Received November 19, 2019, accepted December 9, 2019, date of publication December 24, 2019,
date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961941

K_net: Lysine Malonylation Sites Identification With Neural Network

JIN SUN[1], YI CAO[1], DONG WANG[1], WENZHENG BAO[2], AND YUEHUI CHEN[1]

[1]School of Information, University of Jinan, Jinan 250024, China

[2]School of Information and Electrical Engineering, Xuzhou University of Technology, Xuzhou 221018, China

Corresponding author: Wenzheng Bao (baowz55555@126.com)

This work was supported in part by the National Science Foundation of China under Grant 61902337, and in part by the National Key Research and Development Program of China under 2016YFC0106000.

ABSTRACT Lysine Malonylation (Kmal) is a newly discovered protein post-translational modifications (PTMs) type, which plays an important role in many biological processes. Therefore, identifying and understanding Kmal sites is very critical in the studies of biology and diseases. The typical methods are time-wasting and expensive. Nowadays, many researchers have proposed machine learning (ML) methods to deal with PTMs's identification issue. Especially, some deep learning (DL) methods are also utilized in this field. In this work, we proposed K_net, which employed Convolutional Neural Network to identify the potential sites. Meanwhile, we proposed a new verification method Split to Equal Validation (SEV), which can well solve the impact of sample imbalance on prediction results. More Specifically, Acc, Sn, Sp, MCC and AUC values were adopted to evaluate the prediction performance of predictors. In total, CNN_Kmal achieved the better performance than other methods.

INDEX TERMS Lysine malonylation, deep learning, convolutional neural network, split to equal validation.

I. INTRODUCTION

Protein post-translational modification (PTM) is a key mechanism that influence almost all aspects of cell biology and pathogenesis [1], [2]. PTMs can regulate protein functions by covalent addition of functional groups or small molecules, which plays important role in some internal life processes of organisms like metabolism and material transportation [3]–[5]. For instance, Lysine glycation is characterized as an important regulator for aging and pathogenesis of diabetes. acetylation, another important type of lysine PTM, is associated with protein stability, protein–protein interactions and cellular metabolism [6]–[8]. In fact, many protein residues owe various PTM sites, and there are almost various modification types in the same residue [9]–[14]. More than 20 type of PTM s have been characterized, such as lysine acetylation, glycation and methylation [15]–[18]. As an original identified modification type, lysine malonylation has been widely existed on lysine residues. There are many researches related to Kmal in recent researches: For instance, malonylation on K184 of glyceraldehyde 3-phosphate dehydrogenase regulates the activity of this key metabolic enzyme, and several Kmal sites in histone proteins have potential connections with cancer. Therefore, identifying and understanding Kmal sites become very urgent in the studies of biology and diseases [9], [19]–[21].

Due to the rapid development of computational methods and their cross- disciplinary characteristics, many machine learning (ML) models have become very popular tools in many fields including bioinformatics. As for PTMs researches, Support vector machine (SVM) was utilized in MaloPred for malonylation site prediction using 10-fold cross-validation [22], [23]. NetGlycate-1.0, a predictor made by combining 60 artificial neural networks (ANN), was proposed for Prediction of glycation sites [23]–[25]. The flexible neural tree algorithm, which is a new neural network model proposed by Chen et al, was also applied in prediction of phosphorylation sites using the features according to different encoding schemes [26], [27].

In addition to traditional machine learning methods, with the deepening of theoretical research and wider application of deep learning method, it has become a hot tool for classification and regression problems. Many researchers have utilized deep learning tools to do PTM experiments: the recently published tool MusiteDeep is utilized for general and kinase-specific phosphorylation site

The associate editor coordinating the review of this manuscript and approving it for publication was Yongtao Hao.

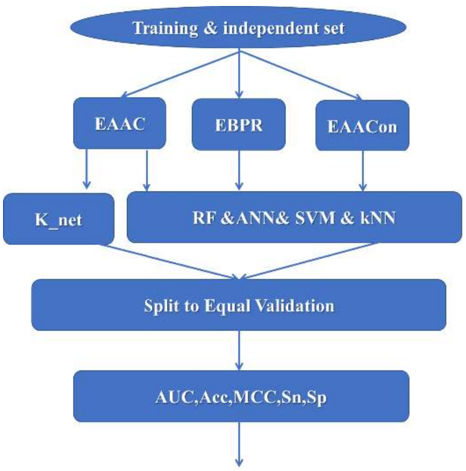

FIGURE 1. The activity flow of Classifier development.

prediction [28]. Muscadel successfully applies multi-layer deep neural network to predict nitration and nitrosylation sites, and long short-term memory (LSTM) recurrent neural networks (RNNs) were utilized in DeepNitro to predict eight types of lysine PTMs [29].

