BMC Medical Genomics

RESEARCH

Open Access

N6-methyladenine identification using deep learning and discriminative feature integration

Salman Khan[1], Islam Uddin[1], Sumaiya Noor[2], Salman A. AlQahtani[3] and Nijad Ahmad[4*]

Abstract

N6-methyladenine (6 mA) is a pivotal DNA modification that plays a crucial role in epigenetic regulation, gene expression, and various biological processes. With advancements in sequencing technologies and computational biology, there is an increasing focus on developing accurate methods for 6 mA site identification to enhance early detection and understand its biological significance. Despite the rapid progress of machine learning in bioinformatics, accurately detecting 6 mA sites remains a challenge due to the limited generalizability and efficiency of existing approaches. In this study, we present Deep-N6mA, a novel Deep Neural Network (DNN) model incorporating optimal hybrid features for precise 6 mA site identification. The proposed framework captures complex patterns from DNA sequences through a comprehensive feature extraction process, leveraging k-mer, Dinucleotide-based Cross Covariance (DCC), Trinucleotide-based Auto Covariance (TAC), Pseudo Single Nucleotide Composition (PseSNC), Pseudo Dinucleotide Composition (PseDNC), and Pseudo Trinucleotide Composition (PseTNC). To optimize computational efficiency and eliminate irrelevant or noisy features, an unsupervised Principal Component Analysis (PCA) algorithm is employed, ensuring the selection of the most informative features. A multilayer DNN serves as the classification algorithm to identify N6-methyladenine sites accurately. The robustness and generalizability of Deep-N6mA were rigorously validated using fivefold cross-validation on two benchmark datasets. Experimental results reveal that Deep-N6mA achieves an average accuracy of 97.70% on the *F. vesca* dataset and 95.75% on the *R. chinensis* dataset, outperforming existing methods by 4.12% and 4.55%, respectively. These findings underscore the effectiveness of Deep-N6mA as a reliable tool for early 6 mA site detection, contributing to epigenetic research and advancing the field of computational biology.

Keywords Deep Learning, DNA Modifications, N6-methyladenine (6 mA), Epigenetics, Sequence Analysis, DNA Methylation Detection, Deep Neural Network

Introduction

DNA modifications marked by N6-methyladenine (6 mA) emerged as notable epigenetic markers that respond to environmental factors. The study on hypoxic-stressed human cells shows a dramatic increase in mitochondrial DNA (6 mA) levels. In the mouse brain, scientists detected an inverse relationship between 6 mA levels and stress-responsive neuronal genes, thus indicating their importance in adaptation to stress events. Caenorhabditis elegans experiences mitochondrial stress, which leads to elevated 6 mA levels that maintain adaptive mechanisms that are available between generations. In rice cells,

*Correspondence:
Nijad Ahmad
Nijad@khurasan.edu.af
[1] Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan
[2] Business and Management Sciences Department, Purdue University, West Lafayette, IN, USA
[3] Department of Computer Engineering, New Emerging Technologies and 5g Network and Beyond Research Chair, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[4] Department of Computer Science, Khurasan University, Jalalabad, Afghanistan

the levels of 6 mA exhibit an opposite pattern concerning cold resistance, but they emerge as positively linked to salt and heat adaptation mechanisms [1]. The active 6 mA signaling underlies vital stress adaptation mechanisms within eukaryotic organisms. DNA N6-methyladenine (6 mA) analysis in molecular biology and stress response research requires multiple experimental approaches, which researchers have developed due to their fundamental role in epigenetics [2, 3]. The experimental approaches demonstrate effective results but encounter limitations due to their high costs, extensive effort input, and lengthy duration requirements [4, 5]. Identifying DNA N6-methyladenine (6 mA) sites requires fast, reliable computational approaches. Machine learning and deep learning provide efficient, cost-effective alternatives to experimental methods, enhancing detection accuracy and biological insights [6, 7].

Several computational models have been developed to predict DNA N6-methyladenine (6 mA), employing machine learning and deep learning techniques, i.e., SNNRice6mA [8], DNA6mA-MINT [9], SpineNet-6 mA [10], and ENet-6 mA [11]. For example, SNNRice6mA, proposed by Yu et al. [8], is a neural network model that eliminates the need for manually crafted features, allowing the model to learn directly from sequence data. This approach achieved 93% and 92% accuracy on two benchmark genome datasets (i.e., *R. chinensis* and *F. vesca*). Similarly, Rehman et al. [9] presented DNA6mA-MINT, a deep learning-based tool that follows Chou's 5-step rule and integrates CNN and Long Short-Term Memory (LSTM) layers to capture high-level features and sequential patterns. The proposed model achieved an average accuracy of 92.53% and 93.2% using cross-validation across combined-species genomes. Further, Li et al. [12] introduced Deep6mA, a deep learning framework that leverages convolutional neural networks (CNNs) to extract sequence features automatically. Deep6mA demonstrated impressive performance, achieving an accuracy of 94% on the rice genome while showcasing strong generalization capabilities. The model maintained over 90% accuracy when applied to other species, including *Arabidopsis thaliana*, *Fragaria vesca*, and *Rosa chinensis*, highlighting its potential for cross-species applications. Similarly, Hasan et al. [13] presented i6mA-Fuse, which combines a random forest (RF) model with a linear regression fusion approach. The i6mA-Fuse model integrates several encoding methods, including mononucleotide binary, dinucleotide binary, k-space spectral nucleotide, k-mer, and electron–ion interaction pseudo potential (EIIP), yielding AUC values of 0.982 and 0.978, along with MCC scores of 0.869 and 0.858 for Rosa *chinensis* and *Fragaria vesca*, respectively. Recently, Khanal et al. [14] proposed i6mA-Stack, a stacking ensemble-based model incorporating multiple feature representations to improve predictive performance across Rosaceae genomes. i6mA-Stack demonstrated accuracy, achieving 94.09% for *Fragaria vesca* and 93.44% for *Rosa chinensis*, outperforming several existing 6 mA prediction tools. These models have shown strong performance; however, further improvements can still be achieved, especially in addressing more complex data patterns. Additionally, the prediction performance of these models is limited when nonlinearity exists in the dataset.

In this study, we propose Deep-N6mA, a novel deep learning-based framework for identifying N6-methyladenine (6 mA) sites in DNA sequences. The proposed model integrates a DNN with optimally hybrid features to enhance predictive accuracy. To construct the hybrid feature vector, multiple feature extraction techniques are employed, including k-mer, Dinucleotide-based Cross Covariance (DCC), Trinucleotide-based Auto Covariance (TAC), Pseudo Single Nucleotide Composition (PseSNC), Pseudo Dinucleotide Composition (PseDNC), and Pseudo Trinucleotide Composition (PseTNC). The hybrid approach incorporates diverse features but also introduces redundancy and noise. Feature selection techniques, such as Principal Component Analysis (PCA), mitigate these issues by retaining only the most relevant attributes, improving model performance. Finally, the classification framework of Deep-N6mA is based on a multilayer Deep Neural Network (DNN) designed to achieve high precision in identifying 6 mA sites. Extensive experimental evaluations demonstrate that Deep-N6mA significantly outperforms conventional machine learning classifiers, including Support Vector Machine (SVM), k-nearest Neighbors (KNN), Naive Bayes (NB), and Random Forest (RF), using both benchmark datasets. Moreover, Deep-N6mA surpasses state-of-the-art models by achieving superior performance metrics and establishing its robustness and reliability in 6 mA site prediction—the design and implementation of the proposed model, as shown in Fig. 1.

