

RESEARCH ARTICLE

DTL-NeddSite: A Deep-Transfer Learning Architecture for Prediction of Lysine Neddylolation Sites

DELI XU^{1,2}, YAFEI ZHU¹, QIANG XU³, YUHAI LIU⁴, YU CHEN¹, YANG ZOU⁵, AND LEI LI^{1,5}¹College of Computer Science and Technology, Qingdao University, Qingdao 266071, China²Sino Genomics Technology Company Ltd., Qingdao 266510, China³Department of Geriatrics, The Affiliated Hospital of Qingdao University, Qingdao 266000, China⁴Dawning International Information Industry Company Ltd., Qingdao 266101, China⁵Faculty of Biomedical and Rehabilitation Engineering, University of Health and Rehabilitation Sciences, Qingdao 266000, China

Corresponding authors: Lei Li (lileime@hotmail.com) and Yang Zou (yangzou@uor.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 31770821, Grant 32071430, and Grant 32271504; and in part by the Innovation Capability Improvement Project of Science and Technology for Small- and Medium-Sized Enterprises in Shandong Province under Grant 2021TSGC1295.

ABSTRACT Neddylolation, as a reversible post-translational modification (PTM), plays a role in various cellular processes. Defects in neddylolation are related to human diseases. Detecting neddylolation sites is necessary for revealing the mechanisms of protein neddylolation. As identifying such sites through experimental methods is expensive and time-consuming, it is essential to develop *in silico* methods to predict neddylolation sites. In this study, we constructed a few classifiers integrating various algorithms and encoding features. However, they performed poorly (AUC \approx 0.767), mainly due to the limited number (\sim 1000) of identified neddylolation sites. The large number ($>$ 100,000) of other lysine PTM sites inspired us to employ a deep transfer learning (DTL) strategy for performance improvement. We constructed a predictor, dubbed DTL-NeddSite, which adopted the DTL-based convolution neural network using the one-hot encoding approach. Specifically, the massive number of lysine PTM sites were used to build the source model, followed by the fine-tuning of the target model using neddylolation sites. DTL-NeddSite compared favourably with the corresponding model without the DTL strategy in cross-validation and independent tests. For instance, the AUC value increased to 0.818. Contrary to a general DTL model that combines frozen and unfrozen layers, all the layers in DTL-NeddSite were unfrozen to re-train. We expect the DTL strategy to be widely used in newly discovered modification types with limited known sites. Furthermore, DTL-NeddSite is freely accessible at <https://github.com/XuDeli123/DTL-NeddSite>.


INDEX TERMS Neddylolation, post-translational modification, modification site prediction, deep learning, deep transfer learning.

I. INTRODUCTION

Neddylolation is a post-translational modification (PTM) process in which the ubiquitin-like protein NEDD8 is covalently attached to the lysine side chain of a target protein [1]. Protein neddylolation regulates many critical biological processes and has become a novel anti-tumour therapeutic target [2], [3], [4], [5], [6], [7], [8], [9]. Identifying protein neddylolation

sites is essential to explore their characteristics. Recently, a few experimental approaches have been developed to study proteome-wide neddylolation sites [10], [11]. Nevertheless, they have disadvantages, such as laboriousness and expansiveness. An alternative is the construction of *in silico* prediction models.

Two predictors, NeddPred and NeddPreddy [12], [13], have been developed as the first attempt to predict neddylolated sites in protein sequences. Both predictors are based on the same dataset, including 51 neddylolation sites from