In this article, we proposed a deep learning classifier named K_net, which employed Convolutional Neural Network to identify the potential sites. 4 traditional machine learning classifiers were constructed to compare the performance. Meanwhile, we proposed a new verification method Split to Equal Validation (SEV), which can well solve the impact of sample imbalance on prediction results. In total, As AUC value, accuracy (Acc), sensitivity (Sn), specificity (Sp), and Matthew's correlation coefficient (MCC) were adopted to evaluate the prediction performance of predictors, CNN_Kmal performed better than other classifiers.

II. METHODS AND MATERIALS

The working flow of our works is demonstrated in Figure 1. The first step is data procession, through which the dataset would be refined. The following procedure is feature extraction, which converted the input data into feature vectors. And then, we will train different models on the training set and evaluated by 10-fold cross validation as well as Split to Equal validation (SEV) [30], [31]. Finally, the trained model will be validated on independent test data, and then 5 metrics would be used to evaluate the performance of predictors.

In details, EAAC, EBPR and EAACon are different feature extraction shames, EAAC were developed by chen et al. based on AAC encoding, EBPR were proposed by Han et al. and EAACon were proposed by us based on EAAC and Convolutional Neural Network. Besedes, K_net is the model proposed by us based on CNN, and Split to Equal Validation were also proposed by us, which can both achieve two functions of data preprocessing and cross-validation.

In order to develop an effective predictor, we derived a non-redundant dataset with high confidence from the database established by Chen et al. [23] In details, (1)The initial set consists of a total of 10368 Kmal sites with high confidence (i.e., Kmal peptides with Andromeda scores > 50 and localization probability > 0.75) and 142830 negative sites belonging to different human and mice proteins. With the lysine site in the center, 31-residue peptides (-15 to +15) were extracted from the representatives. It's worth emphasizing that if the peptides containing positive sites were identical to the peptides containing negative sites, both peptides were removed. And then, the dataset was separated into two groups: one for training and other for testing. In total, 5032 positive and 62299 negative peptides sequence were retained for training and then were subjected to ten-fold cross-validation and Split to Equal validation (SEV) proposed by us, other 1046 positive peptides and 16827 negative peptides were employed as the independent test dataset.

A. FEATURE ENCODINGS

1) EAAC ENCODING

The EAAC encoding that proposed by Chen et. Al reflects the frequency of 20 amino acid residues surrounding the modification site. In details, we can define an 8-size window continuously sliding from the N-terminus to C-terminus of a peptide, and the frequency of the 20 amino acid residues appeared in each 8-dimensional peptide chain fragment was counted. Accordingly, the dimension of features can be calculated as follows:

$$N\_s = L\_p - L\_s + 1 \tag{1}$$

$$D\_eaac = N\_s \times 20 \tag{2}$$

where, L_p refers to the length of each peptide, L_s is the length of sliding windows and D_eaac is the dimension of feature vector. As all of the peptides owe 31 residues, each peptide in the dataset would be converted to a matrix of 24 (31-8 + 1) $*$ 20 dimensions, and then turned into a 480-dimensional vector after stretched by rows.

2) EBAG ENCODING

It has been proved that amino acid residues can be grouped according to their various physical and chemical properties in previous research. Based on this theory, we adopt a feature encoding method called Encoding Based on Attribute Grouping (EBAG) [24], [25], which divide 20 amino acid residues into 4 groups containing hydrophobic group, polar group, acidic group and basic group. Although some intervals existed in the amino acid sequences have no separate physical and chemical properties, they can be a basis for identifying whether a site can be modified or no, and we divide them into the fifth group denoted by "X" (Table 1):

3) EBAG + PROFILE ENCODING

As each peptide contains 31 amino acid residues, Profile Encoding can calculate the frequency of each residue and then generate the frequency sequence for every peptide. The frequency of each residue can be calculated as follows:

$$F\_i = C\_i/L \tag{3}$$

TABLE 1. The groups generated by EBAG method.

Group	Amino Acid Residue	Label
C1	A, F, G, I, L, M, P, V, W	Hydrophobic
C2	C, N, Q, S, T, Y	Polar
C3	D, E	Acidic
C4	H, K, R	Basic
C5	X	Intervals

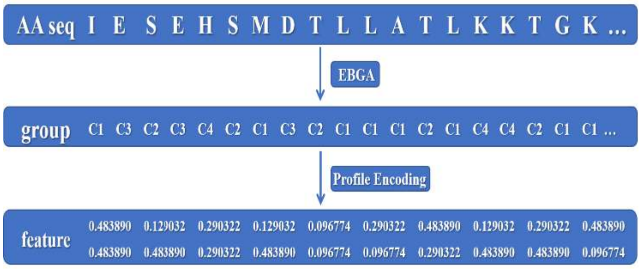

FIGURE 2. Example of the EBAG + Profile encoding method utilizing to an amino acid sequence.

where, L is length of the sample; i is type of amino acid residue, and C_i is its times appeared in the peptide. And then, a sample can be converted a feature vector PV as follows:

$$PV = [F\_i] i \in [1, 2, 3, \ldots 20] \tag{4}$$

There is a novel strategy to integrate Profile encoding method with EBAG method, called EBPR method, which shows better performance than each of it. The first step is splitting a 31-dimension peptide into a new sequence with 5 groups, which generates a new sequence by EBAG method. And then Profile encoding method was utilized to encode the generative EBAG sequence, where all residues are assigned values according the frequency calculated by this method. The example of EGPR feature encoding is shown in Figure 2.