The remainder of the paper is structured as follows: Sect. " Methods and Materials" explains the material and methods, including the benchmark dataset, feature extraction, and classification algorithms. The performance evaluation metrics are presented in Sect. " Performance Evaluation". Sect. " Experimental Result and Analysis" discusses the experimental findings and discussions. Finally, Sect. " Conclusions" includes the paper's conclusion and future work.

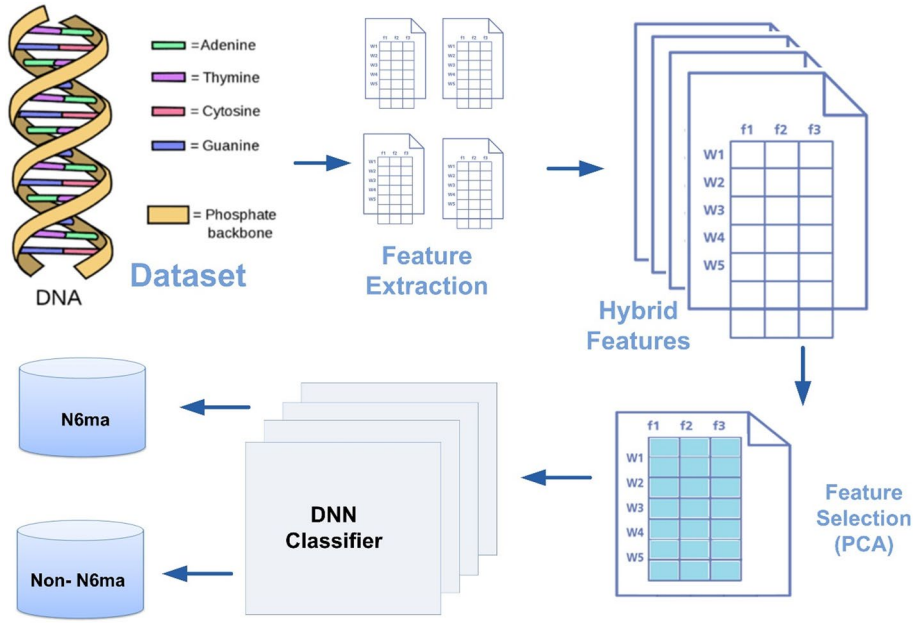

Fig. 1 Framework of the proposed model

Methods and materials

Benchmark dataset preparation

The development of a robust and efficient computational model necessitates the availability of a valid and reliable benchmark dataset. Such datasets are fundamental for training, validating, and testing machine learning algorithms under standardized conditions, ensuring reproducibility and comparability of results [14]. For this study, positive samples (6mAs) were sourced from the well-established MDR database for *F. vesca* and *R. chinensis* [http://mdR.xieslab.org/] [15]. Each sequence consisted of 41 base pairs, with an adenine "A" positioned centrally and a modification score of at least 30. To minimize redundancy and ensure data diversity, we applied the CD-HIT tool with a similarity threshold 0.7, filtering out highly similar sequences [13]. A similar approach was followed for selecting negative samples. After this screening, we obtained 4,626 and 1,912 positive and negative samples for *R. chinensis* and *F. vesca*, respectively. Both datasets were curated to ensure balanced classes, reduced noise, and suitability for benchmarking machine learning models.

Furthermore, to evaluate the model's generalizability and to reflect better real-world conditions, we perform independent validation, i.e., balance-independent validation. For balance-independent validation, 15% of the benchmark dataset was set aside. We used a balanced set of 694 (i.e., 347 positive and 347 negative samples) for the *F. vesca* dataset and 286 (i.e., 143 positive and 143 negative samples) for the *R. chinensis* dataset. Table 1 shows the overall statistics of the two benchmark datasets utilized in the study.

Table 1 Statistical summary of the datasets for the two species

Genomes	Samples	Training Dataset	Independent Dataset
F. vesca	Positive	1966	347
	Negative	1966	347
R. chinensis	Positive	813	143
	Negative	813	143

Feature extraction methods

The feature extraction process enables biological sequences to evolve into numerical data for machine-learning model integration. Bioinformatics specialists have developed multiple sequence transformation techniques that convert biological nucleotide strings into mathematical models while maintaining their structural uniqueness. The bioinformatics algorithms transform DNA sequences into separate statistical forms without disrupting their original patterns and measurement values [16, 17]. In alignment with Chou's 5-step guidelines, this study employs six feature extraction techniques: The PseKNC compound features PseSNC (K = 1), PseDNC (K = 2), and PseTNC (K = 3) together with k-mer series, Dinucleotide-based Cross Covariance (DCC) and Trinucleotide-based Auto Covariance (TAC).

The Pseudo K-Tuple Nucleotide Composition Sequences approach categorizes provided DNA sequences into a function vector while suppressing order information and suggesting a similarity between DNA samples [18]. Let consider a DNA sequence D with N number of nucleotide is represented as:

$$D = D_1, D_2, D_3, \ldots, D_h, \ldots, D_N \tag{1}$$

The number of nucleotides in a DNA sequence, or its dimension, is denoted by the letter N:

$$D_h\{T, G, C, A\}\,(h = 1, 2, 3, \ldots, N) \tag{2}$$

Various models have been suggested for DNA sequences while preserving their biological significance. [*A, C, G, T*] represents Adenine, Cytosine, Guanine, and Thymine, respectively, and D_h denotes a nucleotide at the h^{th}, the position of a sequence. The general form for Pseudo K-Tuple Nucleotide Composition (PseKNC) [19] is as follows:

$$\left[\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_x, \ldots, \varphi_y\right] T \tag{3}$$

T seems to be the transposed vector, y is numeric, and ϕ_x, is the value of the DNA sequence's function vector, calculated using Eq. (4).

$$\phi_x = \begin{cases} \frac{f_x^{K-tuple}}{\sum_{h=1}^{4^K} f_x^{K-tuple} + w\sum_{h=1}^{\lambda}\theta_j} \left(1 \le x \le 4^K, x = 1, 2, \ldots\right) \\ \frac{w\theta_{x-4^K}}{\sum_{h=1}^{4^K} f_x^{K-tuple} + w\sum_{h=1}^{\lambda}\theta_j} \left(4^K + 1 \le x \le 4^K + \lambda\right) \end{cases} \tag{4}$$

where, θ_j represent the h^{th} tier correlation factor or h^{th} rank correlation factor that reflects the sequence order correlation in most contiguous K-tuple nucleotides. λ represents the total number correlation rank, and w represents the weight. In Eq. 4, the total correlation rank λ weight w was selected through empirical analysis, and the experimental result exhibits that $w=0.1$ and $\lambda=1$ achieved high-performance results.