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

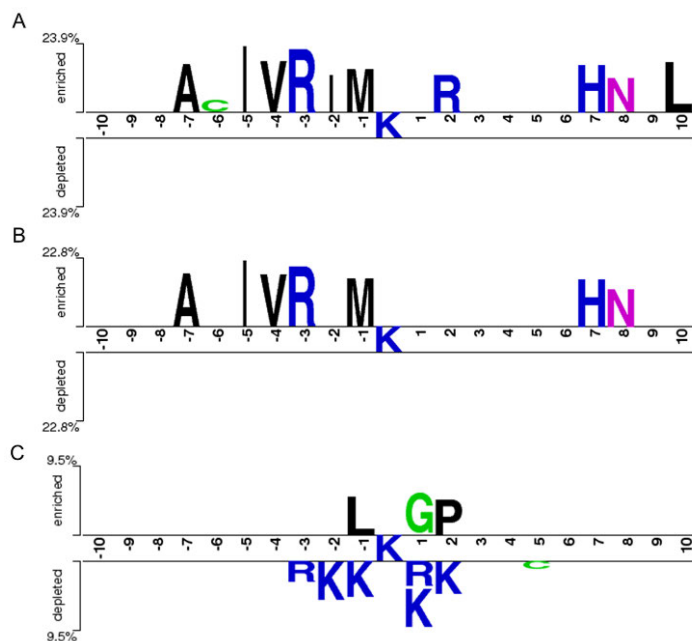


FIGURE 1. Sequence patterns of neddylolation-containing peptides, showing enriched and depleted residues, were generated using the Two Sample Logo program: (A) The pattern of the neddylolation peptides using the previously published dataset (51 samples) compared with our collected dataset (> 1000 samples); (B) The pattern was based on the previous neddylolation dataset and non-modification peptides [12], [13]; (C) The pattern was based on our collected neddylolation and non-modification peptides ($p < 0.05$, student's t-test with Bonferroni correction).

various organisms and 1031 non-neddylolation sites. They contain the same algorithmic architecture (*i.e.*, a support vector machine (SVM)), but they have different encoding schemes. It has been observed that flanking sequences of the neddylolation sites are commonly flexible and enriched with positively charged amino acids [12]. Nevertheless, they have two limitations. First, the neddylolation sites (51) for model construction are limited and may not fully reflect the neddylolation characteristics. We compared these data with 1715 recently identified neddylolation sites [10], [11] using the Two Sample Logo program [14]. Figure 1A shows that they are significantly different. Specifically, 11 residues are enriched in the former at different positions (*e.g.*, A@P-7 and L@P10), while no residue is enriched in the latter. Furthermore, we compared each of them with K-centric peptides without modification annotation, and we found that they demonstrated different features (Figure 1B and 1C). Therefore, it is essential to construct novel classifiers using the latest and most extensive data. Second, the SVM algorithm applied to the reported predictors is a type of traditional machine-learning (ML) algorithm. It has been found that these algorithms demonstrate a relatively poorer level of performance than deep learning (DL) algorithms in the field of bioinformatics, such as the prediction of RNA-binding sites [15] and PTM sites [16], [17], [18], [19]. Therefore, it is necessary to develop prediction models based on DL algorithms and to compare them with the reported models.

In this study, we collected known human neddylolation sites as a new dataset and constructed various prediction models (Figure 2). The classifiers combined various algorithms (*e.g.*, RF, SVM, CNN, and long short-term memory (LSTM)) and different encoding features (*e.g.*, one-hot and word embedding). We split the dataset into cross-validation (CV) and independent test datasets for model evaluation. The best model was the convolution neural network with the one-hot encoding approach (CNN_{OH}). It had area-under-the-curve (AUC) values of 0.767 and 0.771 in the cross-validation (CV) and independent tests, respectively. Because nearly half of the neddylolation sites had *in situ* crosstalk with other lysine PTM types, there existed commonalities between different lysine PTM types. These commonalities may assist in the construction of new models with superior performance. Accordingly, we applied a deep transferlearning (DTL) strategy to model construction based on the large number (> 100,000) of known lysine PTM sites. Specifically, the massive amount of human lysine PTM data was used to pre-train a CNN_{OH} classifier, followed by fine-tuning using human neddylolation sites. The final model achieved an AUC value of 0.818. Therefore, the CNN_{OH} classifier with the DTL strategy compared favourably to the corresponding model that lacked such a strategy. Interestingly, the DTL-based model demonstrated better performance than did the Apriori algorithm. It also showed good predictive ability when applied to a real situation. In summary, the massive number of lysine PTM sites could improve the performance in predicting neddylolation

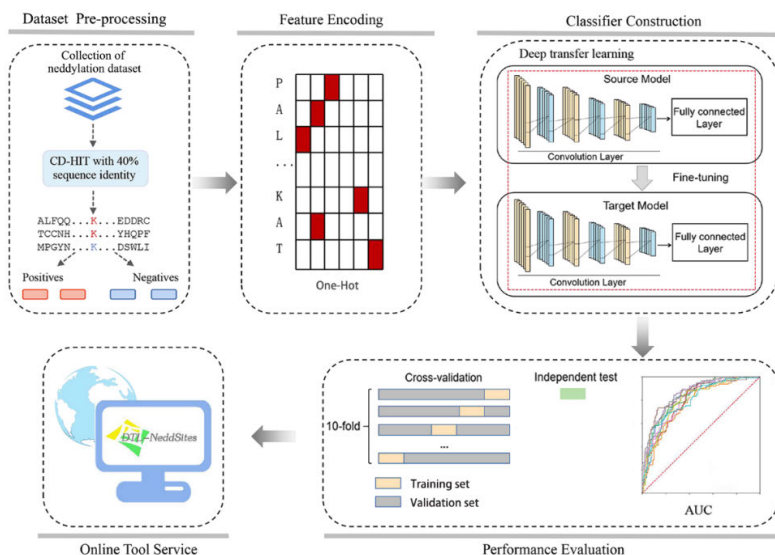


FIGURE 2. The flowchart of model construction and evaluation.

