ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

Identifying Transcription Factors That Prefer Binding to Methylated DNA Using Reduced *G*-Gap Dipeptide Composition

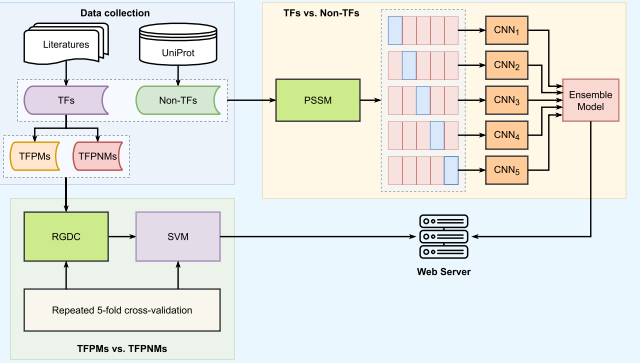
Quang H. Nguyen,[‖] Hoang V. Tran,[‖] Binh P. Nguyen, and Trang T. T. Do*

Cite This: *ACS Omega* 2022, 7, 32322−32330

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Transcription factors (TFs) play an important role in gene expression and regulation of 3D genome conformation. TFs have ability to bind to specific DNA fragments called enhancers and promoters. Some TFs bind to promoter DNA fragments which are near the transcription initiation site and form complexes that allow polymerase enzymes to bind to initiate transcription. Previous studies showed that methylated DNAs had ability to inhibit and prevent TFs from binding to DNA fragments. However, recent studies have found that there were TFs that could bind to methylated DNA fragments. The identification of these TFs is an important steppingstone to a better understanding of cellular gene expression mechanisms. However, as experimental methods are often time-consuming and labor-intensive, developing computational methods is essential. In this study, we propose two machine learning methods for two problems: (1) identifying TFs and (2) identifying TFs that prefer binding to methylated DNA targets (TFPMs). For the TF identification problem, the proposed method uses the position-specific scoring matrix for data representation and a deep convolutional neural network for modeling. This method achieved 90.56% sensitivity, 83.96% specificity, and an area under the receiver operating characteristic curve (AUC) of 0.9596 on an independent test set. For the TFPM identification problem, we propose to use the reduced *g*-gap dipeptide composition for data representation and the support vector machine algorithm for modeling. This method achieved 82.61% sensitivity, 64.86% specificity, and an AUC of 0.8486 on another independent test set. These results are higher than those of other studies on the same problems.



1. INTRODUCTION

Transcription factors (TFs) are proteins that play an important role in gene expression. TFs directly control gene expression as they have a special property of being able to bind to specific sequences of DNA.[1] Some TFs bind to DNA promoters to form transcription initiation complexes from which the polymerase can bind to the DNA fragment and initiate transcription. Other TFs bind to enhancer DNA fragments that can either stimulate or inhibit gene transcription. In addition to their role in gene expressions, TFs can act as protein anchors, thereby helping regulate 3D genome conformation.[2] Therefore, determining the binding sites of TFs plays an important role in understanding the gene expression mechanism and regulating 3D genome conformation. Previous studies showed that methylated DNA fragments inhibited the binding of TFs;[3−5] for example, methylation at cytosine−guanine dinucleotides (CpGs) has ability to prevent TFs from binding to DNA fragments.[3] However, recent experimental studies have shown that there are TFs that can still bind to methylated DNA fragments.[6,7] The identification of TFs is an important step in understanding the activities of TFs in detail and thereby better understanding the role of methylated DNAs in gene expressions and in the regulation of 3D genome structures.

There are several experimental methods to identify TFs and TFs that prefer binding to methylated DNA targets (TFPMs).[7] However, these methods are often costly and time-consuming. In addition, the database of TFs is increasingly being expanded. Automated methods are therefore needed to quickly and accurately identify TFs and TFPMs. At the present time, there are not many tools that can do this. In 2020, Liu et al.[7] introduced a dataset of TFs and TFPMs and proposed a machine learning method to recognize TFPMs. First, the protein sequences are encoded into a vector of 201 features selected from the composition/transition/distribution, split AAC, and dipeptide composition (DC) features. Then, the support vector machine (SVM) algorithm is used to determine if the sequence is a TF. If the protein is a TF, DC will be used to encode the protein, and the XGBoost algorithm will be used

Received: June 14, 2022
Accepted: August 23, 2022
Published: August 30, 2022

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

to determine if that TF protein is a TFPM. After training on the training set using fivefold cross-validation, their TF classification model using the SVM was tested on an independent test set and achieved 80.19% sensitivity and 85.85% specificity. For the model using XGBoost to classify TFPMs, the results on the independent test set were 71.01% sensitivity and 64.86% specificity. In a more recent study from the same group, Li et al.[8] represented each amino acid sequence as a sequence of tripeptide word vectors by using the skip-gram model and then fed this sequence into a long short-term memory model for both classifications of TFs and TFPMs. They reported better performances compared to those of their previous work with an area under the receiver operating characteristic curve (AUC) of 0.9130 compared to 0.9116 for classification of TFs and an AUC of 0.8324 compared to 0.7356 for classification of TFPMs.

In this study, we propose a method with better performance in classifying TFs and TFPMs. In this method, for TF classification, we use the position-specific scoring matrix (PSSM) to encode protein sequences and use a deep convolutional neural network (CNN) to recognize TFs. For classifying TFPMs, we use the reduced g-gap amino acid composition method with optimal schemes and use the SVM algorithm for classification. We use the same dataset as that used by Liu et al.[7] and Li et al.[8] in order to compare the performance of our proposed models with theirs.

