Article:
Liu, R., Ou, L., Qi, J. et al. (10 more authors) (2022) Mixed-weight neural bagging for detecting m6A modifications in SARS-CoV-2 RNA sequencing. IEEE Transactions on Biomedical Engineering, 69 (8). pp. 2557-2568. ISSN 1558-2531

https://doi.org/10.1109/TBME.2022.3150420

Mixed-weight Neural Bagging for Detecting m^6A Modifications in SARS-CoV-2 RNA Sequencing

Ruhan Liu, Liang Ou, Jun Qi,*Member,IEEE*, Bin Sheng,*Member,IEEE*, Huating Li, Ping Li,*Member,IEEE*, Pei Hao, Xiaokang Yang, *Member,IEEE*, Guangtao Xue,*Member,IEEE*, Jinman Kim,*Member,IEEE*, Ping Zhang,*Senior Member,IEEE*, Po Yang,*Senior Member,IEEE*, and David Dagan Feng,*Life Fellow,IEEE*

***Abstract*— It has had a big influence in the aftermath of the Corona Virus Disease 2019 (COVID-19) outbreak, which was caused by the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) virus. Recent research has established that a ribonucleic acid (RNA) modification called m6A is connected with viral infections and that it has a strong correlation with the structure and function of several essential proteins. It is critical to have a thorough understanding of RNA modification from the standpoint of viral diseases—however, simple approaches for discovering RNA alterations within the transcriptome are lacking. In addition, nanopore single-molecule direct RNA sequencing (DRS) also provides better data support for RNA modification detection, which directly detects the original samples and preserves the potential m^6A signature compared to second-generation sequencing. We present a methodology for precisely identifying m6A alterations that incorporates both extracted features from direct RNA sequencing data and raw current and quality data. To discover m6A alterations, we present a multi-model fusion method called mixed-weight neural bagging (MWNB). Model development was accomplished through the use of modified and unmodified m6A synthetic sequences. Our Bagging-LightGBM model achieves the highest classification accuracy of 97.85%, precision of 98.37%, and AUC of 0.9968. Additionally, we apply the suggested model to the COVID-19 dataset; the experiment results revealed a strong association with biomedical experiments. Our strategy enables the prediction of m6A alterations using sequencing data and the identification of m6A modifications on the COVID-19.**

***Index Terms*— m^6A RNA modifictions, ensemble learning, COVID-19, SARS-COV-2**

R. Liu, B. Sheng, and Guangtao Xue are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (Email: shengbin@sjtu.edu.cn, liurh996@sjtu.edu.cn, xue-gt@cs.sjtu.edu.cn).

L. Ou and P. Hao are with the Key Laboratory of Molecular Virology and Immunology, Institut Pasteur of Shanghai, 200031, China (Email: liangou@ips.ac.cn, phao@ips.ac.cn).

J. Qi is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (Email: jun.qi@xjtlu.edu.cn).

H. Li is with the Shanghai Jiao Tong University Affiliated Sixth People's Hospital, Shanghai 200233, China (Email: huarting99@sjtu.edu.cn).

P. Li is with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (Email: p.li@polyu.edu.hk).

X. Yang is with the Shanghai Key Laboratory of Digital Media Processing and Communication, Department of Electronic Engineering, and he is also with the Institute of Image Communication and Network Engineering, Shanghai Jiaotong University, Shanghai, China (Email: xkyang@sjtu.edu.cn).

J. Kim is with the Biomedical and Multimedia Information Technology Research Group, School of Information Technologies, The University of Sydney, Sydney, NSW, Australia (Email: jinman.kim@sydney.edu.au)

P. Zhang is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio 43210, USA; and also with the Department of Biomedical Informatics, The Ohio State University, Columbus, Ohio 43210, USA (Email: zhang.10631@osu.edu).

P. Yang is with the Department of Computer Science, The University of Sheffield, Sheffield S1 4DP, U.K. (Email: po.yang@sheffield.ac.uk).

D. D. Feng is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (Email: dagan.feng@sydney.edu.au).

I. Introduction

THE Corona Virus Disease 2019 (COVID-19) outbreak has spread throughout the world, claiming a large number of lives and affecting global economic and social stability [1]. Vaccine development and anti-infection strategies have emerged as critical components of the global response to this pandemic [2]. Due to our limited understanding of the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), there are currently no specific drugs available for the treatment of SARS-CoV-2. Thus, understanding the genetic information of SARS-CoV-2 enables us to analyze the virus's characteristics and aid in developing and implementing measures.

It is critical to establish a link between RNA modification and protein expression because the structure and state of proteins have a significant impact on gene expression. Additionally, locating RNA modification sites can be extremely beneficial for analyzing the relationship between RNA modifications and protein expression. SARS-CoV-2 is an enveloped virus with a 30 kb single-stranded RNA genome found in COVID-19 [3]. SARS-CoV-2, Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV), and Middle East Respiratory Syndrome Coronavirus (MERS-CoV) all belong to the β coronavirus genus and share approximately 80% and 50% homology, respectively. The global outbreak of COVID-19 demonstrates SARS-CoV-2's ability to cross species barriers and spread between humans [4]. Theoretically, SARS-CoV-2 has the potential to mutate, with the mutation occurring due to changes in protein structure and properties. Additionally, changes in the structure and properties of proteins can be reflected by changes in the m6A modification status. As a result, it is critical for research to understand the location and magnitude of m6A modifications. Additionally, Kim's Cell article [4] predicted possible base modification sites in the SARS-CoV 2 transcriptome. The disclosure of relevant

information about RNA modification has enormous biological significance and provides potential resources for understanding how coronaviruses appear to regulate themselves.

RNA modifications are modifications to the chemical structure of RNA. RNA modifications have been implicated in the life cycles of numerous viruses and the cellular response to viral infection in recent studies [5], [6]. N6-methyladenosine ($m6A$) is the most abundant mRNA modification, accounting for more than 170 different types of RNA modifications [7]. For several decades, $m6A$ has been identified on a variety of virus-encoded transcripts. Numerous studies examining the function of $m6A$ in viral-host interactions have identified distinct roles, implying widespread regulatory control over viral life cycles [8]. As a result, elucidating the location and magnitude of $m6A$ modifications contributes to our understanding of the regulatory mechanisms that govern viral replication. It is advantageous for vaccine development and anti-infection strategy development in the era of the COVID-19 pneumonia pandemic.

While some experimental methods have been adapted for $m6A$ detection, some limitations remain. There are frequently issues with resolution and immune specificity [7], and the procedure is frequently unsatisfactory in terms of cost and precision [9]. As a result, the field of $m6A$ modification detection in RNA is a high-value one. In addition, compared to second-generation sequencing that requires polymerase chain reaction (PCR) which is an amplification technique that loses m^6A modification information, DRS directly detects the original samples and preserves the underlying m^6A signature. In DRS, nanopores are used to move uniformly from the beginning to the end of the RNA sequence in a sliding window of base length five, and the current value at each moment determines by the composition of the five nucleotides inside the slide window. The capacity of quarantine nanopore single-molecule direct RNA sequencing (DRS) to detect base modification in RNA is demonstrated [10]. DRS has been shown to record traces of base modification in the form of electrical signals. The following section provides an overview of the currently used DRS-based RNA $m6A$ detection method. The method for extracting features is based on Basecall-Error, a hypothesis testing technique that was first described in [5]. While these methods produce results, they have limitations in terms of robustness and generalizability to larger datasets, and this type of method is highly dependent on the reliability of the control sample [11]. Additionally, these methods' performance is entirely dependent on the sensitivity of potential modification types to features. The type of modification [12] is unknown.

Machine learning techniques have been enormously successful in biomedical engineering [13]–[17]. Many classical machine learning algorithms perform exceptionally well at detecting base modification. For instance, Hidden Markov Models (HMM) [18] and Support Vector Machines (SVM) [19] have been used to identify specific base modifications in DNA and RNA in some previous work. However, certain issues impair their ability to generalize. To begin, DRS is a novel technique, and the samples obtained in vivo are heterogeneous. As a result, despite the higher accuracy of DRS compared to previous generation sequencing technologies, accurately labeling them is challenging, and suitable training samples are scarce. Due to a lack of training data, the deep learning model performance is insufficient.

Therefore, it is critical to investigate a more effective method for handling this novel data. In this study, we propose an ensemble learning framework for $m6A$ detection using DRS data. We obtain features from raw sequencing data, including current and quality data, as well as extracted and screened mapping base features, according to DRS data. The raw data features and mapping features are then fed into the proposed integrated learning model (Mixed-weight neural bagging) to obtain $m6A$ prediction results. Additionally, we compare the performance of the model we introduced to that of state-of-the-art methods. All methods employ parameter tuning techniques to produce the best models. Finally, we use our model to predict $m6A$ base modification in the most recent COVID-19 data set and obtain illuminating results for gene mutation problems. We make the following contributions to our work:

1) We propose a pipeline for detecting $m6A$ modifications using DRS that includes an end-to-end processing flow based on a well-trained mixed-weight neural bagging (MWNB) model. The MWNB model achieves superior performance by providing dedicated feature extraction modules for both raw and mapped features. When compared to current state-of-the-art $m6A$ RNA detection methods, the accuracy is approximately increased by 8 $\%$.
2) We investigate the MWNB model's optimal parameters. Additionally, we compare the performance of the MWNB model, which utilizes both raw data and extracted features, to that of the best models that utilize only raw data or only extracted features. For raw data, models such as LSTM, RNN, and GRU are compared. For extracted features, we compare SVM, Decision Tree (DT), Extra Tree (ET), LightGBM, and random forest (RF).
3) In all models, we tune parameters using the grid search algorithm. To compare models, the best performance of each is used. Additionally, metrics such as accuracy, precision, specificity, sensitivity, F1-score, G mean1, G mean2, and AUC are considered during the evaluation process. To determine the possible $m6A$ positions on the SARS-CoV-2 transcription, we applied our model to the DRS data of a SARS-CoV-2 sample and determined the location of the potential gene mutation.

The remainder of the paper is divided into the following sections. In Section II, we introduce the nanopore sequencing technology and discuss related work on identifying RNA modifications. Then, in Section III, we detail the methodology for detecting $m6A$ modification using our MWNB model. Section IV presents our experimental results and comparison for MWNB with other state-of-the-art methods on the DRS and COVID-19 datasets. We discuss the shortcomings of our framework and future work direction in Section V. Finally, in Section VI, we conclude the paper in Section VI.

