BERT_PLPS: A BERT-based Model for Predicting Lysine Phosphoglycerylation Sites

Songning Lai[1], Yankun Cao[2], Pengwei Wang[1*], Lan Ye[3*], Zhi Liu[1*]

[1]the School of Information Science and Engineering, Shandong University, Qingdao, Shandong, 266237, China.
[2]the School of Software, Shandong University, Jinan, Shandong, 250101, China.
[3]the Cancer center, the Second Hospital of Shandong University, Jinan, Shandong, 250358, China.

*Corresponding author(s). E-mail(s): wangpw@sdu.edu.cn; sdeyyelan@email.sdu.edu.cn; liuzhi@sdu.edu.cn;
Contributing authors: 202000120172@mail.sdu.edu.cn; kunkun@sdu.edu.cn;

Abstract

As one of the most important post-translational modification processes, lysine phosphoglycerylation modifications affect many important biosynthetic processes in the human body. However, traditional experimental methods for the recognization of lysine phosphoglycerylation sites are not only expensive but also time-consuming. Computational techniques may provide an economical and efficient way to predict lysine phosphoglycerylation sites. Therefore, it is extremely necessary and meaningful to study and establish prediction models with high accuracy. In the present study, we propose a BERT-based model, BERT_PLPS, which could predict accurately lysine phosphoglycerylation sites. This model extracts amino acid sequence features with three algorithms: CKSAAP, AAC, and BE. Sample equalization is performed using the ADASYN and KNN algorithms. The data are dimensionalized by the ISOMap algorithm, and the features are encoded into feature sequences by an encoder as the input to a BERT-based prediction model. To learn better the intrinsic biological language of lysine, we replaced the original static mask with a dynamic random mask. Compared to other machine learning or deep learning-based models, BERT_PLPS exhibits up to 99.53% accuracy and outperforms the most advanced model (PLP_FS) with an increase of approximately 0.35% on ACC and approximately 0.93% on MCC.

Keywords: BERT, post-translational modification, phosphoglycerylation, data dimensionality reduction

1 Introduction

Post-translational modifications (PTM)[1], including phosphoglycerylation, ubiquitination, crotonylation and so on[2], play an important role in protein biosynthesis. The process involves adding functional groups to proteins on multiple amino acid residues during protein translation. It makes the protein gain a more unique steric structure, leading to it getting more elaborate and complex regulatory functions[3, 4]. This process is closely associated with physiological processes such as gene expression and biological signaling in organisms.

Organisms use 20 amino acids, among which lysine is most often post-translationally modified because of its unique structure[5]. Lysine is associated with a variety of human diseases, such as heart disease and rheumatoid arthritis[6, 7]. Phosphoglycerylation is a PTM of lysine, which affects numerous vital processes in organisms, including glucose metabolism[8–10]. Phoshoglycerylation of lysine has been reported to be associated with cardiovascular diseases in humans[11, 12]. Further study of this PTM will facilitate understanding deeply the mechanism and regulatory role of lysine phosphoglycerylation, which would help to develop new corresponding drugs and therapeutic regimens.

Nowadays an increasing number of methods are developed to identify phosphoglycerylation sites. These methods can be roughly divided into two types: experimental and computational methods. The experimental methods, represented by mass spectrometry [13–18], are very expensive and inefficient. Instead, computational methods are cheaper and more effective.

In past decades, many effective calculation methods have been proposed to identify PTM sites. The Phogly PseAAC model[19] was the first tool to be carried out to predict the phosphoglycerylation sites of amino acids. Xu et al predicted phosphoglycerylation sites in pseudo amino acid feature data with the K nearest neighbor algorithm (KNN). The CKSAAP_Phoglysite model[11] utilized the CKSAAP protein feature extraction technique for amino acid feature construction, and the sample feature matrix was used to train a support vector machine (SVM) for the task of predicting amino acid phosphoglycerylation sites. The PLP_FS model[20] used three protein feature extraction techniques, including the composition of K-spaced residue pairs (CKSAAP), amino acid composition (AAC), and binary encoding (BE), to construct protein feature vectors. It uses the machine learning algorithm SVM for the prediction of the feature vectors. GPS-Palm[21] predicted various PTMs, such as succinylation, using a deep learning framework. The LSTMCNNsucc[22] model combined bidirectional long short-term memory (LSTM) and deep learning algorithms based on convolutional neural networks (CNN). The DeepSuccinySite[22] model used word embedding to award amino acid sequences into feature vectors and used CNN-based deep learning models for feature extraction and site prediction.

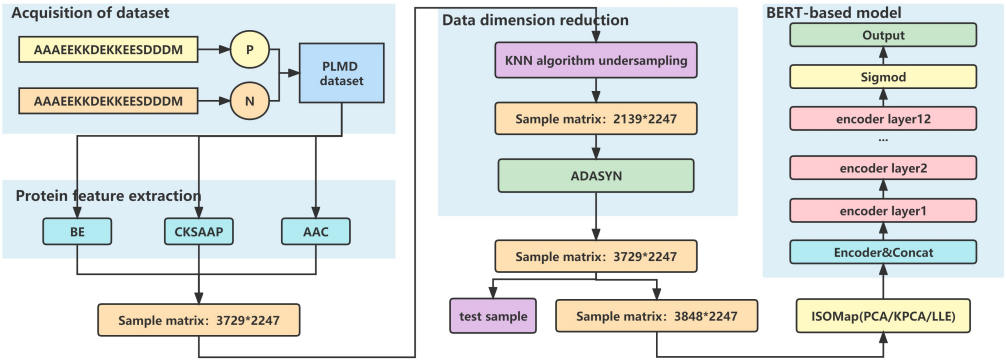

Fig. 1: The complete flow chart of BERT_PLPS model

In recent years there is an era of explosive development of deep learning. In the fields of computer vision and natural language processing, Transformer[23] became compelling. The Transformer-derived model soon became the SOTA in many tasks in both fields[24–26], showing its amazing ability. Bidirectional encoder representations for transformers (BERT)[27], which works well in text classification tasks and text translation tasks, has overtaken the Transformer encoder and won many natural language processing competitions. The transformer-based BERT model has taken off in the field of natural language processing, which made us wonder whether its ability to process human language can be transferred to process biological languages, such as protein sequence information.

In the present study, the innovative contributions of our work can be summarized as follows.

1. We propose a novel and efficient computational method, BERT_PLPS, to predict lysine phosphoglycerylation sites.

2. The dynamic masking strategy is used to replace the original static masking strategy, which effectively improves the anti-noise ability of the model.

3. Several groups of experiments are designed to compare and explore the adaptability of various data dimensionality reduction algorithms for amino acid sequence task learning.

