC-GA, an ensemble learning based model for the identification antifungal peptides. In the context of the investigation mentioned, three distinct ensemble models were applied including blending, bagging, and boosting.

Blending ensemble

Blending is an ensemble technique that combines the outputs of multiple classification or regression models using a meta-classifier or meta-regressor [34, 35]. In this approach, the base-level models are first trained, and their outputs are then used as features for the meta-model. This meta-model leverages the knowledge of the base models to make more accurate and robust predictions. The current investigation employed four base models, including an artificial neural network (ANN), a k-nearest neighbor (KNN), a support vector machine (SVM), and a decision tree (DT). The gradient boost classifier was chosen as the meta-classifier to combine the outputs of these base models. Hyperparameter optimization is an essential step in machine learning, as it ensures that each model performs at its best. Table 3 presents the details of the hyperparameter optimization process for all the classifiers used in the blending ensemble deployment.

Table 3 Parameters tuning of the blending ensemble 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'			

Bagging ensemble

The bagging ensemble methods in the research deployed in such a way that the trained samples were divided into smaller subsamples for the base models using a subsampling approach with replacement and row sampling. This strategy ensures that each base model is trained on a different subset of the data, promoting diversity among the individual models and reducing the overall variance of the ensemble [36].

The test data were evaluated using the trained base models, and the final forecast was obtained through a voting mechanism, which typically involves majority voting for classification tasks or averaging for regression tasks. Four bagging models, namely the bagging classifier, random forest, extra tree, and decision tree classifier, were developed and trained as part of the investigation. For improved results, all the bagging classifiers were subjected to hyperparameters optimization. The hyperparameters such as 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* were considered. Table 4 contains the hyper-parameter optimization information of the aforementioned bagging models.

Boosting ensemble

The boosting ensemble approach is designed to optimize the model based on the output of the preceding model in the sequence. It operates sequentially, with each model focusing on reducing the differentiable loss by learning from the errors of the previous model. This process helps boost the overall performance of the ensemble by combining the strengths of multiple weak learners. In the current investigation, several boosting ensemble training approaches were employed, including gradient boosting, histogram-based gradient boosting (HGB), AdaBoost, and extreme gradient boosting (XGB). To optimize the performance of the boosting ensemble models, various hyperparameters were fine-tuned, as shown in Table 5. Figure 4 depicts the concept diagram of ensemble

Table 4 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

Table 5 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

model implementation for the current research study, which includes blending, boosting, and bagging.

Results and discussion

The trained models were subjected to validation using independent set testing and tenfold cross validation. The independent test was carried out using the standard "Train-Test" split method. However, tenfold cross validation is a rigorous test that divides the whole dataset into subsamples, where one sample is subjected to testing while the other nine are used for training. Different accuracy metrics were used to score the performance of all ensemble models, including *ACC*, *Sp*, *Sn*, and *MCC*.