II. Related Work

The nanopore RNA sequencing process preserves the modification of $m6A$ and faithfully records the disturbance of the $m6A$ molecule to the background current in the form of an electrical signal [20]. Several studies confirmed the difference in electrical signals between $m6A$ and normal adenylate in theory and in practice [18] . Smith et al. accurately performed direct RNA-seq on two samples with high degree m^6A and low degree m^6A, respectively [19]. Their work indicated that they observed alternation of current signals and base calling error near the location of $m6A$. As a result, it has a certain research foundation for determining the base position by comparing the differences in electrical signals. Thus, we introduce related works in $m6A$ modification detection by following three parts. To begin, a novel sequencing technology, nanopore sequencing, is introduced. Following that, we demonstrate one of the most frequently used statistical methods: statistical hypothesis testing, which has evolved into one of the primary algorithms for detecting RNA $m6A$ modifications using nanopore technology [12]. Additionally, we demonstrate several novel machine learning techniques that have been applied in this field.

A. Nanopore sequencing technology

The detection and identification of RNA sequences in living organisms is a challenging and significant research topic. Nanopore single-molecule direct RNA sequencing (DRS) is a promising and advanced technology for solving this problem. The basic principle of DRS is as follows. When RNA sequences pass sequentially through a nanopore which is a protein-electron coupler, different sequences will excite different current patterns. The relationship between these patterns and corresponding sequences have been studied in current researches. In recent studies, observed current signals can be fed into machine learning models to obtain predicted RNA canonical base sequences, when do not considering RNA modifications [22]. However, in real organisms, in addition to the canonical base, there are also some chemical groups that are modified base, such as m^6A. Therefore, the existing recognition models will have large generalization errors and poor performance capabilities when modified bases are present in the sequences.

B. Statistics methods for base modification detection

Several previous studies demonstrated statistical methods for detecting base modifications using direct sequencing with promising results [24]. Stoiber M H et al. [11] used the Mann–Whitney U test to detect $m5C$ on DNA/RNA, which is also a modification, and $m6A$ on DNA in all sequence contexts without requiring unmodified samples in addition to de novo detection. When $m5C$ occurs and when it does not, the electrical signal characteristic distribution of $m5C$ is significantly different, even more so than the distribution of $m6A$ [25]. Liu et al. [12] used the Kolmogorov–Smirnov test to demonstrate that NanoMod outperformed Tombo at detecting $m5C$ in E. coli. The statistics-based modification detection method has a number of advantages, including low computational resource consumption and a wide detection range [26]. Nonetheless, its flaws remain insurmountable. To ensure precision, a completely clean sample must be prepared (free of any modification). [5] must be performed using the same sequencing experiments and data pre-processing steps as in the experimental sample. Without a doubt, this will significantly increase the difficulty and cost of the preliminary sample preparation stage, particularly for some highly valuable biological samples. Additionally, statistical methods lack improvement space; it is difficult to improve the performance of statistical methods at the algorithm level.

C. Machine learning approaches in modification detection

Researchers gradually shifted their focus with the development of machine learning algorithms and their widespread application in bioinformatics. They attempted to implement RNA modification detection using machine learning techniques [27]–[29]. Garalde et al. developed a tool called the Nanopolish that uses the HMM (hidden Markov model) to accurately call $m5C$ on DNA in the CpG context. SignalAlign [**?**] also a modification detection tool because it is based on the HMM with the hierarchical Dirichlet process. Rand et al. used SignalAlign to identify $m5C$ and $m6A$ sequences in E. coli DNA. The mCaller, which doubles as a modification detector, detected $m6A$ on DNA using four machine learning classifiers (neural network, random forest, logistic regression, and naive Bayes classifiers). Mclntyre et al. [20] demonstrated that the most accurate predictor (84%) used the mCaller with the neural network. Prior research has concentrated on DNA base modification, particularly $m5C$. Due to structural similarity and the difficulty of obtaining accurate data sets, the $m6A$ modification in RNA has not been investigated previously. Huanle Liu et al. recently constructed a labeled dataset using in vitro transcription of $m6A$ and adenosine, respectively. Additionally, the SVM classifier they proposed produced acceptable results (90% accuracy). However, novel machine algorithms in this area should be investigated to improve the solution to this problem.

III. Proposed Method

The purpose of this work is to develop a practical model for identifying $m6A$ RNA modifications using nanopore single-molecule direct RNA sequencing (DRS). We combine the extracted feature classification model (Bagging-LightGBM) and the raw sequencing classification model (Bagging-LSTM) using a weight bagging strategy implemented by the neural network. The combined model, dubbed Mixed-Weight Neural Bagging (MWNB), is used to assess $m6A$ RNA modifications via DRS. The following sections introduce the MWNB model, which is divided into three sections: capturing various features from DRS, pre-processing and selecting features, and the MWNB classifier methodology.

The proposed method, which serves as a framework for $m6A$ modification recognition in RNA sequencing, requires that the first step extract base features from RNA sequences.

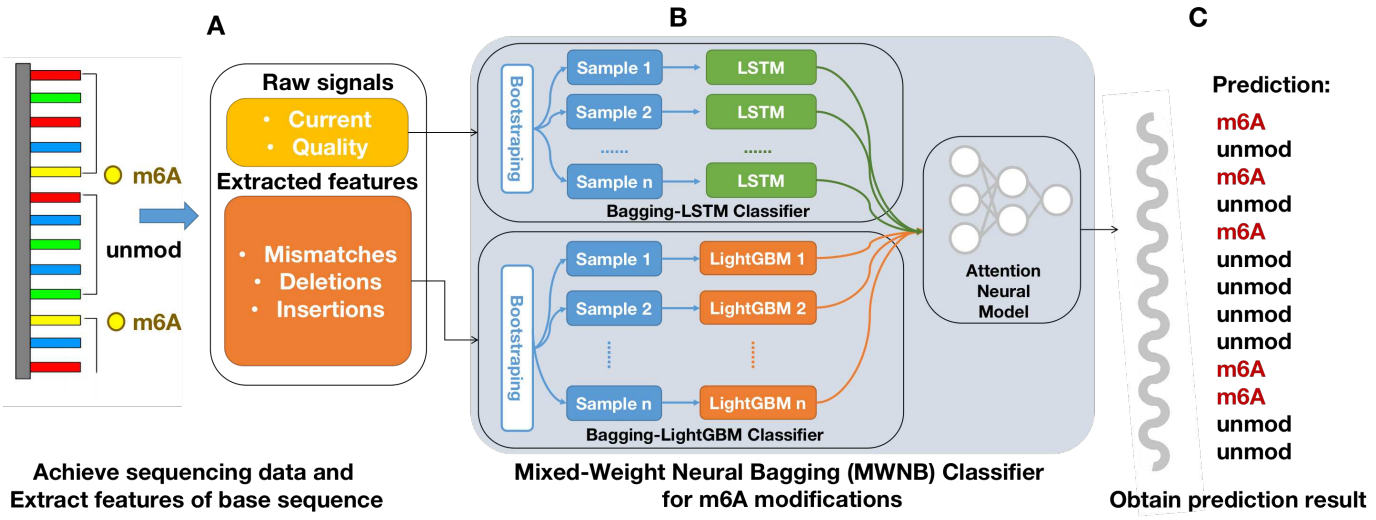

Fig. 1. A overview of the framework for analyzing $m6A$ RNA modifications using RNA sequencing. The framework is divided into three sections: gaining raw sequencing data and extracting features (Part A), classifying the data using Mixed-Weight Neural Bagging (Part B), and obtaining the prediction results (Part C) (Part C). In Part A, critical features such as base current intensity, base quality, mismatches frequency, delete frequency, and insert frequency are introduced. In Part B, the mixed-weight bagging network (MWBN) is proposed for detecting $m6A$ RNA modifications in five-base DRS fragments. Additionally, Part C details the procedure for obtaining prediction results.

This step introduces the features that are used and defines their meaning. The following step is to identify appropriate base features, thereby improving classification performance and minimizing information loss. We demonstrate how we handle extracted feature measurements and how we select essential features. Following that, four distinct types of features are employed based on their biological significance and experimental results. We illustrate the Bagging-LightGBM classifier used to discover the relationship between features and $m6A$ modifications in the third section. We introduce the Bagging-LSTM model in the fourth section, which is used to classify $m6A$ modifications based on direct sequencing. Finally, we demonstrate how to fuse the Bagging-LightGBM feature extraction algorithm with the Bagging-LSTM raw sequencing algorithm to obtain fused results.

Additionally, in the classifier design, a sequence handled model: LSTM is utilized in our DRS data. To optimize the result, we use the grid search algorithm to optimize the parameters of the LSTM to improve classification accuracy. Also, for extracted features, LightGBM [30] is applied. We used the grid search algorithm to optimize the parameters of the LightGBM. Then, to improve model ability, we proposed a fusion model to integrate multiple LightGBM [30] models and LSTM models by weight bagging strategy to identify m^6A modification. The weight bagging strategy is implemented by a neural network to obtain the voting results. Based on the above fusion model: MWNB, not only is better performance obtained, but also the best performance technology for current problems can be determined. Finally, we used our model on the COVID-19 data set and displayed the potential m^6A sites in SARS-CoV 2 RNA sequencing. Fig 1 shows the complete work of the framework.

A. Feature extraction

The downloaded compressed DRS data is decompressed using the NCBI-recommended ”FastQ-dump” software and mapped to the complete synthetic sequences using the ”Minimap2” software with the ”-ax map-ont” pre-settings option. ”Samtools” software was used to sort and index the mapped readings. We acquire the raw data of quality and current after the sorting operation. We next extracted each position’s characteristics in reference using two independent Epinano scripts (`https://github.com/enovoa/EpiNano`). The feature table was constructed using a sliding window with a length of five bases and a step of one base, as well as the feature of the next location, which included base quality, base current, mismatches frequency, insert frequency, and delete frequency. We exhibit the features we derived from the RNA sequence and explain each feature’s meaning in Table I.

B. Feature pre-processing and selecting

We get five related features after feature extraction: C, Q, Mis, Ins, and Del. The choice of features has a significant impact on classification accuracy and is a necessary step before clustering or classification. According to earlier research [19], the five characteristics of bases listed above are primarily associated with whether or not m^6A modification takes place. The duration of sliding windows also has an impact on the accuracy of forecasting m^6A RNA modifications. We determined the sliding window length of the base, which is five bases, based on [19], and deleted the fragments picked from the sliding window that did not fit the standards by referring to the [19] base matching rules.