4. Experiments on the public benchmark data sets verify the feasibility of the model. Moreover, our current approach outperforms the current state-of-the-art methods. The complete flowchart of the BERT_PLPS model is shown in Figure 1.

2 Methods

2.1 Preparation of the dataset

To make a fair and balanced comparison with existing models, the dataset employed in this study was used in the previous study[20], in which the researchers developed a machine learning model PLP_FS for phosphoglycerylation site prediction. The

dataset made use of the protein synthesis dataset in the Protein Lysine Modifications Database (PLMD)[28]. It contains 103 annotated lysine phosphoglycerylation sites (40% redundant proteins have been removed using the CD-HIT tool[29]) and 3626 lysine nonphosphoglycerylation sites. The segmentation of protein sequences was completed using a sliding window technique with a window size of 21 to form peptides. Finally, the baseline dataset contained 103 positive samples and 3626 negative samples.

2.2 Protein feature extraction

To extract the protein sequence features completely and effectively, we use three methods, CKSAAP, BE and AAC, for protein feature extraction of the lysine dataset samples.

CKSAAP:

This method[30] is proven to be an effective and fast method for protein feature extraction. It is invented by Chen et al. and based on Tung et al[31]. A window of size 11-31 was considered as an optimal range. We used 21 of amino acids, including a blank dummy amino acid, and intercepted a peptide fragment of length 21. The equations of CKSAAP[32] are as follows:

$$\left(\frac{N_{A*A}}{N_{total}}, \frac{N_{A*B}}{N_{total}}, \frac{N_{A*D}}{N_{total}}, \ldots, \frac{N_{X*X}}{N_{total}}\right)_{441} \tag{1}$$

$$N_{total} = L - k - 1 \tag{2}$$

where N_{total} is the total number of 1 spaced amino acid pairs in each fragment, and L is the length corresponding to each peptide. The five values of k are 0, 1, 2, 3, and 4, respectively. The five values of k are 0, 1, 2, 3, and 4, respectively, and the five values of N_{total} dimension are 20,19,18,17, and 17, respectively[33, 34]. With this algorithm, a 2205-dimensional feature vector is obtained for each protein fragment.

AAC:

This is a more parsimonious way to represent protein characteristics. The AAC[35, 36] protein feature extraction algorithm is used to represent the probability of the occurrence of each amino acid in a protein sequence by means of descriptors. Twenty-one amino acids (where X indicates a virtual amino acid) are used to generate protein fragment samples. A 21-dimensional protein feature vector is generated based on the number of occurrences of each amino acid in the protein sequence. The specific equations are as follows:

$$V_X = [P_1, P_2, P_3, ..., P_{21}] \tag{3}$$

$$P_i = \frac{AA_i}{L} \tag{4}$$

where P_i denotes the probability of occurrence of the ith amino acid, AA_i is the number of occurrences of the ith amino acid, and L denotes the length of the amino acid sequence.

BE:

This is a straightforward and effective technique for protein sequence feature extraction[37–40]. In this algorithm, we present a virtual amino acid X and represent each amino acid as a 21-dimensional feature vector. Each amino acid corresponds to a specific sequence. For example, the virtual amino acid is represented by "000000000000000000000." We obtain an $L * 21 - dimensional$ feature matrix for this peptide segment of length L. Regarding the previously described method, we use a window of size 21 for peptide fragment interception. With this algorithm, we will obtain a $21 * 21$ dimensional feature matrix to encode each protein fragment.

2.3 Balance the dataset

The present study used a baseline dataset with a ratio of positive to negative samples of approximately 1:36. The data collectively show an extreme imbalance. Various studies[41, 42] have shown that it is difficult to construct models with highly imbalanced datasets. The KNN algorithm is firstly used to undersample the negative samples. The positive samples are then over-sampled using the ADASYN algorithm to balance the dataset.

KNN algorithm undersample:

The KNN algorithm[43] is a parameter-free and fast algorithm. The Euclidean distance between all sample points was calculated. Negative samples with k=36 near neighbors for each positive sample are excluded. It reduces the number of negative samples to alleviate the data imbalance problem. Meanwhile, it can effectively solve the problem of negative samples possibly containing positive data (the public dataset cannot guarantee the purity of the negative data, so these negative samples may also be phosphoglycerylation sites). The date was iterated by decreasing the value of k until 2036 negative samples remain.

ADASYN algorithm oversampling:

The ADASYN algorithm[44] is an improved algorithm based on the conventional oversampling algorithm Synthetic Minority Oversampling Technique (SMOTE)[45]. This algorithm assigns different weights to different samples, which generates the corresponding number of different samples. The exact flow of the algorithm is as follows.

The equation for calculating the number of positive samples to be synthesized is:

$$G = (m_1 - m_s) \times \beta \tag{5}$$

where m_1 denotes the number of negative samples (2036), m_s denotes the number of positive samples (103), and $\beta = 1$ is the selected parameter.

The ratio of negative samples in the K-nearest neighbors (Euclidean distance) of each positive sample was calculated separately. The equation is:

$$r_i = \frac{\Delta_i}{K} \tag{6}$$

where r_i denotes the proportion and Δi denotes the number of negative samples in the K-nearest neighbors of the ith positive sample.

r_i is normalized, and the number of new samples to be generated for each positive sample is calculated based on the weight of each positive sample (the normalized r_i is

multiplied by G as the weight of each sample). The SMOTE algorithm is finally used to generate new samples. The equation is:

$$s_i = x_i + (x_{zi} - x_i) \times \varepsilon \tag{7}$$

where s_i denotes the new positive sample synthesized, x_i denotes the ith positive sample, x_{zi} denotes a positive sample randomly selected from the k-nearest neighbor of x_i, and ε is a random number in [0,1]. This algorithm has a greater preference for increasing positive samples in regions with a low density of positive samples compared to the original SMOTE algorithm. This facilitates the enhancement of positive sample points at the decision boundary. The ADASYN algorithm not only effectively reduces the challenges posed by data imbalance but also adaptively pushes the decision boundaries of the subsequent model to more challenging samples.

We increased the number of positive samples to 2033 by the ADASYN algorithm in the current experiment. A balanced dataset with a ratio between positive and negative samples tending to 1:1 was constructed. All samples were randomly divided into a training dataset (2848) and a test dataset (1221) at a ratio of 7:3.

2.4 Data Dimensionality Reduction

After extracting protein features from the samples and balancing the dataset, a sample feature matrix with dimensions of 2848*2247 is generated. Each sample contains 2247 dimensional features. Using high-dimensional data to train the prediction model will cause a "dimensional disaster" (exponential growth in computation as the number of dimensions increases). The noise in the features will also be trained, and the useful feature information will be buried in the high-dimensional features. Therefore, finding a suitable data dimension reduction algorithm is necessary[46]. Data dimensionality reduction can help solve the "dimension disaster" and enable subsequent deep learning models to understand the data better.

