IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 7, 2020, accepted August 16, 2020, date of publication September 22, 2020, date of current version October 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025990

A Deep Learning Model for Predicting DNA N6-Methyladenine (6mA) Sites in Eukaryotes

LOKUTHOTA HEWAGE ROLAND AND CHAMPI THUSANGI WANNIGE
Department of Computer Science, University of Ruhuna, Matara 81000, Sri Lanka
Corresponding author: Champi Thusangi Wannige (wannige@dcs.ruh.ac.lk)

ABSTRACT DNA N6-methyladenine (6mA) is an epigenetic modification, which is involved in many biological regulation processes like DNA replication, DNA repair, transcription, and gene expression regulation. The widespread presence of this 6mA modification in *eukaryotes* has been unclear until recently. Studying the genome-wide distribution of 6mA can provide a deeper understanding of the epigenetic modification process and the biological processes it involves. Existing experimental techniques are time-consuming and computational machine learning methods have room for performance improvement. DNA N6-methyladenine prediction in eukaryotic cross-species shows low performance. Hence, there is a need for a more accurate, time-efficient method to predict the distribution of 6mA sites in *eukaryotes*. Since deep learning architectures have shown higher accuracy, we develop a customized VGG16 architecture-based model named 6mAVGG using convolution neural networks for the prediction of DNA 6mA sites in eukaryotes. We introduce a novel 3-dimensional encoding mechanism extending the one-hot encoding method to support the input of the VGG16 model. Specifically, the 10-fold cross-validation on the benchmark datasets for the proposed model achieves higher accuracies of 98.01%, 97.44%, 99.56% respectively for cross-species, Rice, and M. musculus genomes. The proposed model outperforms existing tools for the prediction of 6mA sites and has enhanced accuracies by 2.88%, 4.2%, 0.9% respectively for cross-species, *Rice*, and *M. musculus* genomes compared to the state of the art method SNNRice6mA. The model trained with benchmark data predicts 6mA sites of other species *ArabidopsisThaliana*, *RosaChinensis*, *Drosophila*, and *Yeast* with prediction accuracy over 70%. Thus, this model can be used for the genome-wide prediction of 6mA sites in *eukaryotes*.

INDEX TERMS DNA N6-methyladenine, sequence analysis, deep learning, eukaryotes, DNA sequence encoding method.

I. INTRODUCTION

Epigenetics is the study of chemical modifications to DNA that change the way genes are expressed without altering the underlying genetic sequence [1], [76]. DNA methylation is a type of epigenetic modification that results in unexpected activation or repression of genes [76]. DNA methylation controls gene expression by causing changes in the chromatin structure, DNA stability, DNA-protein interactions and DNA conformation [70], [10]–[13], [16], [17]. The most common types of DNA methylation modifications are N4-methylcytosine (4mC), 5-methylcytosine (5mC), and N6-methyladenine (6mA) that have been found in both prokaryotic and eukaryotic genomes [2], [71]. One of the most abundant DNA methylations in eukaryotes is 6mA [3], [14], [15] which is involved in biological processes like DNA replication [4], DNA repair [5], [8], transcription [6], [9], and gene expression regulation [7]. The prevalence and significance of this 6mA modification in eukaryotes have been unclear until 2015 [67]. In 2015 it was proposed that 6mA as a new epigenetic mark in eukaryotes [67], opening the field to investigate approaches to precisely identify these 6mA sites. Therefore, for eukaryotes, the study of DNA 6mA is still in its early stages [68]. Understanding the epigenetic modification process of DNA N6-methyladenine (6mA) in eukaryotes and the biological processes it involves in eukaryotes is a problem due to the lack of research in this area [67]. Finding the genome-wide distribution of DNA 6mA sites can provide a good understanding of the epigenetic modification process and the biological processes it involves.

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Hugo Albuquerque.

Though 6mA is found as a new epigenetic mark in eukaryotes [67], the field is frequently investigating highly accurate, time-efficient experimental and computational approaches to precisely identify these 6mA sites [22]–[27], [30], [31].

6mA sites in eukaryotes can be detected by highly sensitive experimental techniques. These experimental techniques contribute to studying DNA 6mA modification sites including immunoprecipitation [18], ultra-high-performance liquid chromatography with mass spectrometry [21], capillary electrophoresis and laser-induced fluorescence (CE-LIF) [19], single-molecule real-time sequencing (SMRT) [20]. Although many experimental techniques [18]–[21] have been proposed to identify 6mA sites, these techniques are labor-intensive, complex, time-consuming, and expensive [30].

To solve the time, cost, and complexity problems of finding DNA 6mA sites, computational methods have been introduced. Several machine learning methods contribute for studying and predicting DNA 6mA modification sites including iDNA6mA-PseKNC [22], i6mA-Pred [23] based on support vector machines (SVM). The iDNA6mA-PseKNC was the first computational tool established for identifying 6mA sites in DNA sequences. Although the SVM based tool which uses a manual feature extraction method was trained by using only the benchmark dataset of M.musculus genome, it has high success rates for identifying 6mA sites in many other species. The SVM based tool i6mA-Pred was developed to identify 6mA sites in rice genome. By using a manual feature extraction method i6mA-Pred achieved an accuracy of 83.13% on the rice genome dataset. MM-6mAPred [24] based on the Markov model outperformed i6mA-Pred in predicting 6mA sites of the rice genome. However, MM-6mAPred uses a manual feature extraction method which only achieved an accuracy of 89.72% on the rice genome. iDNA6mA [25] based on convolutional neural networks uses automatic feature extraction to predict 6mA sites in rice genome. However, it achieved only an accuracy of 86.64% on the rice benchmark dataset. SDM6A [26] is based on an ensemble approach using several features encoding methods and machine learning classifiers. SDM6A explored various manual feature extraction methods to generate an optimal feature set. However, it achieved only an accuracy of 88.2% on the rice genome dataset. iDNA6mA-rice [27] based on random forest algorithm outperformed i6mA-Pred in predicting 6mA sites in rice genome. However, it also uses a manual feature extraction method. 6mA-RicePred [72] employs a feature fusion method to combine advantageous features from other methods and obtained a new feature to identify 6mA sites. However, it only achieved an accuracy of 87.27% for the prediction of 6mA sites in rice genome with 10-fold cross validation. 6mA-Finder [73] incorporates seven features derived from sequence and three physicochemical-based features through recursive feature elimination (RFE) strategy. 6mA-Finder achieved high sensitivity, with an improvement of 37.89% AUC value compared to iDNA6mA-Rice and SDM6A using the rice benchmark data. However, the accuracy was low. p6mA [74] performed maximum relevance maximum distance (MRMD) analysis to select key features and used the Extreme Gradient Boosting (XGBoost) algorithm. P6mA performed better than iDNA6mA-PseKNC and MM-6mAPred. However, there is room for performance improvement. Although, several machine learning methods [22]–[27], [72]–[74] have been developed to identify 6mA sites genome-wide, they were validated separately for the benchmark datasets of the rice genome and mice genome. Cross-Species evaluation for existing methods shows low performance [30].

Recent studies have shown that deep learning is a powerful technique [33]–[37] for sequence analysis and classification in bioinformatics [28], [29], [38]–[42]. Based on Convolutional Neural Networks (CNN), Yu and Dai (2019) proposed a method named SNNRice6mA [30] to predict the 6mA sites of rice [23] and showed its advantages over other methods [30]. They tested whether model trained on rice datasets can be used to predict DNA 6mA sites in other species. They used the M. musculus 6mA dataset proposed in a previous study [22] and denoted this dataset as 6mA-mouse-Feng. The 6mA-mouse-Feng dataset contains 1,934 6mA site containing sequences and 1,934 non-6mA site containing sequences. They used this independent dataset as test data. They evaluated the performance of SNNRice6mA, which was trained on the rice 6mA-rice-Lv dataset [30], on the M. musculus test data. They found that SNNRice6mA achieved predicted accuracy of 61.81%, which was higher than those of the other three methods (52.43% for i6mA-Pred, 41.93% for iDNA6mA, 44.11% for MM-6mAPred). However, there was room for improvement in the performance of SNNRice6mA on the M. musculus dataset. In this regard, a novel deep learning-based computational model named iIM-CNN [31] using convolutional neural networks (CNN) was proposed for the identification of N6-methyladenine sites from DNA sequences [31]. The proposed model achieved the Mathew correlation coefficient (MCC) of 0.651, 0.752, and 0.941 for cross-species, Rice, and M. musculus genome respectively. The comparison of the outcomes of this model showed that their new model outperformed the existing computational tools for the prediction of the 6mA sites.

