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

Ruhan Liu, Liang Ou, Bin Sheng, *Member, IEEE*, Pei Hao, Ping Li, *Member, IEEE*, Xiaokang Yang, *Fellow, IEEE*, Guangtao Xue, *Member, IEEE*, Lei Zhu, *Member, IEEE*, Yuyang Luo, Ping Zhang, *Senior Member, IEEE*, Po Yang, *Senior Member, IEEE*, Huating Li, and David Dagan Feng, *Life Fellow, IEEE*

Abstract*—*Objective: **The m6A modification is the most common ribonucleic acid (RNA) modification, playing a role in prompting the virus's gene mutation and protein structure changes in the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). Nanopore single-molecule direct RNA sequencing (DRS) provides data support for RNA modification detection, which can preserve the potential m^6A signature compared to second-generation sequencing. However, due to insufficient DRS data, there is a lack of methods to find m6A RNA modifications in DRS. Our purpose is to identify m^6A modifications in DRS precisely.** ***Methods:*** **We present a methodology for identifying m^6A modifications that incorporated mapping and extracted features from DRS data. To detect m^6A modifications, we introduce an ensemble method called mixed-weight neural bagging (MWNB), trained with 5-base RNA synthetic DRS containing modified and unmodified m^6A.** ***Results:*** **Our MWNB model achieved the highest classification accuracy of 97.85% and AUC of 0.9968. Additionally, we applied the MWNB model to the COVID-19 dataset; the experiment results reveal a strong association with biomedical experiments.** ***Conclusion:*** **Our strategy enables the prediction of m^6A modifications using DRS data and completes the identification of m^6A modifications on the SARS-CoV-2.** ***Significance:*** **The Corona Virus Disease 2019 (COVID-19) outbreak has significantly influence, caused by the SARS-CoV-2. An RNA modification called m^6A is connected with viral infections. The appearance of m^6A modifications related to several essential proteins affects proteins' structure and function. Therefore, finding the location and number of m^6A RNA modifications is crucial for subsequent analysis of the protein expression profile.**

***Index Terms*—COVID-19, ensemble learning, m^6A RNA modifictions, nanopore single-molecule direct RNA sequencing (DRS), SARS-CoV-2.**

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 to treat SARS-CoV-2. Thus, understanding the genetic information of SARS-CoV-2 enables us to analyze the virus's characteristics and aids in developing and implementing measures.

Manuscript received April 12, 2021; revised November 20, 2021 and January 7, 2022; accepted January 31, 2022. Date of publication February 11, 2022; date of current version July 19, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 61872241 and 61572316, in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, and in part by Shanghai Science and Technology Commission under Grant 21511101200. *(Ruhan Liu, Liang Ou, and Bin Sheng contributed equally to this work.) (Corresponding authors: Bin Sheng; Pei Hao; Huating Li.)*

Ruhan Liu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, and also with the MoE Key Lab of Artificial Intelligence, Artificial Intelligence Institute, Shanghai Jiao Tong University, China.

Liang Ou is with the Key Laboratory of Molecular Virology and Immunology, Institut Pasteur of Shanghai, China.

Bin Sheng is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the MoE Key Lab of Artificial Intelligence, Artificial Intelligence Institute, Shanghai Jiao Tong University Shanghai 200240, China (e-mail: shengbin@cs.sjtu.edu.cn).

Pei Hao is with the Key Laboratory of Molecular Virology and Immunology, Institut Pasteur of Shanghai, Shanghai 200031, China (e-mail: phao@ips.ac.cn).

Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

Xiaokang Yang is with the Shanghai Key Laboratory of Digital Media Processing and Communication, Department of Electronic Engineering, Shanghai Jiao Tong University, China, and also with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, China.

Guangtao Xue is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.

Lei Zhu is with ROAS Thrust, The Hong Kong University of Science and Technology (Guangzhou), China.

Yuyang Luo is with the Shanghai Sharee Technology Company, Ltd., China, and also with the Department of Electonic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

Ping Zhang is with the Department of Computer Science and Engineering, The Ohio State University, USA, and also with the Department of Biomedical Informatics, The Ohio State University, USA.

Po Yang is with the Department of Computer Science, The University of Sheffield, U.K.

Huating Li is with Shanghai Jiao Tong University, Affiliated Sixth People's Hospital, Shanghai 200233, China (e-mail: huarting99@sjtu.edu.cn).

David Dagan Feng is with the School of Computer Science, The University of Sydney, Australia.

This article has supplementary downloadable material available at https://doi.org/10.1109/TBME.2022.3150420, provided by the authors.

Digital Object Identifier 10.1109/TBME.2022.3150420

Ribonucleic acid (RNA) modifications are closely linked to gene activation and expression, affecting the structure and morphology of proteins through several actions [3], [4]. Additionally, locating modification sites can be highly beneficial for analyzing the relationship between RNA modifications and protein expression. Numerous studies examining the function of m^6A in viral-host interactions have identified distinct roles, implying widespread regulatory control over viral life cycles [5]. SARS-CoV-2 is a single-stranded RNA genome found in COVID-19 [6]. Theoretically, SARS-CoV-2 has the potential to mutate due to changes in protein structure and properties. Changes in the structure and properties of proteins also can be reflected by changes in the m^6A modification status. As a result, research needs to understand the location and magnitude of m^6A modifications. Moreover, Kim's article published in Cell [7] predicted possible base modification sites in the SARS-CoV-2 transcriptome. Additionally, as demonstrated in experiments of [8], m6A negatively regulates SARS-CoV-2 infection. It shows that m6A may have the impact on the biological function of the spike protein, which is a key material for the SARS-CoV-2 immunoassay kit. Thus, elucidating the location and magnitude of m^6A modifications contributes to understanding 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 m^6A detection, some limitations remain. There are frequently issues with resolution and immune specificity [9], and the procedure is frequently unsatisfactory in terms of cost and precision [10]. In addition, compared to second-generation sequencing that requires polymerase chain reaction (PCR), an amplification technique that loses m^6A modification information, DRS preserves the underlying m^6A signature. In DRS, nanopores move uniformly from the beginning to the end of the RNA sequence in a sliding window of base length five. The current value at each moment is determined by the composition of the five nucleotides inside the sliding window. DRS can record traces of base modification in the form of electrical signals, and the capacity of DRS to detect base modification in RNA is demonstrated [11]. The following section provides an overview of the currently used DRS-based RNA m^6A detection method. In [3], [12]–[14], methods are proposed to detect modifications using hypothesis testing technique. 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 [13]. Additionally, these methods' performance is entirely dependent on the sensitivity of potential modification types to features. The type of modification is unknown, which means we can not distinguish m^6A with other modifications.