Following the previous feature selection, we list all of the features used in creating our model. First, we took the mean, median, and standard deviation values of based quality and base current intensity as feature values to represent the

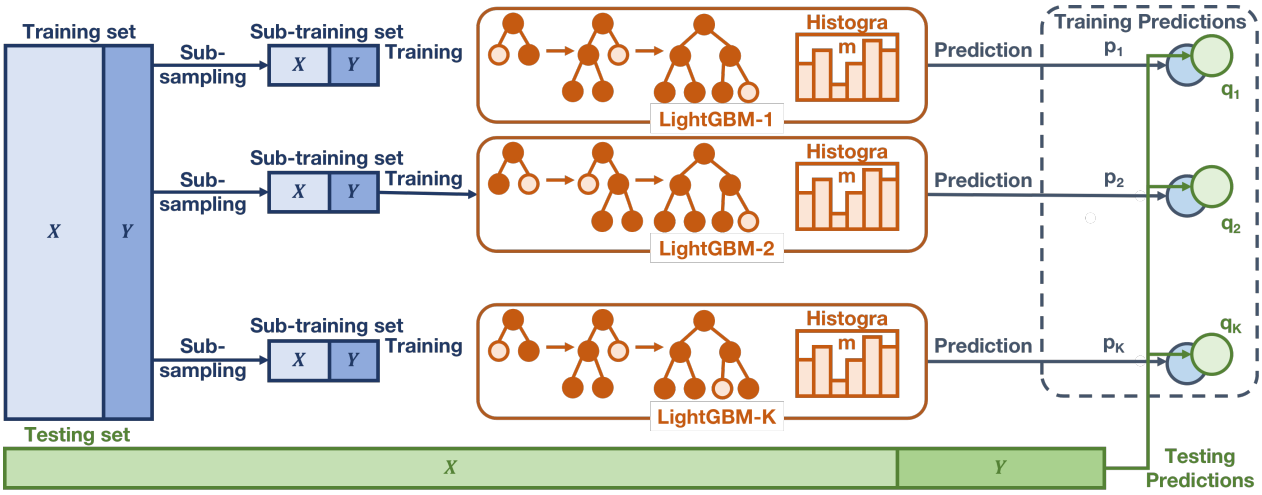

Fig. 2. The architecture of the Bagging-LightGBM model is depicted in the model structure image. LightGBM [30] models are used as basic learners in the Bagging-LightGBM model, and they are trained using different subsets of the training set. They employed the same testing set for model evaluation. The Bagging-LightGBM model produces numerous predictions about whether $m6A$ will emerge.

TABLE I

WE RETRIEVED FEATURES FROM THE RNA SEQUENCE'S RAW DATA. C AND Q INDICATE THE BASE CURRENT FEATURE AND BASE QUALITY FEATURE, RESPECTIVELY, CONTAINING THE MEAN, MEDIAN, AND STANDARD DEVIATION OF FIVE BASES (C=$c1,c2,c3,c4,c5$, AND c_i=$c-mean_i$, $c-median_i$, $c-std_i$, Q=q_1,q_2,q_3,q_4,q_5, AND q_i=$q-mean_i$, $q-median_i$, $q-std_i$). THE PROBABILITY OF MISMATCHES, INSERT, AND DELETE ARE MIS, INS, AND DEL. (MIS=$mis_1,mis_2,mis_3,mis_4,mis_5$, INS =$ins_1,ins_2,ins_3,ins_4,ins_5$, DEL = $del_1,del_2,del_3,del_4,del_5$).

Feature	Abs	Description
Base current	C	Per-base estimates of current intensity emitted by the sequencing machines.
Base quality	Q	Per-base estimates of quality emitted by the sequencing machines.
Mismatches frequency	Mis	A base of the database which is different from the query base called "mismatch". Mis = $\frac{\text{num of mismatch}}{\text{total num of base}}$
Insert frequency	Ins	A base of the database is not be mapped a base corresponding to the query sequence called "insert". Ins = $\frac{\text{num of insert}}{\text{total num of base}}$
Delete frequency	Del	A base of the query sequence is not mapped a base of database called "delete". Del = $\frac{\text{num of delete}}{\text{total num of base}}$

fundamental information of base pieces. Furthermore, the frequency of mismatches, insert, and delete in each base from the base fragment is considered expanded information. Table I lists all of the features we ended up using in our model.

C. Bagging-LightGBM feature classification

We used the light gradient boosting machine (LightGBM) as the base classifier for predicting $m6A$ RNA modifications utilizing attributes of base fragments. LightGBM [30] is a unique Gradient Boosting Decision Tree (GBDT)-based approach. Through iteration, GBDT builds weak decision tree classifiers, each of which is trained based on the residual error of the previous round of classifiers and continuously improves the accuracy of the final classifier by lowering the deviation. LightGBM provides the advantages of faster training efficiency, higher accuracy, and the ability to analyze massive amounts of data when compared to GBDT.

For training dataset $X = \{(x_i, y_i | x_i \in R^k, y_i \in R, k = 15, |X| = n\}$, where $x = \{x_1, x_2, ..., x_i, ..., x_n\}$ is the input feature set, k is the dimension of input features and $y = \{y_1, y_2, ..., y_i, ..., y_n\}$ is the corresponding label. The input features of Bagging-LightGBM include mismatch frequency, delete frequency, and insert frequency. The goal of LightGBM algorithm in training base learner is to optimize a loss function L. Considering $F(x)$ as an estimated function, the optimization goal is given as:

$$G = arg_F min E_{x,y}[L(y, \epsilon)] \tag{1}$$

where ϵ is the initial constant function value of the algorithm.

After training base classifier, the boosting process is used to improve the model performance. From iteration $M = \{1, 2, ..., j, ..., m\}$, the pseudo residuals or gradient is $g_j = \{g_{1j}, g_{2j}, ..., g_{ij}, ..., g_{nj}\}$ in each iteration, and the modified dataset called $MX = \{mX_1, mX_1, ..., mX_j, ..., mX_m\}$ in each iteration. The formula of g_{ij} and mX_j is:

$$\begin{aligned} g_{ij} &= -\frac{\partial L(y_i, F_{j-1}(x_i))}{\partial F_{j-1}(x_i)} \\ mX_j &= \{(x_i, g_{ij}) | i = 1, 2, ..., n\} \end{aligned} \tag{2}$$

where $F_j(x) = F_{j-1}(x) + \epsilon \cdot h_j(x)$.

The ϵ is updated iteratively according to $\epsilon = arg_\epsilon min \sum_n^{i=n} L(y_i, F_{j-1}(x_i) + \epsilon \cdot h_j(x_i))$. $h_j(x)$ is the fitted decision tree model using modified dataset mX_j to train. The decision tree model is the base learner in LightGBM algorithm.

We apply the bagging approach to bootstrap additional model integration. Bagging, also known as bootstrap aggregation, is a type of integrated learning model (Fig. 2).

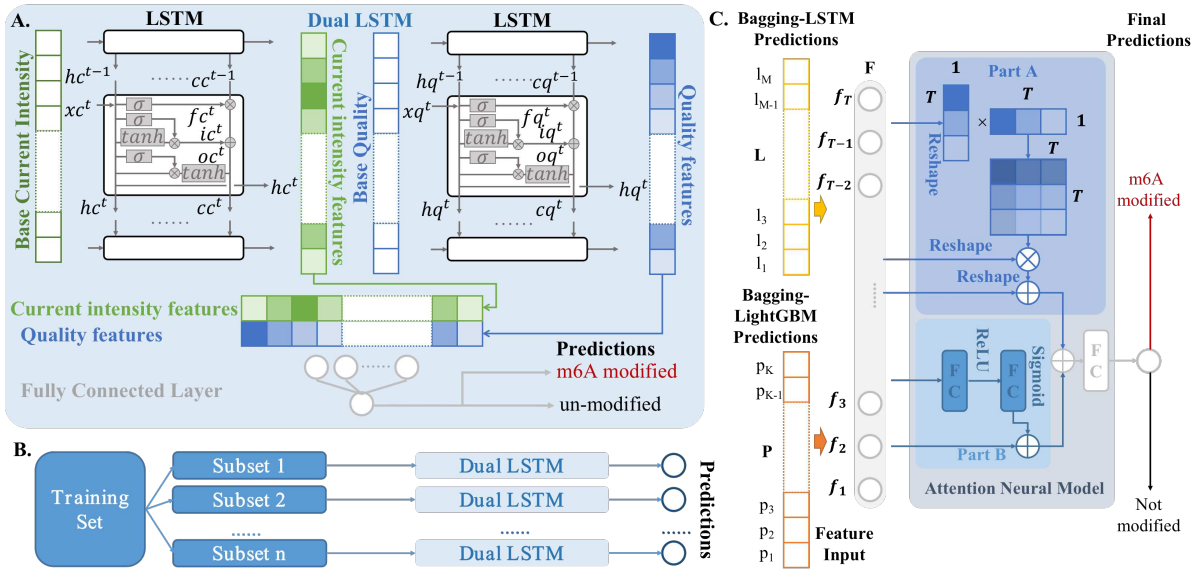

Fig. 3. The model architecture figure shows the architecture of the MWNB model. ***A.*** Figure shows the structure of a dual LSTM model for detecting m^6A RNA alterations. The architecture plot of the Bagging-LSTM model. ***B.*** In the bagging model, the training set is chosen at random and divided into many subsets that are used to train various dual LSTM models and make predictions. ***C.*** The attention neural model structure for features classification.

Algorithm 1 Bagging-LightGBM classifier

Require:

Given trainning dataset, $X = \{(x_i, y_i | x_i \in R^k, y_i \in R, |X| = n\}$.

Base LightGBM classifier, Φ.

The number of sub-sampling, K.

Ensure: Aggregation of K sub-sampling

for each $i = 1 : K$ **do**

  bootstrap sample in X to obtain modified training dataset $train_X$ and validation dataset val_X.

  train the i^{th} expert (classifier Φ) in $train_X$ and val_X.

end for

The predictions $P = \{p_1, p_2, ..., p_K\}$ is obtained by K expert models.

return P

The fact that it may be used with other classification and regression methods to improve accuracy and stability is its most major advantage. This method divides the training set into different training subsets, trains the sub-models with the training subsets, and ultimately integrates the sub-models to obtain comprehensive prediction results. To examine the hyperparameters in our job, we utilize the LightGBM model listed below as the basic learner. The number of base learners, the sample ratio when the base learner is trained, the feature ratio during training, whether to extract samples and replace them, and whether to extract features and replace them are among the parameters. Algorithm 1 illustrates the bagging classifier.