From the previous literature [22]–[27], [30], [31], it is clear that, there is room for improvement in the performance for cross-species of eukaryotes. Further, previous studies have used Rice and M. musculus genomes for building their cross-species trained model and the trained model has shown low accuracies in predicting DNA 6mA sites in other eukaryotic species. There are other eukaryotic genomes such as *ArabidopsisThaliana*, *RosaChinensis*, *Drosophila*, and *Yeast* that are available for studies of DNA 6mA sites of eukaryotes [53]. The objective of this research is to find a more accurate, time-efficient method in order to predict the presence of DNA 6mA sites in eukaryotic genomes.

Deep learning models have shown good performance in creating accurate bioinformatics predictors in the field of bioinformatics [38]–[42]. To develop an accurate deep learning model to identify DNA 6mA sites in eukaryotes,

TABLE 1. Summary of datasets used for training.

Species	Dataset	Number of Samples
Cross-species	Positive	2768
	Negative	2716
Rice	Positive	880
	Negative	880
M. musculus	Positive	1934
	Negative	1934

we experimented with several robust CNN Architectures [69], [78] which claimed higher accuracies. This model needs no prior knowledge of 6mA or manually crafted sequence features. We build the model named 6mAVGG based on rice, mice, cross-species benchmark datasets, and independently validate the model based on *ArabidopsisThaliana, RosaChinensis, Drosophila*, and *Yeast* 6mA datasets. This method gets an average prediction accuracy higher than the existing methods on the benchmark datasets [23], [22], [29]. A comparison with existing 6mA prediction tools shows that this model outperforms the existing high-performance methods [30], [31].

The main contributions of this research work are as follows.

- A high accurate deep learning based model for 6mA site prediction in eukaryotic genomes that has enhanced accuracies by 2.88%, 4.2%, 0.9% respectively for cross-species, Rice, and M. musculus genomes compared to the state of the art method SNNRice6mA.
- A novel approach to encoding DNA sequences to use with Convolutional Neural Network Architectures. This novel approach introduces a 3D representation of an input DNA sequence to be used with CNNs which was not available in previous studies.

This paper is divided into sections as follows. Section II discusses, the approach followed in terms of benchmark datasets used, sequence representation technique, model creation approach, and the techniques used in model evaluation. Section III demonstrates and discusses the performance of the proposed model. Section IV discusses the validation techniques followed and demonstrates the results of the evaluation. Finally, section V discusses the potential success and limitations of the proposed model for identifying DNA 6mA sites in eukaryotes.

II. METHODOLOGY

To develop a useful bioinformatics predictor, a series of recent publications [43], [44]–[50] and review papers [51], [52] demonstrate that one needs to follow Chou's 5-steps rule [51] that includes the following five steps: (1) construction of a gold standard dataset to train and test the model; (2) encoding samples with effective formulations; (3) conducting the prediction model with a powerful classifier; (4) evaluating model performance by using cross-validation tests and standard measures; (5) establishing a user-friendly tool for the predictor that can be accessible to the public. A new bioinformatics predictor presented according to the five-step rules [51] would have the following advantages: (i) clear logic deduction; (ii) better demonstration in stimulating other relevant tools; (iii) useful in practical application [56]. We followed this, five-step procedures to develop our new prediction method.

A. BENCHMARK DATASETS

MethSMRT [53] is an integrative database containing DNA 6mA and 4mC site information generated from SMRT sequencing [20] and it is commonly used by researchers in 6mA site prediction research. In this study, we used the 6mA benchmark datasets of the rice genome [23] and the M. musculus genome [22]. The cross-species dataset [56] which we use in this study was constructed by integrating these two benchmark datasets [31]. Then for reducing the sequence redundancy in the dataset, the threshold value 0.8 was set using CD-HIT software [57]. The benchmark datasets were downloaded from https://github.com/NWAFU-LiuLab/csDMA which were also used in other studies [30], [31] on DNA 6mA site prediction. The benchmark datasets contain DNA sequences in positive and negative samples which are 41bp long. The benchmark dataset of the rice genome contains 880 positive samples and 880 negative samples [23]. Altogether, it contains 1760 sequences. The benchmark dataset of M. musculus genome contains 1934 positive samples and 1934 negative samples [22]. It contains 3868 sequences. The cross-species dataset contains 2768 positive samples and 2716 negative samples [56]. It contains 5484 sequences. Datasets used for training are summarized in table 1. In all the benchmark datasets, the positive samples contain a 6mA modification site at the center and the negative samples collected from NCBI [83] contain no 6mA modification site. To demonstrate that this method can be used to detect the 6mA sites of other eukaryotes, the DNA 6mA sequences of Arabidopsis thaliana, Rosa chinensis, Drosophila, and Yeast were collected from https://github.com/lwzyb/SICD6mA [77].

To build a model with more generalized performance, a sensible data splitting strategy is needed for model validation [75]. Data size has an impact on generalization performance [75]. There is a significant gap between performance from the validation set and test set for all data splitting methods applied on small data sets [75]. This gap can be decreased when more samples are available for training and validation [75]. Model performance has a negative effect when the training set has too many or too few samples [75]. There should be a good balance between the sizes of training and validation set for a reliable estimation of model performance [65]. So the benchmark dataset was split as 80% training data (8 folds), 10% validation data (1 fold), and 10% testing data using K-fold cross-validation after shuffling the positive and negative datasets [30], [87]–[89]. The model was validated

using both independent dataset validation test and K-fold cross-validation test [58], [59].

B. SEQUENCE REPRESENTATION

The samples in our dataset are DNA sequences of length 41bp and we choose the VGG16 model for this study since the idea of much deeper networks and with much smaller filters tend to be more powerful [78]. VGG16 is based on CNN, which has been successful in image classification tasks [78]. Normally, the input of a VGG16 takes tensors of shape (image_height, image_width, color_channels), ignoring the batch size. The color_channels refers to (R, G, B).

The input size of the customized VGG16 based model we use here is (48, 48, 3) which experimentally gave us the highest performance (see Table 5). This is also the minimum input size accepted by the VGG architecture without performance degradation as mentioned in previous work [89]. The DNA sequences must be converted to the vector form of the above-mentioned dimensions before using it with the model. Instead of using manually extracted features, we propose a novel extended one-hot encoding method to convert the input DNA sequences into encoding tensors. Normally, each nucleotide A, C, G, T, and N are encoded as (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1) and (0,0,0,0) respectively in one-hot encoding method [30]. However, here we extend this approach to represent a DNA sequence as a 3-dimensional structure.

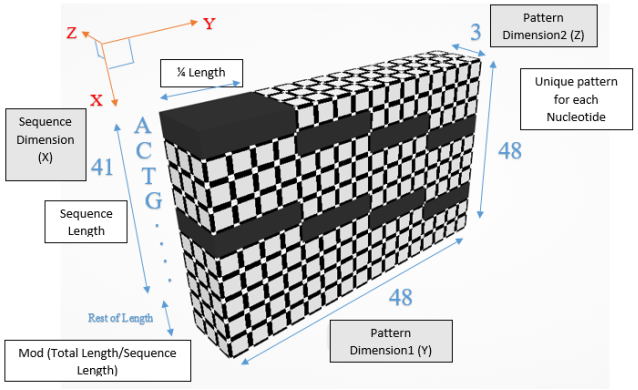

FIGURE 1. Example sequence representation: The DNA sequence with length 41bp is represented as an image with 3 dimensions. The black blocks indicate regions encoded with 1 while the gray blocks indicate regions encoded with 0. The sequence dimension repeats the 41bp length sequence only once to cover the maximum length of the dimension and the rest of the dimension is encoded with zero. pattern dimension 1 represent a unique pattern for each nucleotide A, C, T, G and pattern dimension 2 repeat the pattern in pattern dimension 1.