Machine learning techniques have been enormously successful in biomedical engineering [15]–[19]. Many classical machine learning algorithms perform exceptionally well at detecting base modification. For instance, Hidden Markov Models (HMM) and Support Vector Machines (SVM) have been used to identify specific base modifications in DNA and RNA in some previous work [20], [21]. 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 labelling them is challenging. Moreover, suitable training samples are scarce due to the lack of DRS samples, leading to insufficient performance of deep-learning models.

Therefore, it is critical to investigate an effective method for handling this novel data. In this study, we propose an ensemble learning framework for m^6A detection using DRS data. According to DRS data, we obtain extracted features from raw sequencing data, including current and quality data, as well as screened mapping base features, including mismatches frequency, delete frequency and insert frequency. The extracted and mapping features are then fed into the proposed integrated learning model (Mixed-weight neural bagging) to obtain m^6A prediction results. Additionally, we compare the performance of the model 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 m^6A 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 m^6A 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 extracted and mapping features. When compared to current state-of-the-art m^6A 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. The best performance obtained from the grid search algorithm of each model is used to evaluate models' performance. Additionally, metrics such as accuracy, precision, specificity, sensitivity, F1-score, G mean1, G mean2, and AUC are considered during the evaluation process. Moreover, we apply our model to the DRS data of a SARS-CoV-2 sample and determine 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 m^6A modification using our MWNB model. Section IV presents our experimental results and compares 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 m^6A and faithfully records the disturbance of the m^6A molecule to the background current in the form of an electrical signal [22]. Several studies confirmed the difference in electrical signals between m^6A and normal adenylate in theory and practice [20]. Smith *et al.* accurately performed direct RNA-seq on two samples with high degree m^6A and low degree m^6A, respectively [21]. Their work indicated that they observed modification of current signals and base calling error near m^6A. As a result, it has a research foundation for determining the base position by comparing the differences in electrical signals. Thus, we introduce related works in m^6A 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 m^6A modifications using nanopore technology [14]. 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 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 consider RNA modifications [23]. However, in real organisms and the canonical base, there are also some chemical groups that are modified base, such as m^6A. When modified bases are present in the sequence, DRS can clearly show the difference between them and the canonical base.

B. Statistics Methods for Base Modification Detection

Several previous studies demonstrated statistical methods for detecting base modifications using direct sequencing with promising results [12]. Stoiber M H *et al.* [13] used the Mann–Whitney U test to detect m^5C on DNA/RNA, which is also a modification, and m^6A on DNA in all sequence contexts without requiring unmodified samples in addition to de novo detection. When m^5C occurs and when it does not, the electrical signal characteristic distribution of m^5C is significantly different, even more so than the distribution of m^6A [24] Liu *et al.* [14] used the Kolmogorov–Smirnov test to demonstrate that NanoMod outperformed Tombo at detecting m^5C in E. coli. Statistics-based modification detection methods have several advantages, such as low computational resource consumption [25]. Nonetheless, Their flaws remain insurmountable. Completely clean samples must be prepared (free of any base modification) for ensuring detection precision. [3] must be performed using the same sequencing experiments and data pre-processing steps as the experimental sample. Without a doubt, statistical methods significantly increase the difficulty and cost of the preliminary sample preparation stage, particularly for some precious 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 [26]–[28]. Garalde *et al.* developed a tool called the Nanopolish that uses the HMM (hidden Markov model) to identify m^5C on DNA in the CpG context accurately. SignalAlign [?] also a modification detection tool based on the HMM with the hierarchical Dirichlet process. Rand *et al.* used the SignalAlign to detect m^5C and m^6A sequences in E. coli DNA. The mCaller, which doubles as a modification detector, detected m^6A on DNA using four machine learning classifiers (neural network, random forest, logistic regression, and naive Bayes classifiers). Mclntyre *et al.* [22] demonstrated that the most accurate predictor (84%) used the mCaller with the neural network. Prior research has concentrated on DNA base modification, particularly m^5C. Due to structural similarity and the difficulty of obtaining accurate data sets, the m^6A modification in RNA has not been investigated previously. Huanle Liu *et al.* recently constructed a labelled dataset using in vitro transcription of m^6A and classical adenosine, respectively. Additionally, the SVM classifier they proposed produced acceptable results (90% in accuracy). Moreover, 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 m^6A 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 m^6A RNA modifications via DRS. The following sections introduce the MWNB model, divided into three sections: capturing various features from DRS, preprocessing and selecting features, and the MWNB classifier methodology.