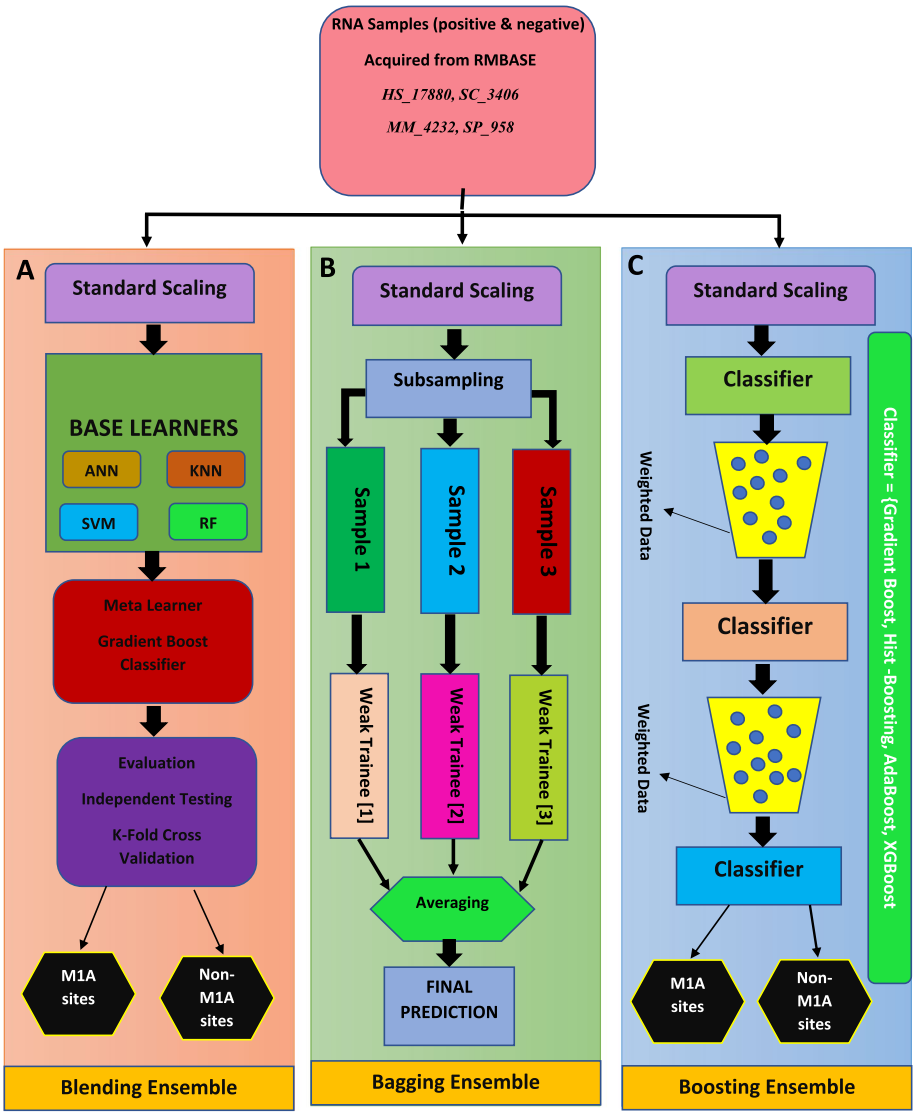

Fig. 4 Ensemble models Development and Training/Testing for the Current research study using RNA samples from RMBase (**A**). Blending Ensemble (**B**). Bagging Ensemble (**C**). Boosting Ensemble

Metrics for evaluation

In this research, four metrics, S_n, S_p, Acc, and MCC were used to evaluate the prediction models [37, 38]. The effectiveness of a categorization model may be measured in terms of its Acc. The Acc rate is the ratio of the model's correct predictions to the total number of forecasts. It is the fraction of the dataset that was properly predicted relative to the total number of occurrences. Whereas Specificity (S_p) is a metric used to evaluate the performance of a binary classification model, particularly in cases where the negative class is of greater importance. It measures the proportion of true negatives (TN) that are correctly identified by the model out of all negative instances. Sensitivity (S_n) is a metric used to evaluate the performance of a binary classification model, particularly in cases where the positive class is of greater importance. It measures the proportion of true positives (TP) that are correctly identified by the model out of all positive instances. Matthews Correlation Coefficient (MCC) is a metric used to evaluate the performance of a binary classification model, particularly when the classes are imbalanced. MCC takes into account the number of true and false positives and negatives to give a balanced measure of the model's performance. The accuracy metrics equations have been mentioned in [22].

$$\begin{cases} S_n = \frac{\mathrm{TP}}{\mathrm{TP+FN}} 0 \leq S_n \leq 1 \\ S_p = \frac{\mathrm{TN}}{\mathrm{TN+FP}} 0 \leq S_p \leq 1 \\ Acc = TP + TN/(TP + FP + FN + TN) 0 \leq Acc \leq 1 \\ MCC = (\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN})/\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})} - 1 \leq MCC \leq 1 \end{cases} \tag{21}$$

The TP denotes the m1A sites, whereas the TN denotes the non-m1A sites. A similar notation, FN, represents the total number of modified sites that were indeed actual m1A sites but were misidentified as false m1A sites. Furthermore, FP stands for the total number of false m1A sites that were misidentified. However, it's important to note that the measurements only apply to systems with a single class [39]. The false positive and false negative value have crucial roles in the performance evaluation of the system. A wrong detection of false positive leads to the wrong m1A site detection within a given RNA sample. Similarly, the increase in false negatives may result into the increase in non-m1A sites abnormally.

Data preprocessing

The obtained feature set was subjected to data preprocessing by using standard scaling of sklearn preprocessing [40]. All the missing values were removed using standard scaling before input to the machine learning model.

Independent set testing

An Independent test set was carried out to validate all the ensemble models, including blending, bagging, and boosting. The independent set was created using the standard "train-test split" method with a 70% training and 30% testing dataset [41, 42]. There were 8385 positive and 8901 negative train samples. The test samples were 3593 positives and 3814 negatives. It is important to mention that training and test samples were

separate frofutm each other. Table 6 contains the results revealed by all the ensemble models deployed for the current research. Whereas Fig. 5 depicts the area under curve (AUROC) of the ensemble model in independent testing.

10-Fold cross validation

The cross-validation approach is used to test all the samples while splitting the dataset into "k" disjoint folds [43, 44]. 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 [45]. 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 7.