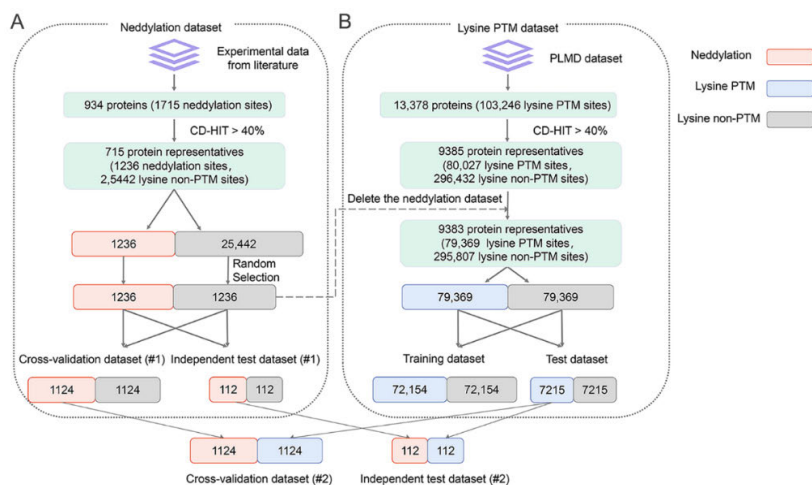


FIGURE 3. Schematic diagram of human neddylolation dataset construction and pre-processing: (A) Neddylolation dataset construction; (B) Other lysine PTM dataset construction.

sites. The DTL-based model and the dataset are accessible via <https://github.com/XuDeli123/DTL-NeddSite>.

II. MATERIALS AND METHODS

A. DATASET CONSTRUCTION AND PRE-PROCESSING

Figure 3 shows the process of dataset construction. Specifically, we collected 1715 experimentally verified lysine neddylolation sites on 934 human proteins from the literature [10], [11], [12]. After CD-HIT clustering [20], [21] with a sequence identity of 40%, we obtained 715 protein clusters, which contained 1236 neddylolation sites as positives and 25,442 lysine non-modified sites. From these non-modified sites, we randomly selected 1236 sites as the negatives. Finally, we split the data into 11 groups: 10/11 (1124 positives and 1124 negatives) as the cross-validation

dataset and 1/11 (112 positives and 112 negatives) as the independent test dataset.

In order to estimate the optimal sequence window for model construction, we set the sequence window to different sizes and compared them through ten-fold cross-validation. We found that a window size of 41 had the best result. This is consistent with our previous studies of lysine PTM site prediction [17], [19], [22], [23], where prediction performance increased with the protein sequence length and reached a plateau when the length generally ranged from 31 to 41. Accordingly, we represented each site using a sequence window of 41 amino acids in length, with the modified/unmodified “K” in the centre. If the central “K” is located near the N-terminus or C-terminus of the protein sequence, “X” residues are added to the sequence at the affected terminus to ensure that the length is maintained.

B. FEATURE ENCODING SCHEMES

1) ENCODING OF ENHANCED AMINO ACID COMPOSITION (EAAC)

EAAC [24] calculates amino acid frequencies of a fixed-length sequence window (the default length being 5), sliding continuously from each sequence's N-terminus to the C-terminus. Each sequence is encoded as a vector of 740 ($= (41 - 5 + 1) \times 20$) items. The related formula for each sequence is listed below:

$$f(win, t) = \frac{N(win, t)}{N(win)} win \in \{\text{window1}, \dots, \text{window37}\}, t \in \{A, \dots, Y\} \quad (1)$$

where $N(win, t)$ is the number of the amino acid t in the sliding window (win), and $N(win)$ is the length of the sliding window (win) (*i.e.*, 5).

2) ONE-HOT (OH) ENCODING

Every amino acid in a sequence is encoded by a 21-dimensional binary vector, where each position corresponds to a specific amino acid, or "X". In the vector related to a given amino acid, the position corresponding to the amino acid is set as 1, and the other positions are set as 0. For example, the amino acid "A" is represented by the vector "10000000000000000000", according to the alphabet "AVLIFWMPGSTCYNQHKRDEX."