2. MATERIALS AND METHODS

2.1. Benchmark Dataset. The dataset introduced by Liu et al.[7] can be freely accessed at http://lin-group.cn/server/TFPred. The dataset includes a training set and a test set for classifying TFs and another training set and another test set for the problem of classifying TFPMs (Table 1). For the problem

Table 1. Datasets for Classification of TFs Versus Non-TFs and TFPMs Versus TFPNMs

	TFs vs non-TFs		TFPM vs TFPNM	
dataset	positive	negative	positive	negative
training set	416	416	270	146
independent test set	106	106	69	37

of classifying TFs, the training set included 416 TF protein sequences (the positive class) and 416 non-TF protein sequences (the negative class), while the independent test set had 106 TF protein sequences and 106 non-TF protein sequences. For the problem of classifying TFPMs, the training set was unbalanced with 270 TFs attached to methylated DNAs (the positive class) and 146 TFs not attached to methylated DNAs (the negative class), and the independent test set also had the same imbalance ratio as that of the training set with 69 TFs attached to methylated DNAs and 37 TFs not attached to methylated DNAs.

2.2. Identifying TFs. An overview of data representation for our TF classification framework is shown in Figure 1. First, the input protein is encoded by a PSSM. Then, our proposed CNN is used for TF prediction. The CNN was trained using fivefold cross-validation, leading to five CNN models using the same network architecture but with different parameters. Finally, ensemble learning to combine the predictions from the five CNN models is used to determine if the input sequence is a TF.

2.2.1. Position-Specific Scoring Matrix. In bioinformatics, a PSSM is a common method to encode protein sequences.[9−17] Although this encode provides a lot of useful information for research, when using a PSSM to encode data, the common problem is that the difference in the amino acid number of the protein sequences leads to the difference in the size of the PSSM data representing the sequences. Therefore, there have been many studies performing transformations to standardize the size of the PSSM. Cheol Jeong et al.[9] reduced the matrix size from L × 20 (L is the number of amino acids in the sequence) to 20 × 20 by averaging the PSSM values at positions of the same amino acid. Wang et al.[17] proposed a method to calculate the correlation values from the PSSM and position-specific frequency matrix to produce a new feature vector with 2000 dimensions. This method not only helps ensure the characteristic vector size for protein sequences but also improves the efficiency of the subsequent machine learning model. There is also an auto-cross covariance transform and discrete wavelet transform used by Zhang et al.[13] Due to the rather large PSSM data with a size of L × 20 (L is the protein chain length), Chen et al.[15] proposed an secondary structure element (SSE)-PSSM method that not only reduces the size of the feature vector but also improves performance of predicting the secondary structure of proteins with the most important modification being the SSE transformation. After searching for strings that are similar to the query string, they first converted the strings to an SSE form and then computed the position propensity matrix and finally the PSSM. Although there are initially 20 amino acids, after the SSE transform, only H (α-helices), E (β-strand sheets), and C (others) types remain. This method helps reduce the complexity of the PSSM from L × 20 to L × 3. There is also a weighting method for the PSSM using Gaussian distribution implemented by Ge et al.[12]

For TF classification problem, we use the PSSM scheme to encode protein sequences, and then, PSSMs are used to train the CNN model. PSSMs have the form $P = P_{ij}$: i = 1, ..., L and j = 1,..., 20 where L is the length of the protein sequence and each P_{ij} represents the score of the jth amino acid in the 20 amino acids and the acid the ith amino acid in the query sequence. The PSSM is created using Position-Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) software.[18] PSI-BLAST searches for protein sequences that match the query sequence in a large database and then performs multiple sequence alignment on these sequences for generating the corresponding PSSM. In this study, we used PSI-BLAST with the Swiss-Prot database to calculate the PSSM for each protein sequence.

CNNs are widely used in many research fields. In this study, we use a deep CNN model to learn patterns in the PSSMs. Although the CNN requires a fixed-size matrix as input, the number of amino acids of the protein sequences in the training set is not fixed and ranges from 51 to 4834 amino acids. The distribution in Figure 2 shows that up to 90.74/97.23/98.67% of strings have lengths less than 1000/1500/2000, respectively. We converted all the sequences to the same length of N. This length was determined through cross-validation (described in Section 3.1). The conversion was performed either by cutting off amino acids at the end of the sequence at the positions after the position N for sequences having lengths greater than N or by zero-padding on sequences having lengths less than N.