Several statistical tests were conducted to verify the effectiveness of the ensemble models implemented in this study. The primary goal of these tests was to compare the performance of various learning algorithms in achieving accurate classification outcomes. One of the tests conducted was a two-proportion test, commonly referred to as the Z test, on the ensemble models. This Z test was utilized to assess whether there existed a significant distinction between the two sets of samples. To establish such a distinction, the critical value (p) needed to be below 0.05, indicating the rejection of the null hypothesis. Furthermore, a resampled paired t-test was employed, using a predetermined set of trials, to measure the accuracy of the algorithms. McNemar's test, another statistical test, was applied to evaluate the significance of the difference between two

Table 6 Independent testing result

Model		*Acc*	S_p	S_n	*MCC*	*F1-score*	*AUROC*
Bagging	*Random Forest*	0.88	0.93	0.85	0.77	0.87	0.95
	Extra Tree Classifier	0.81	0.87	0.75	0.63	0.80	0.89
	Decision Tree	0.87	0.87	0.87	0.74	0.86	0.87
	Bagging classifier	0.92	0.97	0.86	0.84	0.91	0.97
Boosting	*Gradient Boost*	0.93	0.97	0.89	0.87	0.93	0.98
	HGB	0.99	0.98	0.97	0.98	0.97	0.99
	AdaBoost	0.94	0.97	0.92	0.89	0.94	0.98
	XGBoost	0.93	0.97	0.93	0.87	0.93	0.98
Blending		0.91	0.86	0.94	0.81	0.90	0.96

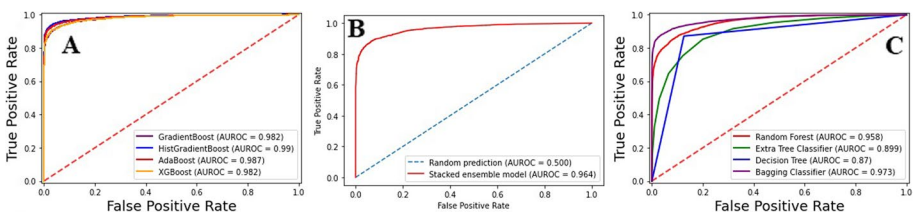

Fig. 5 ROC curve of independent testing (**A**) Boosting Ensemble (**B**) Blending Ensemble (**C**) Bagging Ensemble

Table 7 10-Fold cross validation results

Model		*Acc*	S_p	S_n	*MCC*	*F1-score*	*AUROC*
Bagging	*Random Forest*	0.82	0.85	0.81	0.77	0.80	0.86
	Extra Tree Classifier	0.91	0.92	0.93	0.89	0.83	0.94
	Decision Tree	0.87	0.87	0.87	0.74	0.86	0.87
	Bagging classifier	0.94	0.98	0.88	0.86	0.93	0.98
Boosting	*Gradient Boost*	0.90	0.94	0.87	0.85	0.92	0.96
	HGB	0.98	0.97	0.95	0.97	0.96	0.98
	AdaBoost	0.99	0.98	0.96	0.97	0.96	0.99
	XGBoost	0.92	0.96	0.92	0.86	0.92	0.97
Blending		0.91	0.93	0.85	0.92	0.82	0.93

proportions in a 2×2 contingency table. The resulting "p" values from these tests are listed in Table 8.

The violin plot is a graphical representation that combines elements of a box plot and a kernel density plot to display the distribution of numerical data for one or more groups [46]. It uses density curves to illustrate the probability density of the data at different values, giving a clear visualization of the data distribution, including its central tendency, dispersion, and shape. Key elements of a violin plot include (1) a central white dot representing the median of the data, which indicates the middle value when the data is sorted in ascending order. (2) A black bar in the middle of the violin, showing the interquartile range (IQR), which represents the spread of the middle 50% of the data. () Dark black lines extending from the black bar to the lower and higher neighboring values, indicating the range of the data within 1.5 times the IQR from the lower and upper quartiles. Figure 6 displays the violin plots for the accuracy values obtained in each fold for the best ensemble models in the blending, bagging, and boosting categories.

The application of supervised machine learning models can prove beneficial in various categorization tasks. Nonetheless, relying solely on numerical predictions might not be enough. Gaining a comprehensive understanding of the actual decision boundary that delineates the different groups is crucial. Consequently, the classification algorithms employed in this research were examined using a decision surface to enhance their accuracy. A decision surface map is a visual representation where a trained machine learning system predicts a coarse grid covering the input feature space. This method allows for a

Table 8 Statistical test results of blending, boosting and bagging ensemble models

Model		*Z-test*	*Resampled paired t-test*	*McNemar's test*
Bagging	*Random Forest*	0.00156	0.00089	0.0017
	Extra Tree Classifier	0.00162	0.00052	0.0019
	Decision Tree	0.00137	0.00059	0.0033
	Bagging classifier	0.00130	0.00090	0.0087
Boosting	*Gradient Boost*	0.00144	0.00090	0.0080
	HGB	0.00170	0.00075	0.0069
	AdaBoost	0.00149	0.00055	0.0049
	XGBoost	0.00129	0.00034	0.0038
Blending		0.00159	0.00045	0.0029