At present, there are many data dimension reduction algorithms[47]. It is unclear which kind of data dimension reduction algorithm is more suitable for the protein sequence characteristics of the sample points. The data dimensionality reduction algorithm is extremely sensitive to the distribution of the data. It is conducive to exploring the appropriate algorithm for data dimensionality reduction, for our further understanding of protein feature sequence information. Data dimensionality reduction algorithms include include PCA[48], KPCA[49], ISOMap[50], LLE[51], etc. PCA is a typical linear dimensionality reduction algorithm. KPCA is a typical nonlinear dimensionality reduction algorithm based on the kernel function. ISOMap and LLE are nonlinear dimensionality reduction algorithms based on eigenvalues. These four data dimensionality reduction algorithms are very sensitive to the distribution of the data points[52].

The main idea of PCA is to project the data points to the area where the variance of the projection is sufficiently large. The purpose is to retain the original information of the data as much as possible[53] so that the important information can be extracted from the data points. In essence, the PCA algorithm is a problem of eigen-decomposition of a positive semi-definite matrix and singular value decomposition of

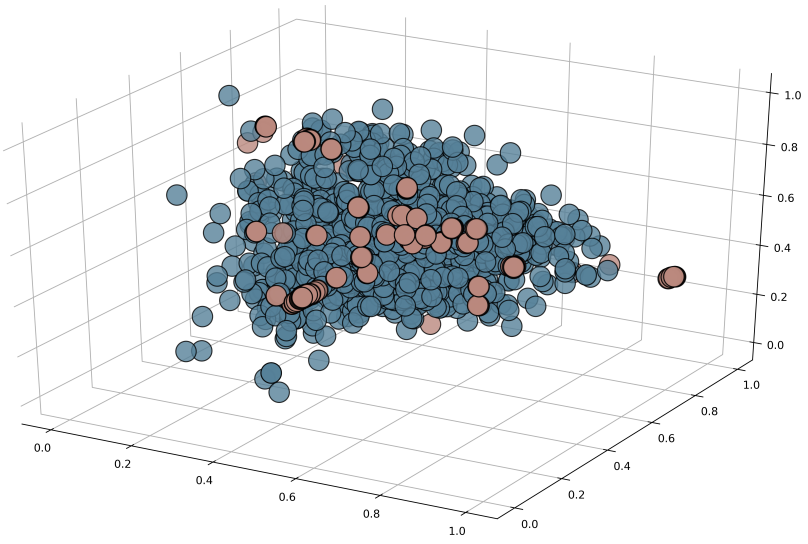

Fig. 2: Sample point visualization

a matrix. However, this algorithm requires the data to present a Gaussian distribution. As a typical linear data dimension reduction algorithm, the effectiveness of the PCA algorithm is worth exploring through experiments.

The main idea of the KPCA algorithm is to find a kernel function that can map data from high-dimensional space to low-dimensional space as a mapping function so that the data can be linearly divisible[54]. The kernel is a nonlinear function. It can give the algorithm the ability to analyze nonlinear data. After the introduction of the nonlinear kernel function, although the data are still linear in the feature space, it can produce nonlinear interpretations in the original space.

The ISOMAP algorithm is used to reduce the data dimension from the perspective of reconstruction. By analyzing the high-dimensional popular structure of the data itself, it can obtain the low-dimensional embedding corresponding to the high-dimensional popular, so that the adjacent structure of the data point on the high-dimensional popular can be more completely reproduced in the low-dimensional embedding[55]. Mathematically, the algorithm is a combination of the shortest path problem and the eigenvalue problem. It requires that the data points be well sampled to reflect the true structure of its prevalence.

The LLE algorithm is similar to the ISOMap algorithm. However, the main difference is that the LLE algorithm reconstructs data based on logarithmic data points of local information and tries to preserve the linear relationship between samples in the field[56].

We chose the ISOMap algorithm for data dimension reduction. The dimensional features was reduce from 2247 to 50. These features are used in the training of models

for predicting the glycerylation sites of amino acid phosphate. The ISOMap algorithm was used to reduce the dimensionality of the sample points to three-dimensional space. The visualization is shown in Figure 2. The eigenmatrix obtained by separation, and the dimensionality reduction algorithm gives relatively clear decision boundaries for the negative and positive samples. We also compared the other dimensionality reduction algorithms, to clarify the relationship between the different algorithms and protein characteristics.

For the other dimensionality reduction algorithms, we also conducted comparative experiments to explore the relationship between different dimensionality reduction algorithms and protein characteristics. The comparative experimental results and corresponding discussion of the data dimensionality reduction algorithm are given in the Results and discussion section.

2.5 BERT_PLPS model

Convolutional neural networks (CNNs) have been used to predict post-translational modification sites in previous attempts to solve this task through deep learning. Although the results achieved by CNNs is good, the accuracy and performance need to be improved. Transformer[23] has emerged in the field of natural language processing in recent years. It adopts the self-attention mechanism. It completely abandons the communication mode in the previous deep learning network, and only propagates in the vertical direction. Moreover, it can constantly superimpose the self-attention layer so that it can be calculated in parallel. BERT[27] uses the encoder of a multi-layer transformer. It is particularly brilliant in the field of natural language processing. Our lysine phosphorylation site prediction model is based on BERT.

The feature of each sample point is encoded into a feature sequence that can be effectively learned by the deep learning model. The encoder concatenates all feature data into a complete feature sequence.

For this sequence, a static mask operation is required to remove the influence of various paddings in the training process. We replaced the original static mask with a dynamic mask[57]. 60% of the dynamic masks continue to use the original static mask, 20% of the parts without masks are randomly covered, and 20% do not use masks. With a dynamic mask, the model generates new mask patterns each time a sequence is fed to the model. During the process of continuous input of large amounts of data, the model will adapt to different mask strategies to learn different amino acid characteristics. This method can effectively improve the generalization ability of the model.

The feature sequence is decomposed into elements with location information in the model. Each element is embedded by word and position. Word embedding converts the original input feature sequence into a sequence of length 449. The equations of position embedding are:

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{8}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{9}$$

where pos represents the absolute position of each element in the feature sequence, and d_{model} is 449, which represents the maximum length of the input sequence and the dimension of the feature vector of the whole sample.