B. CONSTRUCTION OF CLASSIFIERS

1) RANDOM FOREST

RF is a bagging type of ensemble methods that integrate lots of decision trees. The basic idea of RF is to train the model with the original data until it completely split and finally get multiple decision trees, and the prediction of new data is to average all decision trees. Although there are many parameters in the random forest that can affect the final prediction accuracy of the random forest, the number of trees is undoubtedly the most important. In our experiments, we selected 1000 decision trees to build a RF model after continuous attempts, which makes the random forest model stand out in many traditional machine learning classifiers.

2) NEURAL NETWORK

NN, imitated from neurons and their connectivity in animal brains, is another commonly employed machine learning algorithm. Here, we developed an ANN classifier that is composed of three layers: the input layer was the initial layer which received the 31 residues of the peptide sequence fragments; the second layer is hidden layer, which is contained by 100 neurons; the last layer is output layer that a single unit is activated by the ‘‘sigmoid’’ function, outputting the probability score of Kmal modification.

3) SUPPORT VECTOR MACHINE

SVM is another widely employed algorithm for different classification problems in various field. It can transform the samples into a high dimensional feature space, and then construct an Optimal Separating Hyperplane (OSH) that maximize its distance from the closest training samples. In our SVM model, the kernel function was set to Radial Basis Function (RBF), while the penalty coefficient of the objective function is set to *1e-05* to balance classification interval margin and misclassification samples.

4) K-NEAREST NEIGHBORS ALGORITHM

kNN is a basic classification method that can obtain higher classification accuracy for unknown and non-normal distribution data. There are some basic elements of kNN, i.e. the choice of k-value, distance measure and classification decision rules. In our kNN model build in this work, we choose 4 as k, which represents the number of neighbors. Besides, the measure distance of our model is set to Euclidean Distance, which is the power parameter for the Minkowski metric.

5) CONVOLUTIONAL NEURAL NETWORKS

In addition to the traditional machine learning methods mentioned above, we also applied depth learning algorithm to our experiments. CNN is one of the most successful applications of depth learning algorithm, which includes one-dimensional, two-dimensional and three-dimensional convolution neural network. Among them, one-dimensional CNN is mainly applied in data processing of sequence classes, while two-dimensional CNN is often utilized to image recognition filed. Based on EAAC feature extraction method, we propose a new idea to integrate the EAAC encoding and the CNN model, and then we establish two CNN models through this ideal: One is named CNN1d$_{\text{eaac}}$, which is generated by EAAC and a one-dimensional CNN model, the other one is named CNN2d$_{\text{eaac}}$, which combines EAAC and a two-dimensional CNN model. Among them, CNN is the K net in our topic.

C. PERFORMANCE ASSESSMENT OF THE PREDICTORS

1) CROSS VALIDATION

Cross Validation (CV) is a statistic analysis strategy. In this work, 10-fold Cross Validation was employed to evaluate the performance of classifiers. The basic idea of CV is dividing the original dataset into 10 parts, where 9 parts were converted as training set and the last part as validation set. The classifiers will be trained with training set, and then the trained model will be tested by the independent set. There are 10 epochs in whole Cross Validation produces, and each part