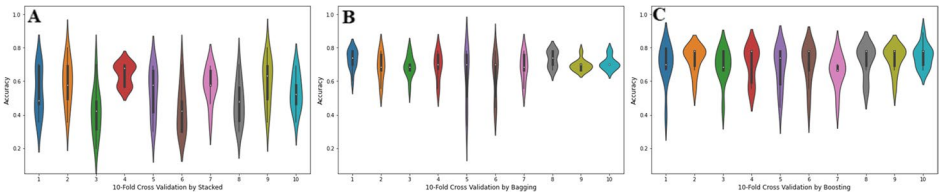

Fig. 6 Violin plots of 10-Fold cross validation accuracy (Acc) metric results for (**A**) Blending ensemble (**B**) Bagging ensemble and (**C**) Boosting ensemble

better understanding of the model's decision-making process by illustrating the regions in which the model assigns a particular class to input data points. Figure 7 displays the decision surface plots of the classification algorithms used in this research. By examining these plots, one can gain insights into how the algorithms differentiate between the various classes and the effectiveness of their decision-making process. This information can be valuable for refining the models, improving their accuracy, and ensuring more reliable outcomes in categorization tasks.

Comparison with preexisting predictors

The proposed model was built on the best performing HGB ensemble model and compared with preexisting predictors to assess the model's efficacy on the independent datasets. The predictors were RamPred, Deepmrmp, irna3typeA, and ISGm1A. It was observed that the proposed model, m1A-Ensem, outperformed exhibiting 0.99 *ACC* , 0.98 *Sp*, 0.97 *Sn,* and 0.98 *MCC*. The comparative results have been mentioned in Table 9. The use of vectors and matrices helped in extracting obscured features within the sequences. Moreover, the hyperparameter optimization of ensemble models helped in gaining promising accuracy scores. The identification of m1A sites is vital as this RNA modification has been implicated in various diseases such as Mitochondrial respiratory chain defects, Neurodevelopmental regression, X-linked intractable epilepsy, and Obesity. Moreover, m1A sites help in gene regulation procedures such as gene splicing, RNA stability and regulatory mechanisms. This modification is also involved in RNA folding and structure stability. Detecting these sites accurately is a critical step towards understanding the mechanisms behind these diseases and developing effective biomarkers for drug discovery. To address this issue, researchers have developed a comprehensive

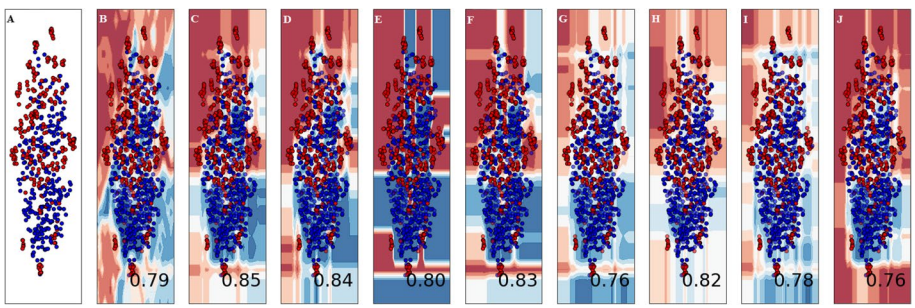

Fig. 7 Boundary visualization of ensemble models used in this study as follows: (**A**). Input data **(B)**. Blending (**C**). Random Forest (**D**) ExtraTree (**E**) Decision Tree (**F**) Bagging (**G**) Gradient Boost (**H**) Histo Gradient Boost (**I**) Adaboost (**J**) XGBoost

Table 9 Comparison with preexisting predictors

Model	Independent set test			
	Acc (%)	S_p	S_n	MCC
RAMPred	98.73	0.99	0.95	0.96
irna-3typeA	84.6	0.93	0.88	0.91
Deepmrmp	70.5	0.95	0.85	0.83
ISGm1A	83.5	0.83	0.83	0.67
m1A-Ensem	99.9	0.98	0.97	0.98

strategy that involves feature development and representation, merging multiple computational models, and testing the model using a variety of methodologies. This approach has resulted in the creation of a predictive model that outperforms existing models in identifying m1A sites. Extensive trials have shown that the proposed model has a high degree of precision, resilience, and scalability. Its accuracy in identifying modified m1A sites has been demonstrated through various testing methodologies, indicating its potential usefulness in research. Overall, the development of this predictive model represents a significant advancement in the field of RNA modification research, providing a valuable tool for researchers and clinicians in their efforts to better understand and treat diseases associated with m1A sites.