Embedding elements allows the model to effectively capture the relationship between sample features and incorporate this relationship into the data for model training. The architecture of the whole model is a multi-layer bidirectional conversion encoder, which uses the attention mechanism of the encoder layer to join the sample features and processes all of the features in the sample in parallel. Each encoding layer includes a multi-head attention layer and a feed-forward neural network layer, which also introduces a residual structure to avoid the vanishing gradient of back-propagation when the model is deep. The model uses layer normalization[58] after the self-attention layer, which is a normalization approach. In each encoding layer, the multihead attention layer is the core, and the specific equations are as follows:

$$MultiHead(Q, K, V) = C(head_1, ..., head_h)W^O \tag{10}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{11}$$

$$Attention(Q, K, V) = softmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right)V \tag{12}$$

Q,K,V are obtained from the input sequence by linear transformation to represent the information in the space. d_k denotes the dimension of k; W_i^Q, W_i^K and W_i^V are the weight matrices of Q, K and V, respectively.

In addition, each multi-head attention layer is connected with a feed-forward neural network layer. The equation is:

$$FFN(x) = max\left(0, xW_1 + b_1\right)W_2 + b_2 \tag{13}$$

where x is the output of the multihead attention layer, and W_1 and W_2 are the weight parameters to be learned.

After multiple identical transformer-based encoder transfers are completed, the mapping from embedding dim to dimension 1 is performed in the last layer. The Sigmod function is used for activation, thus the task of predicting the lysine phosphate glycerylation site was completed. Figure 3 shows the schematic diagram of the entire transformer-based deep learning model, which can provide a clearer understanding of its specific structure.

The dotted line in the model diagram is the residual connection, which is equivalent to taking the input to the output position untouched and adding the output. In the residual part can be expressed as follows.

$$x_{l+1} = x_l + F\left(x_l, W_l\right) \tag{14}$$

where x_{l+1} represents the output of the residual block, x_l represents the input of the residual block, and $F(x_l, W_l)$ represents the result obtained x_l by W_l after entering the residual block. The expression of this residual block can be obtained as follows.

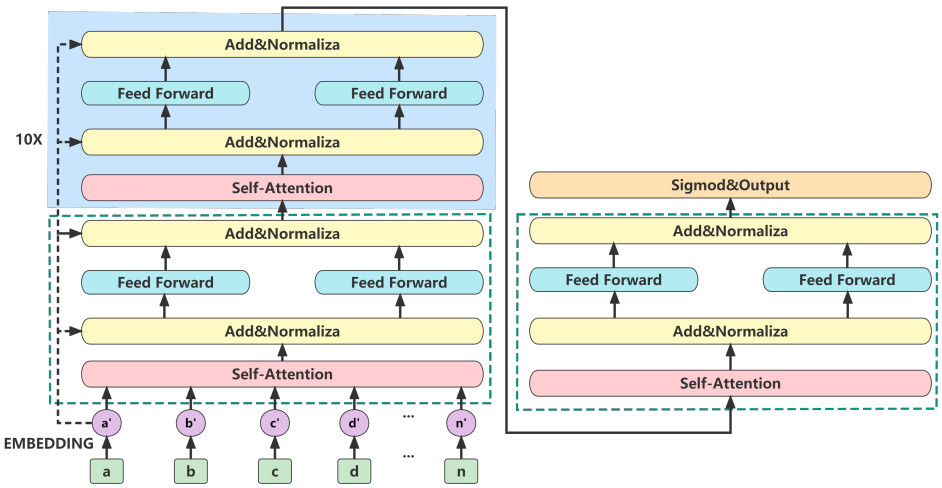

Fig. 3: Schematic of the entire Transformer-based deep learning model

$$
\begin{aligned}
x_1 &= x_0 + F\left(x_0, W_0\right) \\
x_2 &= x_0 + F\left(x_0, W_0\right) + F\left(x_1, W_1\right) \\
&\cdots \\
x_L &= x_l + \sum_{i=1}^{L-1} F\left(x_i, W_i\right)
\end{aligned}
\tag{15}
$$

This expression also vividly illustrates that the residual structure can obtain features while retaining the original information during the training process. Let the Loss function be *Loss*, and according to the chain formula in the back propagation algorithm, the following can be obtained as follows.

$$
\begin{aligned}
\frac{\partial Loss}{\partial x_l} &= \frac{\partial Loss}{\partial x_L} * \frac{\partial x_L}{\partial x_l} \\
&= \frac{\partial Loss}{\partial x_L} * \frac{\partial \sum_{i=1}^{L-1} F\left(x_i, W_i\right)}{\partial x_l} + \frac{\partial Loss}{\partial x_L}
\end{aligned}
\tag{16}
$$

According to the gradient calculation result of the loss function, the original signal can be directly transmitted to the bottom layer without the intermediate weight matrix transformation during the back propagation of the network, which alleviates the problem of gradient disappearance to a certain extent. It is this structure that makes the protein sequence features more smoothly in the forward propagation and back propagation.

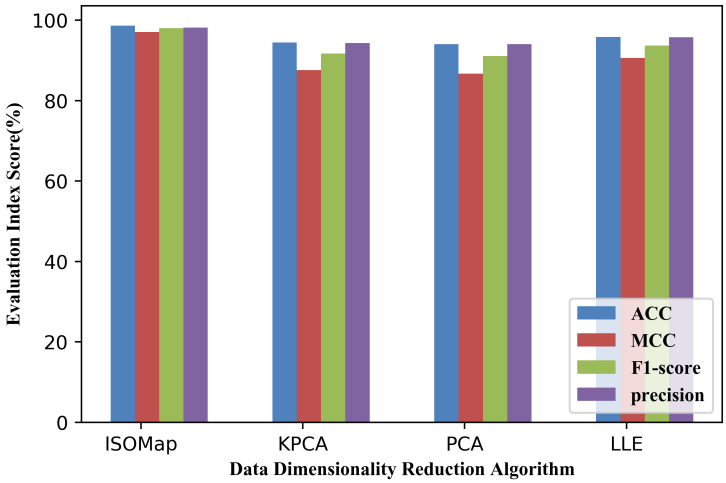

Fig. 4: Histogram comparison of data dimensionality reduction algorithms

2.6 Model evaluation parameters

We objectively evaluated the performance of the model using six different model evaluation parameters: The equations of accuracy (ACC), Matthews correlation coefficient (MCC), precision (PRE), sensitivity (Sn), specificity (Sp) and F1-score are:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{17}$$

$$A = (TP + FP)(TP + FN)(TN + FP)(TN + FN) \tag{18}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{A}} \tag{19}$$

$$PRE = \frac{TP}{(TP + FP)} \tag{20}$$

$$recall(Sn) = \frac{TP}{(TP + FN)} \tag{21}$$

$$Sp = \frac{TN}{(TN + FP)} \tag{22}$$

$$F1 - score = \frac{2 * PRE * recall}{(PRE + recall)} \tag{23}$$

where TP denotes the number of correctly predicted phosphoglyceration sites, FP denotes the number of predicted phosphoglyceration sites that are nonphosphoglyceration sites, TN denotes the number of correctly predicted nonphosphoglyceration sites, and FN denotes the number of predicted nonphosphoglyceration sites that are phosphoglyceration sites.