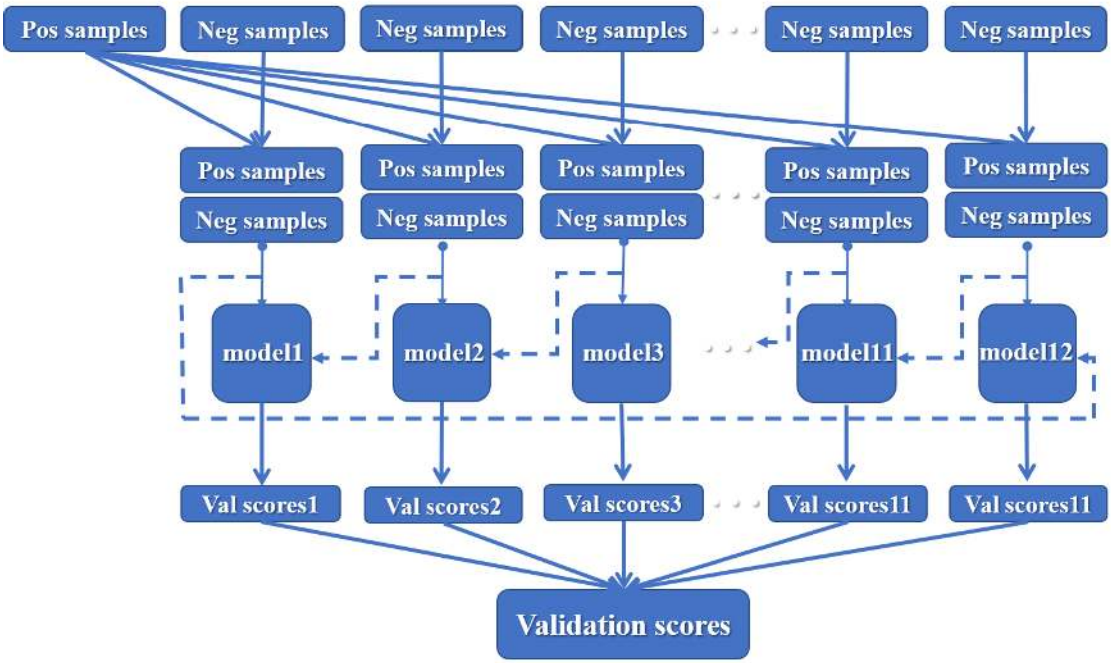

FIGURE 3. Activity flow of Split to Equal Validation (SEV).

in training set will be utilized as a verification set in each specific epoch. This validation strategy takes advantage of making full use of all of samples, however the calculation process is complicated for requiring training and testing for 10 times.

2) SPLIT TO EQUAL VALIDATION

As typically classifiers are more sensitive to detecting the majority class and less sensitive to the minority class, there is a need to preprocess imbalanced data before feed them into classifiers. If we don't take care of the issue, the classification output will be biased, and resulting in always predicting the majority class in many cases. Inspired by oversampling strategy and k-fold cross-validation, we proposed a novel validation schemes named Split to Equal validation (SEV) that can both achieve two functions of data preprocessing and cross-validation. In details, the first step is calculating the rate of majority samples to minority samples in training set:

$$r = N\_maj \div N\_min \tag{5}$$

$$Sub_{data} = [data_1, data_2, \cdots, data_i, \cdots data_r] \tag{6}$$

where, N_maj refers to the total number of negative samples and N_pos is the number of positive samples in our training set. And then the majority class (positive samples) will be split to r groups and each group will be combined with the minority class (positive samples) so that there are i new sub-training sets been generated. As the rate of positive samples to negative samples might be even in the order of 1 to 1, each sub-dataset will be utilized to train a current classifier and validate the previous classifier.

In this article, SEV contained five steps. As the ratio of negative and positive samples in training set approach to 12:1, the first step is to split the negative samples into 12 groups. The second step is to combine each positive group with the positive samples, so that 12 balanced sub-sets were generated. In the third step, model1 would be trained by the sub-set1 and validated by the sub-set2; model 2 would be trained by the sub-set2 and validated by the sub-set3, and do on. In total, 12 models were trained and validated according to these 12 balanced sub-sets. And then, each model would be tested by the independent dataset, the average of their scores would be utilized to assess their performance.

There are four measurements adopted to evaluate the prediction performance of predictors, i.e., accuracy (Acc), sensitivity (Sn), specificity (Sp), and Matthew's correlation coefficient (MCC). The definition of these four metrics is as follows:

$$Sn = \frac{TP}{TP + FN} \tag{7}$$

$$Sp = \frac{TN}{TN + FP} \tag{8}$$

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{9}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FN) \times (TN+FP)}} \tag{10}$$

Moreover, we also shown the Receiver-Operating Characteristic (ROC) curves and calculated Area Under the Curve (AUC) values to reflect the prediction performance.

TABLE 2. The results of different classifiers based on SEV and 10-fold CV.

Classifier	Validation method	AUC	Acc	MCC	Sn	Sp
RF	10-fold CV	0.7489	92.32%	0.0577	0.36%	100.00%
	SEV	**0.7642**	66.50%	0.2178	73.55%	65.91%
ANN	10-fold CV	0.7065	90.31%	0.1218	11.88%	96.86%
	SEV	**0.7237**	59.65%	0.1735	73.84%	58.47%
SVM	10-fold CV	0.5813	92.29%	0.0000	0.00%	100.00%
	SEV	**0.6394**	31.35%	0.0897	88.04%	26.62%
kNN	10-fold CV	0.6163	92.13%	0.1401	6.25%	99.30%
	SEV	**0.6626**	49.73%	0.1141	73.59%	47.73%

III. RESULTS AND DISCUSSION

A. PERFORMANCE OF THE SPLIT TO EQUAL VALIDATION (SEV) AND 10-FOLD CROSS VALIDATION

In previous researches, validation strategies, such as the cross validation, are widely utilized to overcome the intrinsic overfitting limits of various predictors. In this work, we proposed a novel validation strategy, Split to Equal Validation (SEV), for both dataset process and performance evaluation as well as k-fold cross validation. In details, we built 12 balanced, fragment-level non-redundant subsets by splitting the negative samples into 12 pieces, and each piece is combined with the positive sample to creating a new set of balanced set. All of sub-sets were utilizing for model training and model validation, respectively.