The proposed method, which serves as a framework for m^6A modification recognition in RNA sequencing, requires that the first step extract base features from RNA sequences. This step

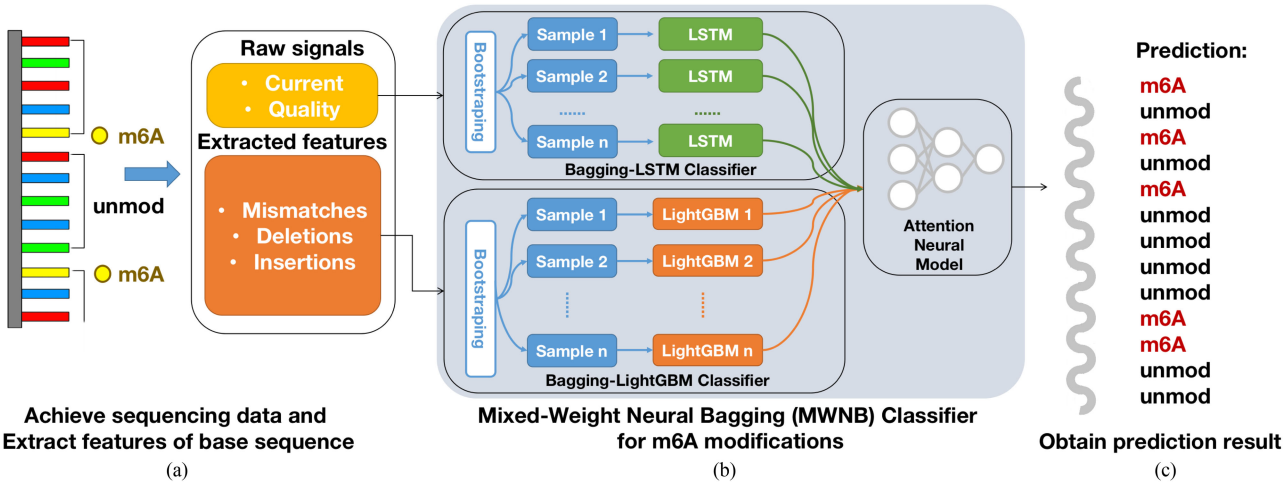

Fig. 1. Overview of the framework shows the procedure of detecting m^6A RNA modifications using RNA sequencing. The framework is divided into three sections: gaining raw sequencing data and extracting features (Part A), classifying the data using Mixed-Weight Neural Bagging (Part B), and obtaining the prediction results (Part C) (Part C). Critical features such as base current intensity, base quality, mismatches frequency, delete frequency, and insert frequency are introduced in Part A. In Part B, the mixed-weight bagging network (MWBN) is proposed for detecting m^6A RNA modifications in five-base DRS sequences. Additionally, Part C details the procedure for obtaining prediction results.

introduces the features that are used and defines their meanings. 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, five 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 m^6A modifications in the third section. We introduce the Bagging-LSTM model in the fourth section, which is used to classify m^6A modifications based on features extracted from direct sequencing. Finally, we demonstrate how to fuse the Bagging-LightGBM algorithm with the Bagging-LSTM 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 [29] 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 [29] 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 use our model on the COVID-19 data set and display 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

After feature extraction, we get five related features: C, Q, Mis, Ins, and Del. The selection of features has a significant impact on classification accuracy and is a necessary step before clustering or classification. According to earlier research [21], 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 [21], and deleted some 5-base sequences that did not fit the standards of base matching rules by referring to the [21].

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

TABLE I
WE RETRIEVE 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=$c - Meani$, $c - Mediani$, $c - Stdi$, 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 = $\frac{\text{num of delete}}{\text{total num of base}}$

C. Bagging-LightGBM Feature Classification

We use the light gradient boosting machine (LightGBM) as the base classifier for predicting m^6A RNA modifications utilizing attributes of base fragments. LightGBM [29] 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. Compared to GBDT, LightGBM provides the advantages of faster training efficiency, higher accuracy, and the ability to analyze massive amounts of data.

For training dataset $X = \{(x_i, y_i | x_i \in R^k, y_i \in R, |X| = n\}$, where $x = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ is the input feature set, k is the dimension of input features and $y = \{y_1, y_2, \ldots, y_i, \ldots, y_n\}$ 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 $M = \{1, 2, \ldots, j, \ldots, m\}$, the pseudo residuals or gradient is $g_j = \{g_{1j}, g_{2j}, \ldots, g_{ij}, \ldots, g_{nj}\}$ in each iteration, and the modified dataset called $MX = \{mX_1, mX_1, \ldots, mX_j, \ldots, mX_m\}$ in each iteration. The formula of g_{ij} and mX_j are:

$$g_{ij} = -\frac{\partial L(y_i, F_{j-1}(x_i))}{\partial F_{j-1}(x_i)} \tag{2}$$

$$mX_j = \{(x_i, g_{ij}) | i = 1, 2, \ldots, n\} \tag{3}$$

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

Algorithm 1: Bagging-LightGBM Classifier.

Require:
1: Given trainning dataset, $X = \{(x_i, y_i | x_i \in R^k, y_i \in R, |X| = n\}$.
2: Base LightGBM classifier, Φ.
3: The number of sub-sampling, K.
Ensure: Aggregation of K sub-sampling
4: **for**each $i = 1 : K$ **do**
5: bootstrap sample in X to obtain modified training dataset $train_X$ and validation dataset val_X.
6: train the i^{th} expert (classifier Φ) in $train_X$ and val_X.
7: **end for**
8: The predictions $P = \{p_1, p_2, \ldots, p_K\}$ is obtained by K expert models.
9: **return** P

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), 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

The previous section introduces the Bagging-LightGBM model, which uses extracted characteristics to categorize m^6A 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 backwards. This model can access complete, sequential information about all context information after each time step in a given sequence. This study proposes a dual LSTM