3 Results and discussion

3.1 Selection of data dimensionality reduction algorithms and a discussion of loss curves

To explore the relationship between the data dimensionality reduction algorithm and the protein features, we changed the data dimensionality reduction algorithm to conduct comparative experiments. Considering the fairness and complexity of multiple experiments, we used PCA, KPCA, LLE and ISOmap algorithms to reduce the dimension of protein features to 5 dimensions. The encoder was entered to encode the feature sequence into the embedding layer of the prediction model. A 31-dimensional word sequence was obtained. All models are set with the same parameters: the batch size is set to 32, the learning rate to 1e-5, epoch to 50, etc. The results of the comparative experiments are shown in Table 1.

The histogram of the data is shown in Figure 4. It can be intuitively observed that all of the metrics of the model using the ISOMap dimensionality reduction algorithm are higher than other algorithms. Linear data dimensionality reduction (PCA) is the worst in terms of linear and nonlinear data. In the mainstream methods of the nonlinear dimensionality reduction algorithm, the KPCA algorithm based on the kernel function is inferior to LLE and ISOMap, which are based on the feature value. The kernel function can make the algorithm have non-linear analysis ability, however, the different non-linear characteristics need specific kernel functions. It is lack of theoretical guidance about how to find an applicable kernel function. We tried multiple kernels, but it is difficult to achieve an excellent result. So we think that the existing kernel functions are not good enough to analyze non-linear protein sequence features.

ISOMap is slightly better compared to LLE. Both algorithms assume that the data distribution is conformed to a popularity distribution. ISOMap is better than LLE in the matter of maintaining a global information advantage. The LLE algorithm shifts the attention to the situation inside each local region. Although this can give the LLE algorithm advantages while analyzing some linear data, it cannot grasp the global characteristics well.

Table 1: Comparing experimental results

Algorithm Name	ACC	MCC	F1-score	PRE
ISOMap	**98.69%**	**97.12%**	**98.04%**	**98.17%**
KPCA	94.49%	87.61%	97.71%	94.32%
PCA	94.07%	86.74%	91.13%	94.07%
LLE	95.80%	90.68%	93.74%	95.81%

Figure 5 shows the loss curves of the four dimensionality reduction algorithms and BERT_PLPS with dynamic masks or static masks. It can be observed that the features obtained by ISOMap following dimensionality reduction are better. Compared to the PCA, KPCA and LLE algorithms, ISOMap's loss decreases faster and converges at a lower loss value. This determine that an appropriate data dimensionality reduction

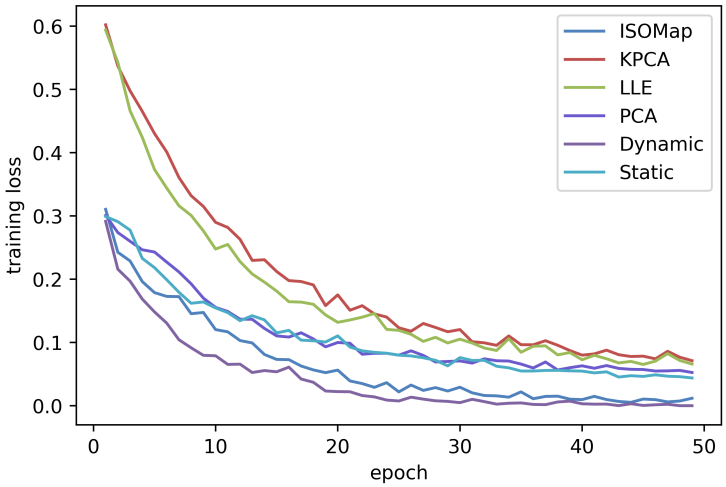

Fig. 5: The loss curves of dimensionality reduction algorithms and BERT-PLPS

algorithm is beneficial for training a deep learning model and the convergence of loss. The loss curve of the dynamic mask fluctuates more violently than the static mask model, but it still maintains a faster convergence rate resulting in a better result. We infer that this dynamic mask mechanism enables the model to obtain the ability to mitigate noise during the training process, which makes the model more robust to noise and generalization. Therefore, the model can more accurately predict phosphoglyceration sites in protein sequences with many interference factors.

3.2 Comparison and evaluation with other existing models

In the past few years, several prediction models for protein phosphoglyceration sites have been developed. We compared the model evaluation results of BERT_PLPS on the same dataset to eight other existing models, including PLP_FS[20], RAM_PGK[59], IDPGK[60], PhoglyPred [61], Bigram-PGK[8], Phogly PseAAC[19], EvolStruct-phogly[62], and predPhogly Site[63]. We also compared the dynamic mask to the static mask using the same dataset. The results is shown in Table 2.

To observe intuitively the prediction ability of different models with the benchmark datasets, the histogram plots of ACC, MCC, Sn and Sp was depicte in Figure 6.

In our results, the ACC, MCC, SN and SP of BERT_PLPS with a dynamic mask are 99.53%, 99.07%, 99.21% and 99.85%, respectively. It is 0.37%, 0.74%, 0.29%, and 0.44% higher than BERT_PLPS using a static mask. During the process of continuous input of a large amount of data, the model will learn different mask strategies, which can effectively improve the anti-noise ability of the model, leading to the empowment of its overall generalization ability. Reffered to the previous studies, the ACC, MCC, SN, and SP of BERT_PLPS using a dynamic mask are increased by 16.72%, 38.50%, 24.21%, and 13.49%, respectively, compared to Bigram-PGK. Compared with PhoglyPred, the ACC, MCC, SN, and SP of BERT_PLPS using a dynamic mask are increased by

Table 2: Comparison results of the evaluation parameters of the model

Model name	ACC	MCC	Sn	Sp
BERT_PLPS(Dynamic)	**99.53%**	**99.07%**	**99.21%**	**99.85%**
BERT_PLPS(Static)	99.16%	98.33%	98.92%	99.41%
Bigram-PGK	82.81%	60.57%	75.00%	86.36%
PhoglyPred	93.69%	92.88%	94.78%	95.63%
Phogly_PseAAC	83.33%	63.12%	85.00%	82.69%
iDGK	77.72%	55.40%	88.24%	73.34%
RAM_PGK	81.79%	59.35%	76.70%	84.16%
EvolStruct-Phogly	83.92%	63.74%	77.78%	86.84%
predPhogly-Site	98.50%	96.47%	97.43%	98.41%
PLP_FS	99.18%	98.14%	98.99%	99.27%