D. Bagging-LSTM raw data classification

In the previous section, we introduced the Bagging-LightGBM model, which uses extracted characteristics to categorize $m6A$ RNA modifications. The current intensity and quality are sequence data, according to DRS data. RNN models, such as RNN, LSTM, and GRU, have exceptional sequence classification performance. To categorize the feature obtained using direct sequencing data, we propose the Bagging-LSTM models.

LSTM network is an elegant solution to capture the information forward and backward. This model can access complete, sequential information about all context information after each time step in a given sequence. In this study, We propose a dual LSTM model to classify the m^6A modifications based on the raw current intensity and quality signals. The architecture of the dual LSTM model can be seen in Fig. 3. In the dual LSTM, at each time step t, hidden state is hc^t for current intensity and is hq^t for quality. The input current intensity data is xc^t at the time step t, and the input quality data is xq^t. The hidden state at the previous time step $t-1$ is hc^{t-1} and hq^{t-1} for current intensity and quality. Also, in the LSTM cell for current intensity and quality, the input gate is ic^t and iq^t in time step t, the forget gate is fc^t and fq^t, the output gate is oc^t and oq^t, and the memory cell is cc^t and cq^t, respectively. The following updating equations are given as follows:

$$
\begin{aligned}
ic^t &= \sigma\left(W^{(ic)} \cdot [hc^{t-1}, xc^t] + b^{(ic)}\right) \\
fc^t &= \sigma\left(W^{(fc)} \cdot [hc^{t-1}, xc^t] + b^{(fc)}\right) \\
oc^t &= \sigma\left(W^{(oc)} \cdot [hc^{t-1}, xc^t] + b^{(oc)}\right) \\
cc^t &= fc^t \times cc^{t-1} + ic^t \times tanh\left(W^{(cc)} \cdot [hc^{t-1}, xc^t] + b^{(cc)}\right) \\
hc^t &= oc^t \times \tanh\left(cc^t\right)
\end{aligned}
\tag{3}
$$

Where $W^{(ic)} \in R^{\omega \times d}$, $W^{(fc)} \in R^{\omega \times d}$, $W^{(oc)} \in R^{\omega \times d}$, $W^{(cc)} \in R^{\omega \times d}$ are the weight matrices for different gates for input current intensity xc^t and hidden state is hc^{t-1} for time step $t-1$ and hc^t for time step t. Here $\times$ is the element-wise multiplication, $\sigma(\cdot)$ and the $tanh(\cdot)$ and are the element-wise activation function. The LSTM handling quality data is used

the same structure as the current intensity LSTM. The details is as follows:

$$\begin{aligned} iq^t &= \sigma\left(W^{(iq)} \cdot [hq^{t-1}, xq^t] + b^{(iq)}\right) \\ fq^t &= \sigma\left(W^{(fq)} \cdot [hq^{t-1}, xq^t] + b^{(fq)}\right) \\ oq^t &= \sigma\left(W^{(oq)} \cdot [hq^{t-1}, xq^t] + b^{(oq)}\right) \\ cq^t &= fq^t \times cq^{t-1} + iq^t \times tanh\left(W^{(cq)} \cdot [hq^{t-1}, xq^t] + b^{(cq)}\right) \\ hq^t &= oq^t \times \tanh\left(cq^t\right) \end{aligned} \tag{4}$$

Further, we use the same bagging strategy to generate the predictions of Bagging-LSTM. The final prediction is $L = \{l_1, l_2, ..., l_M\}$. M is the number of base learner (dual LSTM).

E. Mixed-Weight Neural Bagging(MWNB)

Bagging is an easy-to-use strategy that has a high rate of success in reducing generalization errors. The model averaging technique is used in the classic bagging approach to increase the model's accuracy and stability. We propose a neural network to learn the weights of different weak learners' output in order to get a better fitting effect in this challenge. The procedure is depicted in Fig. 3 C.

Two attention branches are included in ANM to supplement the characteristics recovered by Bagging-LSTM and Bagging-LightGBM, and to provide a prediction of the presence of m6A modifications. With reference to [31], component A of Fig. 3 C constructs the self augmentation part of the features, and Fig. 3 C constructs the self augmentation part of the features. Part B of Fig. 3 C eliminates the [32] recommended attentional enhancement module and learns the significance coefficients of the features through the complete concatenation layer to better uncover the ultimate relationship between the features and the classification results.

IV. Experiments

The experimental results are listed in this section. We begin by describing the dataset's basic information before moving on to the implementation details. Third, evaluation metrics and model evaluation measurement are briefly explored. Following that, we describe the impact of parameter selection on model performance and introduce parameter adjustment and feature selection in all models. We also show how models perform in different classifiers with different settings. Finally, we apply our best model to the recognition of $m6A$ modifications in COVID-19 data.

A. Dataset

We used two data sets in this study: one from Epinano [19], and the other from Kim et al. [3] for the original SARS-CoV-2 data. African green monkey kidney cells (vero cells) infected with the COVID-19 were used as the source sample. They go through the same sequencing technique and upstream pre-treatment process as the training set after mRNA purification and extraction. The signal value at a given time is defined by around four bases (A, T, C, G), and all about 1024 bases are organized and combined to generate a signal pattern in the nanopore sequencing process. To cover as many scenarios as feasible, Liu et al. [19] created a sequence master comprising all signal patterns and employed synthetic substrates with and without N6-methyladenosine in 2019. There are two readings in our data collection that contain $m6A$ and two reads that do not contain $m6A$. We classified it into 19,806 positive samples and 19,964 negative samples based on early data.

B. Implemental Details

In this segmentation task, we use two datasets to train, validate and test the model. The two datasets are the Epinano dataset [19] and the original data [3] of the novel coronavirus provided by Kim et al. The Epinano dataset is used for training, validating and testing the model, while the COVID-19 dataset is used for validation only.

To improve the validity of the data, we performed feature extraction and data preprocessing with reference to Section III A and Section III B. Specifically, for each base, we extracted three features (mismatch frequency, insert frequency, and delete frequency) and two features (base current, base quality) from the raw data by the mapping tool. The two features extracted for the raw data will extract the mean, median, and variance for each base, respectively. Therefore, in our 5-base fragment, all feature inputs for each fragment include mismatch frequency, insert frequency, delete frequency in 5 dimensions, and base current and base quality in 15 dimensions. Thus, features of the 5-based fragment have 45 dimensions in total.

In model training, other state-of-the-art comparison experimental models are trained using the feature extraction methods mentioned above. In the training of the MWNB model, the features extracted from the original data are input to Bagging-LSTM for feature extraction, and the mapped features are input to Bagging-LightGBM for feature extraction. The final extracted bagging features are used in the attention neural model to discriminate whether m6A modifications occur. All experiments were implemented on an Intel XeonE5-2630 v4 @ 2.20GHzz CPU and NVIDIA GeForce RTX 2080 Ti ArchLinux. All models are implemented in Scikit-learn and Pytorch.

In tuning the parameters, we use the leave-one-out 5-fold cross-validation to develop and evaluate the model ability. First, we randomly split the dataset into six folds, and each fold contains an almost equal number of samples. The data in the test set is one of the six-folds, and the training and validation sets are the remaining five folds. In the training process, four folds are used, and the fifth fold uses for testing. The process is repeated five times, picking the different folds for testing each time, and the other four folds are used in training. The data in the test set is one of the six-folds, and the training and validation sets are the remaining five folds.

C. Evaluation metrics

To assess the performance of models, we use 6 metrics: Accuracy (acc), precision (pre), sensitivity (se), specificity (sp), F1-score, G mean_1, and G mean_2. The accuracy is $Accuracy = \frac{(T_P+T_N)}{(P+N)}$, and precision is $Precision = \frac{T_P}{(T_P+F_P)}$. Sensitivity and specificity is $Sensitivity =$

TABLE II

AVERAGE CLASSIFICATION PERFORMANCE OF LEAVE-ONE-OUT 5-FOLD CROSS-VALIDATION OF ALL MACHINE LEARNING METHODS IN OUR m^6A MODIFICATION TASK. THE PERFORMANCE OF MODELS IS SHOWN IN TESTING DATASET AND 5-FOLD CROSS VALIDATION DATASET.

Model	acc (%)	pre (%)	sp (%)	se (%)	F1-score (%)	G mean$_1$ (%)	G mean$_2$(%)	AUC
Testing dataset								
MWNB (Ours)	**97.85**	**98.37**	**97.20**	**98.41**	**98.39**	**98.39**	**97.80**	**0.997**
SVM (linear)	79.63	82.69	75.24	84.07	83.37	83.38	79.53	0.871
SVM (rbf)	77.68	79.88	74.34	81.06	80.46	80.47	77.63	0.873
SVM (poly)	78.19	75.31	84.22	72.09	73.66	73.68	77.92	0.859
SVM (sigmoid)	73.28	73.25	73.67	72.90	73.07	73.07	73.28	0.806
DT [33]	81.64	82.52	79.93	83.32	82.92	82.92	81.61	0.816
RF [34]	91.15	93.95	87.82	94.43	94.19	94.19	91.07	0.970
ET [35]	94.44	96.65	91.99	96.86	96.75	96.75	94.39	0.989
RNN	87.33	94.47	95.44	79.1	86.10	86.44	86.89	0.950
GRU	89.34	90.22	90.59	88.07	89.13	89.14	89.32	0.951
LSTM	91.79	95.61	87.48	96.04	95.82	95.82	91.66	0.977
[19]	90	-	-	-	-	-	-	0.944
[34]	78.58	-	79.65	-	-	-	-	-
Leave-one-out 5-fold cross validation dataset								
MWNB (Ours)	**97.89 ± 0.15**	**98.23 ± 0.18**	**98.25 ± 0.18**	**97.52 ± 0.30**	**97.87 ± 0.17**	**97.87 ± 0.17**	**97.88 ± 0.15**	**0.997 ± 0.0004**
SVM (linear)	79.08±0.32	80.77±0.34	80.95±0.28	77.21±0.42	78.95±0.37	78.97±0.37	79.06±0.34	0.837±0.0012
SVM (rbf)	77.54±0.17	79.97±0.19	80.27±0.18	74.80±0.34	77.31±0.22	77.35±0.22	77.49±0.18	0.830±0.0010
SVM (poly)	78.38±0.29	80.46±0.24	80.70±0.19	76.04±0.48	78.20±0.35	78.23±0.35	78.34±0.31	0.834±0.0012
SVM (sigmoid)	73.55±0.15	75.59±0.35	75.81±0.29	71.27±0.41	73.38±0.37	73.40±0.37	73.51±0.34	0.785±0.0015
DT [33]	81.27±0.34	82.22±0.21	83.29±0.17	80.23±0.60	81.21±0.38	81.22±0.38	81.26±0.35	0.844±0.0009
RF [34]	91.88±0.17	94.34±0.28	94.68±0.29	89.06±0.27	91.62±0.19	91.66±0.19	91.83±0.17	0.976±0.0009
ET [35]	94.35±0.08	96.37±0.18	96.55±0.20	92.13±0.25	94.21±0.12	94.23±0.12	94.32±0.09	0.984±0.0007
RNN	87.63±0.22	90.04±0.25	90.34±0.22	84.91±0.39	87.40±0.28	87.44±0.28	87.58±0.24	0.931±0.0010
GRU	89.16±0.10	91.16±0.13	91.35±0.16	86.96±0.26	89.01±0.14	89.04±0.14	89.13±0.11	0.936±0.0008
LSTM	90.08±0.29	91.57±0.35	91.70±0.32	88.44±0.37	89.98±0.33	89.99±0.33	90.06±0.30	0.940±0.0011