In image classification, frequently 3D representations are used for image representation [78], [80]–[82]. We name the 3 dimensions of the structure as X, Y, and Z as shown in figure 1. Each nucleotide A, C, G, T are represented as a 2D array along Y and Z dimensions as shown in figure 1. Here, dimension X represents the nucleotide sequence of length 41 nucleotides. Since the length along X-axis is 48, the rest of the length along X is encoded with 0 which is a padding mechanism commonly used in Convolutional Neural Networks (CNN) [79]. The other two dimensions (Y and Z) encode each nucleotide into a 2D array representing a unique pattern for each nucleotide A, C, T, G. In this case, it is converted into a 2D array of size 48 by 3. Here the dimension Y is divided into 4 equal parts each of length 12 and for nucleotide A, the first quarter of the dimension is encoded with 1 (see black colored boxes in Figure 1) while the rest of the length is encoded with 0. For nucleotide C, the second quarter of the dimension is encoded with 1 while the rest of the length is encoded with 0. For nucleotide G, the third quarter of the dimension is encoded with 1 while the rest of the length is encoded with 0. For nucleotide T, the final quarter of the dimension is encoded with 1 while the rest of the length is encoded with 0. For the lengths along the dimension X, that doesn't represent any nucleotide the other two dimensions (Y and Z) are encoded with 0 in a 2D array which is a padding mechanism commonly used in CNN [79]. Here we use this proposed method for representing each input sequence as a 3D array of size $48 \times 48 \times 3$ to use with the VGG16 model.

C. MODEL CREATION

Convolution Neural Network architectures have shown great success in solving complex problems in the field of image classification [78], [80]–[82]. There are several high performing architectures proposed in recent past for classification problems such as LeNet [80], AlexNet [81], VGGNet [78], GoogLeNet [82]. We choose the VGGNet architecture for this problem since the idea of much deeper networks with much smaller filters tend to be more powerful in terms of accuracy [78] and it has a simple uniform structure serially stacked which is easy to understand and which has performed very well in image classification tasks [78].

After selecting VGGNet architecture, we evaluated the impact on the results using different types of VGGNet architectures proposed under the standard VGGNet architecture. We choose the best performing two proposed architectures under the standard VGGNet architectures [78] which are VGG16 [78] and VGG19 [78] architectures. We compare these two architectures in terms of five metrics accuracy, sensitivity, specificity, MCC, and AUC. Both models were trained based on the 6mA-rice-Chen dataset [23] with 5 fold cross-validation. The results (see supplementary table 1) show that the performance of the customized VGG16 based architecture is high compared to the customized VGG19 based architecture in terms of all metrics. This proves the fact that VGG16 has a lower error rate compared to VGG19 and the addition of layers makes the model converge slowing down the accuracy improvement as stated in the original paper [78]. Therefore, the customized VGG16 model was chosen as the final 6mAVGG model.

The model parameters were tuned to achieve higher accuracy by carrying out number of experiments. The selected parameters are given in Table 2 and the reasons for their choice is discussed in Section II C. The results are discussed

in section III C. The model was trained based on the rice genome benchmark dataset [23], mouse genome benchmark dataset [22], and finally the cross-species genome dataset [56].

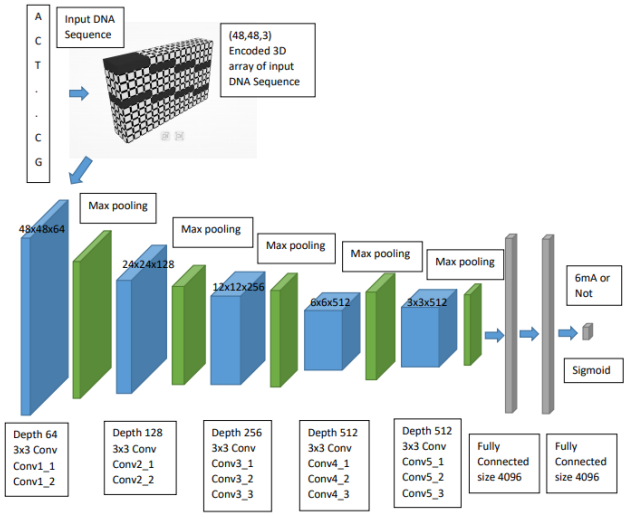

FIGURE 2. Architecture of the proposed model: Includes layers of the standard VGG16 architecture consisting of convolutional layers and max pooling layers for an input dimension (48,48,3) and a final sigmoid layer that predicts 6mA or not.

Adapting the VGG16 based standard convolutional neural network architecture [78] as shown in figure 2, 16 layers were defined. The layers consist of convolutional layers, max-pooling layers, activation layers, and fully-connected layers. It consists of 13 convolutional layers, 5 Max pooling layers, and 3 Dense layers which sum up to 21 layers. However, it consists only of 16 weight layers including the convolutional layers and dense layers as max-pooling layers don't have weight parameters. Convolution layer 1 has 64 filters while Convolution layer 2 has 128 filters, Convolution layer 3 has 256 filters while Convolution layer 4 and 5 have 512 filters in the standard VGG16 based model.

The model was defined using TensorFlow [86] Keras Sequential API [55]. The sequential API, groups the linear stack of layers into the TensorFlow Keras model. The convolution layers create a convolution kernel that is convolved with the layer input to produce a tensor of outputs. Each filter in the convolutional layers preserves or detect important features from the DNA sequence and pass to corresponding filters in the corresponding layers, identifying sequence motifs of 6mA sites from non 6mA sites. The argument filters in each convolution layer define the dimensionality of the output space and they were defined following the standard VGG16 architecture [78].

The kernel_size (3, 3) specifies the height and width of the 2D convolution window that filters out sequence patterns as a small 2D image. In this study, we use the Rectified Linear Unit (ReLu) activation function in this network since it is a simple, fast function that empirically seems to work well [84]. Early researchers observed that training a deep neural network with ReLu tended to converge much more quickly and reliably than training a deep network with sigmoid activation [84]. The max-pooling layers downsample the input sequence pattern representation captured by taking the maximum value over the window defined by pool_size for each dimension along the features axis. The window is shifted by strides in each dimension. The pool size (2, 2) takes the max value over a 2×2 pooling window. The strides (2, 2) specifies how far the pooling window moves for each pooling step. After defining the convolutional layers and the max-pooling layers with the above parameters, based on the standard VGG architecture, next, we define a flatten layer and three dense layers.

The flatten layer flattens the input. The dense layers define a regular densely-connected neural network layer. We define the dimensionality of the output shape in this study as 4096 units, according to the standard VGG16 architecture. We use the sigmoid activation function [54] for the final layer. The final dense layer defines the sigmoid activation function because it exists between (0 to 1) and it is used for models where we have to predict the probability as an output [54]. In this research, the output of the model is the probability of being a 6mA site. For probabilities greater than 0.5, the model will identify the input sequence as a 6mA site with the 6mA site present on the center of the sequence, and if it's less than 0.5, the model will identify the site as a non 6mA site.

In this study, the gradient descent (with momentum) (SGD) optimizer [66] is used to compile the model as it converges better with longer training time. After evaluating performance on different learning rates ranging from 0.90 to 0.95 using grid search with a step size of 0.05, we define momentum as 0.95 in this study that accelerates gradient descent in the relevant direction and dampens oscillations. After evaluating accuracy on different learning rates ranging from 0.01 to 0.0001 using grid search with a step size of multiple of 10, we choose 0.001 as the learning rate of the model. We also evaluated the accuracy for the learning rate 0.005. We choose BinaryCrossentropy [65] as the loss value that computes the cross-entropy loss between true labels and predicted labels. The metric accuracy is used to calculate how often predictions equal labels. To train the model we have used two callbacks: ModelCheckpoint and EarlyStopping [55]. The model was trained for 100 epochs and batch size was defined as 32. Early stopping will stop the training process when the prediction accuracy stops improving on the validation set. Early stopping is defined as 30 epochs so that the training is stopped when the prediction accuracy on the validation set does not improve after 30 training epochs. This value was chosen based on the observation of the behavior of validation loss. As the change of loss is slow, a higher value of this parameter was chosen. The model checkpoint technique saves the model which has the highest prediction accuracy on the validation set. During the training process, the learning rate is reduced when the value of loss function on the validation set no longer decreased. We define the reduced factor as 0.1 and the patience as 20 epochs so that the learning rate is reduced

TABLE 2. Hyper parameter preferences.