Limitations and future work

The limitation of the current work is the availability of RNA samples from a few species only. The number of available samples also limits the possibility of training computational models. Moreover, the discovery of new m1A sites related to samples will require the development of new models and training of those models on latest data samples. This will be affecting the results obviously. Moreover, the scope of the study is limited to the development of ensemble models for the identification of m1A sites. The prediction of m1A sites through deep learning models using the available data samples can be attempted in the future.

Web server 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. The m1A-Ensem, a free online web server for the suggested model, was created with this objective in mind and is accessible at https://taseersuleman-m1a-ensem1.streamlit.app/. It has four tabs including "*Home*", "*Predictor*", "*Dataset*" and "*Citations*". The "*Home*" tab contains the m1A prediction model description. Figure 8 represents the screenshot of the webserver for the proposed model. The "*Predictor*" tab contains the sample sequence and input area. A user can input any length of sequence in the Input area. Figure 9 shows the "*Predictor*" tab with "Example" sequence button and Input area. The user has to click "submit" button and the result generated for each *Adenosine* (A) site as it is m1A site or non-m1A site. Figure 10 represent a sequence showing their actual position within the sequence and their status (m1A site of non-m1A site).

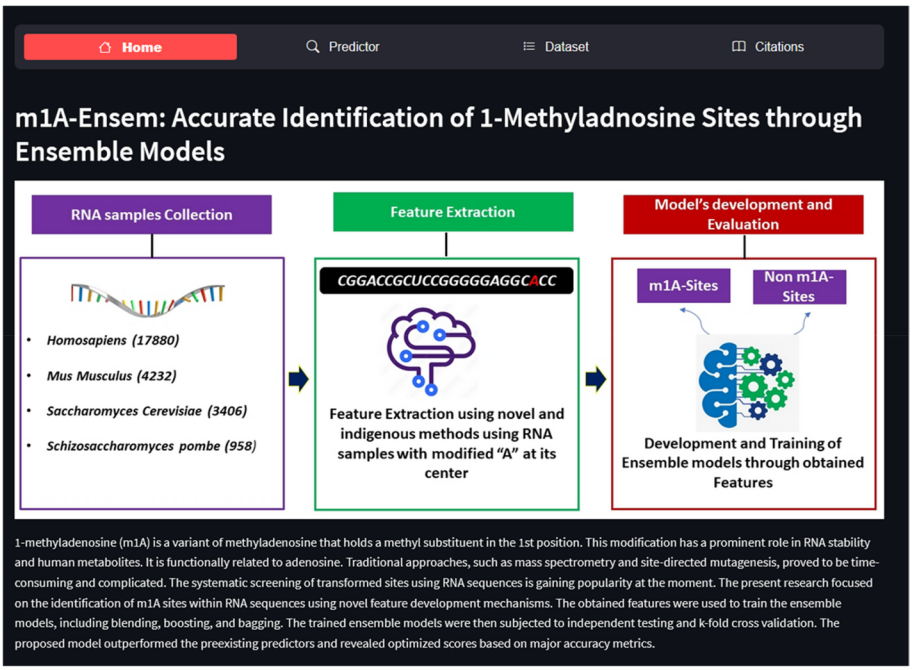

Fig. 8 Screenshot of m1A-Ensem prediction webserver

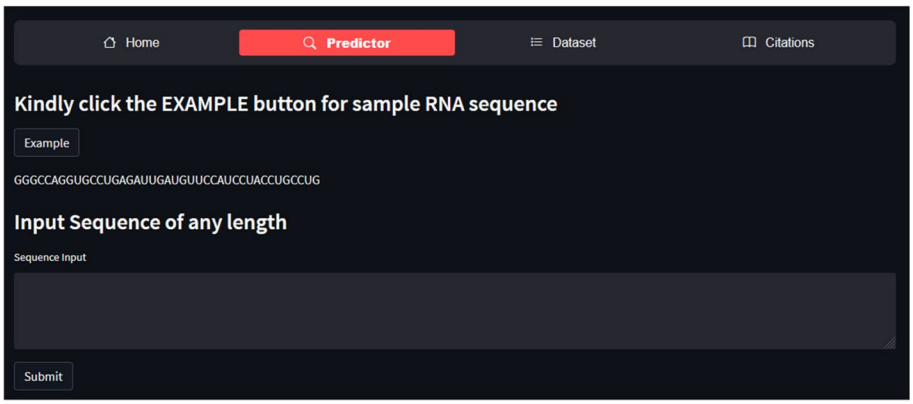

Fig. 9 Image showing webserver "Predictor" page with "Example" sequence

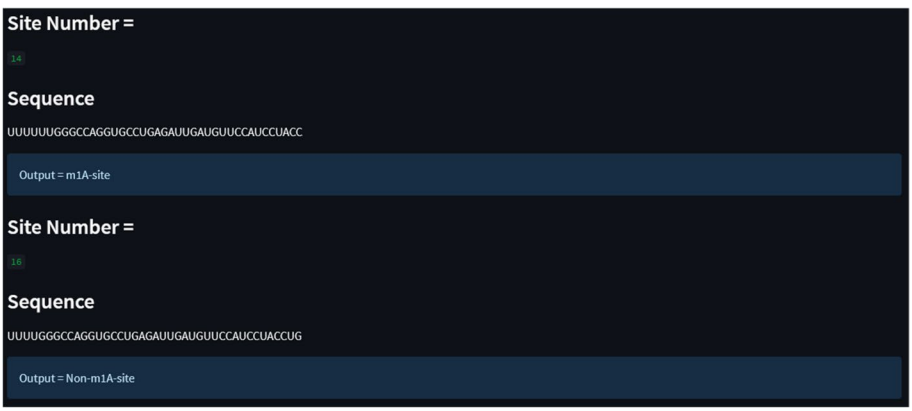

Fig. 10 Webserver identifying m1A and non-m1A sites within RNA sample

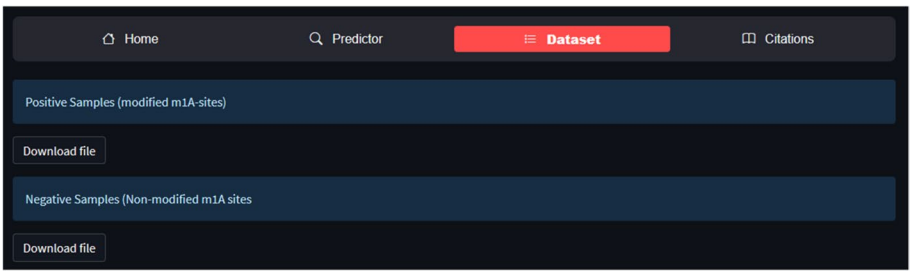

Fig. 11 "Dataset" Tab representing the positive and negative samples

Similarly, the "*Dataset*" tab contains the dataset samples used for training and testing the models. Figure 11 depicts the "*Dataset*" image.

Conclusion