2.2.2. CNN Model Architecture. The CNN model (Figure 3) consists of four convolution layers with 32, 64, 128, and 256

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

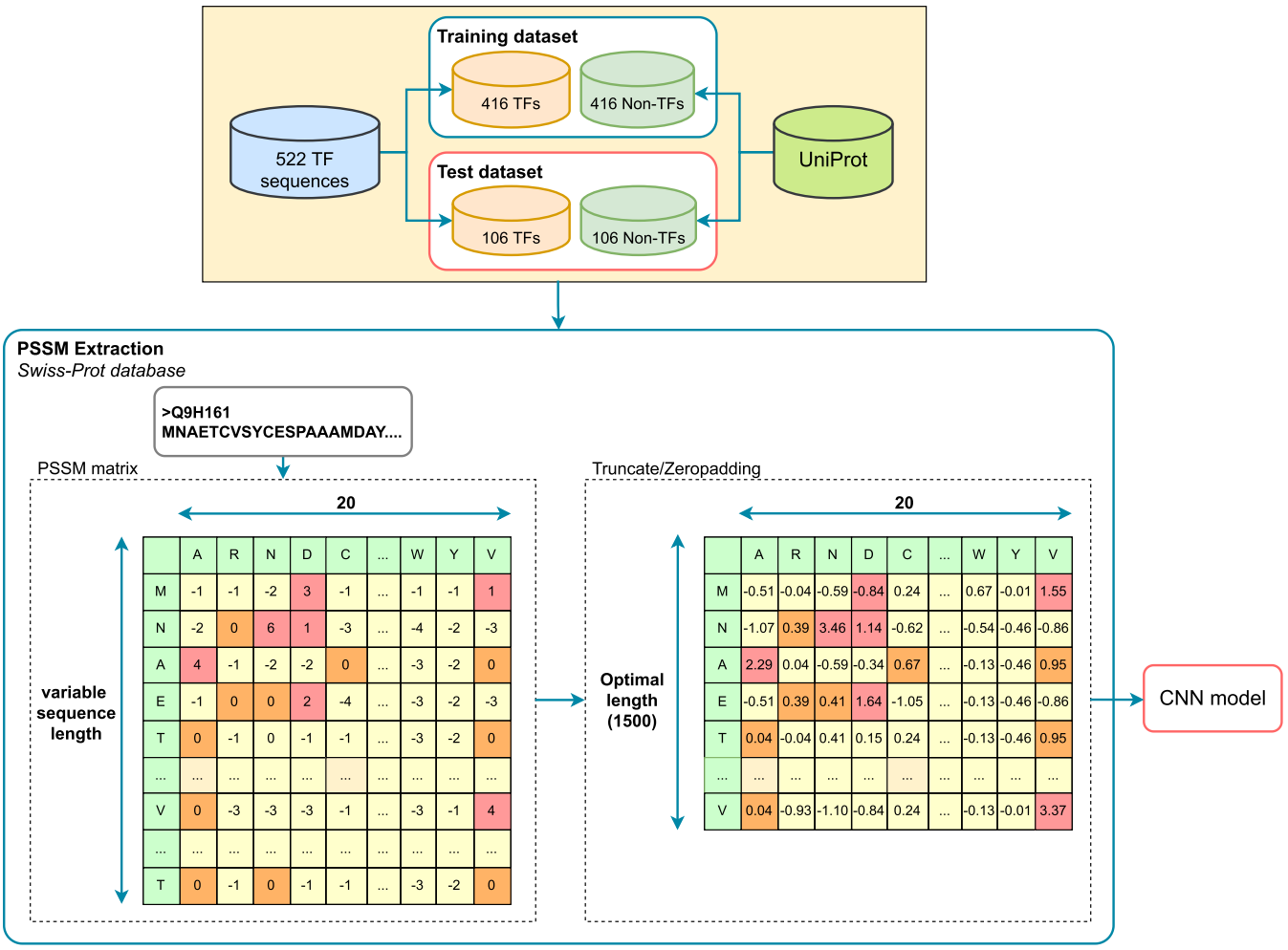


Figure 1. Data representation for TF classification.

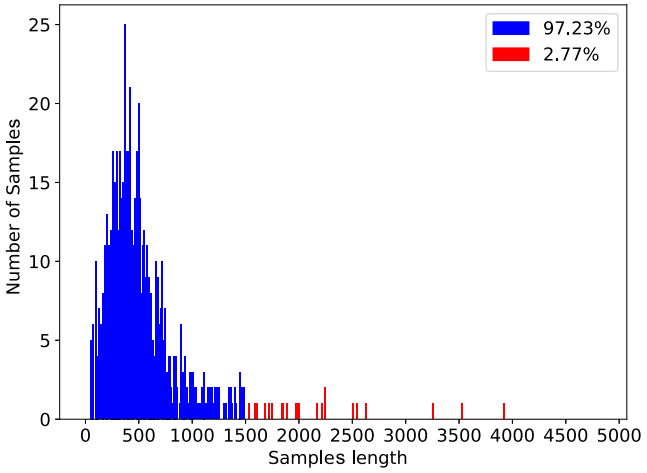


Figure 2. Distribution of amino acids in the protein sequences in the training set, showing that 97.23% of the sequences have lengths less than 1500.

filters, with a kernel size of 3 × 3 and a stride of (1, 1). After each convolution layer, there are MaxPooling2D layers with a kernel size of 2 × 2 and a Dropout layer with a ratio of 0.2. The features extracted from the convolutional network layers are then passed through a GlobalAveragePooling2D layer and then two fully connected layers with 256 neurons. Finally, the output layer with the Softmax function produces probabilities for the two classes: TF (positive) and non-TF (negative).

Our CNN models were trained using the Adam optimization algorithm with sparse categorical cross-entropy loss and an initial learning rate of 0.001. During model training, we use the *early stopping* method with a patient length of 10 to stop training if the validation loss does not decrease after 10 epochs. In addition, the *reduce on plateau* technique with a patient length of 5 is also used to reduce the learning rate when the validation loss does not decrease after five epochs.

2.2.3. Ensemble Learning. We used fivefold cross-validation to train five CNN models, and then, ensemble learning is used to combine the predictions by the five models. The final outcome is determined by the median of the outputs of the five models (Figure 4).

2.3. Identifying TFs That Prefer Binding to Methylated DNA. An overview of data representation for our TFPM classification framework [with Op(13) and 2-gap as a setting example] is shown in Figure 5. First, proteins were parametrically extracted using the reduced *g*-gap DC method. Then, the SVM algorithm is used for TFPM classification. The best hyperparameters of the data representation step and the SVM model were determined from fivefold cross-validation.

2.3.1. Reduced G-Gap DC. 2.3.1.1. Reduced Amino Acids. Reduced amino acids (RAAs) is a method of grouping 20 basic amino acids into different groups and then representing the