To compare the performance of SEV and conventional validation strategy in PTMs, we have built two models of each classifier (i.e. RF, ANN, SVM, kNN) based on different validation strategies of 10-fold cross validation strategy and Split to Equal validation. For the model utilized 10-fold cross-validation, we have one dataset which we divide into 10 parts, using 9 of those parts for training and reserve one tenth for validating. We repeated this procedure 10 times each time reserving a different tenth for validation test, and then the final models were tested by independent dataset. For the SEV model, we also have one dataset which utilized to build 12 balanced subsets. Each subset will be utilized as input to train the corresponding model and then utilized as validation set to test the former model. Finally, the SEV model was also tested by independent dataset, in terms of AUC, Acc, Sn, Sp, and MCC. The results are depicted in **Table 2.**

It can be seen that $\mathrm{RF}_{\mathrm{SEV}}$ got the AUC score of 0.7642, which is nearly 2 percent higher than 10-fold cross validation. Besides, it is obviously unreasonable that the Acc and Sp score of many models based on 10-fold cross validation are coming close to 1, while the MCC and Sn score are approaching to 0. This fallacious phenomenon can be attributed to the training set where the rate of positive samples to negative samples might be even in the order of 1 to 12. As we know, if 90% of the samples of the training set belong to the same class, the trained classifier is like to classify all the new samples into one class. In this case, the classifier would be invalid despite the final classification accuracy of 90%. As 10-fold cross validation cannot effectively solve this problem of unbalanced data, the four metrics of Acc, MCC, Sn, Sp consequently lose their reference significance; On the other hand, the performance of the trained classifier becoming neither well reflected nor fully compared with other models. However, SEV not only achieved higher AUC value, but also obtained more objective and reliable MCC, Acc, Sn and Sp values, which is not as unreliable as 10-fold cross validation.

On the other hand, as a new verification method, SE Validation can also be extended to other binary classification problems besides PTMs. Besides, this validation method can also be utilized to validate multi-classification problems as long as SEV is slightly modified according to the actual situation.

B. FEATURE EXTRACTION METHOD HAS GREAT INFLUENCE ON PREDICTION RESULTS

Although the scores of prediction approach is affected by the selection of the validation approaches, we reason that the most important determinant likely comes from the encoding scheme. Therefore, we trained the classifiers above based on two diffiernt encoding methods of EAAC and EBAG+ Profile(EBPR) methods. The results were shown in **Table 3**.

From the table 3, we can see that the type of different feature extraction methods really has great influence on the classification effect of all the classifiers. For instance, the AUC value of the same NN classifier that utilized EAAC encoding reaches 0.7237, while it is only 0.6538 in EBPR-based models. Similar situation was also occurred in SVM, kNN and other classifiers. Not only AUC value, other evaluation metrics have made great differences under different coding methods, which futher verifies our conjecture that the scores of the prediction method is greatly influenced by the feature extraction methods. Besides, we can see that each classifier that utilized EAAC method are plays better performance than the same classifier utilized another feature encoding scheme, which suggests that the EAAC schemes captured the unique information from Kmal-containing peptides and is independent of the type of classifier.

TABLE 3. The results of different classifiers based on EAAC and EBPR feature extraction schemes.

Classifier	Encoding schemes	AUC	Acc	MCC	Sn	Sp
RF	EAAC	**0.7642**	66.50%	**0.2178**	7355%	65.91%
	EBPR	0.7205	**67.60%**	0.1788	63.98%	**67.90%**
ANN	EAAC	0.7237	59.65%	0.1735	73.84%	58.47%
	EBPR	0.6538	55.84%	0.1179	67.16%	54.90%
SVM	EAAC	0.6394	31.35%	0.0897	**88.04%**	26.62%
	EBPR	0.5734	45.80%	0.0652	68.17%	43.93%
kNN	EAAC	0.6626	49.73%	0.1141	73.59%	47.73%
	EBPR	0.6119	58.53%	0.0791	55.92%	58.75%

In addition to verifying the excellent importance of validation and feature extraction method, these two groups of experiments showed that RF_{EAAC} performed the best in terms of AUC and MCC values among these different classifiers. when all other conditions, including data set selection, feature extraction method and verification strategy are the same and only the classifier is different, RF can achieve higher AUC value. As a result, based on SE Validation and EAAC encoding simultaneously, RF achieves the highest AUC value of 0.7642, while the highest AUC values of ANN, SVM and kNN are 0.7237, 0.6394 and 0.6626, respectively. Therefore, when we need a traditional machine learning classifier in Kmal research, either as a separate model to predict the results or as a part to participate in integrated learning, RF is available as the first choice.

C. DEEP LEARNING APPROACH SHOWED SUPERIOR PERFORMANCE

Although several traditional classifiers established above have achieved good classification scores, we have still established two deep learning models, one is CNN1Dwe(K_net), the other is $CNN2D_{we}$.