This study focused on detecting one of the most common post-transcriptional modifications, 1-methyladenosine (m1A), in RNA sequences using ensemble methods. Identifying m1A sites is crucial as this modification is associated with various human diseases, including mitochondrial respiratory chain defects, neurodevelopmental regression, X-linked intractable epilepsy, and obesity. A novel feature extraction mechanism was developed, taking into account both the positional and compositional attributes of nucleotides within RNA sequences. By calculating statistical moments, feature dimensionality reduction was achieved, streamlining the analysis. The resulting feature set was used to train several ensemble models based on stacking, bagging, and boosting techniques. The trained models underwent evaluation through cross-validation and independent testing. Performance was assessed using well-known accuracy metrics such as accuracy, sensitivity, specificity, and Matthew's correlation coefficient. Based on the best-performing ensemble model, the proposed model, m1a-ensem, was constructed. A comparative analysis of m1A-Ensem was conducted against existing predictors to gauge its effectiveness. The results demonstrated that m1A-Ensem outperformed other predictors in all accuracy metrics. Consequently, it can be concluded that the proposed model successfully enhanced the ability to identify modified m1A sites by employing the techniques described above. In summary, the research developed a novel approach to detect m1A sites in RNA sequences, which has implications for understanding and potentially treating various human diseases. By incorporating ensemble methods and a unique feature extraction mechanism, the m1A-Ensem model demonstrated superior performance in comparison to existing predictors, highlighting its potential for further applications in this field.

Acknowledgements
Researchers would like to thank the Deanship of Scientific Research, Qassim University for funding publication of this project.

Authors' contributions
The research investigation and original draft was prepared by Muhammad Taseer Suleman. Fahad Alturise worked on research methodology, investigation and reviewing the final draft. Tamim Alkhalifah validated the results and reviewed the final draft. Yaser Daanial Khan supervised the research and reviewed the final draft.

Funding
This research received no external funding.

Availability of data and materials
The data and code of the current research study is available at https://github.com/taseersuleman/m1A-ensem-model.

Declarations

Ethics approval and consent to participate
Ethical approval and consent was not required for the current research study.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 30 June 2023 Accepted: 31 December 2023
Published online: 15 February 2024