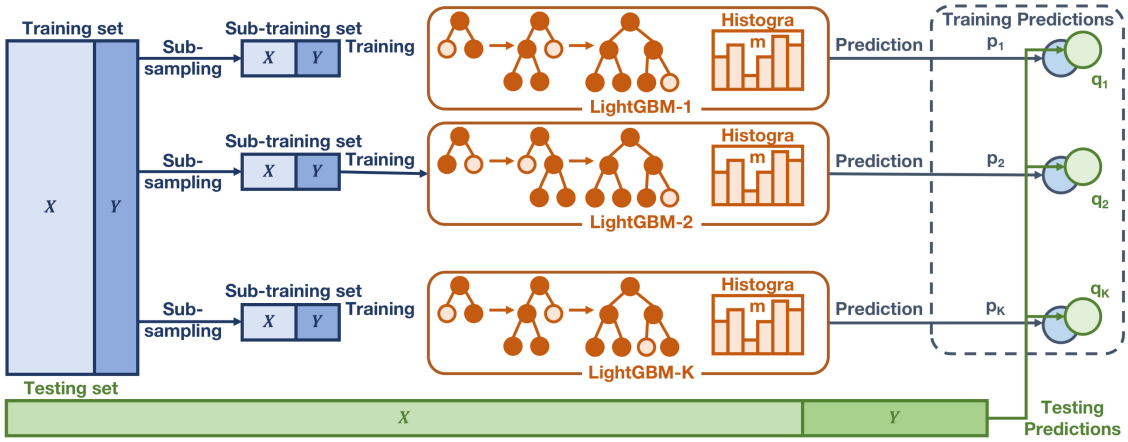

Fig. 2. Architecture of the Bagging-LightGBM model is depicted in the model structure image. LightGBM [29] models are used as the base learners in the Bagging-LightGBM model, and they are trained using different subsets of the training set. They employ the same testing set for model evaluation. The Bagging-LightGBM model produces numerous predictions about whether m^6A emerge.

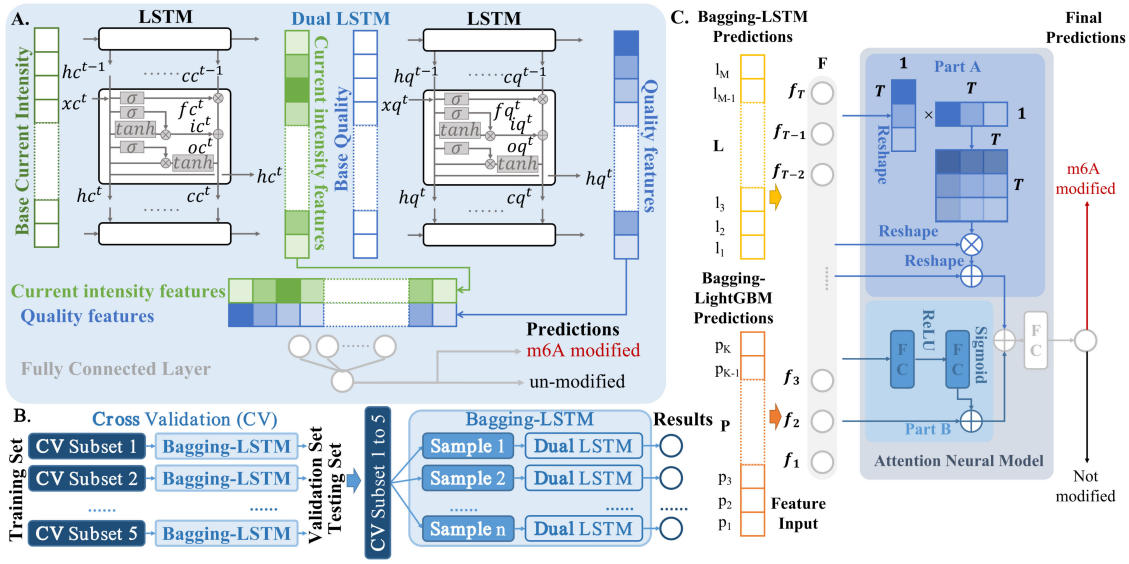

Fig. 3. Model architecture figure shows the architecture of the MWNB model. ***A.*** Figure shows the structure of a dual LSTM model for detecting m^6A RNA modifications. ***B.*** The architecture plot of the Bagging-LSTM model with cross validation. In the bagging model, the training set is chosen at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

model to classify the m^6A modifications based on the extracted features in 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 updating equations are given as follows:

$$ic^t = \sigma\left(W^{(ic)} \cdot [hc^{t-1}, xc^t] + b^{(ic)}\right)$$

$$fc^t = \sigma\left(W^{(fc)} \cdot [hc^{t-1}, xc^t] + b^{(fc)}\right)$$

$$oc^t = \sigma\left(W^{(oc)} \cdot [hc^{t-1}, xc^t] + b^{(oc)}\right)$$

$$cc^t = fc^t \times cc^{t-1} + ic^t \times tanh\left(W^{(cc)} \cdot [hc^{t-1}, xc^t] + b^{(cc)}\right)$$

$$hc^t = oc^t \times \tanh\left(cc^t\right) \tag{4}$$

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 in 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 features are

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

$$
\begin{aligned}
iq^t &= \sigma\left(W^{(iq)} \cdot [hq^{t-1}, xq^t] + b^{(iq)}\right) \\
fq^t &= \sigma\left(W^{(fq)} \cdot [hq^{t-1}, xq^t] + b^{(fq)}\right) \\
oq^t &= \sigma\left(W^{(oq)} \cdot [hq^{t-1}, xq^t] + b^{(oq)}\right) \\
cq^t &= fq^t \times cq^{t-1} + iq^t \times tanh\left(W^{(cq)} \cdot [hq^{t-1}, xq^t] + b^{(cq)}\right) \\
hq^t &= oq^t \times \tanh\left(cq^t\right) \qquad (5)
\end{aligned}
$$

Further, we use the same bagging strategy to generate the predictions of Bagging-LSTM. The final prediction is $L = \{l_1, l_2, \ldots, l_M\}$. 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 m^6A modifications. With reference to [30], 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) illustrates the attention enhancement module and learns the significant 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 m^6A modifications in COVID-19 data.

A. Dataset

We used two data sets in this study: one from Epinano [21], and the other from Kim *et al.* [6] 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. After mRNA purification and extraction, they went through the same sequencing technique and upstreamed the pretreatment process as the training set. The signal value at a given time was defined by around four bases (A, T, C, G), and all about 1024 bases were organized and combined to generate a signal pattern in the nanopore sequencing process. To cover as many scenarios as feasible, Liu *et al.* [21] created a sequence master comprising all signal patterns and employed synthetic substrates with and without N6-methyladenosine in 2019. Two readings in our data collection contained m^6A and two reads that did not contain m^6A. We classified them into 19,806 positive samples and 19,964 negative samples based on previous research [21].