Parameters	Range
Convolution Layers	13
Filters in each block of Convolutional Layers	[64, 128, 256, 512, 512]
Size of the filters	3
Maxpooling pool size	2
Maxpooling stride length	2
Dense units	[1, 4096]
learning rate	[0.01,0.001,0.005]
momentum	[0.90, 0.95]

when the value of loss function on the validation set does not improve after 20 training epochs.

The model was implemented using Python and TensorFlow Keras [55]. The model was run on Google Colab [85] using GPU. The Matplotlib library was used for visualization. Hyperparameter preferences are shown in Table 2.

D. PERFORMANCE EVALUATION

According to previous studies [30]–[32], to evaluate the quality of a new predictor or its performance, we need to consider the following two problems. First, the metrics that should be used to measure the predictor's quality need to be decided. Secondly, the method that should be adopted to calculate the metrics needs to be decided [32].

For the evaluation metrics, we used the five metrics. The metrics used were, including accuracy, sensitivity, specificity, Matthews correlation coefficient (MCC), and area under the curve (AUC). Accuracy is used as it is the most commonly used measure for classification performance [30], [31], [64] defined as a ratio between the correctly classified samples to the total number of samples. Sensitivity and specificity are used as they are commonly used for evaluating classification performance with imbalanced data [64]. MCC is used as it represents the correlation between the observed and predicted classifications [64]. AUC is used as it is a significant measure to calculate the success rate of the prediction model [64].

The metric accuracy is a measure of the ratio of correct predictions out of the total number of predictions. The accuracy is defined as in equation E 01 [32], [64]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(E 01)}$$

True positive (TP) is the number of predictions that classify the actual positive samples correctly. True negative (TN) is the number of predictions that classify the actual negative samples correctly. False-positive (FP) calculates the number of predictions that misclassify the negative ones as the positive ones. False-negative (FN) calculates the number of predictions that misclassify the positive ones as the negative ones. The positive means the samples containing the 6mA sites, and vice versa.

The metric sensitivity indicates the ratio of correctly identified positive samples out of the whole actual positive data. The sensitivity is defined as in equation E 02 [32], [64]:

$$sensitivity = \frac{TP}{TP + FN} \quad \text{(E 02)}$$

The metric specificity is the ratio of correctly identified negative samples out of the whole actual negative data. The specificity is defined as in equation E 03 [32], [64]:

$$specificity = \frac{TN}{TN + FP} \quad \text{(E 03)}$$

MCC measures the quality of the binary classification model [60]. MCC considers TP, TN, FP, and FN. MCC is considered as a balanced measure that can be used with imbalanced data [61].

It is a measure of the correlation between the actual and predicted binary classifications. MCC has a coefficient value ranging between -1 and $+1$, where $+1$ represents the binary classification model as perfect, 0 represents the prediction is random, and -1 represents the binary classification model as poor. MCC is defined as in equation E 04 [32], [64]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad \text{(E 04)}$$

AUC stands for the area under the receiver operating characteristic curve [64]. AUC is used to measure the success of a binary classification model. Higher the AUC value, the better the performance of the classification model. It's value ranges between 0 and 1. The AUC score 1 indicates the model is perfect in prediction, while 0.5 means no discrimination which is similar to randomly flipping a coin. The value 0 indicates the classifier is perfectly incorrect.

To be consistent with the previous studies [30], [31], the following two different validation methods are used to examine a predictor's performance [62], [75]: (i) independent dataset test, (ii) subsampling (or K-fold cross-validation) test. For evaluating the performance of the proposed model, we use the 10-fold cross-validation method, that have empirically yield test error estimates that neither suffers from high bias nor high variance [63], [87], [88], [90]. Each subset is iteratively chosen as a test set and validation set in a separate cross-validation fold, while the remaining eight subsets will be used for the training of the model. The average results of the ten trials will be finally used as the performance estimation of the proposed model. To avoid overfitting, we used the early stopping technique discussed in section II C during the model training process and the K-fold cross-validation technique discussed here.

E. BIOINFORMATIC PREDICTOR

The implementation code of this research work and a publically accessible version of this tool named 6mAVGG is currently available at https://github.com/RolandHewage/6mAVGG.

TABLE 3. Performance comparison on rice 6mA benchmark dataset.

Method	Sensitivity	Specificity	Accuracy	MCC	AUC
6mAVGG	0.975	0.9739	0.9744	0.9493	0.9924
SNNRice6mA	0.9216	0.9432	0.9324	0.87	0.97
6mA-RicePred	0.9597	0.7533	0.8565	0.73	0.945
i6mA-Pred	0.8295	0.833	0.8313	0.66	0.89
MM-6mAPred	0.8932	0.9011	0.8972	0.79	No
iDNA6mA	0.867	0.8659	0.8664	0.73	0.93
SDM6A	0.852	0.909	0.881	0.76	0.94
iDNA6mA-rice	0.8386	0.8341	0.8363	0.67	0.91

III. RESULTS

A. PERFORMANCE COMPARISON ON RICE 6mA BENCHMARK DATASET

From our literature review, we found that there are seven existing tools including SNNRice-6mA (Yu *et al.*, 2019) [30], i6mA-Pred (Chen *et al.*, 2019) [23], SDM6A (Basith *et al.*, 2019) [26], iDNA6mA (Tahir *et al.*, 2019) [25], MM-6mAPred (Pian *et al.*, 2019) [24], iDNA6mA-rice (Lv *et al.*, 2019) [27], and 6mA-RicePred (Huang *et al.*, 2020) [72] built based on the 6mA sites data in the rice genome, which could predict the 6mA sites in the rice genome. iDNA6mA-PseKNC [22] is a tool built based on the M. musculus dataset and can be applied in many other species (Feng *et al.*, 2019). Zhiming *et al.*, 2019 examined whether iDNA6mA-PseKNC can predict 6mA sites in the rice genome and found that the error rate of iDNA6mA-PseKNC prediction is relatively high (∼58%). In this regard, we compared our method 6mAVGG with the existing tools SNNRice6mA, i6mA-Pred, SDM6A, iDNA6mA, MM-6mAPred, iDNA6mA-rice, and 6mA-RicePred. For a fair comparison, the performance was compared based on the five metrics accuracy, sensitivity, specificity, MCC, and AUC which are commonly used in many studies [30], [31], [64]. The performance results in Table 3 are based on the 6mA-rice-Chen dataset [23] of SNNRice6mA, i6mA-Pred, SDM6A, iDNA6mA, MM-6mAPred, iDNA6mA-rice and 6mA-RicePred directly quoted from previous studies (Zhiming *et al.*, 2019, Basith *et al.*, 2019; Chen *et al.*, 2019; Lv *et al.*, 2019; Pian *et al.*, 2019; Tahir *et al.*, 2019, Huang *et al.*, 2020). The comparison showed that our model 6mAVGG trained based on the 6mA-rice-Chen dataset outperformed SNNRice6mA, i6mA-Pred, SDM6A, iDNA6mA, MM-6mAPred, and iDNA6mA-rice in terms of the five metrics accuracy, sensitivity, specificity, MCC and AUC. This is because our proposed model holds deeper layers, it was able to detect important features from the DNA sequence and pass to corresponding filters in the corresponding layers, identifying sequence motifs of 6mA sites from non 6mA sites. This justifies the theoretical concept that deeper networks with much smaller filters tend to be more powerful in terms of predictive power [78].