In this paper, we use the PseKNC technique to convert the provided sequences into discrete feature vectors while maintaining the sequence order data. By designating different values to K (i.e., $K=1, 2, 3$) in Eq. (3), three distinct modes of PseKNC were emanated, i.e., PseSNC ($K=1$), PseDNC ($K=2$), and PseTNC ($K=3$), defined as follows:

$$PseSNC = \left| f_{j=1,\ldots 4D}^{1-Tuple} \xrightarrow{f} (A, C, G, T) \right. \tag{5}$$

$$PseDNC = \left| f_{j=1,\ldots 16D}^{2-Tuple} \xrightarrow{f} (AA, CC, GG, TT) \right. \tag{6}$$

$$PseTNC = \left| f_{j=1,\ldots 64D}^{3-Tuple} \xrightarrow{f} (AAA, CCC, GGG, TTT) \right. \tag{7}$$

Furthermore, the k-mer method splits a DNA sequence into overlapping substrings of length k, known as k-mers. These k-mers act as key features for sequence analysis. Overlapping k-mers of length k represent a DNA sequence of length L. The distinct k-mers (i.e., $k=2$) and their frequencies are defined as:

$$f(t) = \frac{N(t)}{N}, t \in (AA, CC, GG....TT) \tag{8}$$

Similarly, the dinucleotide-based Cross Covariance (DCC) method calculates the correlation between two physicochemical indices of dinucleotides (pairs of nucleotides) separated by a lag. For each dinucleotide pair separated by a lag, the covariance is determined based on properties like hydrophobicity or polarity. The resulting cross-covariance matrix is represented as:

$$DCC(u_1, u_2, lag) = \sum_{i=1}^{L-lag-1} \frac{(P_{u_1}(R_iR_{i+1}) - \overline{P}_{u_1})(P_{u_2}(R_{i+lag}R_{i+lag+1}) - \overline{P}_{u_2})}{(L - lag - 1)} \tag{9}$$

where u_1 and u_2 are different physicochemical indices, L is the length of the nucleotide sequence, $(P_{u_1}(R_iR_{i+1}))$ $(P_{u_2}(R_iR_{i+1}))$ is the numerical value of the physicochemical index $u_1(u_2)$ for the dinucleotide R_iR_{i+1} at position$i$,$\overline{P}_{u_1}(\overline{P}_{u_2})$, is the average value for physicochemical index $u_1(u_2)$ along the whole sequence:

$$\overline{P_u} = \sum_{j=1}^{L-1} \frac{P_u(R_jR_{j+1})}{L - 1} \tag{10}$$

The dimension of the DCC feature vector is $N * (N - 1) * LAG$, where N, is the number of physicochemical indices and LAG is the maximum of *lag* ($lag = 1, 2, \ldots, LAG$). In this paper, we select $LAG=2$ and six physicochemical properties (i.e., $N=6$), so the feature vector length is 60.

Finally, the Trinucleotide-based Auto Covariance (TAC) encoding measures the correlation of the same physicochemical index between trinucleotides separated by lag nucleic acids along the sequence and can be calculated as:

$$TAC(lag, u) = \sum_{i=1}^{L-lag-2} \frac{(P_u(R_iR_{i+1}R_{i+2}) - \overline{P}_u)(P_u(R_{i+lag}R_{i+lag+1}R_{i+lag+2}) - \overline{P}_u)}{L - lag - 2} \tag{11}$$

where u is a physicochemical index, L is the length of the nucleotide sequence, $P_u(R_iR_{i+1}R_{i+2})$ is the numerical value of the physicochemical index u for the trinucleotide $R_iR_{i+1}R_{i+2}$, at position i, $\overline{P}_u$ is the average value for physicochemical index u along the whole sequence:

$$\overline{P_u} = \sum_{j=1}^{L-2} \frac{P_u(R_j R_{j+1} R_{j+2})}{L-2} \tag{12}$$

The dimension of the TAC feature vector is $N * LAG$, where N is the number of physicochemical indices, and LAG is the maximum of lag ($lag = 1, 2, \ldots, LAG$). In this paper, we select LAG=2 and six physicochemical properties (i.e., N=6), so the feature vector length is 12. The selected physiochemical properties (i.e., N=6) for DCC and TAC are shift, roll, rise, slide, twist, and tilt.

Hybrids features

This study used six distinct feature extraction methods to encode DNA sequences into discrete feature vectors, as summarized in Table 2. All individual features were integrated to construct a comprehensive hybrid feature vector by capturing diverse sequence-derived attributes. Machine learning models leveraging hybrid features benefit from combining multiple extraction techniques, enhancing predictive performance by effectively capturing complex data patterns. This approach remains a widely adopted strategy in bioinformatics and genomics for improving model interpretability and accuracy.

Features optimization

Feature vectors often contain noisy, redundant, or irrelevant features that negatively impact a classifier's performance. To address this, we utilize Principal Component Analysis (PCA) for feature selection, a dimensionality reduction method that processes multivariate data by computing covariance matrices and eigenvectors to reduce the feature vector's dimensions. The primary goal of PCA is to preserve as much important data as possible while minimizing dimensionality [20]. Feature selection generally aims to reduce the number of input variables, decrease computational costs, and eliminate noisy features. Statistically based feature selection methods evaluate the relationships between input variables and the target variable, selecting those with the most significant associations. PCA highlights variations within the dataset, identifying key characteristics and making the data more interpretable. PCA works by computing the covariance matrix C of the feature vector:

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(x_i - \overline{x})^T \tag{13}$$

where x_i represents the i^{th} feature vector, $\overline{x}$, is the mean vector, and n is the total number of samples. Calculating the eigenvalues (λ) and corresponding eigenvectors (v) of C:

$$Cv = \lambda v \tag{14}$$

Eigenvectors with the largest eigenvalues represent the directions of maximum variance in the dataset. In this study, we consider the hybrid feature vector with dimensions $p \times q$, where p represents the number of features and q denotes the number of sequences. The dimensions for the hybrid feature vector are 176*3932 for the *F. vesca* dataset and 176*1626 for the *R. chinensis* dataset. Let k represent the number of desired features after selection. In our case, the desired feature dimensions are 86*3932 for the *F. vesca* dataset and 68*1626 for the *R. chinensis* dataset. It is important to note that the value of k must be smaller than p (i.e.,$k < p$) to ensure that the selected features form a subset of the original feature set.

Table 2 Dimension of feature vector with different values of K

Method	Number of Features
K-mer	16
Dinucleotide-based Cross Covariance (DCC)	60
Trinucleotide-based Auto Covariance (TAC)	12
Pseudo Single Nucleotide Composition (PseSNC)	4
Pseudo Dinucleotide Composition (PseDNC)	18
Pseudo Trinucleotide Composition (PseTNC)	66
Hybrid Features	176

Deep neural network architecture

DNN is a subfield of machine learning algorithms in artificial intelligence inspired by the human brain's working mechanism and activities. A DNN model topology consists of an input layer, an output layer, and multiple hidden layers, as shown in Fig. 2. A Deep Neural Network (DNN) model requires its hidden layers for learning processes, which substantially determine model performance outcomes. Model efficiency benefits from additional hidden layers, but these enhancements create higher computational demand, risks of overfitting, and, more significantly, expenses [21]. With DNN models, scientists can automatically recognize important features from data sets without human handling by applying conventional learning techniques [22]. DNN models deliver superior results to traditional methods in complex classification applications. DNN models successfully achieve results across the bioengineering domain [23], image and speech recognition categories [24], and natural language processing applications [25].

Model training

This study used two benchmark datasets to determine 6 mA sites through a developed DNN modeling system.