B. Implementation Details

In this identification task, we used two datasets to train, validate and test the models. The two datasets are the Epinano dataset [21] and the original data [6] of the SARS-CoV-2 provided by Kim *et al.* The Epinano dataset was used for training, validating and testing the model, while the COVID-19 dataset was 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 mapping features (mismatch frequency, insert frequency, and delete frequency) and two extracted features (base current, base quality) from the raw data by the mapping tool. The two extracted features (current and quality) for the raw data extracted the mean, median, and variance for each base, respectively. The three mapping features (mismatch, insert, delete frequency) were obtained from the mapping tool. In our 5-base sequences, all feature inputs for each sequence 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 were trained using the feature extraction methods mentioned above. In the training of the MWNB model, the features extracted from the original data were input to Bagging-LSTM for feature extraction, and the mapped features were input to Bagging-LightGBM for feature extraction. The attention neural model used the final extracted bagging features to discriminate whether m^6A modifications occur. All experiments were implemented on an Intel XeonE5-2630 v4 @ 2.20GHzz CPU and NVIDIA GeForce RTX 2080 Ti ArchLinux. All models were implemented in Scikit-learn and Pytorch.

In tuning the parameters, we used 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 were the remaining five folds. In the training process, four folds were used, and the fifth fold uses for testing. The process was repeated five times, picking the different folds for testing each time, and the other four folds were used in training. The data in the test set was one of the six-folds, and the training and validation sets were the remaining five-folds.

C. Evaluation Metrics

To assess the performance of models, we used 6 metrics: Accuracy (acc), precision (pre), sensitivity (se), specificity (sp),

TABLE II
AVERAGE CLASSIFICATION PERFORMANCE OF LEAVE-ONE-OUT 5-FOLD CROSS-VALIDATION OF ALL MACHINE LEARNING METHODS IN OUR m^6 A 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
DT [31]	81.64	82.52	79.93	83.32	82.92	82.92	81.61	0.816
RF [32]	91.15	93.95	87.82	94.43	94.19	94.19	91.07	0.970
ET [33]	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
[21]	90	-	-	-	-	-	-	0.944
[32]	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
DT [31]	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 [32]	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 [33]	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

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 are $Sensitivity = \frac{T_P}{(T_P+F_N)}$ and $Specificity = \frac{T_N}{(T_N+T_P)}$, relatively. In above equations, T_P represents that the prediction results of the model are a positive examples (P) and the ground truth are right examples (T), T_N stands for that the prediction results of the model are negative examples (N) and the judgment results are right examples (T), F_P represents that the prediction results of the model are positive examples (P) and the judgment results are wrong examples (F), F_N is that the prediction results of the model are negative examples (N) and the judgment results are wrong examples (F). Moreover, $F1score$, $Gmean_1$, and $Gmean_2$ are also calculated in evaluation of models:

$$F1score = 2 \times \frac{Precision \times Sensitivity}{(Precision + Sensitivity)} \tag{6}$$

$$Gmean_1 = \sqrt{Sensitivity \times Precision} \tag{7}$$

$$Gmean_2 = \sqrt{Sensitivity \times Specificity} \tag{8}$$

Furthermore, 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:

$$AUC = \frac{\sum_{ins_i \in positiveclass} rank_{ins_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{9}$$

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 extracted and mapping from raw sequencing data were used as input features in our MWNB model. The m^6A RNA modification categorization findings were obtained using two models (Bagging-LightGBM for extracted features from raw sequencing and Bagging-LSTM for mapping features). Because of the peculiarities of our MWNB model, we first compared it against classic machine learning models. Traditional machine learning models did 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 compared our MWNB model to a strategy based on ensemble learning to make more comprehensive comparisons. Additionally, we compared deep learning-based methods with our MWNB model..

TABLE III
AVERAGE CLASSIFICATION PERFORMANCE OF LEAVE-ONE-OUT 5-FOLD CROSS-VALIDATION OF LIGHTGBM [29], BAGGING-DT [34], BAGGING-LIGHTGBM, AND OUR MWNB MODELS. THE RESULTS SHOW MODELS' PERFORMANCE IN TESTING DATASET

Model	acc (%)	pre (%)	sp (%)	se (%)
LightGBM [29]	94.25	97.56	97.77	90.67
Bagging-DT [34]	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

Firstly, we compared 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 model's results mentioned above 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 models discussed above, the model classification accuracy attained on this task was approximately 80%. We employed two ensemble learning models: RF and ET, which performed well on this challenge. These approaches achieved accuracies of approximately 91% to 94%. We estimated 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 to the unique ensemble learning model. The table below compares

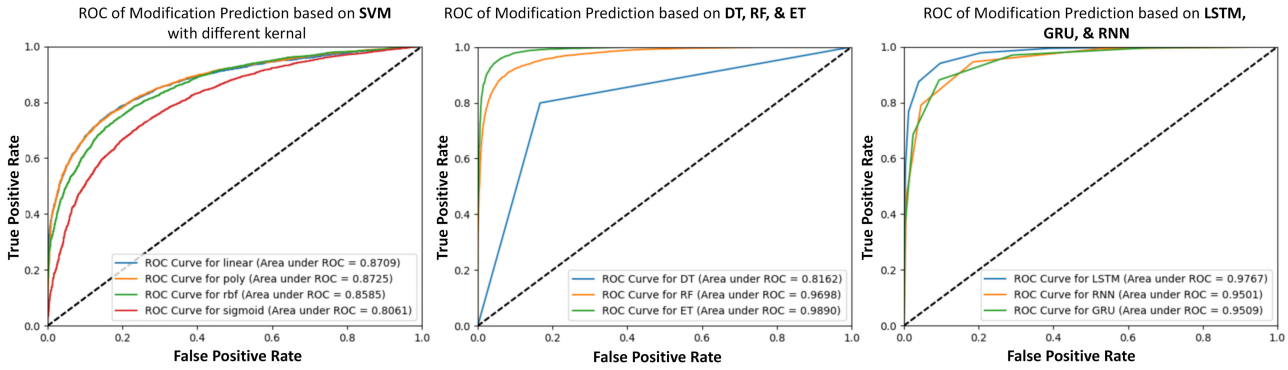

Fig. 4. Best-performing models' ROC chart. The ROC figure of SVM [21] with different kernel types, such as linear, poly, RBF, and Sigmoid, is shown on the left. The ROC chart of decision tree (DT) [31], random forest (RF) [32], and extremely random trees (ET) [33] is shown in the middle figure, and the ROC plot of RNN, LSTM, and GRU models is shown in the right figure.