B. PERFORMANCE COMPARISON ON THE CROSS-SPECIES DATASET

According to our literary survey, there are three existing tools including iIM-CNN [31], csDMA [56], and iDNA6mA-PseKNC [22], which could predict the 6mA sites in cross-species. Wahab *et al.* 2019 [31] compared the performance of these three tools on the same datasets. The performance was compared on the models trained with 6mA sites data in the rice genome [23], M. musculus genome [22], and the cross-species dataset [56] built by integrating the rice and M. musculus datasets. Recently (2020), 6mA-Finder [73], p6mA [74], SICD6mA [77] were developed to predict 6mA sites in cross-species. The performance results based on the datasets of rice, M. musculus, and the cross-species for iIM-CNN, csDMA and iDNA6mA-PseKNC were directly quoted from the previous study by Wahab *et al.* 2019 [31]. In addition, to examine whether SNNRice-6mA (Zhiming *et al.*, 2019) can predict 6mA sites across eukaryotic species with high accuracy we measured the performance results of SNNRice-6mA on the above datasets. The comparison in Table 4 shows that 6mAVGG trained based on the 6mA-rice-Chen dataset, M. musculus dataset, and the cross-species dataset outperformed SNNRice-6mA, iIM-CNN, csDMA, and iDNA6mA-PseKNC in terms of the five metrics accuracy, sensitivity, specificity, MCC and AUC. These results show that 6mAVGG outperforms the state-of-the-art predictor iIM-CNN in terms of all five metrics. The graphical illustration of the performance of each model on the cross-species dataset is shown in figure 3. The accuracy comparison of our proposed model 6mAVGG is illustrated in figure 4.

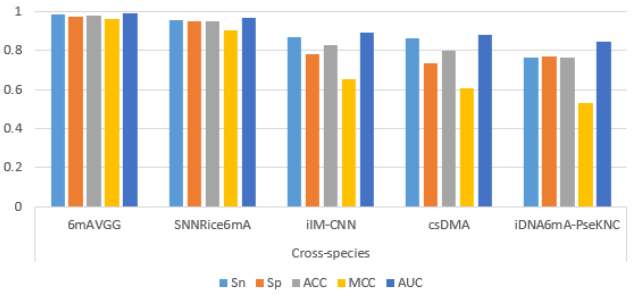

FIGURE 3. Performance comparison on cross species dataset.

6mA-Finder [73] was not compared in this study as it has evaluated only the AUC score. However, the reported AUC score of 0.9207 is less than 0.9801 of this study. P6mA [74] was also difficult to be compared in this study as it was trained using four different species: Oryza Sative (Rice), Drosophila melanogaster (fruit fly), Caenorhabditis Elegans (worm) and Homo sapiens (human). The highest reported AUC score was 0.8871 which is less than 0.9801 of this study. SICD6mA [77] was also not compared in this study as it was trained on human and rice dataset. However, the recorded AUC score of SICD6MA for the rice dataset was 0.9903 which is less than 0.9924 of this study. Our model outperforms the performance metrics: sensitivity, specificity, accuracy,

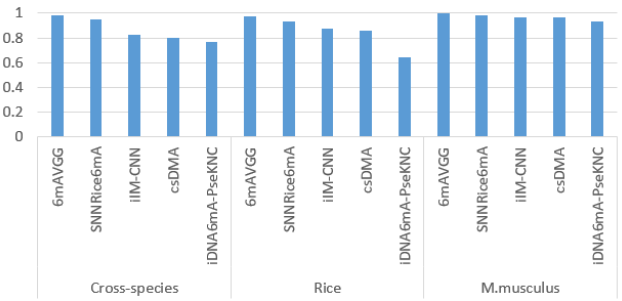

FIGURE 4. Accuracy comparison of 6mAVGG with SNNRice6mA, iIM-CNN, csDMA, iDNA6mA-PseKNC on benchmark dataset of different species.

TABLE 4. Performance comparison on benchmark datasets.

Species	Models	Sn	Sp	ACC	MCC	AUC
Cross-species	6mAVGG	0.9856	0.9746	0.9801	0.9604	0.9871
	SNNRice6mA	0.9545	0.9482	0.9513	0.9037	0.9686
	iIM-CNN	0.869	0.78	0.824	0.651	0.892
	csDMA	0.863	0.735	0.799	0.603	0.879
	iDNA6mA-PseKNC	0.762	0.769	0.765	0.531	0.844
Rice	6mAVGG	0.975	0.9739	0.9744	0.9493	0.9924
	SNNRice6mA	0.9216	0.9432	0.9324	0.87	0.97
	iIM-CNN	0.841	0.914	0.875	0.752	0.934
	csDMA	0.842	0.88	0.861	0.723	0.923
	iDNA6mA-PseKNC	0.569	0.721	0.641	0.394	0.896
M.musculus	6mAVGG	0.9938	0.9974	0.9956	0.9912	0.9972
	SNNRice6mA	0.9757	0.9974	0.9866	0.9734	0.9908
	iIM-CNN	0.938	1	0.969	0.941	0.971
	csDMA	0.932	1	0.966	0.935	0.974
	iDNA6mA-PseKNC	0.869	1	0.935	0.877	0.974

and MCC for rice with the values of 0.975, 0.9739, 0.9744, 0.9493 which are greater than the highest values recorded for each metric 0.9697, 0.95, 0.9599, 0.9199 respectively by SICD6mA. Although SICD6mA shows some success in predicting 6mA sites across eukaryotic species, the prediction accuracy for the dataset of yeast and tolypocladium showed values of 0.5706 and 0.4920 respectively which are less than 0.7657 and 0.7351 of our model trained on M. musculus dataset.

C. PERFORMANCE IMPACT BY CHANGING INPUT DIMENSION OF VGG16 ARCHITECTURE

In this section, we evaluated the impact of changing the input dimension of the customized VGG16 based model thereby testing the input shapes: (224, 224, 3), (128, 128, 3), (48, 48, 3) were used to build each customized VGG16 based model. (224, 224, 3) is the standard input size and (48, 48, 3) is the minimum input size supported by the standard VGG16 model. (128, 128, 3) was chosen as an intermediate size between the above input sizes. We experimented with the performance with and without sequence pattern repetition. High performance was obtained in the experiment without sequence pattern repetition. Therefore, to evaluate the impact by changing the input dimension, we considered only the single representation of the nucleic acid sequence without repetition. The input dimensions were defined bypassing the argument input_shape to the first layer. The models were trained based on the 6mA-rice-Chen dataset [23] and the performance was evaluated in terms of the five metrics accuracy, sensitivity, specificity, MCC, and AUC. By using 10-fold cross-validation, the performance was evaluated for the customized VGG16 based models with input shapes (128, 128, 3), (48, 48, 3) respectively. 5-fold cross-validation was used to evaluate the performance for the customized VGG16 based model with input shape (224, 224, 3). The results show that the performance increases when the input dimension decreases. The experiment verifies that the customized VGG16 model with input shape (48, 48, 3) has the highest performance. Therefore, the customized VGG16 model with input shape (48, 48, 3) was chosen as the final 6mAVGG model. The experiment verifies that repeating the pattern along the sequence length dimension does not increase the performance in terms of the five metrics accuracy, sensitivity, specificity, MCC, and AUC. The comparison in table 5 shows the performance impact by changing the input dimension.

TABLE 5. Performance impact by changing input dimension of VGG16 architecture.