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

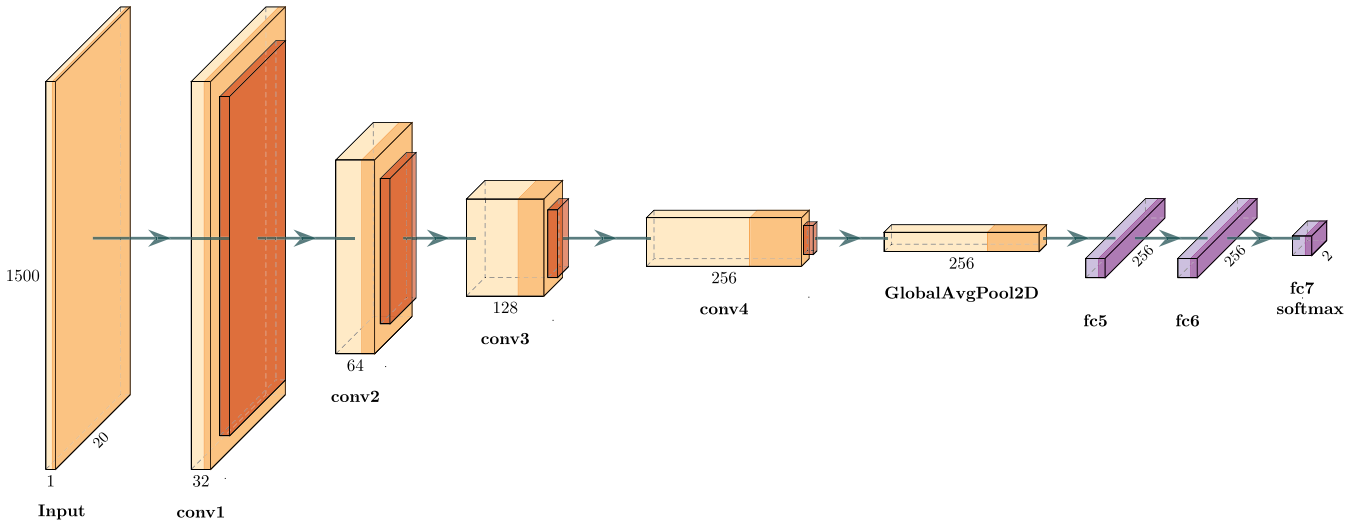


Figure 3. Architecture of the CNN models used for TF classification.

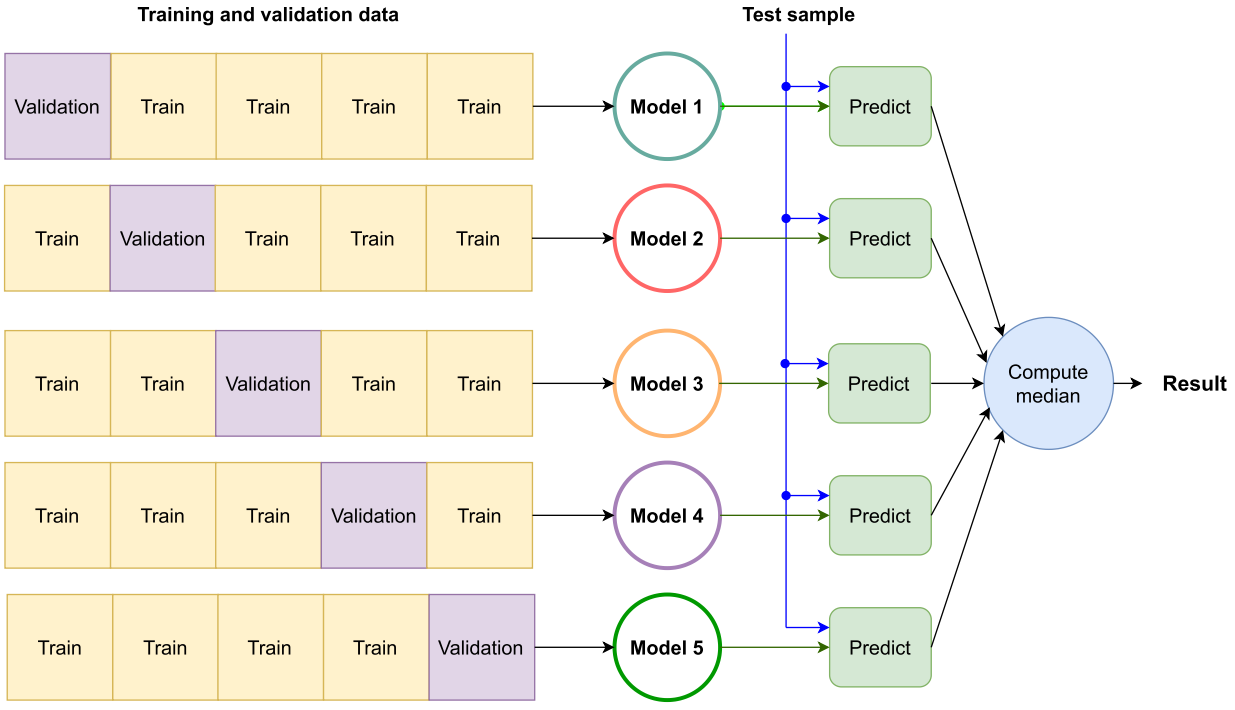


Figure 4. Ensemble learning for TF classification.

amino acids in each group by the corresponding group symbol. Many studies and experiments have shown that reducing the complexity of the original protein sequence helps reduce information redundancy and improve the computational efficiency of the subsequent machine learning model.[19−21] The most important thing in the RAA method is the way to group amino acids. Zheng et al.[22] proposed the RAACBook method in which the authors used up to 74 different evaluation methods to classify amino acids. With each evaluation method, there were up to 18 different grouping methods, from which up to 673 schemes were created. This method was applied in classification of human enzymes[21] and in classification of antimicrobial peptides.[23] Another way of grouping amino acids was proposed by Takabatake et al.[24] in which the BLOSUM62 matrix was used to calculate the number of similarity scores between amino acids prior to calculating the correlation coefficients of each pair of amino acids and grouping the amino acids using hierarchical clustering.

In the study by Etchebest et al.,[19] the authors based on their previous work on 3D protein structures built from a limited number of different amino acid blocks to identify 16 protein blocks (PBs). From the 16 PBs found, they encoded the PDB-REPRDB dataset into sequences of PBs. Then, they computed 16 amino acid occurrence matrices of size 20 × 15 using windowing, followed by transformations and combinations to get a final matrix of size 20 × 240. From this matrix, they calculated the distances between amino acids and clustered the amino acids using R software. After analysis, they proposed Op(13) with the division of 20 amino acids into 13 groups (Table 2). They reported that if some information in the analysis could be ignored, they could reduce the number of