K_net includes the following 7 layers the first two layers is a convolutional layer followed by a pooling layer, which directly serves as the input layers of the entire network. In these two layers, the 24∗20 feature matrix generated by EAAC is converted into intermediate horizontal features, and then further feature abstraction is carried out in the following layers. The third and fourth layers are also composed of convolutional layer and pooling layer. Similar to the prior two layers, they are also utilized to increase the expression ability of classifier. In detail, we set the convolution kernel size of both the first layer and the third layer to 3, so that the first convolution layer contains 32 convolution kernels, while the number of convolution kernels in the third layer is 64. The fifth layer is a flatten layer, which is the transition from convolution-pooling layer to full connection layer, and is used to uniformize multi-dimensional output. After passing through this layer, the multi-dimensional feature matrix will be converted into a 1∗64 one-dimensional feature vector. The next layer is a fully connected layer, where 64 neuron units were built with the rectified linear unit (ReLU) chosen for its activation function. The last layer is output layer, in which a single unit is activated by the "sigmoid" function, outputting the probability score to judge whether the result is positive or negative.

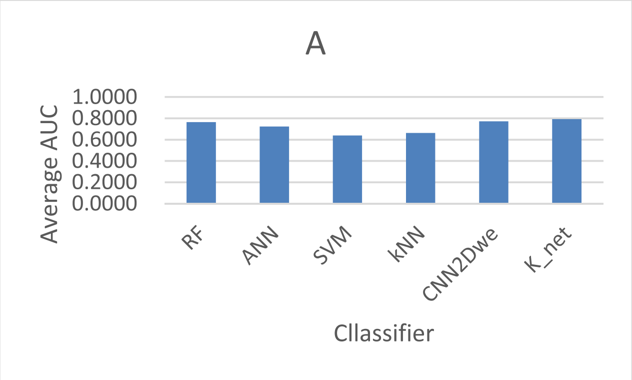

FIGURE 4. The average AUC values of different classifiers, where the $CNN2D_{we}$ and K_net were both based on deep learning approach.

As K_net had been constructed, we practiced another idea: we regard the two-dimensional characteristic matrix as a picture and then constructed the $CNN2D_{we}$ model accordingly. Similar to K_net, this model is also composed of two convolutional layers, two pooling layers, a flatten layer, a full connection layer and an output layer composed of a sigmoid unit. However, the convolutional layer of K_net adopts an intuitive sequence processing strategy, while the convolutional layer of $CNN2D_{we}$ adopts an image recognition idea. Therefore, in $CNN2D_{we}$, the size of convolutional kernel in the first and the third convolutional layers are both 3∗3, which is obviously different from K_net.

After all the models were built and trained, we tested their performance on independent data sets. In order to get a more intuitive analysis result, we draw two histograms to show the scores of two representative evaluation metrics (AUC and MCC values) that can best reflect the model performance were shown in the form of histogram in figure 4 and figure 5.

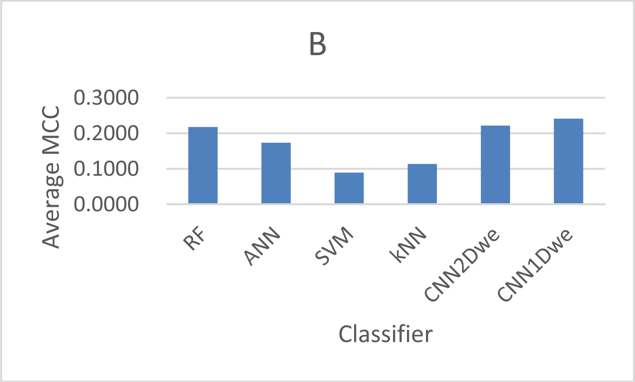

FIGURE 5. The average MCC values of different classifiers.

These two charts show that the deep learning approach–K_net based on CNN1d scored the highest in terms of AUC value and won the first place. The second place is the deep learning model CNN2D$_{\text{we}}$, and the traditional classifiers are all behind. The phenomenon that the scores of that deep learning model is higher than that of all traditional classifiers can also be reflected in the MCC value. This shows that compared with traditional machine learning methods, deep learning methods can achieve better results. The strong point of deep learning lies in its perfect fitting ability, which can approximate any complex function. This may explain why DL models demonstrated superior performance. As a result, K_net obtained the best PTM prediction performance with the AUC value of 0.7927 and MCC value of 0.2412.

IV. CONCLUSION

Although many shallow machine learning methods are still applied to various classification problems, deep learning methods continuously refresh the classification accuracy of various problems with its excellent performance. In this work, we construct two CNN models to predict Kma sites. Deep neural network not only has perfect fitting ability and can approximate any complex function, but also contains many hidden layers with lots of hidden nodes, which makes the expression ability of neural network very stronger.