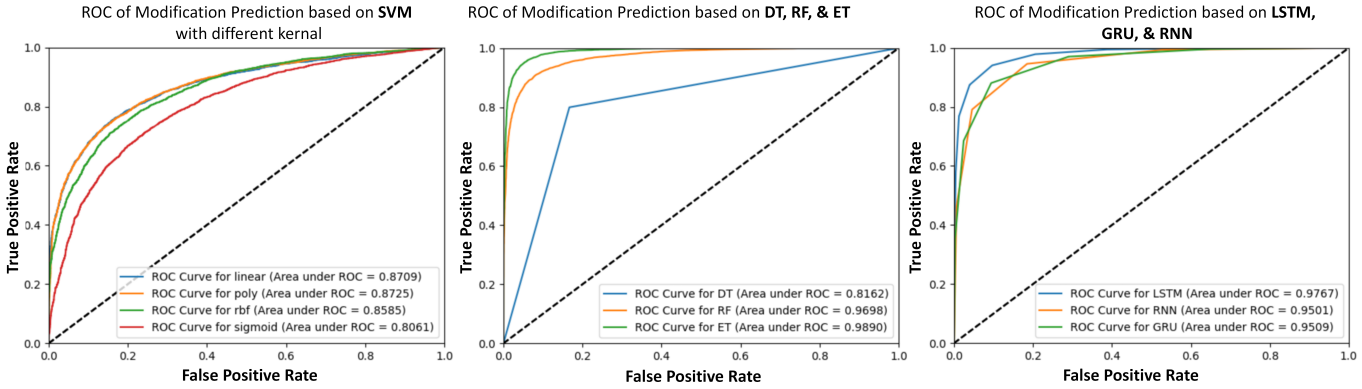

Fig. 4. The best-performing models' ROC chart. The ROC curve of SVM [19] with several kernels, such as linear, poly, rbf, and sigmoid, is shown on the left. The ROC chart of Decision Tree (DT) [33], Random Forest (RF) [34], and Extremely Random Trees (ET) [35] is shown in the middle figure, while the ROC of RNN, LSTM, and GRU models is shown in the right figure.

$\frac{T_P}{(T_P+F_N)}$ and $Specificity = \frac{T_N}{(T_N+T_P)}$ relatively. In above equations, T_P represents the prediction result of the model is a positive example (P), but in fact the judgment result is right (T), T_N stands for the prediction result of the model is a negative example (N), but in fact the judgment result is right (T), F_P represents the prediction result of the model is a positive example (P), but in fact the judgment result is wrong (F), F_N is the prediction result of the model is a negative example (N), but in fact the judgment result is wrong (F). Also, $F1score$, $Gmean_1$, and $Gmean_2$ are calculated in evaluation of models:

$$F1score = 2 \times \frac{Precision \times Sensitivity}{(Precision + Sensitivity)} \tag{5}$$

$$Gmean_1 = \sqrt{Sensitivity \times Precision} \tag{6}$$

$$Gmean_2 = \sqrt{Sensitivity \times Specificity} \tag{7}$$

A basic evaluation metric for assessing classification performance is the receiver operator characteristic curve (ROC).The area under ROC (AUC) also can show the model performance. The calculation formula of AUC is as follows:

$$AUC = \frac{\sum_{ins_i \in positiveclass} rank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{8}$$

where $rank_{ins_i}$ represents the number of the i-th sample. (Probability scores are ranked from small to large, ranked in the rank position), M is the number of positive samples, and N is the number of negative samples.

D. Comparison results of MWNB with other models

The features from raw sequencing data and extracted mapping features are both used as input features in our MWNB model. The $m6A$ RNA modification categorization findings are obtained using two models (Bagging-LightGBM for features from raw sequencing and Bagging-LSTM for extracted mapping features). Because of the peculiarities of our MWNB model, we first compare it against classic machine learning models. Traditional machine learning models do not choose a more appropriate feature extraction approach for the difference of features, which is the most significant distinction between our MWNB model and them. We also compare our MWNB model to a strategy based on ensemble learning to make more comprehensive comparisons. Additionally, deep learning-based methods also compare to our MWNB model as well.

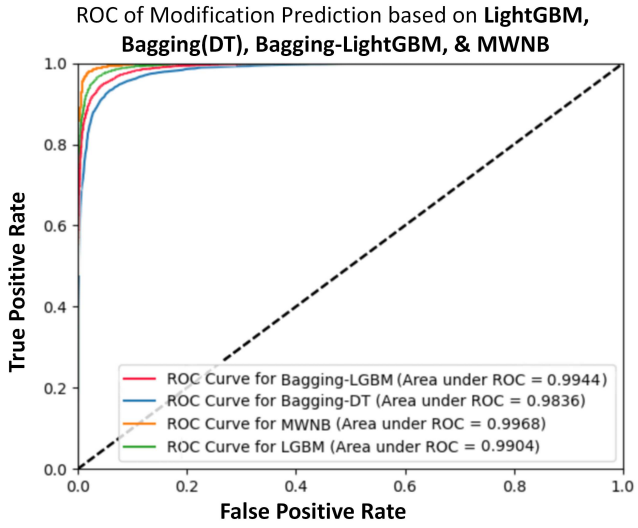

Fig. 5. The AUC figure shows the performance of LightGBM, Bagging-DT, Bagging-LightGBM, and our MWNB.

Firstly, we compare our MWBN to classic machine-learning models. The evaluation metrics for all the best models produced through parameter tuning (Section IV.D) are shown in Table II. The results of the above-mentioned model training and exploration of the optimal model indicate that: simple machine learning models such as DT have the advantage of being fast to train and easily interpretable; however, the classification accuracy obtained is insufficient; the SVM model's training time is lengthy. While the model is sophisticated, the precision gained in this study is also insufficient. For the two types of models discussed above, the model classification accuracy attained on this task is approximately 80%. We employ two ensemble learning models: RF and ET, both of which perform well on this challenge. These approaches achieve accuracy of approximately 91% to 94%. We estimate the AUC of each best model and provided their ROC graphs in Fig. 4. As illustrated in Fig 4, all ensemble learning methods (RF, ET) achieved an AUC value greater than 0.95.

In addition to comparing our MWNB model to regularly used classical machine learning methods, we compared it

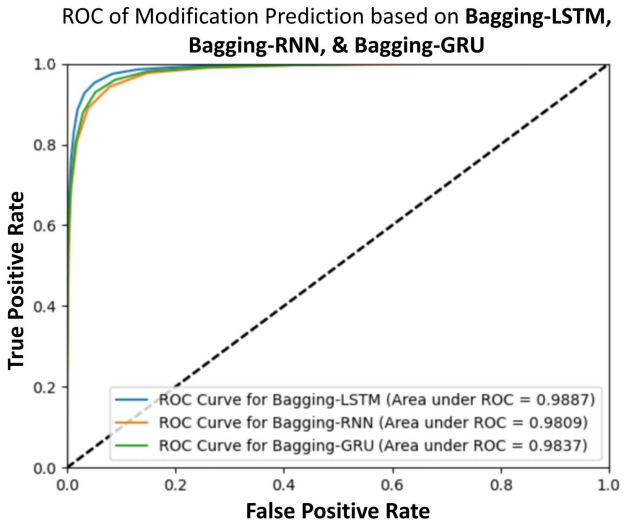

Fig. 6. The AUC figure shows the performance of Bagging-LSTM, Bagging-GRU, and Bagging-RNN.

TABLE III
AVERAGE CLASSIFICATION PERFORMANCE OF LEAVE-ONE-OUT 5-FOLD CROSS-VALIDATION OF LIGHTGBM [30], BAGGING-DT [36], BAGGING-LIGHTGBM, AND OUR MWNB MODELS. THE RESULTS SHOW MODELS' PERFORMANCE IN TESTING DATASET.

Model	acc (%)	pre (%)	sp (%)	se (%)
LightGBM [30]	94.25	97.56	97.77	90.67
Bagging-DT [36]	93.38	95.39	91.06	95.67
Bagging-LightGBM	96.02	97.68	97.80	94.21
MWNB (Ours)	97.85	98.37	97.20	98.41

to the unique ensemble learning model. The table below compares our proposed MWNB model to its base learner LightGBM, Bagging-DT, and Bagging-LightGBM models. The AUC of these models is depicted in Fig. 5. The best AUC for these models is 0.9968, which our MWNB model achieves.

Additionally, we compare the performance of our MWNB models to that of deep learning models that employ only raw sequence data. In our dataset, we use LSTM, GRU, and RNN. Additionally, these models outperform DT and SVM in terms of performance. Additionally, the RNN and upgraded RNN (GRU and LSTM) models have AUC values greater than 0.95. LSTM, GRU, and RNN performance details are provided in Fig. 4 and Table II. Additionally, we integrated these models using the bagging technique. The performance of Bagging-LSTM, Bagging-GRU, and Bagging-RNN is illustrated in

TABLE IV
AVERAGE CLASSIFICATION PERFORMANCE OF LEAVE-ONE-OUT 5-FOLD CROSS-VALIDATION OF BAGGING-LSTM, BAGGING-RNN, AND BAGGING-GRU. THE RESULTS SHOW MODELS' PERFORMANCE IN TESTING DATASET.