References

1. Metodiev MD, Thompson K, Alston CL, Morris AAM, He L, Assouline Z, et al. Recessive mutations in TRMT10C cause defects in Mitochondrial RNA processing and multiple respiratory chain deficiencies. Am J Hum Genet. 2016;98(5):993–1000.
2. Falk MJ, Gai X, Shigematsu M, Vilardo E, Takase R, McCormick E, et al. A novel HSD17B10 mutation impairing the activities of the mitochondrial Rnase P complex causes X-linked intractable epilepsy and neurodevelopmental regression. RNA Biol. 2016;13(5):477–85.
3. Oie S, Matsuzaki K, Yokoyama W, Tokunaga S, Waku T, Han SI, et al. Hepatic rRNA transcription regulates high-fat-diet-induced obesity. Cell Rep. 2014;7(3):807–20.
4. Madec E, Stensballe A, Kjellstro S, Obuchowski M, Jensen ON, Cladie L, et al. Mass spectrometry and site-directed mutagenesis identify several Autophosphorylated residues required for the activity of PrkC, a Ser / Thr Kinase from Bacillus subtilis. J Mol Biol. 2003;2836(03):459–72.
5. Chen W, Feng P, Tang H, Ding H, Lin H. RAMPred: Identifying the N1-methyladenosine sites in eukaryotic transcriptomes. Sci Rep. 2016;6(August):1–8. https://doi.org/10.1038/srep31080.
6. Chen W, Feng P, Yang H, Ding H, Lin H, Chou KC. iRNA-3type A: identifying three types of modification at RNA's Adenosine sites. Mol Ther - Nucleic Acids. 2018;11:468–74.
7. Liu L, Lei X, Meng J, Wei Z. ISGm1A: integration of sequence features and genomic features to improve the prediction of human m1A RNA methylation sites. IEEE Access. 2020;8:81971–7.
8. Sun P, Chen Y, Liu B, Gao Y, Han Y, He F, et al. DeepMRMP: A new predictor for multiple types of RNA modification sites using deep learning. Math Biosci Eng. 2019;16(6):6231–41.
9. Xuan J, Sun W, Lin P, Zhou K, Liu S, Zheng L, et al. RMBase v2.0: deciphering the map of RNA modifications from epitranscriptome sequencing data. Nucleic Acids Res. 2018;46(D1):D327-D334. https://doi.org/10.1093/nar/gkx934.
10. Che D, Liu Q, Rasheed K, Tao X. Decision tree and ensemble learning algorithms with their applications in bioinformatics. Adv Exp Med Biol. 2011;696:191–9.
11. Malebary SJ, Alzahrani E, Khan YD. A comprehensive tool for accurate identification of methyl-Glutamine sites. J Mol Graph Model. 2022;110:108074.
12. Naseer S, Hussain W, Khan YD, Rasool N. Optimization of serine phosphorylation prediction in proteins by comparing human engineered features and deep representations. Anal Biochem. 2021;615:114069.
13. Naseer S, Hussain W, Khan YD, Rasool N. 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 Bioinforma. 2020;19(3):1703–14.
14. Butt AH, Khan YD. CanLect-Pred: A cancer therapeutics tool for prediction of target cancerlectins using experiential annotated proteomic sequences. IEEE Access. 2020;8:9520–31.
15. Shahid M, Ilyas M, Hussain W, Khan YD. ORI-Deep: improving the accuracy for predicting origin of replication sites by using a blend of features and long short-term memory network. Brief Bioinform. 2022;23(2):bbac001.
16. Malebary SJ, Khan YD. Evaluating machine learning methodologies for identification of cancer driver genes. Sci Rep. 2021;11(1):12281.
17. Hussain W, Rasool N, Khan YD. Insights into Machine Learning-based approaches for Virtual Screening in Drug Discovery: Existing strategies and streamlining through FP-CADD. Curr Drug Discov Technol. 2021;18(4):463-72.
18. Mahmood MK, Ehsan A, Khan YD, Chou K-C. iHyd-LysSite (EPSV): identifying hydroxylysine sites in protein using statistical formulation by extracting enhanced position and sequence variant feature technique. Curr Genomics. 2020;21(7):536–45.
19. Barukab O, Khan YD, Khan SA, Chou K-C. DNAPred_Prot: identification of DNA-binding proteins using composition- and position-based features. Appl Bionics Biomech. 2022;2022:1–17.
20. Akbar S, Hayat M, Iqbal M, Jan MA. iACP-GAEnsC: Evolutionary genetic algorithm based ensemble classification of anticancer peptides by utilizing hybrid feature space. Artif Intell Med. 2017;79:62–70.
21. Suleman MT, Alkhalifah T, Alturise F, Khan YD. DHU-Pred: accurate prediction of dihydrouridine sites using position and composition variant features on diverse classifiers. PeerJ. 2022;10:e14104.
22. Alghamdi W, Attique M, Alzahrani E, Ullah MZ, Khan YD. LBCEPred: a machine learning model to predict linear B-cell epitopes. Brief Bioinform. 2022;23(3):bbac035.