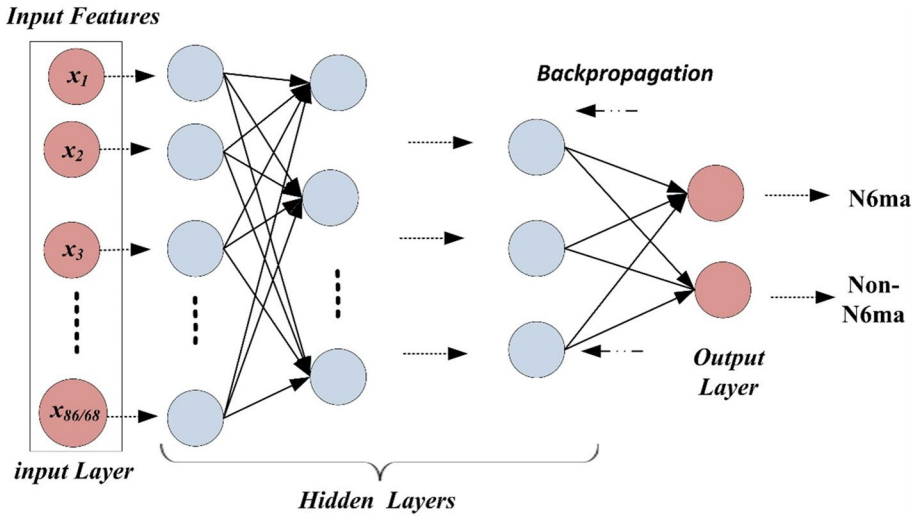

Fig. 2 DNN configuration topology, the circle represents neurons at each layer

The structure of this proposed model includes four hidden layers and an input and output layer. The predictive network uses each layer to process fundamental data elements through multiple computational neurons that generate outcomes.

Firstly, the given feature vector, i.e., $X\{x_i, x_2, x_3, ...x_n\}$ was provided to the input layer. Each neuron in the input layer processes a feature x_i and produces an output Y by using a weight vector W_i, bias vector B_i, and activation function, f_i as shown in Eq. 15. Secondly, the output of the input layer is given as input to the first hidden layer and produced a new output using Eq. (15). Thirdly, the output of the first hidden layer is given as input to the second hidden layer and so forth [26], this process was continued until we reached the output layer. The output layer generated binary values, i.e., 0 and 1. In the case of the first layer, the value 0 denotes N6ma, and 1 denotes Non-N6ma.

$$Y = f(XW_i + B_i) \tag{15}$$

In Eq. (15), each neuron in the model utilizes the Xavier function for weight matrix initialization to maintain uniform variance between layers. The model employs backpropagation for weight matrix adjustment to reduce output prediction error variance with operational target values. Rectified Linear Unit (ReLU) activation functions power both input and hidden layers to detect nonlinear patterns so that Softmax functions at the output layer create probability estimates from 0 to 1, which then determine data point classifications.

Performance evaluation

To evaluate the performance of a machine learning algorithm, performance evaluation parameters are commonly used to check the model's validity and reliability. These parameters include Accuracy (*ACC*), Specificity (*SP*), Sensitivity (*SN*), and Matthews Correlation Coefficient (*MCC*). Accuracy represents the model's overall accuracy, describing the classifier's correctness and general performance. Specificity, also known as the true negative rate, measures the proportion of negative classes correctly identified by the classifier. Sensitivity is the true positive rate, and SN evaluates the proportion of positive classes correctly identified by the classifier. Matthews Correlation Coefficient measures the quality of binary classifications; MCC provides a balanced evaluation that accounts for true positives, true negatives, false positives, and false negatives. This study uses these four performance metrics adopted in a series of publications [27–29]. The mathematical equations for each of these metrics are given below:

$$ACC = \frac{T^+ + T^-}{T^+ + F^+ + T^- + F^-} 0 \leq ACC \leq 1 \tag{16}$$

$$SP = \frac{T^-}{F^+ + T^-} 0 \leq SP \leq 1 \tag{17}$$

$$SN = \frac{T^+}{T^+ + F^-} 0 \leq SN \leq 1 \tag{18}$$

$$MCC = \frac{(T^- * T^+) - (F^- * F^+)}{\sqrt{(f^+ + T^+)(T^+ + F^-)(F^+ + T^-)(T^- + F^-)}} - 1 \leq MCC \leq 1 \quad (19)$$

where,

- T^+ True Positives
- F^+ False Positives
- T^- True Negatives
- F^- False Negatives

Experimental result and analysis

The proposed model efficiency is evaluated and discussed in depth in this section. Several validation tests, including the K-fold validation test, can be utilized to assess the overall performance of the machine learning training algorithm in bioinformatics. The K-fold cross-validation approach is a typical validation technique that uses evenly balanced findings. Consequently, a fivefold cross-validation test using benchmarking datasets was employed to examine the overall accuracy of the proposed approach in this work.

Experimental setup

The system configuration is designed to support the effective implementation and execution of machine learning models. The software setup includes Python 3.6, the primary programming language because it comprehensively supports machine learning libraries and data science capabilities. Deep learning development and model training occur through TensorFlow 2.0 and PyTorch 1.4, though NumPy and Pandas perform numerical work and data handling tasks. With its broad functionality, Scikit-learn enables tasks ranging from classification to regression and data preprocessing. Through its package management and deployment capabilities, Anaconda3 provides users with simplified administration, while Jupyter Notebook delivers an interactive development environment for code creation, prototyping, and visualizations. The machine learning workflows are performed efficiently through an optimized hardware setup. The system runs an HP Core i7 12th-generation high-speed processing unit with 8 GB RAM dedicated to managing data-intensive missions. Storage includes 256 GB SSD for quick performance boosts and 1 TB of storage capacity to handle large datasets and project files. The NVIDIA GeForce GTX 1060 GPU boosts deep learning performance by expediting the operations of neural network training processes.

Nucleotide composition analysis

To analyze the differences in nucleotide composition between sequences containing 6 mA sites and those without, the Two Sample Logos [30, 31] method was employed to identify statistically significant variations. As illustrated in Fig. 3, adenosine and thymine were significantly enriched in sequences with 6 mA sites ($P<0.05$), while sequences lacking 6 mA sites showed a strong preference for cytosine and guanine ($P<0.05$). These findings support the feasibility of developing a computational approach for 6 mA site identification based on sequence characteristics.

Hyper parameters and optimizations

This section aims to find the best configuration values for the hyper-parameters used in the DNN topology. To assess the DNN's performance with different hyper-parameters, we used a grid search technique that tests various combinations of parameters. The analysis focused on hyper-parameters that significantly impact the DNN model's performance. These parameters include the activation function, Learning Rate (LR), and number of iterations. The optimum configuration values for the hyper-parameters are obtained through the grid search, as shown in Table 3.