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

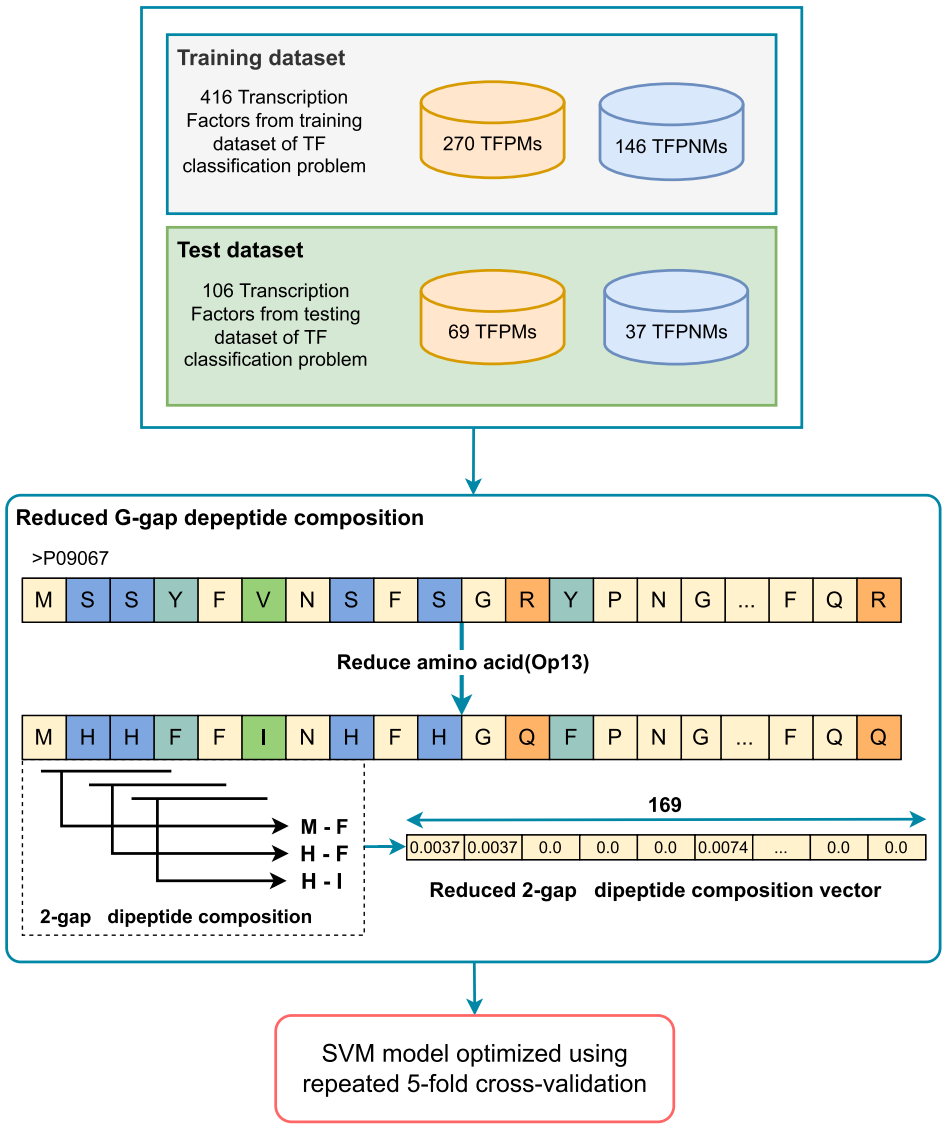


Figure 5. Data representation for TFPM classification.

Table 2. Op(13) Grouping for Amino Acids

index	group of amino acids	symbols for the group
1	G (glycine)	G
2	I (isoleucine), V (valine)	I
3	F (phenylalanine), Y(tyrosine), W (tryptophan)	F
4	A (alanine)	A
5	L (leucine)	L
6	M (methionine)	M
7	E (glutamic acid)	E
8	Q (glutamine), R (arginine), K (lysine)	Q
9	P (proline)	P
10	N (asparagine), D (aspartic acid)	N
11	H (histidine), S (serine)	H
12	T (threonine)	T
13	C (cysteine)	C

groups and thus get Op(11), Op(9), Op(8), and Op(5) with 11, 9, 8, and 5 groups, respectively.

As mentioned above, the RAA has been widely used by many research groups, and there are new proposals on how to divide groups for RAAs. In this study, we extend an efficient clustering method proposed by Etchebest et al.[19] by combining RAAs and *g*-gap DC to obtain an efficient encoding method for the TFPM classification problem.

2.3.1.2. G-Gap DC. *G*-gap DC (GDC) is an extension of the amino acid composition method for encoding protein sequences. In this method, the protein sequence is encoded into a 400-dimensional vector showing the frequency of 400 dipeptides with gaps between two amino acids; these gaps can be 1, 2, 3, and so forth. There have been many studies with different proposals using GDC. Feng et al.[25] used GDC with gaps between 0 and 5 in classifying the super-family of small heat shock proteins. For prediction of antihypertensive peptides, Rauf et al.[26] used a 20 × 20 matrix format with different gaps (from 0 to 3) to generate different encoding matrices. In other studies, GDC was also combined with other encoding schemes including amino acid composition, auto-cross-covariance, and PseAAC.[27,28]

In this study, we propose the reduced GDC (RGDC), an enhanced version of GDC for TFPM classification. We extend the GDC method by combining it with RAAs to obtain an efficient encoding scheme for TFPM classification.

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

2.3.1.3. Reduced GDC. Our proposed RGDC scheme is a combination of two methods: RAA and GDC. The first step of RGDC is to reduce the number of 20 basic amino acids by grouping the amino acids into groups and using one amino acid in each group as the representative of that group.[29] Then, the amino acids in each protein sequence are replaced by their group representatives. This will remove some redundant information while preserving meaningful information of the sequence.[30,31] In order to use the RAA method, we need to group the amino acids together. Here, we use the scheme proposed by Etchebest et al.,[19] in which an optimization procedure (Op) was used to group the 20 basic amino acids into different groups.

After replacing the amino acids in the original protein sequence with their representatives, we compute the GDC feature vector. For N groups, the vector GDC produces N^2 features in the form of $x = [x_1^g, x_2^g, \ldots, x_{N2}^g]$, where each x_i^g represents the frequency occurrence of the ith g-gap dipeptide in the input protein sequence. Each x_i^g is calculated as in eq 1.

$$x_i^g = \frac{n_i^g}{\sum_{j=1}^{N2} n_j^g} = \frac{n_i^g}{L - g - 1} \quad (1)$$

where n_i^g is the total number of occurrences in the input protein sequence of the ith g-gap in the N^2 g-gap dipeptide, L is the length of the input protein sequence, and g is the number of gaps used. $g = 0$ means that there is no gap between two consecutive amino acids; $g = 1$ means that there is a corresponding gap between two consecutive amino acids, similarly for $g = 2$, 3, and so forth.