5.84%, 6.19%, 4.43%, and 4.22%, respectively. Compared with Phogly_PseAAC, the ACC, MCC, SN, and SP of BERT_PLPS using the dynamic mask are increased by 16.20%, 35.95%, 14.21%, and 17.16%, respectively. Compared with iDGK, the ACC, MCC, SN, and SP using the dynamic mask are are increased by 21.81%, 43.67%, 10.97%, and 26.51%, respectively. Compared with RAM_PGK, the ACC, MCC, SN, and SP of BERT_PLPS using the dynamic mask are increased by 17.74%, 39.72%, 22.51%, and 15.69%, respectively. Compared with EvolStruct-Phogly, the ACC, MCC, SN, and SP of BERT_PLPS using the dynamic mask are increased by 15.61%, 35.33%, 21.43%, and 13.01%, respectively. Compared with predPhogly Site, the ACC, MCC, SN and SP of BERT_PLPS using a dynamic mask are increased by 1.03%, 2.60%, 1.78% and 1.44%, respectively. Compared with PLP_FS, the ACC, MCC, SN, and SP of BERT_PLPS with the dynamic mask are increased by 0.35%, 0.93%, 0.22%, and 0.58%, respectively. As displayed in Figure 6 and Table 2, the visualization results show that BERT_PLPS is superior to the other models. The significant improvement of our model results over models in these studies indicates that BERT_PLPS is superior to other studies in predicting lysine phosphoglyceration sites and has the potential to be an effective biological tool.

Experimental results can verify that proper feature extraction is beneficial for model construction. Meanwhile, the BERT based lysine vascularization site prediction model can effectively mine the shared and complementary information between different protein features. Recent state-of-the-art models tend to focus on complex feature extraction, and some tend to build sophisticated deep learning models to learn feature information of amino acid sequences and perform the task of site identification. For the first time, we bring these two focuses together to build a set of site prediction models that integrate excellent protein feature extraction algorithms with sophisticated and applicable deep learning. Based on this, we introduce the random mask module to give the model stronger adversarial noise and generalization performance. By all measures, our model is ahead of recent state-of-the-art models.

In conclusion, the BERT_PLPS model proposed can significantly improve the prediction performance of lysine phosphate glycerylation sites beyond existing models.

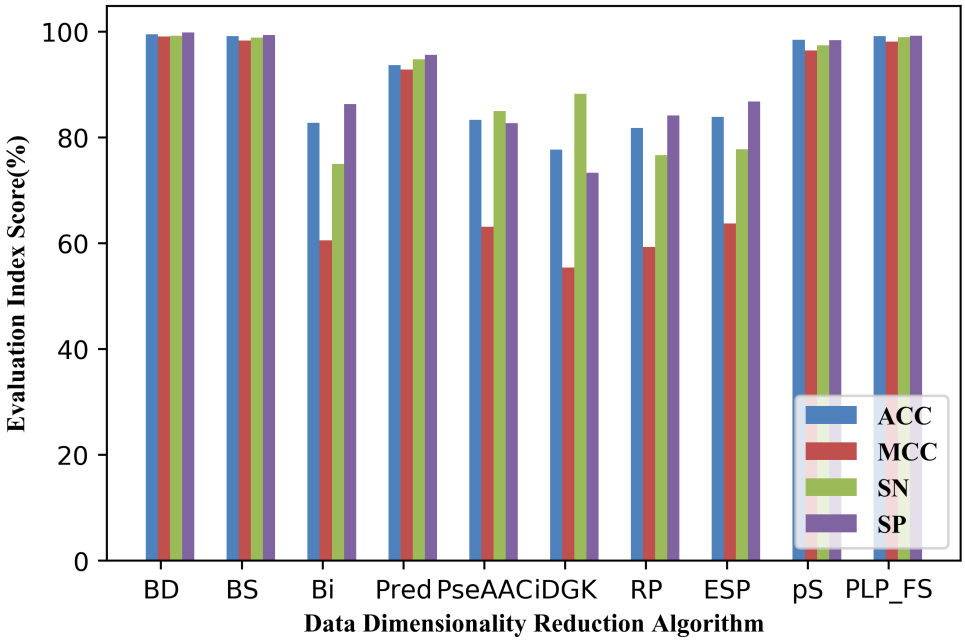

Fig. 6: Histogram comparison of model evaluation metrics

Our results show that transformer models in the field of natural language processing can be effectively used to the task of site prediction in protein sequences.

4 Conclusion

A well-performing protein phosphoglyceration site prediction model can help the scientists to identify actual phosphoglyceration sites. In the past, researchers have proposed computing methods based on machine learning or deep learning. In this work, we identified the most suitable data dimensionality reduction algorithms for protein sequence information, and explored the data distribution of protein feature data points. We also propose BERT_PLPS, which is a lysine phosphate glycerylation site prediction model using a BERT-based encoder with a dynamic mask. Our results show that BERT_PLPS model is the most advanced method for predicting these sites. Therefore, cutting-edge natural language processing models can be transferred to process biological sequence information[64]. In addition, the dynamic mask effectively improves the anti-noise ability of the model and the overall performance of the model. BERT_PLPS is a powerful deep learning system that can accurately predict phosphoglyceration sites.

Acknowledgments. This work was supported in part by Joint found for smart computing of Shandong Natural Science Foundation under Grant ZR2020LZH013; open project of State Key Laboratory of Computer Architecture CARCHA202001; the

Major Scientific and Technological Innovation Project in Shandong Province under Grant 2021CXG010506 and 2022CXG010504; "New University 20 items" Funding Project of Jinan under Grant 2021GXRC108 and 2021GXRC024; Shandong Province Nature Science Foundation under Grant ZR2021MH104; Shandong Province Nature Science Foundation under Grant ZR2020MH208; the Young Taishan Scholars Program under Grant tsqn201909178; Shandong University integrated research and Cultivation project under Grant 2022JC015.