Firstly, we conducted a series of experiments to determine the effects of the activation function and learning rate. The results of the experiments are given in Table 4 using ReLU and Tanh as the activation function and learning rates from 0.01 to 0.03. Table 4 shows that the highest accuracy, i.e., 97.70% and 95.75% on the *F. vesca*

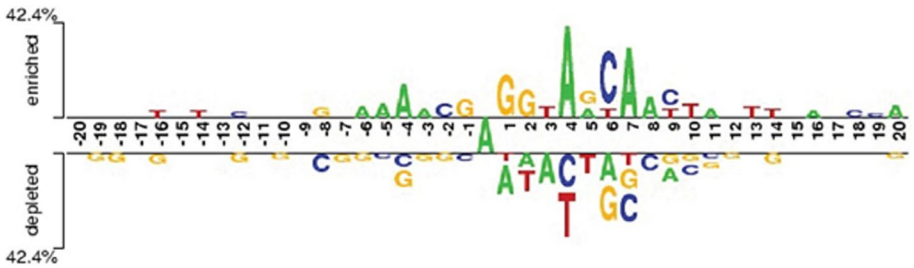

Fig. 3 Nucleotide composition differences between 6 mA and non-6 mA site-containing sequences, identified using the Two Sample Logos method

Table 3 List of optimum hyper-parameters values of the proposed model

List of Parameters	Optimal values
Seed	12345L
Learning rates	0.01
Activation Functions	ReLU and SoftMax
Weight initialization function	XAVIER function
Regularization l2	0.001
Dropout	0.25
Number of Neurons at hidden layers	86–70-45–21-6–2, 68,52,18,12,4,2
Number of hidden layers	4
Updater	ADAGRAD function
Training Epoch	200
Momentum	0.9
Optimizer	SGD Method

Table 4 Impact of different learning rates and activation function ReLU on the performance of the DNN model using a fivefold model

Species	LR	ReLU ACC (%)	Tanh ACC (%)
F. vesca	0.01	97.70	94.91
	0.02	97.34	94.34
	0.03	96.87	93.67
R. chinensis	0.01	95.75	92.65
	0.02	95.05	91.17
	0.03	95.53	90.94

and *R. chinensis* datasets, respectively, is obtained by the DNN classifier at a learning rate value of 0.01 using ReLU as an activation function.

Furthermore, we can observe from the table that the accuracy of the DNN model is continuously improved by decreasing the learning rate. However, after reducing the learning rate from 0.01, the DNN model accuracy could not significantly improve. Hence, we can conclude that the DNN model presented a high accuracy at a learning rate of 0.01 with the ReLU activation function. The optimum values of the various hyper-parameters are shown in Table 3.

Secondly, we performed several experiments to analyze the performance of the DNN model from various training epochs at the model training stage. The results of the research are depicted in the Figs. 4 and 5. According to statistics, the number of errors created decreases as training epochs increase. Consider the *F. vesca* dataset Fig. 4, in which the DNN had 8.78 error losses at the start of the epochs and was regularly reduced to 0.09 as the epochs improved to 200.

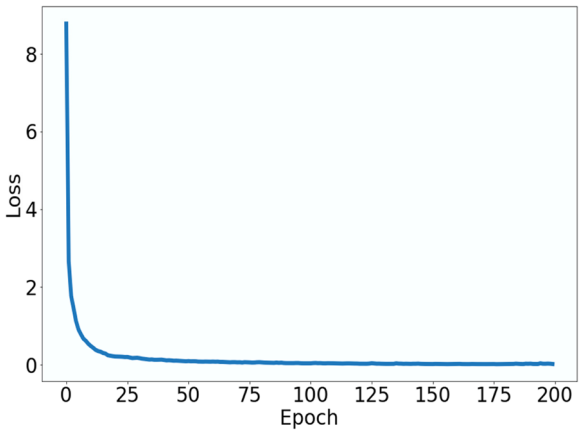

Fig. 4 Performance on the *F. vesca* dataset using the ReLU activation functions and 5-Fold cross-validations

Moreover, Fig. 5 shows the *R. chinensis* dataset, where the first iteration resulted in a cumulative error loss of 4.8, which was reduced to 0.078 when the iteration was increased to 200. It can be concluded from the figures that 200 epochs are the optimum number of iterations as the error losses become stable at this number. Consequently, a set of optimal configurations was obtained through this analysis, as presented in Table 3.

Performance analysis using sequence formulation techniques

In this section, we evaluate the performance of the proposed model using various sequence formulation techniques, as presented in Table 5 with the *F. vesca* and *R. chinensis* datasets. The results show that the proposed model performs best when using hybrid features compared to other individual feature methods with the *F. vesca* dataset. Before feature selection, the model achieved a success rate of 95.87%, Sensitivity of 97.75%, Specificity of 90.86%, and MCC of 0.903%. To enhance the model's performance, feature selection was applied to reduce the dimensionality of the hybrid feature space. After using this dimensionality reduction, the model's performance significantly improved, with the accuracy increasing to 97.70%, Sensitivity 98.01%, Specificity 97.30%, and MCC 0.951.

Similarly, we evaluate the performance of the proposed model using different sequence formulation methods on the *R. chinensis* dataset, as presented in Table 5. Incorporating hybrid features yielded the best results compared to other individual feature methods. Initially, the proposed model achieved an accuracy rate of 91.75%, Sensitivity of 93.09%, Specificity of 90.33%, and MCC of 0.891. To further enhance the performance of the proposed model, the dimensionality of the hybrid feature

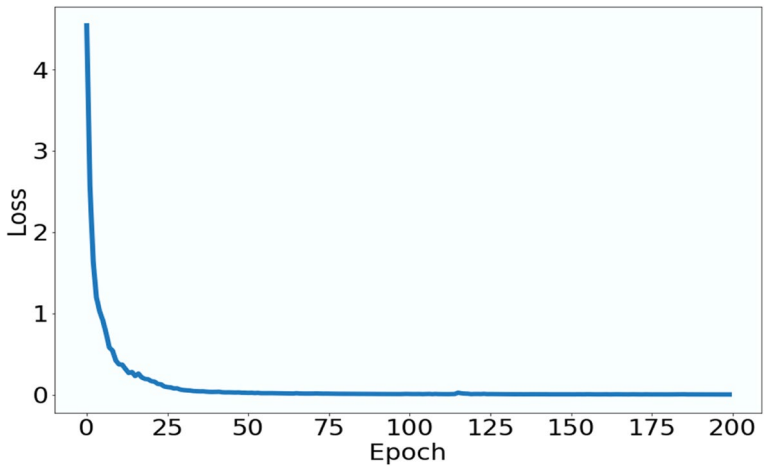

Fig. 5 Performance on the *R. chinensis* dataset using the ReLU activation functions and 5-Fold cross-validations

Table 5 Performance comparison using sequence formulation techniques using *F. vesca* and *R. chinensis*

Species	Methods	ACC (%)	SN (%)	SP (%)	MCC
F. vesca	TAC	90.52	93.32	83.21	0.811
	NAC	85.89	93.87	87.92	0.792
	Kmer	85.90	88.45	80.13	0.786
	PseSNC	80.13	83.34	79.93	0.774
	PseDNC	90.95	92.65	87.83	0.788
	PseTNC	89.32	91.34	85.54	0.808
	Hybrid feature (without feature selection)	95.87	97.75	90.86	0.903
	Hybrid feature (with feature selection)	97.70	98.01	97.30	0.951
R. chinensis	TAC	88.52	90.14	83.21	0.773
	NAC	85.89	75.32	87.32	0.736
	Kmer	85.90	80.45	86.43	0.750
	PseSNC	79.13	81.34	76.63	0.721
	PseDNC	88.95	89.65	81.83	0.792
	PseTNC	84.98	83.56	85.99	0.712
	Hybrid feature (without feature selection)	91.75	93.09	90.33	0.891
	Hybrid feature (with feature selection)	95.75	96.45	94.55	0.921

vector is reduced. This dimensionality reduction significantly improved the model's performance, with accuracy rising to 95.75%, Sensitivity increasing to 96.45%, Specificity improving to 94.55%, and MCC reaching 0.921. The experimental results show that optimal hybrid features significantly boost the proposed model's performance, making it more effective in identifying 6 mA sites using both datasets.