2.3.2. SVM Model. SVM is a machine learning model that has been proven to be effective in many studies.[32−36] In this study, we use an SVM model pre-installed in the scikit-learn library,[37] in which the kernel is the radial basis function with hyperparameters γ and C tuned from repeating 5-fold cross-validation 10 times with different randomization in each repetition. Input data are normalized to the normal distribution $N(0, 1)$, and the model outputs the probability of each class.

2.4. Evaluation Metrics. To assess the model performance, common evaluation metrics including accuracy (ACC), sensitivity (SEN), specificity (SPE), Matthew's correlation coefficient (MCC), and AUC were used. These metrics are computed as in eqs 2−5, in which TPs, FPs, TNs, and FNs are the true positives, false positives, true negatives, and false negatives, respectively.

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (2)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (4)$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (5)$$

3. RESULTS AND DISCUSSION

For classification of TFs, we tested different lengths of 1000, 1500, and 2000 when encoding input sequences. Table 3 shows the performances of the first CNN model from fivefold cross-validation. It can be seen that the length of 1500 gave the best result in terms of the AUC value. We then fixed 1500 as the optimal length when encoding input sequences using PSSMs and used this setting when testing the proposed framework with the independent test set.

Table 3. Performances of the First Model with Different Input Lengths

input length	sensitivity	specificity	accuracy	MCC	AUC
1000	0.7710	0.9518	0.8614	0.7349	0.9577
1500	0.7831	0.9638	0.8734	0.7594	0.9585
2000	0.7590	0.9518	0.8554	0.7244	0.9522

For classification of TFPMs, we tested several amino acid grouping options[19] including Op(5), Op(8), Op(9), Op(11), Op(13), and without using grouping [i.e., using Op(20)] in order to find the optimal scheme. In combination with different amino acid grouping options, we also tested different gaps including 1-gap, 2-gap, 3-gap, 4-gap, and not using g-gap (i.e., $g = 0$). For each pair of Op(.) and g-gap, the performances of SVM models with different hyperparameters γ and C from a pre-defined grid were computed from fivefold cross-validation, and the highest AUC value was recorded as shown in Figure 6.

As seen from Figure 6, the best validation AUC of 0.8529 corresponded to Op(8) with 3-gap, and this corresponded to the SVM model with $\gamma = 0.1$ and $C = 0.01$. This setting was applied when testing the proposed framework with the independent test set.

Table 4 summarizes our results in comparison with those of other state-of-the-art methods using the same training and test datasets.

As can be seen in Table 4, our proposed methods outperform the two recent studies by Liu et al.[7] and Li et al.[8] on both classification problems. Our methods achieve higher AUC, accuracy, sensitivity, and MCC than those of the other two methods while maintaining a competitive specificity with theirs. For the TF classification problem, we use PSSMs to represent input proteins. This representation has been shown to be effective in multiple protein classification problems.[10,11] Each PSSM has a fixed size of 1500 × 20 which is much smaller than the average size of the 100-dimensional tripeptide word vectors in the study of Li et al.[8]—whose average size is about 100 × 500 as the sequences in TF classification have about 500 amino acids on average. On the other hand, the size of our PSSM is much higher than that of the 201-dimensional feature used in the study by Liu et al.[7] However, this PSSM size seems to be suitable for applying deep learning on this specific dataset. In addition, ensemble learning helps improve the predictive performance as it makes full use of the training data while still maintaining certain differences among the individual models. For the TFPM classification problem, the RGDC method is proved to be effective. One possible reason is that this method calculates statistics of amino acid pairs separated by a certain distance (number of gaps) in proteins, and the relationship between these amino acid pairs may be related to their ability to bind to sites on the DNA sequence around the methylated site. We do not use deep learning for TFPM classification, as the dataset is

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

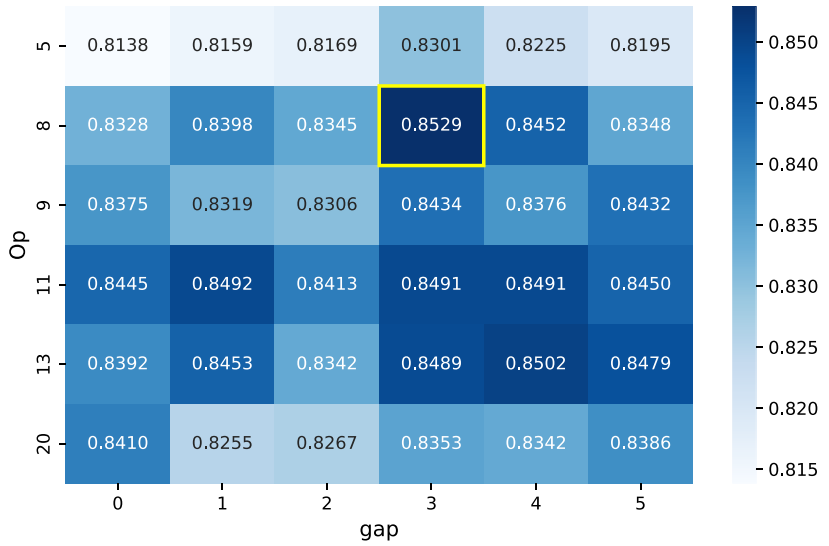


Figure 6. Best AUC values from repeated fivefold cross-validation for different settings of reduced GDC.