3) WORD EMBEDDING (WE) ENCODING

Word embedding [25] relies on the NUM encoding approach, which maps each type of amino acid residue to an integer (Zhang Y. et al., 2019). After the NUM encoding, each integer is encoded into a predefined five-dimensional word vector. Therefore, each sequence is encoded as a vector of 205 ($= 41 \times 5$) items.

C. THE ARCHITECTURE OF MACHINE-LEARNING MODELS

Machine-learning algorithms are commonly used in predicting PTM sites. They include random forest (RF) and SVM. RF integrates multiple independent decision trees, and each decision tree produces a result. RF selects the result with the most votes as its final result by counting the results of each decision tree. This study set the number of decision trees to 150. By contrast, SVM is proposed from the optimal classification surface in the case of linear separability. Specifically, the input space is transformed into a high-dimensional space where the optimal linear classification surface is obtained. This transformation is realized by defining an appropriate inner product function. RF-based and SVM-based classifiers were developed using the Python module "sklearn".

D. THE ARCHITECTURE OF DEEP LEARNING MODELS

Deep learning (DL) models have shown performance superior to traditional ML models for predicting PTM sites [22], [26]. Here, we constructed four DL models integrating the

DL algorithm (*i.e.*, CNN or LSTM) with the encoding approach (*i.e.*, OH or WE).

1) THE CNN MODEL WITH ONE-HOT ENCODING

We constructed a CNN model based on OH encoding (CNN_{OH}), including the following four layers (Figure 4A):

- 1) Input layer. Each sequence of information is converted into a feature vector with OH encoding.
- 2) Convolution layer. This layer includes three convolution sublayers and three max-pooling sublayers arranged alternately. There are 128 convolution kernels with the sizes of 1, 3 and 9 for the first, second and third convolution sublayers, respectively. The rectified linear unit (ReLU) is regarded as the activation function of each convolution sublayer. In the max-pooling sublayers, the parameters' pool_size and padding are set as 2 and "same", respectively.
- 3) Fully connected layer. This layer contains a dense sublayer with 64 neural units without flattening and a global average pooling sublayer to calculate and produce an average value.
- 4) Output layer. This layer contains a single neuron to produce the probability score (within the range from 0 to 1), indicating the likelihood of neddylation. If the probability score of an input sequence is greater than a specified threshold, the central lysine in the sequence is predicted to be neddylated.

2) THE CNN MODEL WITH WORD EMBEDDING ENCODING

This CNN_{WE} model contains five layers. The input layer receives sequences with a window size of 41 and maps each amino acid type to an integer. Next, each integer is converted into a five-dimensional word vector in the embedding layer. The remaining three layers are the same as the second, third and fourth layers of CNN_{OH} (Figure 4B).

3) THE LSTM MODEL WITH ONE-HOT ENCODING

As a recurrent neural network, LSTM [27], [28] can selectively remember patterns for a long time, which is ideal for processing sequential data. We constructed the LSTM model with the OH encoding approach ($LSTM_{OH}$), containing the following five layers (Figure 4C):

- 1) Input layer. Each sequence is converted into a feature vector with OH encoding.
- 2) Recurrent layer. This layer consists of two LSTM sublayers. Each sublayer contains 41 sequentially connected LSTM cells, corresponding to the residues of the input sequence. The LSTM cells in the first/second sublayer contain 128/256 hidden neuron units, respectively. Every cell of the first sublayer processes the information from the corresponding amino acid, and each cell of the second sublayer processes the information from that of the first sublayer cell. Lastly, the vector from the 41st LSTM cell in the second LSTM