23. Hussain W, Rasool N, Khan YD. A sequence-based predictor of Zika virus proteins developed by integration of PseAAC and statistical moments. Comb Chem High Throughput Screen. 2020;23(8):797–804.
24. Awais M, Hussain W, Rasool N, Khan YD. iTSP-PseAAC: identifying tumor suppressor proteins by using fully connected neural network and PseAAC. Curr Bioinform. 2021;16(5):700–9.
25. Suleman MT, Khan YD. m1A-pred: prediction of modified 1-methyladenosine sites in RNA sequences through artificial intelligence. Comb Chem High Throughput Screen. 2022;25:2473.
26. Shah AA, Malik HAM, Mohammad A, Khan YD, Alourani A. Machine learning techniques for identification of carcinogenic mutations, which cause breast adenocarcinoma. Sci Rep. 2022;12(1):11738.
27. Hung TNK, Le NQK, Le NH, Van Tuan L, Nguyen TP, Thi C, et al. An AI-based prediction model for drug-drug interactions in osteoporosis and Paget's diseases from SMILES. Mol Inform. 2022;41(6):2100264.
28. Le NQK, Nguyen TTD, Ou YY. Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties. J Mol Graph Model. 2017;73:166–78.
29. Naseer S, Ali RF, Khan YD, Dominic PDD. iGluK-Deep: computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions. J Biomol Struct Dyn. 2021;40(22):11691-704.
30. Malebary SJ, Khan YD. Identification of antimicrobial peptides using Chou's 5 step rule. Comput Mater Contin. 2021;67(3):2863–81.
31. Khan SA, Khan YD, Ahmad S, Allehaibi KH. N-MyristoylG-PseAAC: Sequence-based prediction of N-Myristoyl Glycine sites in proteins by integration of PseAAC and statistical moments. Lett Org Chem. 2018;16(3):226–34.
32. Akbar S, Ahmad A, Hayat M, Rehman AU, Khan S, Ali F. iAtbP-Hyb-EnC: Prediction of antitubercular peptides via heterogeneous feature representation and genetic algorithm based ensemble learning model. Comput Biol Med. 2021;137:104778.
33. Ahmad A, Akbar S, Tahir M, Hayat M, Ali F. iAFPs-EnC-GA: Identifying antifungal peptides using sequential and evolutionary descriptors based multi-information fusion and ensemble learning approach. Chemom Intell Lab Syst. 2022;222:104516.
34. Butt AH, Alkhalifah T, Alturise F, Khan YD. A machine learning technique for identifying DNA enhancer regions utilizing CIS-regulatory element patterns. Sci Rep. 2022;12(1):15183.
35. Khan YD, Khan NS, Naseer S, Butt AH. iSUMOK-PseAAC: Prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC. PeerJ. 2021;9:e11581.
36. Malebary SJ, Khan R, Khan YD. ProtoPred: advancing oncological research through identification of proto-oncogene proteins. IEEE Access. 2021;9:68788–97.
37. Hassan A, Alkhalifah T, Alturise F, Khan YD. RCCC_Pred: a novel method for sequence-based identification of renal clear cell carcinoma genes through DNA mutations and a blend of features. Diagnostics. 2022;12(12):3036.
38. Shah AA, Alturise F, Alkhalifah T, Khan YD. Evaluation of deep learning techniques for identification of sarcoma-causing carcinogenic mutations. Digit Heal. 2022;8:205520762211337.
39. Thrun MC, Gehlert T, Ultsch A. Analyzing the fine structure of distributions. Plos One. 2020;15(10):e0238835.
40. sklearn.preprocessing.StandardScaler. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Cited 2020 Dec 17
41. Arif M, Ahmed S, Ge F, Kabir M, Khan YD, Yu DJ, et al. StackACPred: Prediction of anticancer peptides by integrating optimized multiple feature descriptors with stacked ensemble approach. Chemom Intell Lab Syst. 2022;220:104458.
42. Baig TI, Khan YD, Alam TM, Biswal B, Aljuaid H, Gillani DQ. Ilipo-pseaac: Identification of lipoylation sites using statistical moments and general pseaac. Comput Mater Contin. 2022;71(1):215–30.
43. Barukab O, Khan YD, Khan SA, Chou K-C. iSulfoTyr-PseAAC: identify tyrosine sulfation sites by incorporating statistical moments via Chou's 5-steps rule and pseudo components. Curr Genomics. 2019;20(4):306–20.
44. Rasool N, Husssain W, Khan YD. Revelation of enzyme activity of mutant pyrazinamidases from Mycobacterium tuberculosis upon binding with various metals using quantum mechanical approach. Comput Biol Chem. 2019;83:107108.
45. Akbar S, Hayat M, Tahir M, Khan S, Alarfaj FK. cACP-DeepGram: Classification of anticancer peptides via deep neural network and skip-gram-based word embedding model. Artif Intell Med. 2022;131:102349.
46. Alghamdi W, Alzahrani E, Ullah MZ, Khan YD. 4mC-RF: Improving the prediction of 4mC sites using composition and position relative features and statistical moment. Anal Biochem. 2021;633:114385.

Publisher's Note