At the same time, in order to solve the problem caused by imbalanced training samples in the classification problem, we propose a new validation method of SEV. SEV not only has the function of making full use of training samples similar to the 10-fold cross-validation, but also can objectively show the final ACC, Sn,Sp and other indicators, instead of making them lose reference significance as the training samples are not balanced. Besides the binary classification problems, SEV method can also be utilized to validate multi-classification problems as long as SEV is slightly modified according to the actual situation.

DATA AVAILABILITY

To data used to support the findings of this study are available from the corresponding author upon request.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

[1] G. Walsh and R. Jefferis, "Post-translational modifications in the context of therapeutic proteins," *Nature Biotechnol.*, vol. 24, no. 10, pp. 1241–1252, Oct. 2006.

[2] A. Zinellu, P. Paliogiannis, C. Carru, and A. A. Mangoni, "Homoarginine and all-cause mortality: A systematic review and meta-analysis," *Eur. J. Clin. Invest.*, vol. 48, no. 8, Aug. 2018, Art. no. e12960.

[3] C. Peng, Z. Lu, Z. Xie, Z. Cheng, Y. Chen, M. Tan, H. Luo, Y. Zhang, W. He, K. Yang, B. M. M. Zwaans, D. Tishkoff, L. Ho, D. Lombard, T.-C. He, J. Dai, E. Verdin, Y. Ye, and Y. Zhao, "The first identification of lysine malonylation substrates and its regulatory enzyme," *Mol. Cell Proteomics*, vol. 10, no. 12, Dec. 2011, Art. no. M111.012658.

[4] M. Lo Monte, C. Manelfi, M. Gemei, D. Corda, and A. R. Beccari, "ADPredict: ADP-ribosylation site prediction based on physicochemical and structural descriptors," *Bioinformatics*, vol. 34, no. 15, pp. 2566–2574, Aug. 2018.

[5] C. Ye, Y. Long, G. Ji, Q. Q. Li, and X. Wu, "APAtrap: Identification and quantification of alternative polyadenylation sites from RNA-seq data," *Bioinformatics*, vol. 34, no. 11, pp. 1841–1849, Jun. 2018.

[6] H. Lodish, A. Berk, C. A. Kaiser, and M. Krieger, *Molecular Cell Biology*. 2004.

[7] H. A. Doyle and M. J. Mamula, "Post-translational protein modifications in antigen recognition and autoimmunity," *Trends Immunol.*, vol. 22, no. 8, pp. 443–449, Aug. 2001.

[8] G. A. Khoury, R. C. Baliban, and C. A. Floudas, "Proteome-wide post-translational modification statistics: Frequency analysis and curation of the swiss-prot database," *Sci. Rep.*, vol. 1, no. 1, pp. 90–98, Dec. 2011.

[9] L. Z. Topol, B. Bardot, Q. Zhang, J. Resau, E. Huillard, M. Marx, G. Calothy, and D. G. Blair, "Biosynthesis, post-translation modification, and functional characterization of Drm/Gremlin," *J. Biol. Chem.*, vol. 275, no. 12, pp. 8785–8793, Mar. 2000.

[10] J.-Y. Shi, J.-X. Li, and H.-M. Lu, "Predicting existing targets for new drugs base on strategies for missing interactions," *BMC Bioinf.*, vol. 17, no. S8, p. 282, Aug. 2016.

[11] S.-P. Shi, J.-D. Qiu, X.-Y. Sun, S.-B. Suo, S.-Y. Huang, and R.-P. Liang, "PLMLA: Prediction of lysine methylation and lysine acetylation by combining multiple features," *Mol. BioSyst.*, vol. 8, no. 5, pp. 1520–1527, 2012.

[12] G. C. Shukla, J. Singh, and S. Barik, "MicroRNAs: Processing, maturation, target recognition and regulatory functions," *Mol. Cellular Pharmacol.*, vol. 3, no. 3, pp. 83–92, 2011.

[13] W. Si, J. Shen, C. Du, D. Chen, X. Gu, C. Li, M. Yao, J. Pan, J. Cheng, D. Jiang, L. Xu, C. Bao, P. Fu, and W. Fan, "A miR-20a/MAPK1/c-Myc regulatory feedback loop regulates breast carcinogenesis and chemoresistance," *Cell Death Differ*, vol. 25, no. 2, pp. 406–420, Feb. 2018.

[14] S. Sidney, C. P. Quesenberry, M. G. Jaffe, M. Sorel, M. N. Nguyen-Huynh, L. H. Kushi, A. S. Go, and J. S. Rana, "Recent trends in cardiovascular mortality in the united states and public health goals," *JAMA Cardiol.*, vol. 1, no. 5, pp. 594–599, Aug. 2016.

[15] E.-J. Jeon, K.-Y. Lee, N.-S. Choi, M.-H. Lee, H.-N. Kim, Y.-H. Jin, H.-M. Ryoo, J.-Y. Choi, M. Yoshida, N. Nishino, B.-C. Oh, K.-S. Lee, Y. H. Lee, and S.-C. Bae, "Bone morphogenetic protein–2 stimulates Runx2 acetylation," *J. Biol. Chem.*, vol. 281, no. 24, pp. 16502–16511, Jun. 2006.