References

[1] Ramazi, S., Zahiri, J.: Post-translational modifications in proteins: resources, tools and prediction methods. Database **2021** (2021)

[2] Yu, H., Bu, C., Liu, Y., Gong, T., Liu, X., Liu, S., Peng, X., Zhang, W., Peng, Y., Yang, J., *et al.*: Global crotonylome reveals cdyl-regulated rpa1 crotonylation in homologous recombination–mediated dna repair. Science advances **6**(11), 4697 (2020)

[3] Bagwan, N., El Ali, H.H., Lundby, A.: Proteome-wide profiling and mapping of post translational modifications in human hearts. Scientific reports **11**(1), 2184 (2021)

[4] Arnaudo, A.M., Garcia, B.A.: Proteomic characterization of novel histone post-translational modifications. Epigenetics & chromatin **6**(1), 1–7 (2013)

[5] Bhat, K.P., Ümit Kaniskan, H., Jin, J., Gozani, O.: Epigenetics and beyond: targeting writers of protein lysine methylation to treat disease. Nature Reviews Drug Discovery **20**(4), 265–286 (2021)

[6] Szondy, Z., Korponay-Szabó, I., Király, R., Sarang, Z., Tsay, G.J.: Transglutaminase 2 in human diseases. BioMedicine **7**(3) (2017)

[7] Li, S., Iakoucheva, L.M., Mooney, S.D., Radivojac, P.: Loss of post-translational modification sites in disease. In: Biocomputing 2010, pp. 337–347. World Scientific, ??? (2010)

[8] Chandra, A., Sharma, A., Dehzangi, A., Shigemizu, D., Tsunoda, T.: Bigram-pgk: phosphoglycerylation prediction using the technique of bigram probabilities of position specific scoring matrix. BMC molecular and cell biology **20**, 1–9 (2019)

[9] Bulcun, E., Ekici, M., Ekici, A.: Disorders of glucose metabolism and insulin resistance in patients with obstructive sleep apnoea syndrome. International journal of clinical practice **66**(1), 91–97 (2012)

[10] Cipriani, C., Colangelo, L., Santori, R., Renella, M., Mastrantonio, M., Minisola, S., Pepe, J.: The interplay between bone and glucose metabolism. Frontiers in endocrinology **11**, 122 (2020)

[11] Ju, Z., Cao, J.-Z., Gu, H.: Predicting lysine phosphoglycerylation with fuzzy svm by incorporating k-spaced amino acid pairs into chou's general pseaac. Journal of Theoretical Biology **397**, 145–150 (2016)

[12] Moellering, R.E., Cravatt, B.F.: Functional lysine modification by an intrinsically reactive primary glycolytic metabolite. Science **341**(6145), 549–553 (2013)

[13] Xu, Y., Ding, Y.-X., Ding, J., Wu, L.-Y., Xue, Y.: Mal-lys: prediction of lysine malonylation sites in proteins integrated sequence-based features with mrmr feature selection. Scientific reports **6**(1), 1–7 (2016)

[14] Xiang, Q., Feng, K., Liao, B., Liu, Y., Huang, G.: Prediction of lysine malonylation sites based on pseudo amino acid. Combinatorial chemistry & high throughput screening **20**(7), 622–628 (2017)

[15] Du, Y., Zhai, Z., Li, Y., Lu, M., Cai, T., Zhou, B., Huang, L., Wei, T., Li, T.: Prediction of protein lysine acylation by integrating primary sequence information with multiple functional features. Journal of proteome research **15**(12), 4234–4244 (2016)

[16] Qiu, W.-R., Xiao, X., Lin, W.-Z., Chou, K.-C.: iubiq-lys: prediction of lysine ubiquitination sites in proteins by extracting sequence evolution information via a gray system model. Journal of Biomolecular Structure and Dynamics **33**(8), 1731–1742 (2015)

[17] Hou, T., Zheng, G., Zhang, P., Jia, J., Li, J., Xie, L., Wei, C., Li, Y.: Lacep: lysine acetylation site prediction using logistic regression classifiers. PloS one **9**(2), 89575 (2014)

[18] Jia, J., Zhang, L., Liu, Z., Xiao, X., Chou, K.-C.: psumo-cd: predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general pseaac. Bioinformatics **32**(20), 3133–3141 (2016)

[19] Xu, Y., Ding, Y.-X., Ding, J., Wu, L.-Y., Deng, N.-Y.: Phogly–pseaac: prediction of lysine phosphoglycerylation in proteins incorporating with position-specific propensity. Journal of Theoretical Biology **379**, 10–15 (2015)

[20] Sohrawordi, M., Hossain, M.A., Hasan, M.A.M.: Plp_fs: prediction of lysine phosphoglycerylation sites in protein using support vector machine and fusion of multiple f_score feature selection. Briefings in Bioinformatics **23**(5) (2022)

[21] Ning, W., Jiang, P., Guo, Y., Wang, C., Tan, X., Zhang, W., Peng, D., Xue, Y.: Gps-palm: a deep learning-based graphic presentation system for the prediction of s-palmitoylation sites in proteins. Briefings in bioinformatics **22**(2), 1836–1847 (2021)

[22] Huang, G., Shen, Q., Zhang, G., Wang, P., Yu, Z.-G.: Lstmcnnsucc: a bidirectional lstm and cnn-based deep learning method for predicting lysine succinylation sites. BioMed Research International **2021** (2021)

[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[24] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)

[25] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., *et al.*: A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence **45**(1), 87–110 (2022)

[26] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)

[27] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[28] Xu, H., Zhou, J., Lin, S., Deng, W., Zhang, Y., Xue, Y.: Plmd: an updated data resource of protein lysine modifications. Journal of Genetics and Genomics **44**(5), 243–250 (2017)

[29] Kondratenko, Y., Korobeynikov, A., Lapidus, A.: Cdsnake: Snakemake pipeline for retrieval of annotated otus from paired-end reads using cd-hit utilities. BMC bioinformatics **21**, 1–7 (2020)

[30] Ju, Z., Wang, S.-Y.: Prediction of lysine formylation sites using the composition of k-spaced amino acid pairs via chou's 5-steps rule and general pseudo components. Genomics **112**(1), 859–866 (2020)

[31] Kao, H.-J., Nguyen, V.-N., Huang, K.-Y., Chang, W.-C., Lee, T.-Y.: Succsite: incorporating amino acid composition and informative k-spaced amino acid pairs to identify protein succinylation sites. Genomics, proteomics & bioinformatics **18**(2), 208–219 (2020)

[32] Wang, R., Wang, Z., Wang, H., Pang, Y., Lee, T.-Y.: Characterization and identification of lysine crotonylation sites based on machine learning method on both plant and mammalian. Scientific Reports **10**(1), 20447 (2020)

[33] Lv, H., Dao, F.-Y., Guan, Z.-X., Yang, H., Li, Y.-W., Lin, H.: Deep-kcr: accurate detection of lysine crotonylation sites using deep learning method. Briefings in bioinformatics **22**(4), 255 (2021)

[34] Lv, H., Zhang, Y., Wang, J.-S., Yuan, S.-S., Sun, Z.-J., Dao, F.-Y., Guan, Z.-X., Lin, H., Deng, K.-J.: irice-ms: an integrated xgboost model for detecting multitype post-translational modification sites in rice. Briefings in Bioinformatics **23**(1), 486 (2022)