Models	Sn	Sp	ACC	MCC	AUC
VGG16 48x48x3	0.975	0.9739	0.9744	0.9493	0.9924
VGG16 128x128x3	0.9148	0.9216	0.9182	0.8372	0.9628
VGG16 224x224x3	0.8886	0.8909	0.8898	0.7804	0.9361

IV. VALIDATION AND EVALUATION

The standard 10-fold cross-validation method was used to evaluate the proposed method on the 6mA-rice-Chen benchmark dataset [23], M. musculus benchmark dataset [22] and the cross-species dataset [56] as $k = 10$ have shown empirically to yield test error estimates that neither suffers from high bias nor high variance [63]. Each benchmark dataset was randomly partitioned into 10 folds with equal size. In each cross-validation iteration, eight folds were used for training, one fold for validating, and the remaining one fold for testing. In each iteration, the specific model with the highest accuracy on the validation fold was saved and the performance was evaluated using the testing fold. The cross-validation iteration was executed 10 times, and the average predicted accuracy of 10 iterations was calculated. The performance results in terms of the five metrics accuracy, sensitivity, specificity, MCC and AUC on each dataset is shown on table 4 and ROC

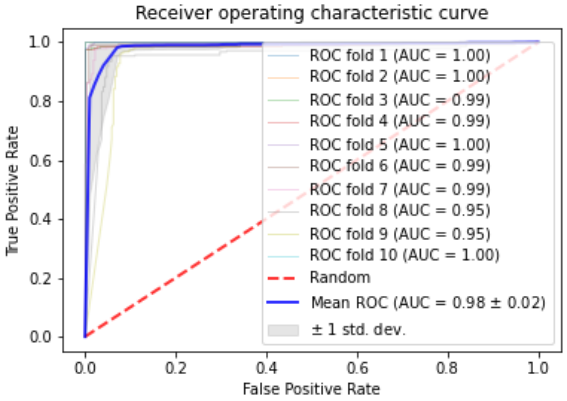

FIGURE 5. Validation ROC on cross species dataset.

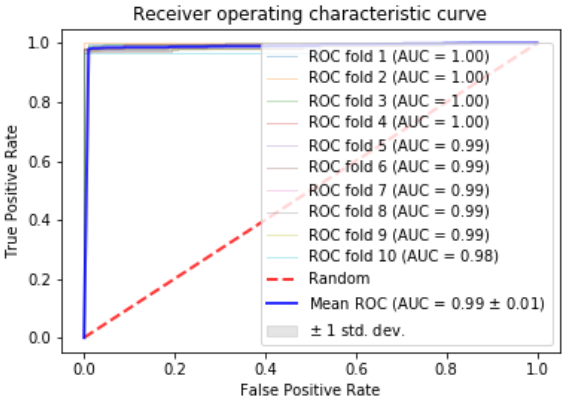

FIGURE 6. Validation ROC on mice benchmark dataset.

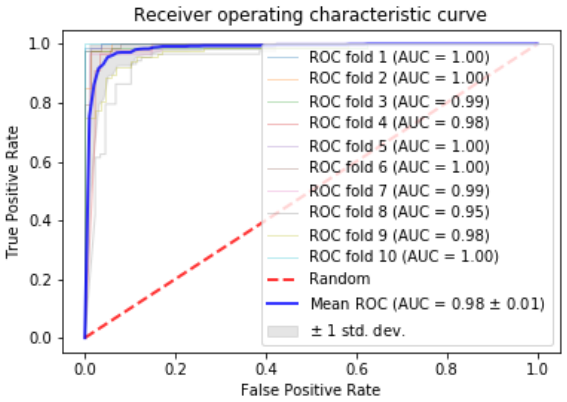

FIGURE 7. Validation ROC on rice benchmark dataset.

curves for the datasets are shown in Figure 5, 6, 7 which shows the success of the model.

The independent dataset validation method was also used to evaluate the proposed model. The independent datasets used in this study consist of species: *ArabidopsisThaliana*, *RosaChinensis*, *Drosophila*, and *Yeast* acquired from https://github.com/lwzyb/SICD6mA [77]. The proposed model was applied to each independent dataset and the accuracy was measured. All these sequences were 41-bp long with the true 6mA site in the center obtained from the "MethSMRT" database [53]. The performance of 6mAVGG, which was trained on the cross-species, rice, and M. musculus dataset was evaluated, on the independent datasets of species: ArabidopsisThaliana, RosaChinensis, Drosophila, and Yeast. The results are shown in Table 6. The model trained with benchmark data predicts 6mA sites of other species ArabidopsisThaliana, RosaChinensis, Drosophila, and Yeast with a prediction accuracy over 70%. This shows that 6mAVGG has good potential in predicting 6mA sites in other eukaryotic genomes.

TABLE 6. Independent dataset validation results.

Trained Species	Arabidopsis Thaliana	Rosa Chinensis	Drosophila	Yeast
Cross Species	0.8018	0.6893	0.7229	0.7493
Rice	0.7646	0.8004	0.7841	0.7168
Mice	0.8132	0.7109	0.7459	0.7657

V. CONCLUSION

In this study, we proposed a customized VGG16 based model called 6mAVGG using convolution neural networks. We also introduced a novel 3-dimensional encoding mechanism extending the one-hot encoding method for the given DNA sequences of length 41bp to support the VGG16 model input. This novel encoding mechanism is generalized to be used in future research which are to apply CNN architectures in a biological DNA sequence analysis domain. Specifically, the 10-fold cross-validation on the benchmark datasets for the proposed model achieves higher accuracies for cross-species, Rice, and M. musculus genomes. This model trained on the cross-species dataset outperforms the existing computational tools SNNRice6mA, ilM-CNN with a current validation accuracy of 98% for the prediction of 6mA sites. The model trained with benchmark data predicts 6mA sites of other species: A*rabidopsisThaliana, RosaChinensis, Drosophila,* and *Yeast* with a prediction accuracy over 70%. The success rates obtained by using the model trained by the benchmark dataset from cross-species to the genomes of other eukaryotic organisms are high, indicating that the proposed model holds a high potential to become a useful tool in genome-wide analysis for identifying 6mA sites in eukaryotes. With the availability of more benchmark data and more computational power, we would be able to further optimize the proposed model. In future work, we can research on other CNN architectures in the problem domain of 6mA prediction to further improve the prediction accuracy.

REFERENCES

[1] Z. K. O'Brown and E. L. Greer, "N6-methyladenine: A conserved and dynamic DNA mark," in *DNA Methyltransferases-Role and Function*. Cham, Switzerland: Springer, 2016, pp. 213–246.

[2] G.-Z. Luo and C. He, "DNA N 6-methyladenine in metazoans: Functional epigenetic mark or bystander?" *Nature Struct. Mol. Biol.*, vol. 24, no. 6, p. 503, 2017.