Model	acc (%)	pre (%)	sp (%)	se (%)
Bagging-LSTM	94.77	96.63	92.70	96.81
Bagging-RNN	92.58	95.75	89.00	96.11
Bagging-GRU	93.86	94.62	92.92	94.80

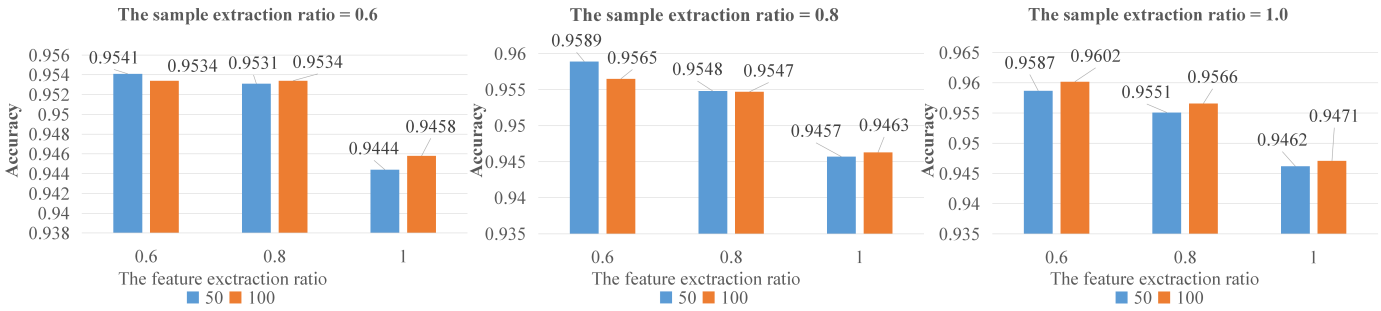

Fig. 7. Accuracy for different paramters in the Bagging-LightGBM model of testing dataset. The orange bar shows the result that the number of base learner is 50, and the blue bar illustrate accuracy which the number of base learner is 100 obtain in different sample ratio and various feature ratio.

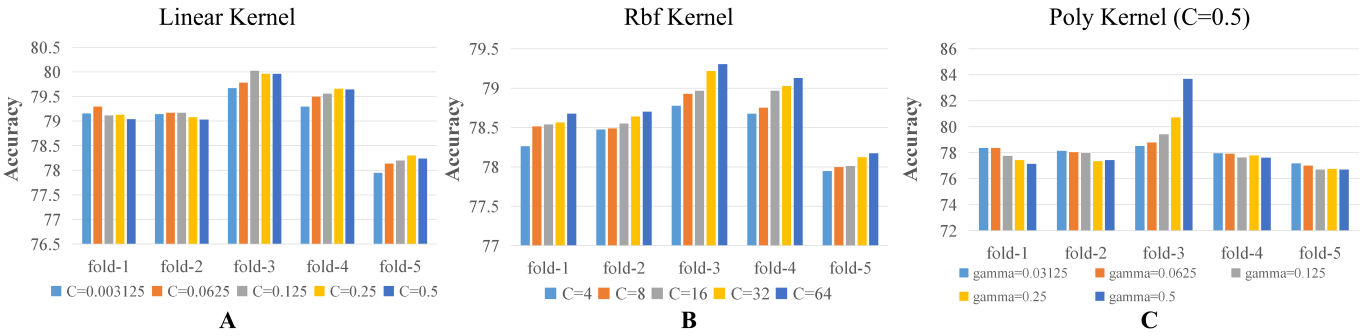

Fig. 8. Average classification performance of SVM's leave-one-out 5-fold cross validation with different penalty coefficient parameter C in 5-fold cross validation datasets (Kernal: Linear, RBF, and Poly).

Table IV and Fig. 6. The best AUC value for these models is 0.9887, which was obtained using Bagging-LSTM.

As a result, the most randomized MWNB approach produces the best outcomes. The LightGBM model has been adjusted and enhanced in numerous ways as an enhancement to the gradient boosting decision tree (GBDT). It offers the advantages of being more efficient in training, having a better degree of accuracy, and processing enormous amounts of data. By fusing many LightGBM models together using the bagging approach, we may further enhance the model's generalization and accuracy. Additionally, the LSTM excels in extracting sequence relationships. The Bagging-LSTM model is used in our MWNB to extract critical information from raw sequencing data. Finally, we merge the Bagging-LSTM and Bagging-LightGBM models using the attention neural network. In our assignment, our MWNB model performs optimally. Our MWNB model obtains the best performance in our task. The MWNB's accuracy is 97.85%, precision is 98.37%, sensitivity is 97.20%, and specificity is 98.41%. The AUC value, which is the highest, is 0.997 obtained by the MWNB method.

E. Parameters tunning

The initial step of the experiment is to determine the proper parameter values for Bagging-LightGBM. To begin, we require a parameter option for the base LightGBM learner. LightGBM has a plethora of parameters that must be selected. The following diagram illustrates the procedures involved in selecting appropriate parameters:

1) We begin by setting the starting parameters. The grid search method determines the learning rate and the number of iterations. The learning rate is between 0.01 and 0.5, and the number of iterations is between 100 and 2000;
2) Then, we investigate the optimal number of leaves between 100 and 500;
3) Finally, the parameters for regularization λL_1 and λL_2 are established. The range of $lambda L_1$ and λL_2 is approximately 1e-5 to 1.0.

The best model achieves an accuracy of 96.55% when the LightGBM parameters are selected. The best model has a learning rate of 0.1, a total of 1400 iterations, a total of 350 leaves, a λL_1 of 1e-3, and a λL_2 of 1e-3. Fig. 7 illustrates the accuracy, precision, sensitivity, and specificity of LightGBM over a range of iterations and leaf counts.

Second, we employ the same method (grid search) to investigate other bagging parameters. Five parameters must be determined throughout the bagging process. The following diagram illustrates the procedures involved in selecting appropriate parameters:

1) The total number of basic learners to be integrated is predetermined. The examined range of base learner numbers is 50 to 200;
2) The sample extraction ratio and feature extraction ratio are next investigated. Both are between 0.5 and 1.0;
3) Finally, we determine the sampling procedure for the sample subset and the feature subset.

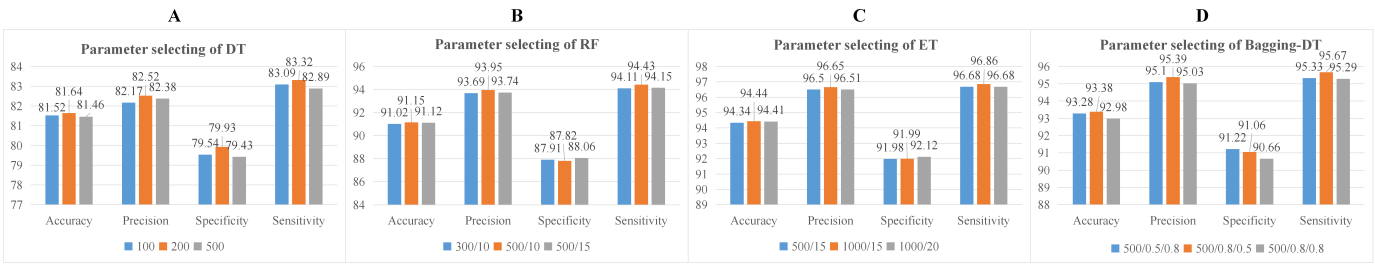

Fig. 9. Accuracy for different paramters in the Bagging-LightGBM model. The orange bar shows the result that the number of base learner is 50, and the blue bar illustrate accuracy which the number of base learner is 100 obtain in different sample ratio and various feature ratio.

The best model achieves an accuracy of 96.02% when the Bagging-LightGBM parameters are chosen. The best bagging model's base learner is LightGBM. We chose 100 base learners for parameter selection, a sample extraction ratio of 1.0, and a feature extraction ratio of 0.6. Additionally, we employ non-replacement sampling to create sample subsets and replacement sampling to create feature subsets. Fig. 7 displays the accuracy of LightGBM over a range of iterations and leaf counts.

We also employ the grid search technique to identify the parameters of Bagging-LSTM in the second round of parameter tuning. To begin, we experiment with varying the number of concealed cells N: from 10 to 200. Additionally, we experiment with various batch sizes (4 to 64) and learning rates (1e-5 to 1e-1). The optimal LSTM parameters are as follows: N=50, batch size=16, learning rate=0.001. We use the same method in Bagging as we do in Bagging-LightGBM. The most accurate Bagging-LSTM model achieves a precision of 94.77%. We chose a base learner count of 20, a sample extraction ratio of 0.8, and a feature extraction ratio of 0.5 for parameter selection. Additionally, we employ non-replacement sampling to create sample subsets and replacement sampling to create feature subsets. Additionally, we investigated the attention neural network's parameters using the best Bagging-LightGBM and Bagging-LSTM. We experiment with various batch sizes (4 to 64) and learning rates (0.00001 to 0.1). The optimal model is defined by the following hyperparameters: batch size = 16, learning rate = 0.001.

We also employ the grid search technique in the three parts of parameter tuning to identify parameters for other comparable classification methods, such as SVM, DT, ET, and RF. We can attain a maximum average classification accuracy of 79.63% using the SVM with a linear kernel after running five-fold cross-validation. The Fig. 8 illustrates the model performance for each fold in linear, rbf, and poly kernel function when the penalty coefficient parameter C is varied.

The Fig. 9 A illustrates the average classification performance of 5-fold cross-validation for DTs with varying maximum depths. The experimental findings demonstrate that when the decision tree's maximum depth is 200, the ideal classification accuracy rate of 81.64% and precision rate of 82.52% are reached. Fig. 9 B displays the average classification performance for various maximal feature counts and subtree counts. When a maximum of ten features are utilized and a maximum of 500 subtrees are employed, the ideal accuracy

TABLE V
THE CLASSIFICATION PERFORMANCE IN DIFFERENT FEATURE SELECTION USING IN MWNB MODEL.

features	Accuracy %	Precision %	Specificity %	Sensitivity %
Q	92.70	94.14	90.88	94.48
Mis	89.97	93.28	85.90	93.95
Ins	52.81	71.87	7.44	97.15
Del	85.86	89.57	80.81	90.80
Q&Mis	97.31	97.33	97.23	97.39
Q&Ins	93.38	94.62	91.81	94.90
Q&Del	97.03	96.97	97.03	97.04
Mis&Ins	91.90	94.80	88.46	95.26
Mis&Del	96.83	97.18	96.39	97.27
Ins&Del	89.13	91.96	85.49	92.70
Q&Mis&Ins	97.41	97.42	97.35	97.48
Q&Mis&Del	97.17	97.08	97.20	97.14
Q&Ins&Del	97.17	97.14	97.13	97.21
Mis&Ins&Del	97.11	97.63	96.50	97.71
Q&Mis& Ins&Del	97.85	98.37	97.20	98.41

is obtained: 91.15 % of the average classification accuracy rate and 93.95 % of the average classification precision rate. Fig. 9 C plots the average classification performance of 5-fold cross-validation with varying maximum feature counts and subtree counts for ET. On ET, the highest accuracy is obtained when the maximum number of features is 15 and the number of subtrees is 1000: 94.44% average classification accuracy rate, 96.65% average classification precision rate.