[35] Dao, F.-Y., Lv, H., Wang, F., Feng, C.-Q., Ding, H., Chen, W., Lin, H.: Identify origin of replication in saccharomyces cerevisiae using two-step feature selection technique. Bioinformatics **35**(12), 2075–2083 (2019)

[36] Feng, C.-Q., Zhang, Z.-Y., Zhu, X.-J., Lin, Y., Chen, W., Tang, H., Lin, H.: itermpseknc: a sequence-based tool for predicting bacterial transcriptional terminators. Bioinformatics **35**(9), 1469–1477 (2019)

[37] Chung, C.-R., Chang, Y.-P., Hsu, Y.-L., Chen, S., Wu, L.-C., Horng, J.-T., Lee, T.-Y.: Incorporating hybrid models into lysine malonylation sites prediction on mammalian and plant proteins. Scientific reports **10**(1), 10541 (2020)

[38] Basith, S., Lee, G., Manavalan, B.: Stallion: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction. Briefings in Bioinformatics **23**(1), 376 (2022)

[39] Ning, Q., Zhao, X., Bao, L., Ma, Z., Zhao, X.: Detecting succinylation sites from protein sequences using ensemble support vector machine. BMC bioinformatics **19**(1), 1–9 (2018)

[40] Sohrawordi, M., Hossain, M.A.: Prediction of lysine formylation sites using support vector machine based on the sample selection from majority classes and synthetic minority over-sampling techniques. Biochimie **192**, 125–135 (2022)

[41] Patel, H., Singh Rajput, D., Thippa Reddy, G., Iwendi, C., Kashif Bashir, A., Jo, O.: A review on classification of imbalanced data for wireless sensor networks. International Journal of Distributed Sensor Networks **16**(4), 1550147720916404 (2020)

[42] Jia, C., Zhang, M., Fan, C., Li, F., Song, J.: Formator: predicting lysine formylation sites based on the most distant undersampling and safe-level synthetic minority oversampling. IEEE/ACM transactions on computational biology and bioinformatics **18**(5), 1937–1945 (2019)

[43] Khorshid, S.F., Abdulazeez, A.M.: Breast cancer diagnosis based on k-nearest neighbors: a review. PalArch's Journal of Archaeology of Egypt/Egyptology **18**(4), 1927–1951 (2021)

[44] Ahmed, G., Er, M.J., Fareed, M.M.S., Zikria, S., Mahmood, S., He, J., Asad, M., Jilani, S.F., Aslam, M.: Dad-net: Classification of alzheimer's disease using adasyn oversampling technique and optimized neural network. Molecules **27**(20), 7085 (2022)

[45] Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., Nappi, M.: Improving the prediction of heart failure patients' survival using smote and effective data mining techniques. IEEE access **9**, 39707–39716 (2021)

[46] Sharma, R., Sungheetha, A., *et al.*: An efficient dimension reduction based fusion of cnn and svm model for detection of abnormal incident in video surveillance. Journal of Soft Computing Paradigm (JSCP) **3**(02), 55–69 (2021)

[47] Ray, P., Reddy, S.S., Banerjee, T.: Various dimension reduction techniques for high dimensional data analysis: a review. Artificial Intelligence Review **54**, 3473–3515 (2021)

[48] Kherif, F., Latypova, A.: Principal component analysis. In: Machine Learning, pp. 209–225. Elsevier, ??? (2020)

[49] Lee, W.J., Mendis, G.P., Triebe, M.J., Sutherland, J.W.: Monitoring of a machining process using kernel principal component analysis and kernel density estimation. Journal of Intelligent Manufacturing **31**, 1175–1189 (2020)

[50] Wang, Y., Zhang, Z., Lin, Y.: Multi-cluster feature selection based on isometric mapping. IEEE/CAA Journal of Automatica Sinica **9**(3), 570–572 (2021)

[51] Zhang, Y., Yang, Y., Li, T., Fujita, H.: A multitask multiview clustering algorithm in heterogeneous situations based on lle and le. Knowledge-Based Systems **163**, 776–786 (2019)

[52] Anowar, F., Sadaoui, S., Selim, B.: Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). Computer Science Review **40**, 100378 (2021)

[53] Uddin, M.P., Mamun, M.A., Afjal, M.I., Hossain, M.A.: Information-theoretic feature selection with segmentation-based folded principal component analysis (pca) for hyperspectral image classification. International Journal of Remote Sensing **42**(1), 286–321 (2021)

[54] Yan, X., Guan, T., Fan, K., Sun, Q.: Novel double layer bilstm minor soft fault detection for sensors in air-conditioning system with kpca reducing dimensions. Journal of Building Engineering **44**, 102950 (2021)

[55] Ghojogh, B., Ghodsi, A., Karray, F., Crowley, M.: Multidimensional scaling, sammon mapping, and isomap: Tutorial and survey. arXiv preprint arXiv:2009.08136 (2020)

[56] Ghojogh, B., Ghodsi, A., Karray, F., Crowley, M.: Locally linear embedding and its variants: Tutorial and survey. arXiv preprint arXiv:2011.10925 (2020)

[57] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

[58] Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.: On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10524–10533 (2020). PMLR

[59] Chandra, A.A., Sharma, A., Dehzangi, A., Tsunoda, T.: Ram-pgk: prediction of lysine phosphoglycerylation based on residue adjacency matrix. Genes **11**(12), 1524 (2020)

[60] Huang, K.-Y., Hung, F.-Y., Kao, H.-J., Lau, H.-H., Weng, S.-L.: idpgk: characterization and identification of lysine phosphoglycerylation sites based on sequence-based features. BMC bioinformatics **21**(1), 1–16 (2020)

[61] Chen, Q.-Y., Tang, J., Du, P.-F.: Predicting protein lysine phosphoglycerylation sites by hybridizing many sequence based features. Molecular BioSystems **13**(5), 874–882 (2017)

[62] Chandra, A.A., Sharma, A., Dehzangi, A., Tsunoda, T.: Evolstruct-phogly: incorporating structural properties and evolutionary information from profile bigrams for the phosphoglycerylation prediction. BMC genomics **19**, 1–9 (2019)

[63] Ahmed, S., Rahman, A., Hasan, M.A.M., Islam, M.K.B., Rahman, J., Ahmad, S.: predphogly-site: Predicting phosphoglycerylation sites by incorporating probabilistic sequence-coupling information into pseaac and addressing data imbalance. Plos one **16**(4), 0249396 (2021)

[64] Qiao, Y., Zhu, X., Gong, H.: Bert-kcr: prediction of lysine crotonylation sites by a transfer learning method with pre-trained bert models. Bioinformatics **38**(3), 648–654 (2022)