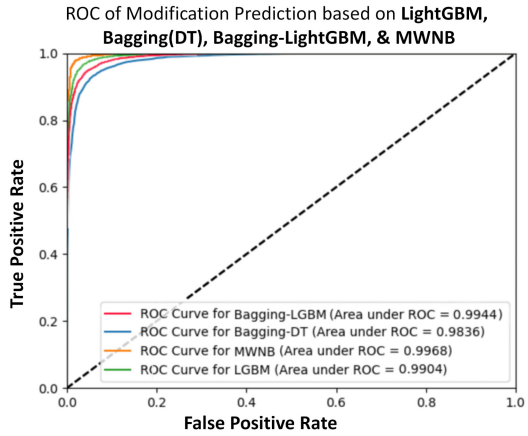

Fig. 5. AUC figure shows the performance of LightGBM, Bagging-DT, Bagging-LightGBM, and our MWNB.

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 compared the performance of our MWNB models to that of deep learning models that were employed in all features. In comparison, we used LSTM, GRU, and RNN with our MWNB. Additionally, these models outperformed DT and SVM in terms of performance. The RNN and upgraded RNN (GRU and LSTM) models also had AUC values greater than 0.95. LSTM, GRU, and RNN performance details are provided in Fig. 4 and Table II. Moreover, we integrated these models using the bagging technique. The performance of Bagging-LSTM, Bagging-GRU, and Bagging-RNN is illustrated in Table IV and Fig. 6. The best AUC value for these models was 0.9887, which Bagging-LSTM obtained.

As shown in Table III, the MWNB approach produces the best outcomes. The LightGBM model has been adjusted and enhanced in numerous ways to enhance the gradient boosting decision tree (GBDT). It offers the advantages of being efficient in training, having outstanding accuracy, and processing enormous amounts of data. We enhanced the model's generalisation

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

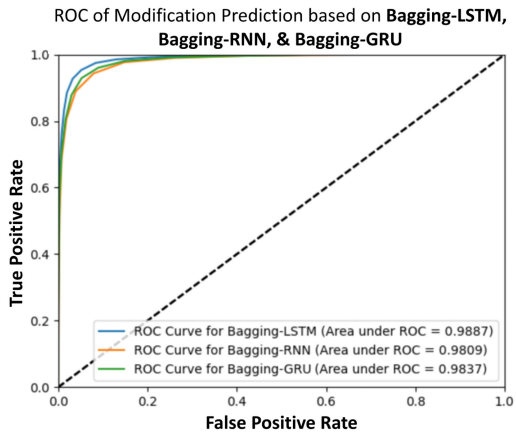

Fig. 6. AUC figure shows the performance of Bagging-LSTM, Bagging-GRU, and Bagging-RNN.

and accuracy by fusing several LightGBM models using the bagging approach. Additionally, the LSTM excels in extracting sequence relationships, and we used the Bagging-LSTM model to extract critical information from extracted features in raw sequencing data. Finally, we merged the Bagging-LSTM and Bagging-LightGBM models using the attention neural network. In our assignment, our MWNB model performs optimally. Our MWNB model obtained the best performance in our task. The MWNB's accuracy was 97.85%, precision was 98.37%, sensitivity was 97.20%, and specificity was 98.41%. The AUC value, which is the highest, was 0.997 obtained by the MWNB method.

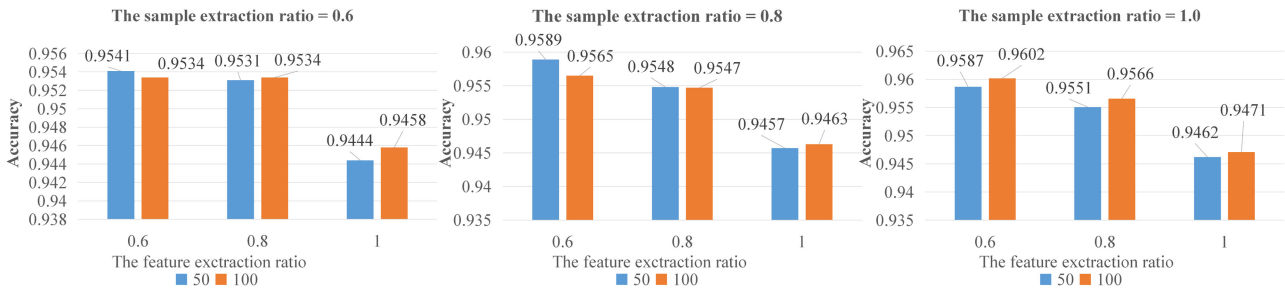

Fig. 7. Accuracy for different paramters in the Bagging-LightGBM model of testing dataset. The orange bar shows the result that the number of base learner is 50, and the blue bar illustrate accuracy which the number of base learner is 100 obtain in different sample ratio and various feature ratio.

E. Parameter Tunning

The initial step of the parameter tuning experiment is to determine the proper parameter values for Bagging-LightGBM. To begin, we required 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 began 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 investigated 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 $lambdaL_1$ 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, 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 employed 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 determined the sampling procedure for the sample subset and the feature subset.