Performance comparison of different classifiers

In this section, the performance of the proposed model is compared with widely used machine learning algorithms using optimally selected hybrid features. For evaluation, classifiers including Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbor (KNN) were employed. The performance of various classifiers on the *F. vesca* and *R. chinensis* datasets was compared to evaluate their effectiveness. A performance comparison of different ML algorithms on both datasets is provided in Table 6.

As shown in Table 6 for the *F. vesca* dataset, the NB, RF, SVM, and KNN models achieved ACC scores of 83.40%, 84.10%, 85.94%, and 89.69%, respectively, with MCC values ranging from 0.722 to 0.753. In contrast, the proposed Deep-N6mA model demonstrated superior performance, achieving an ACC of 97.70%, SN of 98.01%, SP

Table 6 Performance comparison of different classifiers using the *F. vesca* and *R. chinensis* datasets

Species	Methods	ACC (%)	SN (%)	SP (%)	MCC
F. vesca	NB	83.40	88.81	78.41	0.722
	RF	84.10	81.51	88.28	0.734
	SVM	85.94	81.24	86.74	0.735
	KNN	89.69	92.22	85.65	0.753
	Deep-N6mA	97.70	98.01	97.30	0.951
R. chinensis	SVM	78.57	78.03	83.92	0.677
	RF	81.33	81.51	76.82	0.705
	NB	81.89	78.21	87.24	0.727
	KNN	84.43	91.12	83.25	0.714
	Deep-N6mA	95.75	96.45	94.55	0.921

of 97.30%, and MCC of 0.951, significantly outperforming all other classifiers. Similarly, as presented in Table 6, the SVM, RF, NB, and KNN models achieved ACC scores of 78.57%, 81.33%, 81.89%, and 84.43%, respectively, with MCC values ranging from 0.677 to 0.727. The proposed Deep-N6mA model demonstrated superior performance, achieving an ACC of 95.75%, SN of 96.45%, SP of 94.55%, and MCC of 0.921, highlighting its enhanced predictive capability over conventional classifiers.

To analyze further, we evaluate the proposed model's performance using the Area Under the ROC Curve (AUC), as shown in Figs. 6 and 7. AUC is a widely used metric for assessing the performance of binary classifiers, with values ranging from 0 to 1. A higher AUC indicates better predictive capability, with values closer to 1 representing superior performance than those closer to 0 [32]. The AUC analysis plots the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis.

Figures 6 and 7 show that the proposed model achieved an AUC of 0.982 on the *F. vesca* dataset and 0.964 on the *R. chinensis* dataset, indicating excellent performance compared with widely used ML algorithms. The AUC curve visually demonstrates the model's performance, with the Area under the curve increasing as the model's ability to distinguish between positive and negative classes improves. A shrinkage in the Area under the curve would suggest a decline in the model's effectiveness, emphasizing that the proposed model exhibits robust predictive power in both datasets.

Comparison with existing predictors

In this section, the performance of the proposed Deep-N6mA model was evaluated against existing predictors, i.e., i6mA-Fuse [13] and i6mA-stack [14], on the *F. vesca* and *R. chinensis* datasets. Table 7 presents the performance comparison of the proposed Deep-N6mA model with existing predictors.

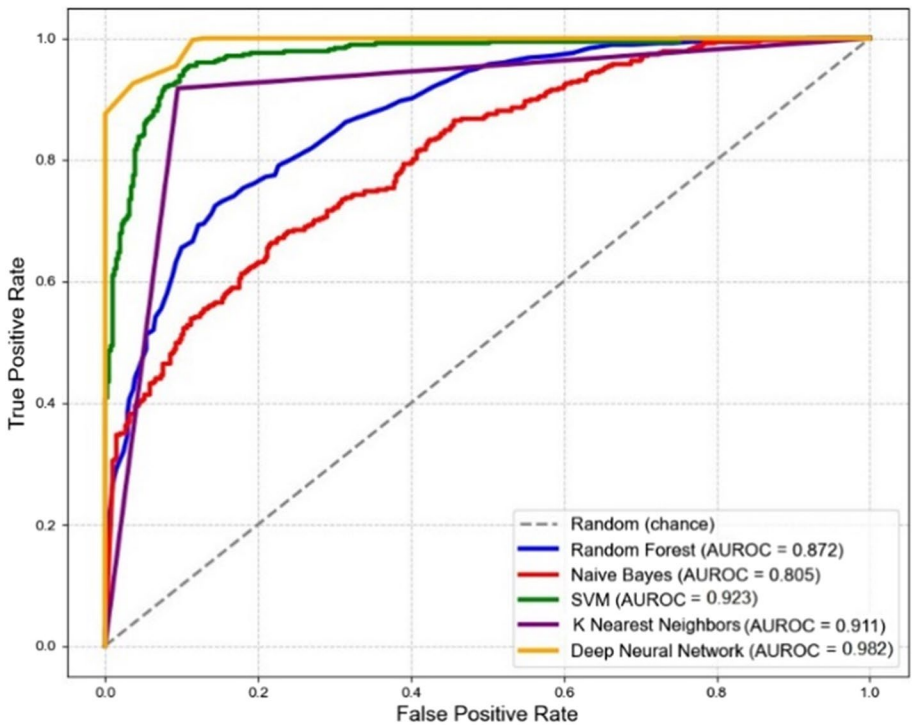

Fig. 6 AUC comparison with different ML algorithms using the *F. vesca* dataset

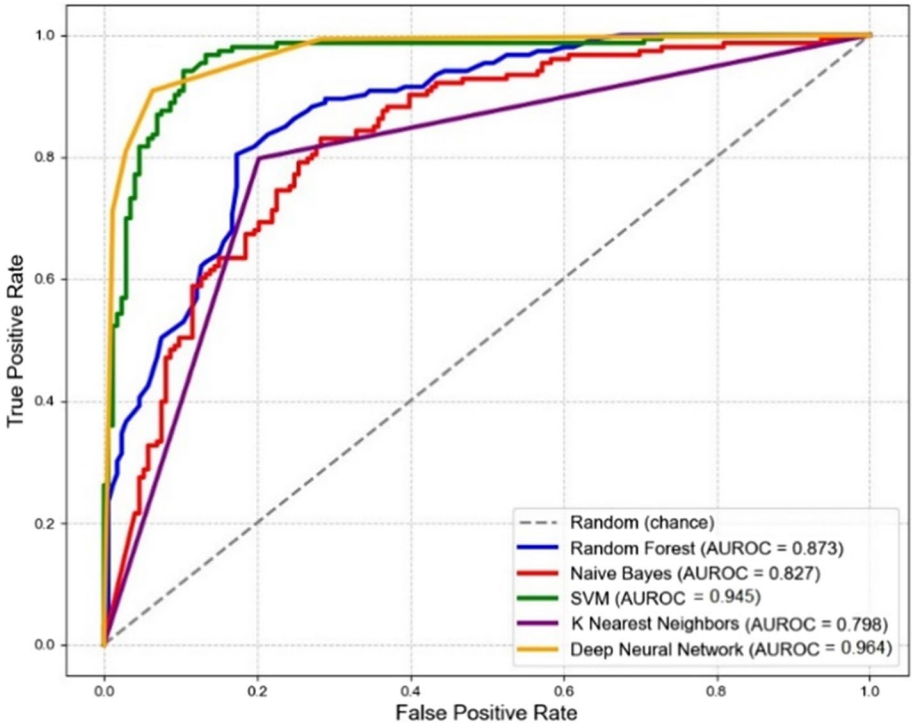

Fig. 7 AUC comparison with different ML algorithms using the *R. chinensis* dataset