[16] J. G. Voet and D. Voet, "A new project for a new year: A biochemistry and molecular biology digital library," *Biochem. Mol. Biol. Educ.*, vol. 31, no. 1, p. 1, Jan. 2003.

[17] J. Eichler and J. Maupin-Furlow, "Post-translation modification in Archaea: Lessons fromHaloferax volcaniiand other haloarchaea," *Fems Microbiol. Rev.*, vol. 37, no. 4, pp. 583–606, Jul. 2013.

[18] J. Gough and A. K. Dunker, "Sequences and topology: Disorder, modularity, and post/pre translation modification," *Current Opinion Struct. Biol.*, vol. 23, no. 3, pp. 417–419, Jun. 2013.

[19] M. S. Rahman, U. Aktar, M. R. Jani, and S. Shatabda, "iPromoter-FSEn: Identification of bacterial σ^{70} promoter sequences using feature subspace based ensemble classifier," *Genomics*, vol. 111, no. 5, pp. 1160–1166, 2019.

[20] N. Yanaihara, N. Caplen, E. Bowman, M. Seike, K. Kumamoto, M. Yi, R. M. Stephens, A. Okamoto, J. Yokota, T. Tanaka, G. A. Calin, C.-G. Liu, C. M. Croce, and C. C. Harris, ''Unique microRNA molecular profiles in lung cancer diagnosis and prognosis,'' *Cancer Cell*, vol. 9, no. 3, pp. 189–198, Mar. 2006.

[21] D.-S. Huang and H.-J. Yu, ''Normalized feature vectors: A novel alignment–free sequence comparison method based on the numbers of adjacent amino acids,'' *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 2, pp. 457–467, Mar. 2013.

[22] L.-N. Wang, S.-P. Shi, H.-D. Xu, P.-P. Wen, and J.-D. Qiu, ''Computational prediction of species-specific malonylation sites via enhanced characteristic strategy,'' *Bioinformatics*, vol. 33, pp. 1457–1463, Nov. 2016.

[23] M. Solayman, M. A. Saleh, S. Paul, M. I. Khalil, and S. H. Gan, ''In silico analysis of nonsynonymous single nucleotide polymorphisms of the human adiponectin receptor 2 (ADIPOR2) gene,'' *Comput. Biol. Chem.*, vol. 68, pp. 175–185, Jun. 2017.

[24] M. B. Johansen, L. Kiemer, and S. Brunak, ''Analysis and prediction of mammalian protein glycation,'' *Glycobiology*, vol. 16, no. 9, pp. 844–853, Sep. 2006.

[25] Z. Ju, J. Sun, Y. Li, and L. Wang, ''Predicting lysine glycation sites using bi-profile Bayes feature extraction,'' *Comput. Biol. Chem.*, vol. 71, pp. 98–103, Dec. 2017.

[26] W. Bao, D. Wang, and Y. Chen, ''Classification of protein structure classes on flexible neutral tree,'' *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 5, pp. 1122–1133, Sep. 2017.

[27] W. Bao, B. Yang, D.-S. Huang, D. Wang, Q. Liu, Y.-H. Chen, and R. Bao, ''IMKPse: Identification of protein malonylation sites by the key features into general PseAAC,'' *IEEE Access*, vol. 7, pp. 54073–54083, 2019.

[28] D. Wang, S. Zeng, C. Xu, W. Qiu, Y. Liang, T. Joshi, and D. Xu, ''MusiteDeep: A deep-learning framework for general and kinase-specific phosphorylation site prediction,'' *Bioinformatics*, vol. 33, no. 24, pp. 3909–3916, Dec. 2017.

[29] Y. Xie, X. Luo, Y. Li, L. Chen, W. Ma, J. Huang, J. Cui, Y. Zhao, Y. Xue, Z. Zuo, and J. Ren, ''DeepNitro: Prediction of protein nitration and nitrosylation sites by deep learning,'' *Genomics, Proteomics Bioinf.*, vol. 16, no. 4, pp. 294–306, Aug. 2018.

[30] J. H. Liebregts, M. Timmermans, M. J. De Koning, S. J. Bergé, and T. J. Maal, ''Three–dimensional facial simulation in bilateral sagittal split osteotomy: A validation study of 100 patients,'' *J. Oral Maxillofacial Surg.*, vol. 73, no. 5, pp. 961–970, May 2015.

[31] S. Dudoit, M. Van Der Laan, S. Keles, A. Molinaro, S. Sinisi, and S. L. Teng, ''Loss-based estimation with cross-validation: Applications to microarray data analysis and motif finding,'' *ACM SIGKDD Explor. Newslett.*, vol. 5, pp. 56–68, Dec. 2003.