[3] B. F. Vanyushin, S. G. Tkacheva, and A. N. Belozersky, "Rare bases in animal DNA," *Nature*, vol. 225, no. 5236, pp. 948–949, Mar. 1970.
[4] J. L. Campbell and N. Kleckner, "E. Coli oriC and the dnaA gene promoter are sequestered from dam methyltransferase following the passage of the chromosomal replication fork," *Cell*, vol. 62, no. 5, pp. 967–979, Sep. 1990.
[5] K. G. Au and K. W. P. Modrich, "Initiation of methyl-directed mismatch repair," *J. Biol. Chem.*, vol. 267, no. 17, pp. 12142–12148, 1992.
[6] J. L. Robbins-Manke, Z. Z. Zdraveski, M. Marinus, and J. M. Essigmann, "Analysis of global gene expression and double-strand-break formation in DNA adenine methyltransferase- and mismatch repair-deficient escherichia coli," *J. Bacteriology*, vol. 187, no. 20, pp. 7027–7037, Oct. 2005.
[7] D. A. Low, N. J. Weyand, and M. J. Mahan, "Roles of DNA adenine methylation in regulating bacterial gene expression and virulence," *Infection Immunity*, vol. 69, no. 12, pp. 7197–7204, Dec. 2001.
[8] P. J. Pukkila, J. Peterson, G. Herman, P. Modrich, and M. Meselson, "Effects of high levels of DNA adenine methylation on methyl-directed mismatch repair in Escherichia coli," *Genetics*, vol. 104, no. 4, pp. 571–582, 1983.
[9] L. Cheng, J. Sun, W. Xu, L. Dong, Y. Hu, and M. Zhou, "OAHG: An integrated resource for annotating human genes with multi-level ontologies," *Sci. Rep.*, vol. 6, no. 1, pp. 1–9, Dec. 2016.
[10] G.-Z. Luo, M. A. Blanco, E. L. Greer, C. He, and Y. Shi, "DNA N^6-methyladenine: A new epigenetic mark in eukaryotes?" *Nature Rev. Mol. Cell Biol.*, vol. 16, no. 12, pp. 705–710, 2015.
[11] G. Lichinchi, S. Gao, Y. Saletore, G. M. Gonzalez, V. Bansal, Y. Wang, C. E. Mason, and T. M. Rana, "Dynamics of the human and viral m6A RNA methylomes during HIV-1 infection of t cells," *Nature Microbiol.*, vol. 1, no. 4, p. 16011, Apr. 2016.
[12] G. Lichinchi, S. Gao, Y. Saletore, G. M. Gonzalez, V. Bansal, Y. Wang, C. E. Mason, and T. M. Rana, "Dynamics of the human and viral m6A RNA methylomes during HIV-1 infection of t cells," *Nature Microbiol.*, vol. 1, no. 4, pp. 1–9, Apr. 2016.
[13] C. L. Xiao, S. Zhu, M. He, D. Chen, Q. Zhang, Y. Chen, G. Yu, J. Liu, S. Q. Xie, F. Luo, and Z. Liang, "N6-methyladenine DNA modification in the human genome," *Mol. Cell*, vol. 71, no. 2, pp. 306–318, 2018.
[14] E. L. Greer, M. A. Blanco, L. Gu, E. Sendinc, J. Liu, D. Aristizábal-Corrales, C.-H. Hsu, L. Aravind, C. He, and Y. Shi, "DNA methylation on N6-adenine in C. Elegans," *Cell*, vol. 161, no. 4, pp. 868–878, May 2015.
[15] G. Zhang, H. Huang, D. Liu, Y. Cheng, X. Liu, W. Zhang, R. Yin, D. Zhang, P. Zhang, J. Liu, C. Li, B. Liu, Y. Luo, Y. Zhu, N. Zhang, S. He, C. He, H. Wang, and D. Chen, "N6-methyladenine DNA modification in drosophila," *Cell*, vol. 161, no. 4, pp. 893–906, May 2015.
[16] D. Wion and J. Casadesús, "N6-methyl-adenine: An epigenetic signal for DNA–protein interactions," *Nature Rev. Microbiol.*, vol. 4, no. 3, pp. 183–192, Mar. 2006.
[17] H. Heyn and M. Esteller, "An adenine code for DNA: A second life for N6-methyladenine," *Cell*, vol. 161, no. 4, pp. 710–713, May 2015.
[18] K. R. Pomraning, K. M. Smith, and M. Freitag, "Genome-wide high throughput analysis of DNA methylation in eukaryotes," *Methods*, vol. 47, no. 3, pp. 142–150, Mar. 2009.
[19] A. M. Krais, M. G. Cornelius, and H. H. Schmeiser, "Genomic N6-methyladenine determination by MEKC with LIF," *Electrophoresis*, vol. 31, no. 21, pp. 3548–3551, Oct. 2010.
[20] B. A. Flusberg, D. R. Webster, J. H. Lee, K. J. Travers, E. C. Olivares, T. A. Clark, J. Korlach, and S. W. Turner, "Direct detection of DNA methylation during single-molecule, real-time sequencing," *Nature Methods*, vol. 7, no. 6, p. 461, 2010.
[21] E. L. Greer, M. A. Blanco, L. Gu, E. Sendinc, J. Liu, D. Aristizábal-Corrales, C.-H. Hsu, L. Aravind, C. He, and Y. Shi, "DNA methylation on N6-adenine in C. Elegans," *Cell*, vol. 161, no. 4, pp. 868–878, May 2015.
[22] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, and K.-C. Chou, "IDNA6 mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC," *Genomics*, vol. 111, no. 1, pp. 96–102, Jan. 2019.
[23] W. Chen, H. Lv, F. Nie, and H. Lin, "I6 mA-pred: Identifying DNA N6-methyladenine sites in the rice genome," *Bioinformatics*, vol. 35, no. 16, pp. 2796–2800, Aug. 2019.
[24] C. Pian, G. Zhang, F. Li, and X. Fan, "MM-6 mAPred: Identifying DNA N6-methyladenine sites based on Markov model," *Bioinformatics*, vol. 36, no. 2, pp. 388–392, Jul. 2019.
[25] M. Tahir, H. Tayara, and K. T. Chong, "IDNA6 mA (5-step rule): Identification of DNA N6-methyladenine sites in the rice genome by intelligent computational model via Chou's 5-step rule," *Chemometric Intell. Lab. Syst.*, vol. 189, pp. 96–101, Jun. 2019.
[26] S. Basith, B. Manavalan, T. H. Shin, and G. Lee, "SDM6A: A Web-based integrative machine-learning framework for predicting 6 mA sites in the rice genome," *Mol. Therapy Nucleic Acids*, vol. 18, pp. 131–141, Dec. 2019.
[27] H. Lv, F.-Y. Dao, Z.-X. Guan, D. Zhang, J.-X. Tan, Y. Zhang, W. Chen, and H. Lin, "IDNA6 mA-rice: A computational tool for detecting N6-methyladenine sites in rice," *Frontiers Genet.*, vol. 10, p. 793, Sep. 2019.
[28] Z. Zhang, Y. Zhao, X. Liao, W. Shi, K. Li, Q. Zou, and S. Peng, "Deep learning in omics: A survey and guideline," *Briefings Funct. Genomics*, vol. 18, no. 1, pp. 41–57, Feb. 2019.
[29] C. Pian, G. Zhang, F. Li, and X. Fan, "MM-6 mAPred: Identifying DNA N6-methyladenine sites based on Markov model," *Bioinformatics*, vol. 36, no. 2, pp. 388–392, Jul. 2019.
[30] H. Yu and Z. Dai, "SNNRice6 mA: A deep learning method for predicting DNA N6-methyladenine sites in rice genome," *Frontiers Genet.*, vol. 10, p. 1071, Oct. 2019.
[31] A. Wahab, S. D. Ali, H. Tayara, and K. To Chong, "IIM-CNN: Intelligent identifier of 6 mA sites on different species by using convolution neural network," *IEEE Access*, vol. 7, pp. 178577–178583, 2019.
[32] M. Vihinen, "How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis," *BMC Genomics*, vol. 13, no. 4, p. S2, 2012.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[34] H. Tayara and K. Chong, "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network," *Sensors*, vol. 18, no. 10, p. 3341, Oct. 2018.
[35] H. Tayara, K. Gil Soo, and K. T. Chong, "Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network," *IEEE Access*, vol. 6, pp. 2220–2230, 2018.
[36] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
[37] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.
[38] X. Pan, P. Rijnbeek, J. Yan, and H.-B. Shen, "Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks," *BMC Genomics*, vol. 19, no. 1, p. 511, Dec. 2018.
[39] I. Nazari, H. Tayara, and K. T. Chong, "Branch point selection in RNA splicing using deep learning," *IEEE Access*, vol. 7, pp. 1800–1807, 2019.
[40] Z. Louadi, M. Oubounyt, H. Tayara, and K. T. Chong, "Deep splicing code: Classifying alternative splicing events using deep learning," *Genes*, vol. 10, no. 8, p. 587, Aug. 2019.
[41] M. Tahir, H. Tayara, and K. T. Chong, "IRNA-PseKNC(2methyl): Identify RNA 2'-O-methylation sites by convolution neural network and Chou's pseudo components," *J. Theor. Biol.*, vol. 465, pp. 1–6, Mar. 2019.
[42] M. Oubounyt, Z. Louadi, H. Tayara, and K. T. Chong, "DeePromoter: Robust promoter predictor using deep learning," *Frontiers Genet.*, vol. 10, p. 286, Apr. 2019.
[43] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, and K.-C. Chou, "IDNA6 mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC," *Genomics*, vol. 111, no. 1, pp. 96–102, Jan. 2019.
[44] W. Chen, P. Feng, H. Ding, H. Lin, and K.-C. Chou, "IRNA-methyl: Identifying N6-methyladenosine sites using pseudo nucleotide composition," *Anal. Biochemistry*, vol. 490, pp. 26–33, Dec. 2015.
[45] W. Chen, H. Tang, J. Ye, H. Lin, and K.-C. Chou, "iRNA-PseU: Identifying RNA pseudouridine sites," *Mol. Therapy-Nucleic Acids*, vol. 5, p. e332, Jan. 2016.
[46] J. Jia, L. Zhang, Z. Liu, X. Xiao, and K.-C. Chou, "PSumo-CD: Predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC," *Bioinformatics*, vol. 32, no. 20, pp. 3133–3141, Oct. 2016.
[47] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, and K.-C. Chou, "IPTM-mLys: Identifying multiple lysine PTM sites and their different types," *Bioinformatics*, vol. 32, no. 20, pp. 3116–3123, Oct. 2016.