Table 7 The proposed predictor is compared to existing predictors

Species	Model	ACC (%)	SN (%)	SP (%)	MCC
F. vesca	I6mA-Fuse [13]	93.40	90.80	95.7	0.873
	I6mA-stack [14]	93.76	93.25	94.30	0.875
	Proposed Deep-N6mA	97.70	98.01	97.30	0.951
R. chinensis	I6mA-Fuse [13]	91.60	88.10	95.0	0.851
	I6mA-stack [14]	90.79	90.00	91.61	0.815
	Proposed Deep-N6mA	95.75	96.45	94.55	0.921

Table 8 The performances of the proposed model on the independent datasets

Species	Methods	ACC (%)	SP (%)	SN (%)	MCC
F. vesca	i6mA-Fuse [13]	93.70	94.80	92.80	0.869
	6 mA-stack [14]	95.10	97.11	91.06	0.880
	Proposed Deep-N6mA	95.65	97.72	93.61	0.892
R. chinensis	i6mA-Fuse [13]	92.90	94.30	91.5	0.858
	6 mA-stack [14]	93.44	92.81	94.12	0.868
	Proposed Deep-N6mA	94.23	95.32	93.14	0.876

From Table 7, the i6mA-Fuse [13] model achieved an accuracy (ACC) of 93.40%, Sensitivity (SN) of 90.80%, Specificity (SP) of 95.70%, and a Matthews correlation coefficient (MCC) of 0.873 on the *F. vesca* dataset. The i6mA-Stack [14] model yielded an ACC of 93.76%, SN of 93.25%, SP of 94.30%, and MCC of 0.875. In comparison, the proposed Deep-N6mA model significantly outperformed these existing methods, achieving an ACC of 97.70%, SN of 98.01%, SP of 97.30%, and MCC of 0.951. Similarly, on the *R. chinensis* dataset, the i6mA-Fuse [13] model achieved an ACC of 91.60% and an MCC of 0.851, while the i6mA-Stack [14] model scored 90.79% ACC and MCC of 0.815. In contrast, the Deep-N6mA model outperformed them with an ACC of 95.75% and an MCC of 0.921. The findings underscore the effectiveness of the proposed Deep-N6mA model, which consistently outperforms existing models on both datasets. This reinforces its outstanding capability in identifying N6-methyladenine modifications in DNA sequences. Notably, the Deep-N6mA model achieves the highest performance, surpassing the average success rate of the benchmark methods by 4.12% and 4.55%, respectively.

Performance comparison on an independent dataset

The accurate measure of a prediction model's generalization is its performance on unseen data. To evaluate the robustness of our developed model, we tested it on an independent dataset, using 80% of the data for training

and 20% for testing. The performance comparison of the proposed Deep-N6mA model on the independent datasets is shown in Table 8.

From Table 8, on the *F. vesca* dataset, the i6mA-Fuse [13] model achieved an ACC of 93.7%, with an MCC of 0.869. The 6 mA-Stack [14] model outperformed i6mA-Fuse by achieving an ACC of 95.10%, with an MCC of 0.880. However, the proposed Deep-N6mA model achieved superior performance, with an ACC of 95.65%, an SP of 97.72%, an SN of 93.61%, and an MCC of 0.892. Similarly, on the *R. chinensis* dataset, the i6mA-Fuse [13] achieved an ACC of 92.9%, with SP of 94.3% and SN of 91.5%, whereas the 6 mA-Stack [14] model achieved an ACC of 93.44%, an SP of 92.81%, and an SN of 94.12%. However, the proposed Deep-N6mA model achieved superior performance: ACC of 94.23%, SP of 95.32%, SN of 93.14%, and an MCC of 0.876. Hence, evaluating the proposed Deep-N6mA on independent datasets demonstrates its superior performance in accurately predicting N6-methyladenine (6 mA) sites, improving the average success rate of the benchmark methods as high as 1.25% and 1.06%, respectively.

Conclusions

N6-methyladenine (6 mA) is an essential DNA modification, playing a pivotal role in regulating key biological processes such as gene expression, DNA replication, and repair mechanisms. Identifying 6 mA sites within DNA sequences is critical for understanding epigenetic regulation and its implications in various organisms. This study introduced Deep-N6mA, a novel computational framework based on a Deep Neural Network (DNN). The proposed model employs a hybrid feature extraction approach, integrating multiple sequence-based features to capture intricate patterns within DNA sequences. Rigorous evaluations were conducted using two datasets, *F. vesca* and *R. chinensis*, with fivefold cross-validation to ensure robust performance assessment. The Deep-N6mA model achieved remarkable outcomes, with accuracy prediction techniques by 3.94% on the *R. chinensis* dataset and 4.64% on the *F. vesca* dataset, indicating its superior effectiveness compared to prior methods. Furthermore, the proposed model demonstrated superior Sensitivity, Specificity, and MCC values, underscoring its ability to accurately and reliably identify 6 mA sites. Compared to earlier models, the Deep-N6mA approach excels in predictive accuracy and generalizability across species, making it a significant advancement in computational biology. This study highlights the effectiveness of leveraging advanced deep learning techniques over classical machine learning models for addressing complex biological problems.

Future work will focus on expanding the datasets to cover a broader range of species, further optimizing the model architecture, and conducting experimental validations to enhance reliability. These steps are anticipated to establish Deep-N6mA as a robust and scalable solution for 6 mA site prediction, paving the way for more advanced epigenetic studies [33–35].

Acknowledgements
This work was supported by Research Supporting Project Number (RSPD2025R585), King Saud University, Riyadh, Saudi Arabia

Authors' contributions
All authors contributed equally. SK and IU wrote the main manuscript text, SQ and SN debugged the code and provided datasets, and NA reviewed the paper comments and syntax/grammar.

Funding
This research is not funded.

Data availability
The datasets used and/or analyzed during the current study are available on the GitHub link: https://github.com/salman-khan-mrd/Deep-N6mA.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Conflicts of interests
The authors declare no competing interests.