Table 4. Comparison of the Proposed Frameworks with Other Methods on the Same Independent Test Set

problem	method	sensitivity	specificity	accuracy	MCC	AUC
TF vs non-TF	Liu et al.[7]	0.8019	**0.8585**	0.8302	0.6614	0.9116
	Li et al.[8]	0.8868	0.8396	0.8663	0.7272	0.9130
	PSSM + CNN (Ours)	**0.9056**	0.8396	**0.8726**	**0.7469**	**0.9596**
TFPM vs TFPNM	Liu et al.[7]	0.7101	0.6486	0.6887	0.3471	0.7356
	Li et al.[8]	0.7826	**0.6487**	0.7359	**0.4831**	0.8324
	RGDC + SVM (Ours)	**0.8261**	0.6486	**0.7642**	0.4778	**0.8486**

relatively small. We propose using SVM, a classification method that often works well with small multidimensional datasets and uses repeated fivefold cross-validation to tune the SVM model for generalization ability. Our preliminary work also confirmed that SVM performed much better than other machine learning algorithms on this problem. Using Op(8), each RGDP feature vector has 64 dimensions which is smaller than the 200-dimensional dipeptide composition feature in Liu et al.[7] and the averaged size of 200 × 470 of the tripeptide word vectors in Li et al.[8] (the sequences in TFPM classification have about 470 amino acids on average). This small size of the RGDP feature vectors is suitable to be used with SVM for this small dataset. All of these factors contribute to the effectiveness of our proposed methods.

4. WEB-BASED APPLICATION

To support the research community in identifying TFs and TFPMs, we deployed our proposed framework as an online web server with a user-friendly interface. A link to the web server is available at https://github.com/ngphubinh/iTFPM-RGDC. Our web-based application supports protein sequences stored in the FASTA format. After input sequences are submitted, the predicted results will be returned.

5. CONCLUSIONS

Although the fact that TFs can bind to methylated DNA has been recently confirmed, the mechanism of this relation is still unclear, and it is often costly and time-consuming to use experimental methods to identify TFs and TFPMs. In this study, we propose two efficient machine learning frameworks for predicting TFs and TFPMs. The two data representation schemes, PSSMs for TF identification and RGDC for TFPM identification, contribute significantly to the superior performance of our frameworks when compared with other state-of-the-art methods. In future, the relation between other protein representation schemes and more advanced deep learning architectures and machine learning algorithms will be considered to further improve the performance of the methods.

AUTHOR INFORMATION

Corresponding Author

Trang T. T. Do − *School of Innovation, Design and Technology, Wellington Institute of Technology, Lower Hutt 5012, New Zealand;* orcid.org/0000-0002-1614-4661; Phone: +64 4 920 2627; Email: trang.do@weltec.ac.nz

Authors

Quang H. Nguyen − *School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam*

Hoang V. Tran − *School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam;* orcid.org/0000-0002-7493-3693

Binh P. Nguyen − *School of Mathematics and Statistics, Victoria University of Wellington, Wellington 6140, New Zealand;* orcid.org/0000-0001-6203-6664

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.2c03696

Author Contributions

‖Q.H.N. and H.V.T. contributed to the manuscript equally.

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

Notes

The authors declare no competing financial interest.

Data used in this study were collected from Liu et al.[7] Source code, data, and a link to our web server are available at https://github.com/ngphubinh/iTFPM-RGDC.

■ ACKNOWLEDGMENTS

H.V.T. was funded by Vingroup and supported by the Master, PhD Scholarship Programme/Postdoctoral Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VinBigdata), code VINIF.2021.ThS.52.

■ REFERENCES

(1) Wang, G.; Wang, F.; Huang, Q.; Li, Y.; Liu, Y.; Wang, Y. Understanding transcription factor regulation by integrating gene expression and DNase I hypersensitive sites. *BioMed Res. Int.* **2015**, *2015*, 757530.

(2) Kim, S.; Shendure, J. Mechanisms of interplay between transcription factors and the 3D genome. *Mol. Cell* **2019**, *76*, 306−319.

(3) Blattler, A.; Farnham, P. J. Cross-talk between site-specific transcription factors and DNA methylation states. *J. Biol. Chem.* **2013**, *288*, 34287−34294.

(4) Kribelbauer, J. F.; Lu, X.-J.; Rohs, R.; Mann, R. S.; Bussemaker, H. J. Toward a mechanistic understanding of DNA methylation readout by transcription factors. *J. Mol. Biol.* **2020**, *432*, 1801−1815.

(5) Héberlé, É.; Bardet, A. F. Sensitivity of transcription factors to DNA methylation. *Essays Biochem.* **2019**, *63*, 727−741.

(6) Wang, G.; Luo, X.; Wang, J.; Wan, J.; Xia, S.; Zhu, H.; Qian, J.; Wang, Y. MeDReaders: a database for transcription factors that bind to methylated DNA. *Nucleic Acids Res.* **2018**, *46*, D146−D151.

(7) Liu, M.-L.; Su, W.; Wang, J.-S.; Yang, Y.-H.; Yang, H.; Lin, H. Predicting preference of transcription factors for methylated DNA using sequence information. *Mol. Ther. Nucleic Acids* **2020**, *22*, 1043−1050.

(8) Li, H.; Gong, Y.; Liu, Y.; Lin, H.; Wang, G. Detection of transcription factors binding to methylated DNA by deep recurrent neural network. *Briefings Bioinf.* **2022**, *23*, bbab533.

(9) Cheol Jeong, J.; Lin, X.; Chen, X.-W. On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2011**, *8*, 308−315.

(10) Nguyen, B. P.; Nguyen, Q. H.; Doan-Ngoc, G.-N.; Nguyen-Vo, T.-H.; Rahardja, S. iProDNA-CapsNet Identifying Protein-DNA Binding Residues using Capsule Neural Networks. *BMC Bioinf.* **2019**, *20*, 634.

(11) Khanh Le, N. Q.; Nguyen, Q. H.; Chen, X.; Rahardja, S.; Nguyen, B. P. Classification of Adaptor Proteins using Recurrent Neural Networks and PSSM Profiles. *BMC Genom.* **2019**, *20*, 996.

(12) Ge, F.; Zhu, Y.-H.; Xu, J.; Muhammad, A.; Song, J.; Yu, D.-J. MutTMPredictor: Robust and accurate cascade XGBoost classifier for prediction of mutations in transmembrane proteins. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 6400−6416.

(13) Zhang, S.; Zhu, F.; Yu, Q.; Zhu, X. Identifying DNA-binding proteins based on multi-features and LASSO feature selection. *Biopolymers* **2021**, *112*, No. e23419.