When we used the Bagging approach, we started with the DT base model, which is a simple and fast classifier. When the number of base learners is 500, the proportion of samples used for each training of the base learner is 80%, the proportion of features used for each training of the base learner is 50%, the samples are drawn without replacement, and the features are drawn with replacement, the best accuracy is 93.38%, and the best precision rate is 95.39%.

F. Feature Selection

We compare the results of various feature selections for each model. When only current intensity features are used to create models, the best model performance is 85.71% accuracy. When just quality features are used to create models, the best performance is an accuracy rate of 84.31%. When two types of feature development models are used, the optimum model performance is 94.77% accuracy. To help you determine the optimal strategy to pick features, we show the impacts

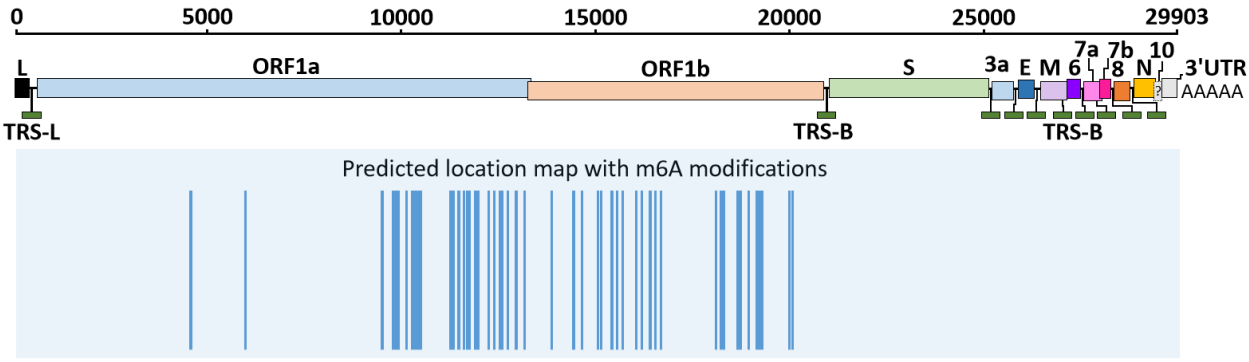

Fig. 10. Schematic diagram of the SARS-CoV-2 Genome. At the top of the picture is the genome's axis (1-29903) of SARS-CoV-2. The bars in the middle represent the different genes of the genome anchored in the corresponding regions. The lower part of the image is the predicted density map of $\boldsymbol{m^6A}$ sites. The higher part(like "peak") means that the $\boldsymbol{m^6A}$ predicted sites are densely distributed.

of several selection measures in Table V. Table V compares the classification performance of the Bagging model when different base feature combinations are used. We examined 16 different feature selection combinations (Table V). When MWNB performances are compared, it is clear that combining all features produces the best classification effect. It achieves the highest accuracy of 97.85% when compared to other combinations.

G. Bagging-LightGBM Model used in COVID-19

To assess our model's performance with respect to $m6A$ modifications inside certain sequence motifs (RRACH). We performed $m6A$ prediction on SARS-CoV-2 RNA data, identified several locations with high confidence, and then examined their possible biological importance, which will aid future research on COVID-19 focused medication development and infection process. Further research will be required to enable single-read detection of RNA modifications and to extend our findings to other RNA alterations.

We studied SARS-CoV-2 DRS data from Korea using the same upstream procedure as previously described. Thus, we used the Korean vero-infected (host and SARS-CoV-2) dataset to demonstrate our female's ability to detect $m6A$ alteration. 65.4% (Fig. 10) of readings were mapped to SARS-CoV-2, indicating that Kim's sample is dependable and reproducible. We observed a modification score draft with noise that was distributed uniformly across the genome. To minimize false positives, we selected a probability threshold of $m6A$ to verify the results' accuracy. The other major peaks suggested the presence of a significant amount of $m6A$. The majority of the high probability loci were discovered to be located in the ORF1b region of the genome. ORF1b encodes a nonstructural protein (nsp) that is required for viral transcription, replication, and inhibition of host immune response and gene expression. Antiviral therapy aims to inhibit RNA-dependent RNA polymerase [37]. Our findings imply that the nonstructural protein mRNA of NCV is highly methylated. It could be connected to RNA stability and amino acid sequence mutations. The inclusion of a synthetic inhibitor of $m6A$ reduces the influenza virus' replication [38]. The model results aid in our understanding of the SARS-life CoV-2's activities at a deeper level, which will aid in the development of targeted antiviral medications.

V. Discussion

This work presented a machine learning-based method to recognize patterns of m^6A in nanopore direct RNA sequencing. Features derived from the raw-signals and their mapping information were utilized as the model input. Ion channels disturbed by m^6A modification was recorded as nanopore raw signals("squiggle") to determine the links between the difference in current value and different bases in nanopores. Several classifiers were utilized, namely: SVM, RF, ensemble learning (RF, ET, and Bagging) and our MWNB to classify the m^6A and normal base based on different features of the positions while mapping the reads to reference sequence. Based on the machine learning techniques and the extracted derivative features, an Integrated framework was developed to detect m^6A modification based on features produced by sequencing patterns. The MWNB model proposed in this paper is a generalized framework for classification detection by targeted feature extraction (Bagging-LightGBM for mapping features and Bagging-LSTM for signal statistical features) to perform signal difference classification of sequencing data. The model is not only applicable to the detection of m6A modifications in RNA sequencing but from the modeling perspective, our model can be directly used in the detection of other modifications with only simple migration.

There is an artificial sequence [19] of about 10kb length was used in this project. Generally, The features used are not the original electrical signal but after base calling and sequence alignment for each site statistic. This finding is consistent with previous studies in which m^6A had a significant effect on the base quality and alignment results. Huanle Liu et al. [19] conducted a study that verified that these features could be used to accurately classify a given site into "m^6A-modified" or "unmodified". Wongsurawat et al. [5] also found a systematic "base call error" difference between modified samples and unmodified samples. Based on this discovery, they developed software called "eligos". The original electrical signals of nanopore direct RNA sequencing and the characteristics derived from the downstream analysis have been widely studied

[26] as an indicator of changes in the detection of chemical modification of bases. Previous studies have confirmed that the base quality [24] (the correct probability given by the base interpretation model) is used as a reverse modification index.

This discovery [5], [24] has been verified by laboratory experiments based on immunoprecipitation. It has been proved that with the decrease of base quality, the probability of base interpretation error increases, and the chemical structure of the corresponding base is different from that of the normal base, which may be due to chemical modification. Therefore, according to the existing literature, the original current value is considered the basic index to judge the nanopore base sequence. Based on this basis, base quality and alignment results are considered the specific expression of each site's methylation degree. In order to further optimize the performance of our model, we limit the motif of the input site to adapt to the context sequence of m^6A in the real environment. Many studies have confirmed that in different biological samples, the modification of m^6A often follows the same motif, which may be related to the relationship between m^6A related proteins.

In addition to the novelty and advantages of this proposal, there are also some disadvantages. In limitations from the data set, our model training does not use real-life mRNA for sequencing, and we use artificial sequences for feature extraction, affecting model prediction behavior. These samples in the dataset are mixed in the laboratory environment and cannot obtain the real label. Moreover, the location of m^6A cannot be accurately identified due to technical limitations. There is also no experimental verification of the predicted m^6A specific locations in the SARS-CoV-2 data. Furthermore, considering the model's limitations, our MWNB model uses a complex ensemble learning approach to separate feature extraction and final prediction, making the workflow more complex. In addition, we propose feature extraction approaches for different features specific to the features' nature, making the model more complex.

Although our model has achieved good detection results, some erroneous predictions still exist, which may be due to the limitations mentioned above of the data and model. First, our dataset was synthesized artificially using in vitro transcription techniques, while the actual predictions used for the model are naturally occurring in the organism. Although the chemical structures of the two are identical in terms of currently available theories, there may be potential systematic differences. Moreover, the current sampling rate for nanopore sequencing is not high enough, with the number of samples obtained per base ranging between 8-9 discrete current observations [39]. Such low-dimensional data are challenging to distinguish between the occurring and non-occurring m6A modifications. Also, compared to the multi-electrode, multi-channel nature of EEG and ECG techniques, nanopore sequencing has only one channel [40], and the number of features in the data itself is too low. We hope that in the near future Oxford Nanopore UK will provide the resolution and sampling accuracy of the device to provide higher dimensional feature information for improving the performance of the model.

In our future work, we will further improve our modification detection framework in two ways. First, from the data side, we will use RNA/DNA sequencing data and corresponding modification labels in real scenarios to validate our model's performance further. In addition, from the footing of model improvement, we will consider end-to-end learning to simplify the complexity of model training and achieve better feature extraction by some feature extraction and enhancement means, including attention mechanism in the context of end-to-end network learning.

VI. Conclusions

This article proposes a Bagging-LightGBM model for m^6A modification detection. In the proposed Bagging-LightGBM, we combine speed-up LightGBM models and Bagging strategy to form a fusion model. The Bagging-LightGBM model is trained and tested on artificially synthesized sequences, which obtains the best performance of 97.85% of accuracy. We used state-of-art machine-learning models such as SVM, DT, RF, ET, and Bagging in our dataset to compare our model ability. To ensure models' performance, we use the same grid search algorithm and 5-fold cross-validation on other state-of-art models and our Bagging-LightGBM. Our Bagging-LightGBM model outperforms other methods. More importantly, we applied the optimal m^6A modification detection model (Bagging-LightGBM) to the SARS-COV-2 sequencing data to obtain the possible m^6A modification site information on SARS-COV-2. The prediction results will help us to find possible location of gene mutation.