Received: 22 December 2024 Accepted: 20 March 2025
Published online: 29 March 2025

References

1. Khan S, Khan M, Iqbal N, et al. Deep-piRNA: Bi-Layered Prediction Model for PIWI-Interacting RNA using discriminative features. Comput Mater Contin. 2022;72:2243–2258. https://doi.org/10.32604/cmc.2022.022901.
2. Pashaei E, Aydin N. Markovian encoding models in human splice site recognition using SVM. Comput Biol Chem. 2018;73:159–70. https://doi.org/10.1016/j.compbiolchem.2018.02.005.
3. Pashaei E, Yilmaz A, Ozen M, et al. Prediction of splice site using AdaBoost with a new sequence encoding approach, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 003853–003858. 2016. https://doi.org/10.1109/SMC.2016.7844835.
4. Khan S, Naeem M, Qiyas M. Deep intelligent predictive model for the identification of diabetes. AIMS Math. 2023;8:16446–62. https://doi.org/10.3934/math.2023840.
5. Uddin I, Awan HH, Khalid M, et al. A hybrid residue based sequential encoding mechanism with XGBoost improved ensemble model for identifying 5-hydroxymethylcytosine modifications. Sci Rep. 2024;14:20819. https://doi.org/10.1038/s41598-024-71568-z.
6. Khan S, Khan MA, Khan M, et al. Optimized feature learning for anti-inflammatory peptide prediction using parallel distributed computing. Appl Sci. 2023;13: 7059. https://doi.org/10.3390/app13127059.
7. Li Y, Hu XG, Li PP, et al. Predicting circRNA-disease associations using similarity assessing graph convolution from multi-source information networks, 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE. 2022:94–101. https://doi.org/10.1109/BIBM55620.2022.9995674.

8. Yu H, Dai Z. SNNRice6mA: a deep learning method for predicting DNA N6-methyladenine sites in rice genome. Front Genet. 2019;10:1–6. https://doi.org/10.3389/fgene.2019.01071.
9. Rehman MU, Chong KT. DNA6mA-MINT: DNA-6mA modification identification neural tool. Genes (Basel). 2020;11: 898. https://doi.org/10.3390/genes11080898.
10. Abbas Z, Tayara H, Chong K to (2020) SpineNet-6mA: A novel deep learning tool for predicting DNA N6-Methyladenine Sites in Genomes. IEEE Access 8: 201450–201457. https://doi.org/10.1109/ACCESS.2020.3036090.
11. Abbas Z, Tayara H, Chong KT. ENet-6mA: identification of 6mA modification sites in plant genomes using ElasticNet and neural networks. Int J Mol Sci. 2022;23: 8314. https://doi.org/10.3390/ijms23158314.
12. Li Z, Jiang H, Kong L, et al. Deep6mA: a deep learning framework for exploring similar patterns in DNA N6-methyladenine sites across different species. PLoS Comput Biol. 2021;17:1–15. https://doi.org/10.1371/JOURNAL.PCBI.1008767.
13. Hasan MM, Manavalan B, Shoombuatong W, et al. i6mA-Fuse: improved and robust prediction of DNA 6 mA sites in the Rosaceae genome by fusing multiple feature representation. Plant Mol Biol. 2020;103:225–34. https://doi.org/10.1007/s11103-020-00988-y.
14. Khanal J, Lim DY, Tayara H, et al. i6mA-stack: A stacking ensemble-based computational prediction of DNA N6-methyladenine (6mA) sites in the Rosaceae genome. Genomics. 2021;113:582–92. https://doi.org/10.1016/j.ygeno.2020.09.054.
15. Liu Z-Y, Xing J-F, Chen W, et al. MDR: an integrative DNA N6-methyladenine and N4-methylcytosine modification database for Rosaceae. Hortic Res. 2019;6:78. https://doi.org/10.1038/s41438-019-0160-4.
16. Chou K-CKC. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics. 2005;21:10–9. https://doi.org/10.1093/bioinformatics/bth466.
17. Bin SH, Chou KC. PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem. 2008;373:386–8. https://doi.org/10.1016/j.ab.2007.10.012.
18. Liu B, Wu H, Chou K-C. Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nat Sci. 2017;09:67–91. https://doi.org/10.4236/ns.2017.94007.
19. Chen W, Lin H, Chou KC. Pseudo nucleotide composition or PseKNC: An effective formulation for analyzing genomic sequences. Mol Biosyst. 2015;11:2620–34. https://doi.org/10.1039/c5mb00155b.
20. Khan S, Uddin I, Khan M, et al. Sequence based model using deep neural network and hybrid features for identification of 5-hydroxymethylcytosine modification. Sci Rep. 2024;14:9116. https://doi.org/10.1038/s41598-024-59777-y.
21. Ma J, Sheridan RP, Liaw A, et al. Deep neural nets as a method for quantitative structure-activity relationships. J Chem Inf Model. 2015;55:263–74. https://doi.org/10.1021/ci500747n.
22. Zhu Z, Albadawy E, Saha A, et al. Deep learning for identifying radiogenomic associations in breast cancer. Comput Biol Med. 2019;109:85–90. https://doi.org/10.1016/j.compbiomed.2019.04.018.
23. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60:84–90. https://doi.org/10.1145/3065386.
24. Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag. 2012;29:82–97. https://doi.org/10.1109/MSP.2012.2205597.
25. Bordes A, Chopra S, Weston J, et al. Question Answering with Subgraph Embeddings, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Stroudsburg, PA, USA, Association for Computational Linguistics. 2014:615–620. https://doi.org/10.3115/v1/D14-1067.
26. Khan S, Noor S, Awan HH, Iqbal S, AlQahtani SA, Dilshad N, et al. DeepProBind: binding protein prediction with transformer-based deep learning model. BMC Bioinformatics. 2025;26:88.
27. Khan F, Khan M, Iqbal N, et al. Prediction of recombination spots using novel hybrid feature extraction method via deep learning approach. Front Genet. 2020;11:1052. https://doi.org/10.3389/fgene.2020.539227.
28. Inayat N, Khan M, Iqbal N, et al. iEnhancer-DHF: identification of enhancers and their strengths using optimize deep neural network with multiple features extraction methods. IEEE Access. 2021;9:40783–96. https://doi.org/10.1109/ACCESS.2021.3062291.
29. Ahmad W, Ahmad A, Iqbal A, et al. Intelligent hepatitis diagnosis using adaptive neuro-fuzzy inference system and information gain method. Soft Comput. 2019;23:10931–8. https://doi.org/10.1007/s00500-018-3643-6.
30. Chen W, Lv H, Nie F, et al. i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome. Bioinformatics. 2019;35:2796–800. https://doi.org/10.1093/bioinformatics/btz015.
31. Vacic V, Iakoucheva LM, Radivojac P. Two Sample Logo: A graphical representation of the differences between two sets of sequence alignments. Bioinformatics. 2006. https://doi.org/10.1093/bioinformatics/btl151.
32. Zhou G-P, Chen D, Liao S, et al. Recent progresses in studying helix-helix interactions in proteins by incorporating the Wenxiang diagram into the NMR spectroscopy. Curr Top Med Chem. 2015;16:581–90. https://doi.org/10.2174/1568026615666150819104617.
33. Noor S, AlQahtani SA, Khan S. Chronic liver disease detection using ranking and projection-based feature optimization with deep learning. AIMS Bioeng. 2025;12:50–68. https://doi.org/10.3934/bioeng.2025003.
34. Noor S, Naseem A, Awan HH, et al. Deep-m5U: a deep learning-based approach for RNA 5-methyluridine modification prediction using optimized feature integration. BMC Bioinformatics. 2024;25:360. https://doi.org/10.1186/s12859-024-05978-1.
35. Khan S, Noor S, Javed T, et al. XGBoost-enhanced ensemble model using discriminative hybrid features for the prediction of sumoylation sites. BioData Min. 2025;18:12. https://doi.org/10.1186/s13040-024-00415-8.

Publisher's Note