(14) Pan, J.; Li, L.-P.; Yu, C.-Q.; You, Z.-H.; Guan, Y.-J.; Ren, Z.-H. Sequence-based prediction of plant protein-protein interactions by combining discrete sine transformation with rotation forest. *Evol. Bioinf.* **2021**, *17*, 11769343211050067.

(15) Chen, T.-R.; Juan, S.-H.; Huang, Y.-W.; Lin, Y.-C.; Lo, W.-C. A secondary structure-based position-specific scoring matrix applied to the improvement in protein secondary structure prediction. *PLoS One* **2021**, *16*, No. e0255076.

(16) Le, N.-Q.-K.; Nguyen, B. P. Prediction of FMN Binding Sites in Electron Transport Chains based on 2-D CNN and PSSM Profiles. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2021**, *18*, 2189−2197.

(17) Wang, N.; Zhang, J.; Liu, B. IDRBP-PPCT identifying nucleic acid-binding proteins based on position-specific score matrix and position-specific frequency matrix cross transformation. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2021**, *19*, 2284.

(18) Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **1997**, *25*, 3389−3402.

(19) Etchebest, C.; Benros, C.; Bornot, A.; Camproux, A.-C.; de Brevern, A. A reduced amino acid alphabet for understanding and designing protein adaptation to mutation. *Eur. Biophys. J.* **2007**, *36*, 1059−1069.

(20) Melo, F.; Marti-Renom, M. A. Accuracy of sequence alignment and fold assessment using reduced amino acid alphabets. *Proteins: Struct., Funct., Bioinf.* **2006**, *63*, 986−995.

(21) Wang, H.; Xi, Q.; Liang, P.; Zheng, L.; Hong, Y.; Zuo, Y. IHEC_RAAC: a online platform for identifying human enzyme classes via reduced amino acid cluster strategy. *Amino Acids* **2021**, *53*, 239−251.

(22) Zheng, L.; Huang, S.; Mu, N.; Zhang, H.; Zhang, J.; Chang, Y.; Yang, L.; Zuo, Y. RAACBook: a web server of reduced amino acid alphabet for sequence-dependent inference by using Chou's five-step rule. *Database* **2019**, *2019*, baz131.

(23) Dong, G.-F.; Zheng, L.; Huang, S.-H.; Gao, J.; Zuo, Y. Amino acid reduction can help to improve the identification of antimicrobial peptides and their functional activities. *Front. Genet.* **2021**, *12*, 669328.

(24) Takabatake, K.; Izawa, K.; Akikawa, M.; Yanagisawa, K.; Ohue, M.; Akiyama, Y. Improved Large-Scale Homology Search by Two-Step Seed Search Using Multiple Reduced Amino Acid Alphabets. *Genes* **2021**, *12*, 1455.

(25) Feng, P.; Liu, W.; Huang, C.; Tang, Z. Classifying the superfamily of small heat shock proteins by using G-gap dipeptide compositions. *Int. J. Biol. Macromol.* **2021**, *167*, 1575−1578.

(26) Rauf, A.; Kiran, A.; Hassan, M. T.; Mahmood, S.; Mustafa, G.; Jeon, M. Boosted Prediction of Antihypertensive Peptides Using Deep Learning. *Appl. Sci.* **2021**, *11*, 2316.

(27) Shen, Z.; Liu, T.; Xu, T. Accurate Identification of Antioxidant Proteins Based on a Combination of Machine Learning Techniques and Hidden Markov Model Profiles. *Comput. Math. Methods Med.* **2021**, *2021*, 5770981.

(28) Zhang, D.; Chen, H.-D.; Zulfiqar, H.; Yuan, S.-S.; Huang, Q.-L.; Zhang, Z.-Y.; Deng, K.-J. iBLP: an XGBoost-based predictor for identifying bioluminescent proteins. *Comput. Math. Methods Med.* **2021**, *2021*, 6664362.

(29) Solis, A. D.; Rackovsky, S. Optimized representations and maximal information in proteins. *Proteins: Struct., Funct., Bioinf.* **2000**, *38*, 149−164.

(30) Wang, J.; Wang, W. A computational approach to simplifying the protein folding alphabet. *Nat. Struct. Biol.* **1999**, *6*, 1033−1038.

(31) Zuo, Y.; Li, Y.; Chen, Y.; Li, G.; Yan, Z.; Yang, L. PseKRAAC: a flexible web server for generating pseudo K-tuple reduced amino acids composition. *Bioinf* **2017**, *33*, 122−124.

(32) Manavalan, B.; Shin, T. H.; Lee, G. PVP-SVM: sequence-based prediction of phage virion proteins using a support vector machine. *Front. Microbiol.* **2018**, *9*, 476.

(33) Dao, T.-K.; Nguyen, T.-T.; Pan, J.-S.; Qiao, Y.; Lai, Q.-A. Identification failure data for cluster heads aggregation in WSN based on improving classification of SVM. *IEEE Access* **2020**, *8*, 61070−61084.

(34) Li, X.; Tang, Q.; Tang, H.; Chen, W. Identifying antioxidant proteins by combining multiple methods. *Front. Bioeng. Biotechnol.* **2020**, *8*, 858.

(35) Nguyen, Q. H.; Nguyen, B. P.; Nguyen, T. B.; Do, T. T. T.; Mbinta, J. F.; Simpson, C. R. Stacking Segment-based CNN with SVM for Recognition of Atrial Fibrillation from Single-lead ECG Recordings. *Biomed. Signal Process Control* **2021**, *68*, 102672.

(36) Ali, L.; Wajahat, I.; Amiri Golilarz, N.; Keshtkar, F.; Bukhari, S. A. C. LDA−GA−SVM improved hepatocellular carcinoma prediction

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330

through dimensionality reduction and genetically optimized support vector machine. *Neural Comput. Appl.* **2021**, *33*, 2783−2792.
(37) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

https://doi.org/10.1021/acsomega.2c03696
ACS Omega 2022, 7, 32322−32330