References

[1] P. Dashraath, J. L. J. Wong, M. X. K. Lim, L. M. Lim, S. Li, A. Biswas, M. Choolani, C. Mattar, and L. L. Su, "Coronavirus disease 2019 (covid-19) pandemic and pregnancy," *American Journal of Obstetrics and Gynecology*, vol. 222, no. 6, pp. 521–531, 2020.

[2] M. Jeyanathan, S. Afkhami, F. Smaill, M. S. Miller, B. D. Lichty, and Z. Xing, "Immunological considerations for covid-19 vaccine strategies," *Nature Reviews Immunology*, vol. 20, no. 10, pp. 615–632, 2020.

[3] D. Kim, J.-Y. Lee, J.-S. Yang, J. W. Kim, V. N. Kim, and H. Chang, "The architecture of sars-cov-2 transcriptome." *Cell*, vol. 181, no. 4, 2020.

[4] C. G. Ziegler, S. J. Allon, S. K. Nyquist *et al.*, "Sars-cov-2 receptor ace2 is an interferon-stimulated gene in human airway epithelial cells and is detected in specific cell subsets across tissues." *Cell*, vol. 181, no. 5, 2020.

[5] P. Jenjaroenpun, T. Wongsurawat, T. D. Wadley, T. M. Wassenaar, J. Liu, Q. Dai, V. Wanchai, N. S. Akel, A. Jamshidi-Parsian, A. T. Franco, G. Boysen, M. L. Jennings, D. W. Ussery, C. He, and I. Nookaew, "Decoding the epitranscriptional landscape from native rna sequences." *Nucleic Acids Research*, 2020.

[6] Y. Liu, Y. You, Z. Lu, J. Yang, P. Li, L. Liu, H. Xu, Y. Niu, and X. Cao, "N6-methyladenosine rna modification–mediated cellular metabolism rewiring inhibits viral replication," *Science*, vol. 365, no. 6458, pp. 1171–1176, 2019.

[7] D. Dominissini, S. Moshitch-Moshkovitz, S. Schwartz, M. Salmon-Divon, L. Ungar, S. Osenberg, K. Cesarkas, J. Jacob-Hirsch, N. Amariglio, M. Kupiec, R. Sorek, and G. Rechavi, "Topology of the human and mouse m6a rna methylomes revealed by m6a-seq," *Nature*, vol. 485, no. 7397, pp. 201–206, 2012.

[8] N. S. Gokhale, A. B. McIntyre, M. J. McFadden, A. E. Roder, E. M. Kennedy, J. A. Gandara, S. E. Hopcraft, K. M. Quicke, C. Vazquez, J. Willer, O. R. Ilkayeva, B. A. Law, C. L. Holley, M. A. Garcia-Blanco, M. J. Evans, M. S. Suthar, S. S. Bradrick, C. E. Mason, and S. M. Horner, "N6-methyladenosine in flaviviridae viral rna genomes regulates infection." *Cell Host and Microbe*, vol. 20, no. 5, pp. 654–665, 2016.

[9] B. Linder, A. V. Grozhik, A. O. Olarerin-George, C. Meydan, C. E. Mason, and S. R. Jaffrey, "Single-nucleotide-resolution mapping of m6a and m6am throughout the transcriptome," *Nature Methods*, vol. 12, no. 8, pp. 767–772, 2015.

[10] E. L. van Dijk, Y. Jaszczyszyn, D. Naquin, and C. Thermes, "The third revolution in sequencing technology," *Trends in Genetics*, vol. 34, no. 9, pp. 666–681, 2018.

[11] M. H. Stoiber, J. Quick, R. Egan, J. E. Lee, S. E. Celniker, R. Neely, N. Loman, L. Pennacchio, and J. B. Brown, "De novo identification of dna modifications enabled by genome-guided nanopore signal processing," *bioRxiv*, p. 94672, 2017.

[12] Q. Liu, D. C. Georgieva, D. Egli, and K. Wang, "Nanomod: a computational tool to detect dna modifications using nanopore long-read sequencing data." *BMC Genomics*, vol. 20, no. 1, p. 78, 2019.

[13] P. Liu, B. Fu, S. X. Yang, L. Deng, X. Zhong, and H. Zheng, "Optimizing survival analysis of xgboost for ties to predict disease progression of breast cancer," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 148–160, 2021.

[14] Y. Tang, S. M. Brown, J. Sorensen, and J. B. Harley, "Physiology-informed real-time mean arterial blood pressure learning and prediction for septic patients receiving norepinephrine," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 181–191, 2021.

[15] A. Moniri, D. Terracina, J. Rodriguez-Manzano, P. H. Strutton, and P. Georgiou, "Real-time forecasting of semg features for trunk muscle fatigue using machine learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 2, pp. 718–727, 2021.

[16] G. Noaro, G. Cappon, M. Vettoretti, G. Sparacino, S. D. Favero, and A. Facchinetti, "Machine-learning based model to improve insulin bolus calculation in type 1 diabetes therapy," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 247–255, 2021.

[17] L. Lu, Y. Tan, M. Klaic, M. P. Galea, F. Khan, A. Oliver, I. Mareels, D. Oetomo, and E. Zhao, "Evaluating rehabilitation progress using motion features identified by machine learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 4, pp. 1417–1428, 2021.

[18] A. M. Smith, M. Jain, L. Mulroney, D. R. Garalde, and M. Akeson, "Reading canonical and modified nucleobases in 16s ribosomal rna using nanopore native rna sequencing." *PLOS ONE*, vol. 14, no. 5, 2019.

[19] H. Liu, O. Begik, M. C. Lucas, J. M. Ramirez, C. E. Mason, D. Wiener, S. Schwartz, J. S. Mattick, M. A. Smith, and E. M. Novoa, "Accurate detection of m6a rna modifications in native rna sequences," *Nature Communications*, vol. 10, no. 1, p. 4079, 2019.

[20] A. B. R. McIntyre, N. Alexander, K. Grigorev, D. Bezdan, H. Sichtig, C. Y. Chiu, and C. E. Mason, "Single-molecule sequencing detection of n6-methyladenine in microbial reference materials," *Nature Communications*, vol. 10, no. 1, pp. 579–579, 2019.

[21] D. Deamer, M. Akeson, and D. Branton, "Three decades of nanopore sequencing," *Nature Biotechnology*, vol. 34, no. 5, pp. 518–524, 2016.

[22] R. M. Leggett and M. D. Clark, "A world of opportunities with nanopore sequencing." *Journal of Experimental Botany*, vol. 68, no. 20, pp. 5419–5429, 2017.

[23] J. J. Kasianowicz, E. Brandin, D. Branton, and D. W. Deamer, "Characterization of individual polynucleotide molecules using a membrane channel," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 24, pp. 13 770–13 773, 1996.

[24] M. T. Parker, K. Knop, A. V. Sherwood, N. J. Schurch, K. Mackinnon, P. D. Gould, A. J. W. Hall, G. J. Barton, and G. G. Simpson, "Nanopore direct rna sequencing maps the complexity of arabidopsis mrna processing and m6a modification." *eLife*, vol. 9, pp. 1–35, 2020.

[25] A. C. Rand, M. Jain, J. M. Eizenga, A. Musselman-Brown, H. E. Olsen, M. Akeson, and B. Paten, "Mapping dna methylation with high-throughput nanopore sequencing," *Nature Methods*, vol. 14, no. 4, pp. 411–413, 2017.

[26] L. Xu and M. Seki, "Recent advances in the detection of base modifications using the nanopore sequencer." *Journal of Human Genetics*, vol. 65, no. 1, pp. 25–33, 2020.

[27] D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello, and J. J. Collins, "Next-generation machine learning for biological networks," *Cell*, vol. 173, no. 7, pp. 1581–1592, 2018.

[28] P. Ni, N. Huang, Z. Zhang, D.-P. Wang, F. Liang, Y. Miao, C.-L. Xiao, F. Luo, and J. Wang, "Deepsignal: detecting dna methylation state from nanopore sequencing reads using deep-learning." *Bioinformatics*, vol. 35, no. 22, pp. 4586–4595, 2019.

[29] R. R. Wick, L. M. Judd, and K. E. Holt, "Performance of neural network basecalling tools for oxford nanopore sequencing." *Genome Biology*, vol. 20, no. 1, p. 129, 2019.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: a highly efficient gradient boosting decision tree," in *NIPS'17 Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, 2017, pp. 3149–3157.

[31] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. CVPR*, 2019, pp. 3141–3149.

[32] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," in *Proc. MICCAI*, vol. 11070, 2018, pp. 421–429.

[33] W. Lu, Y. Cao, H. Wu, H. Huang, and Y. Ding, "Research on rna secondary structure prediction based on decision tree," in *Intelligent Computing Theories and Application*, D.-S. Huang, K.-H. Jo, and Z.-K. Huang, Eds. Cham: Springer International Publishing, 2019, pp. 430–439.

[34] "Detecting n6-methyladenosine sites from rna transcriptomes using random forest," *Journal of Computational Science*, vol. 47, p. 101238, 2020.

[35] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, p. 3–42, 2006.

[36] S. Gupta and M. Kumar, "Forensic document examination system using boosting and bagging methodologies," *Soft Computing*, vol. 24, p. 5409–5426, 2020.

[37] L. Subissi, I. Imbert, F. Ferron, A. Collet, B. Coutard, E. Decroly, and B. Canard, "Sars-cov orf1b-encoded nonstructural proteins 12–16: Replicative enzymes as antiviral targets," *Antiviral Research*, vol. 101, pp. 122 – 130, 2014.

[38] D. G. Courtney, E. M. Kennedy, R. E. Dumm, H. P. Bogerd, K. Tsai, N. S. Heaton, and B. R. Cullen, "Epitranscriptomic enhancement of influenza a virus gene expression and replication," *Cell Host and Microbe*, vol. 22, no. 3, pp. 377 – 386.e5, 2017.

[39] S. Marcus and B. James, "BasecRAWller: streaming nanopore basecalling directly from raw signal," *bioRxiv*, vol. abs/05/01/133058, 2017. [Online]. Available: https://doi.org/10.1101/133058

[40] L. A. H, D. I. M, and G. J. H, "Chapter fifteen - subangstrom measurements of enzyme function using a biological nanopore, sprnt. in m. spies & y. r. chemla (eds.)," *Methods in Enzymology*, vol. 582, pp. 387—414, 2017.