The best model achieved an accuracy of 96.02% when the Bagging-LightGBM parameters were 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 employed 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 employed the grid search technique to identify the parameters of Bagging-LSTM in the second round of parameter tuning. First, we experimented with varying the number of concealed cells N: from 10 to 200. Additionally, we experimented 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 used the same method in bagging as we did in Bagging-LightGBM. The most accurate Bagging-LSTM model achieved 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. Moreover, we employed non-replacement sampling to create sample subsets and replacement sampling to create feature subsets. Furthermore, we investigated the attention neural network's parameters using the best Bagging-LightGBM and Bagging-LSTM. We experimented with various batch sizes (4 to 64) and learning rates (0.00001 to 0.1). The following hyperparameters define the optimal model: batch size $=$ 16, learning rate $=$ 0.001.

Fig. 8 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. 8 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 is obtained: 91.15 % of the average classification accuracy rate and 93.95 % of the average classification precision rate. Fig. 8 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.

In the following settings, we obtained the best accuracy of the bagging method in 93.38% and the best precision rate in 95.39%. The number of base learners (DT) is 500, the proportion of samples used for each training of the base learner is 80%, and the proportion of features used for each training of the base learner is 50%.

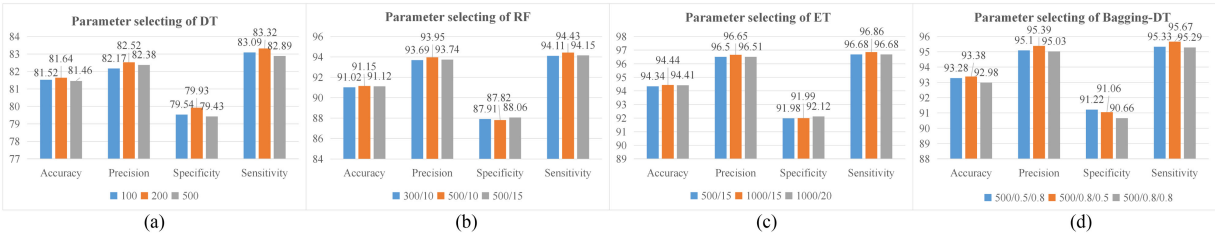

Fig. 8. Accuracy for different paramters in the Bagging-LightGBM model. The orange bar shows the result that the number of base learner is 50, and the blue bar illustrate accuracy which the number of base learner is 100 obtain in different sample ratio and various feature ratio.

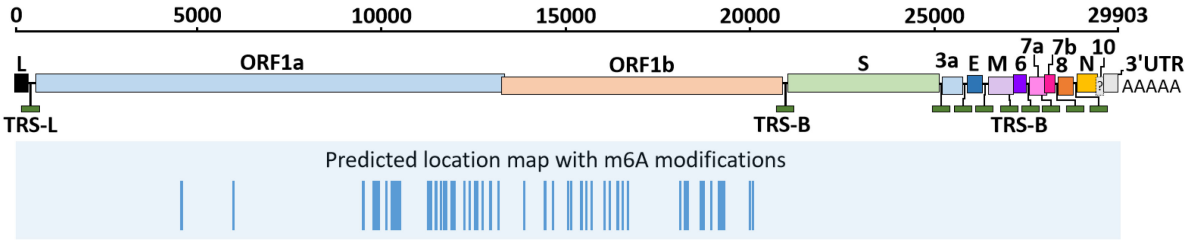

Fig. 9. Schematic diagram of the SARS-CoV-2 Genome. At the top of the picture is the genome's axis (1-29903) of SARS-CoV-2. The bars in the middle represent the different genes of the genome anchored in the corresponding regions. The lower part of the image is the predicted density map of m^6 A sites. The higher part(like "peak") means that the m^6 A predicted sites are densely distributed.

F. Bagging-LightGBM Model Used in COVID-19

To assess our model's performance with respect to m^6 A modifications inside certain sequence motifs (RRACH). We performed m^6 A 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 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 m^6 A alteration. 65.4% (Fig. 9) 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. We selected a probability threshold of m^6 A to minimise false positives to verify the results' accuracy. The other major peaks suggested the presence of a significant amount of m^6 A. The majority of the high probability loci were discovered in the ORF1b region of the genome. ORF1b encodes a nonstructural protein (NCP) required for viral transcription, replication, and inhibition of host immune response and gene expression. Antiviral therapy aims to inhibit RNA-dependent RNA polymerase [35]. 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 m^6 A reduces the influenza virus' replication [36]. The model results help our understanding of the SARS-life CoV-2's activities at a deeper level, which aids in developing targeted antiviral medications.

V. Discussion

This work presented a machine learning-based method to recognize RNA m^6 A modifications in DRS. Features derived from the raw signals and their mapping information were utilized as the model input. Several classifiers were utilized: SVM, RF, ensemble learning (RF, ET, and Bagging) and our MWNB to classify the m^6 A 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 features, an integrated framework was developed to detect m^6 A modification based on features produced by sequencing patterns. We proposed the MWNB model to classify m6A RNA modifications by targeted feature extraction (Bagging-LightGBM for mapping features and Bagging-LSTM for extracted features of current and quality) according to signal difference of sequencing data. The model is not only applicable to the detection of m^6 A modifications in RNA sequencing, but from the modelling perspective, our model can be directly used in the detection of other modifications with only simple migration.

Although our model has achieved good detection results, some erroneous predictions still exist due to the data and model's limitations. 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 [37]. Such low-dimensional data are challenging to distinguish between the occurring and non-occurring m^6 A modifications. Also, compared to the multi-electrode, nanopore sequencing has only one channel [38], and the number of features

in the data itself is too low. We hope that in the near future Oxford Nanopore U.K. 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.

VI. Conclusion

This article proposes a Bagging-LightGBM model for $m^6\ A$ 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 $m^6\ A$ modification detection model (MWNB) to the SARS-COV-2 sequencing data to obtain the possible $m^6\ A$ modification site information. The prediction results will help us to find the possible location of gene mutation.

References

[1] P. Dashraath *et al.*, "Coronavirus disease 2019 (COVID-19) pandemic and pregnancy," *Amer. J. Obstet. Gynecol.*, vol. 222, no. 6, pp. 521–531, 2020.