[48] P. Feng, H. Ding, H. Yang, W. Chen, H. Lin, and K.-C. Chou, "IRNA-PseColl: Identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC," *Mol. Therapy-Nucleic Acids*, vol. 7, pp. 155–163, Jun. 2017.
[49] W. Chen, P. Feng, H. Yang, H. Ding, H. Lin, and K.-C. Chou, "IRNA-3typeA: Identifying three types of modification at RNA's adenosine sites," *Mol. Therapy-Nucleic Acids*, vol. 11, pp. 468–474, Jun. 2018.
[50] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, J.-H. Jia, and K.-C. Chou, "IKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier," *Genomics*, vol. 110, no. 5, pp. 239–246, Sep. 2018.
[51] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *J. Theor. Biol.*, vol. 273, no. 1, pp. 236–247, Mar. 2011.
[52] K.-C. Chou, "Advances in predicting subcellular localization of multi-label proteins and its implication for developing multi-target drugs," *Current Medicinal Chem.*, vol. 26, no. 26, pp. 4918–4943, Oct. 2019.
[53] P. Ye, Y. Luan, K. Chen, Y. Liu, C. Xiao, and Z. Xie, "MethSMRT: An integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D85–D89, Jan. 2017.
[54] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.* Berlin, Germany: Springer, 1995, pp. 195–201.
[55] N. Ketkar, "Introduction to keras," in *Deep Learning With Python*. Berkeley, CA, USA: Apress, 2017, pp. 97–111.
[56] Z. Liu, W. Dong, W. Jiang, and Z. He, "CsDMA: An improved bioinformatics tool for identifying DNA 6 mA modifications via Chou's 5-step rule," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.
[57] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: Accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, Dec. 2012.
[58] K.-C. Chou and H.-B. Shen, "Recent progress in protein subcellular location prediction," *Anal. Biochemistry*, vol. 370, no. 1, pp. 1–16, Nov. 2007.
[59] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Crit. Rev. Biochemistry Mol. Biol.*, vol. 30, no. 4, pp. 275–349, 1995.
[60] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
[61] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.
[62] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Crit. Rev. Biochem. Mol. Biol.*, vol. 30, no. 4, pp. 275–349, 1995.
[63] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. New York, NY, USA: Springer, 2013.
[64] A. Tharwat, "Classification assessment methods," *Appl. Comput. Inform.*, to be published, doi: 10.1016/j.aci.2018.08.003.
[65] A. Creswell, K. Arulkumaran, and A. A. Bharath, "On denoising autoencoders trained to minimise binary cross-entropy," 2017, *arXiv:1708.08487*. [Online]. Available: http://arxiv.org/abs/1708.08487
[66] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.
[67] G.-Z. Luo, M. A. Blanco, E. L. Greer, C. He, and Y. Shi, "DNA N^6-methyladenine: A new epigenetic mark in eukaryotes?" *Nature Rev. Mol. Cell Biol.*, vol. 16, no. 12, pp. 705–710, 2015.
[68] M. J. Koziol, C. R. Bradshaw, G. E. Allen, A. S. H. Costa, and C. Frezza, "Identification of methylated deoxyadenosines in genomic DNA by dA6m DNA immunoprecipitation," *Bio-Protocol*, vol. 6, no. 21, p. e1990, 2016.
[69] A. Khan, A. Sohail, U. Zahoora, and A. Saeed Qureshi, "A survey of the recent architectures of deep convolutional neural networks," 2019, *arXiv:1901.06032*. [Online]. Available: http://arxiv.org/abs/1901.06032
[70] O. Koukoura, D. A. Spandidos, A. Daponte, and S. Sifakis, "DNA methylation profiles in ovarian cancer: Implication in diagnosis and therapy," *Mol. Med. Rep.*, vol. 10, no. 1, pp. 3–9, 2014.
[71] W. Chen, H. Yang, P. Feng, H. Ding, and H. Lin, "IDNA4mC: Identifying DNA N4-methylcytosine sites based on nucleotide chemical properties," *Bioinformatics*, vol. 33, no. 22, pp. 3518–3523, Nov. 2017.
[72] Q. Huang, J. Zhang, L. Wei, F. Guo, and Q. Zou, "6 mA-RicePred: A method for identifying DNA N6-methyladenine sites in the rice genome based on feature fusion," *Frontiers Plant Sci.*, vol. 11, p. 4, Jan. 2020.
[73] H. Xu, R. Hu, P. Jia, and Z. Zhao, "6 mA-finder: A novel online tool for predicting DNA N6-methyladenine sites in genomes," *Bioinformatics*, vol. 36, no. 10, pp. 3257–3259, May 2020.
[74] H.-T. Wang, F.-H. Xiao, G.-H. Li, and Q.-P. Kong, "Identification of DNA N6-methyladenine sites by integration of sequence features," *Epigenetics Chromatin*, vol. 13, no. 1, pp. 1–10, Dec. 2020.
[75] Y. Xu and R. Goodacre, "On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning," *J. Anal. Test.*, vol. 2, no. 3, pp. 249–262, Jul. 2018.
[76] B. Jin, Y. Li, and K. D. Robertson, "DNA methylation: Superior or subordinate in the epigenetic hierarchy?" *Genes Cancer*, vol. 2, no. 6, pp. 607–617, Jun. 2011.
[77] W. Liu and H. Li, "SICD6mA: Identifying 6mA sites using deep memory network," *bioRxiv*, to be published.
[78] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[79] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Into Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.
[80] Y. LeCun. (2015). *LeNet-5, Convolutional Neural Networks*. [Online]. Available: http://yann.lecun.com/exdb/lenet
[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[82] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[83] D. L. Wheeler, "Database resources of the national center for biotechnology information," *Nucleic Acids Res.*, vol. 33, pp. D39–D45, Dec. 2004.
[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[85] E. Bisong, "Google colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Apress, 2019, pp. 59–64.
[86] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Devin, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
[87] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, vol. 26. New York, NY, USA: Springer, 2013.
[88] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: A comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, Aug. 2005.
[89] H. Jia, Y. Xia, Y. Song, W. Cai, M. Fulham, and D. D. Feng, "Atlas registration and ensemble deep convolutional neural network-based prostate segmentation using magnetic resonance imaging," *Neurocomputing*, vol. 275, pp. 1358–1369, Jan. 2018.
[90] S. Barua, M. U. Ahmed, C. Ahlström, and S. Begum, "Automatic driver sleepiness detection using EEG, EOG and contextual information," *Expert Syst. Appl.*, vol. 115, pp. 121–135, Jan. 2019.

LOKUTHOTA HEWAGE ROLAND was born in Galle, Sri Lanka, in 1995. He is currently pursuing the B.C.S. (Special) degree from the University of Ruhuna. He is a Former Software Engineer Intern with WSO2. He is also a Technical Writer with the publications, such as Towards Data Science, The Startup, Gitconnected, Data Driven Investor, Towards AI, CoFoundersTown, and Faun. His research interests include bioinformatics and deep learning. He is a member of the IEEE Quantum Technical Community.

CHAMPI THUSANGI WANNIGE received the B.Sc. (Special) and M.Phil. degrees in computer vision from the University of Colombo, in 2007 and 2010, respectively, and the Ph.D. degree in computational systems biology from Lincoln University, New Zealand, in 2014. She is currently with the Department of Computer Science, University of Ruhuna, as a Senior Lecturer.