[2] M. Jeyanathan *et al.*, "Immunological considerations for COVID-19 vaccine strategies," *Nat. Rev. Immunol.*, vol. 20, no. 10, pp. 615–632, 2020.

[3] T. Wongsurawat *et al.*, "Decoding the epitranscriptional landscape from native RNA sequences," *bioRxiv*, 2018, Art. no. 487819.

[4] Y. Liu *et al.*, "N6-methyladenosine RNA modification–mediated cellular metabolism rewiring inhibits viral replication," *Science*, vol. 365, no. 6458, pp. 1171–1176, 2019.

[5] N. S. Gokhale *et al.*, "N6-methyladenosine in flaviviridae viral RNA genomes regulates infection," *Cell Host Microbe*, vol. 20, no. 5, pp. 654–665, 2016.

[6] D. Kim *et al.*, "The architecture of SARS-CoV-2 transcriptome," *Cell*, vol. 181, no. 4, pp. 914–921, 2020.

[7] C. G. Ziegler *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, pp. 1016–1035, 2020.

[8] J. Liu *et al.*, "The m6A methylome of SARS-CoV-2 in host cells," *Cell Res.*, vol. 31, no. 4, pp. 404–414, 2021.

[9] D. Dominissini *et al.*, "Topology of the human and mouse m6A RNA methylomes revealed by m6A-seq," *Nature*, vol. 485, no. 7397, pp. 201–206, 2012.

[10] B. Linder *et al.*, "Single-nucleotide-resolution mapping of m6A and m6Am throughout the transcriptome," *Nat. Methods*, vol. 12, no. 8, pp. 767–772, 2015.

[11] E. L. van Dijk *et al.*, "The third revolution in sequencing technology," *Trends Genet.*, vol. 34, no. 9, pp. 666–681, 2018.

[12] M. T. Parker *et al.*, "Nanopore direct RNA sequencing maps the complexity of arabidopsis mRNA processing and m6A modification," *eLife*, vol. 9, pp. 1–35, 2020.

[13] M. H. Stoiber *et al.*, "De novo identification of DNA modifications enabled by genome-guided Nanopore signal processing," *bioRxiv*, 2017, Art. no. 94672.

[14] Q. Liu *et al.*, "NanoMod: A computational tool to detect DNA modifications using Nanopore long-read sequencing data," *BMC Genomic.*, vol. 20, no. 1, 2019, Art. no. 78.

[15] P. Liu *et al.*, "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, Jan. 2021.

[16] Y. Tang *et al.*, "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, Jan. 2021.

[17] A. Moniri *et al.*, "Real-time forecasting of sEMG features for trunk muscle fatigue using machine learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 2, pp. 718–727, Feb. 2021.

[18] G. Noaro *et al.*, "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, Jan. 2021.

[19] L. Lu *et al.*, "Evaluating rehabilitation progress using motion features identified by machine learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 4, pp. 1417–1428, Apr. 2021.

[20] A. M. Smith *et al.*, "Reading canonical and modified nucleobases in 16s ribosomal RNA using Nanopore native RNA sequencing," *PLoS one*, vol. 14, no. 5, 2019, Art. no.e0216709.

[21] H. Liu *et al.*, "Accurate detection of m6A RNA modifications in native RNA sequences," *Nat. Commun.*, vol. 10, no. 1, 2019, Art. no. 4079.

[22] A. B. R. McIntyre *et al.*, "Single-molecule sequencing detection of N6-methyladenine in microbial reference materials," *Nat. Commun.*, vol. 10, no. 1, pp. 579–579, 2019.

[23] R. M. Leggett and M. D. Clark, "A world of opportunities with Nanopore sequencing," *J. Exp. Botany*, vol. 68, no. 20, pp. 5419–5429, 2017.

[24] A. C. Rand *et al.*, "Mapping DNA methylation with high-throughput Nanopore sequencing," *Nat. Methods*, vol. 14, no. 4, pp. 411–413, 2017.

[25] L. Xu and M. Seki, "Recent advances in the detection of base modifications using the Nanopore sequencer," *J. Hum. Genet.*, vol. 65, no. 1, pp. 25–33, 2020.

[26] D. M. Camacho *et al.*, "Next-generation machine learning for biological networks," *Cell*, vol. 173, no. 7, pp. 1581–1592, 2018.

[27] P. Ni *et al.*, "DeepSignal: Detecting DNA methylation state from Nanopore sequencing reads using deep-learning," *Bioinformatics*, vol. 35, no. 22, pp. 4586–4595, 2019.

[28] R. R. Wick *et al.*, "Performance of neural network basecalling tools for Oxford Nanopore sequencing," *Genome Biol.*, vol. 20, no. 1, 2019, Art. no. 129.

[29] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3149–3157.

[30] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3141–3149.

[31] W. Lu *et al.*, "Research on RNA secondary structure prediction based on decision tree," in *Intelligent Computing Theories and Application*. Cham, Switzerland: Springer, 2019, pp. 430–439.

[32] A. Khana *et al.*, "Detecting N6-methyladenosine sites from RNA transcriptomes using random forest," *J. Comput. Sci.*, vol. 47, 2020, Art. no. 101238.

[33] P. Geurts *et al.*, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

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

[35] L. Subissi *et al.*, "SARS-CoV ORF1b-encoded nonstructural proteins 12–16: Replicative enzymes as antiviral targets," *Antiviral Res.*, vol. 101, pp. 122–130, 2014.

[36] D. G. Courtney *et al.*, "Epitranscriptomic enhancement of influenza a virus gene expression and replication," *Cell Host Microbe*, vol. 22, no. 3, pp. 377–386, 2017.

[37] S. Marcus and B. James, "BasecRAWller: Streaming Nanopore basecalling directly from raw signal," *bioRxiv*, 2017, Art. no. 133058.

[38] L. A. H. *et al.*, "Chapter fifteen - Subangstrom measurements of enzyme function using a biological Nanopore, SPRNT," in *Methods in Enzymology*, vol. 582, M. Spies and Y. R. Chemla Eds., 2017, pp. 387–414.