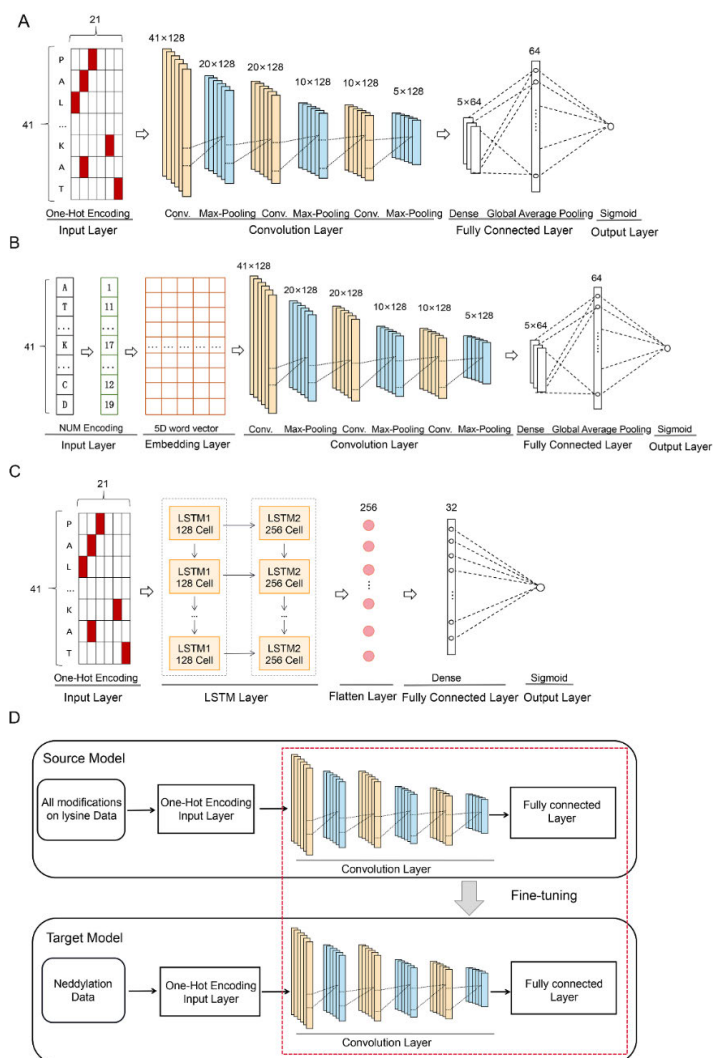


FIGURE 4. The architectures of (A) CNN_{OH} ; (B) CNN_{WE} ; (C) $LSTM_{OH}$; and (D) the DTL-based CNN_{OH} .

sublayer is regarded as the output of the LSTM layer, representing the features of the input peptide sequence.

- 3) Flatten layer. This layer collapses the spatial dimensions of the input into a one-dimensional linear vector.
- 4) Full connection layer. This layer contains 32 neurons with the rectified linear unit (ReLU) as the activation function.
- 5) Output layer. This layer contains a single neuron, activated by the “Sigmoid” function, to produce probability scores.

4) THE LSTM MODEL WITH WORD EMBEDDING ENCODING

The LSTM algorithm based on word embedding encoding ($LSTM_{WE}$) consists of six layers. The input layer receives the sequence with a window size of 41 and maps each amino acid type to an integer using NUM encoding. In the embedding layer, each integer of the vector from the input layer is converted and encoded into a five-dimensional word vector.

The remaining four layers are the same as the second through fifth layers of $LSTM_{OH}$.

E. THE ARCHITECTURE OF THE DEEP TRANSFER LEARNING MODELS

Deep transfer learning (DTL) can leverage knowledge learned in previous tasks (*i.e.*, source tasks) based on a massive semi-related dataset and apply it to new related tasks (target tasks) that have limited target data [29], [30], [31], [32], [33], [34], [35]. We used the DTL strategy for model construction (Figure 4D). Specifically, a source model is pre-trained using massive amounts of lysine PTM data, following by fine-tuning through gradually unfrozen parameters, starting from the last layer using the limited neddylolation sites.

F. THE STRATEGY FOR AVOIDING OVERFITTING

In DL architectures, the “ReLU” function [36] was used as the activation function for the convolution and fully

connected sublayers to avoid gradient dispersion during training. The maximum number of training cycles was set through the optimized number of epochs (300) and the batch size (256 for CNN_{OH} and CNN_{WE}; 512 for LSTM_{OH} and LSTM_{WE}) to ensure that the loss function value converged. The RMSprop algorithm [37] was used to optimize the model parameters. The loss rate was 50% in the convolution and fully connected layers to avoid overfitting. In addition, an early stop strategy was applied to stop the training process when the verification accuracy did not increase in 50 consecutive iterations.

G. PERFORMANCE EVALUATION STRATEGIES

We used five measures (*e.g.*, sensitivity (Sn), specificity (Sp), Matthews correlation coefficient (MCC), accuracy (ACC), and area under the curve (AUC)) to evaluate the performance of the prediction model. The definitions of Sn, Sp, ACC, and MCC are given as follows:

$$Sn = \frac{TP}{TP + FN} \quad (2)$$

$$Sp = \frac{TN}{TN + FP} \quad (3)$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

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

In the above formulas, TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and true negatives, respectively. Generally, the closer the AUC value is to 1, the better the model's performance.

III. RESULTS

A. CONSTRUCTION OF THE NEDDYLOTION DATASET AND EVALUATION OF VARIOUS CLASSIFIERS

We collected 1715 experimentally verified lysine neddylation sites on 934 human proteins [10], [11], [12] (Figure 3A). To remove the redundant proteins, we grouped the proteins into 715 clusters using CD-HIT [20], [21] with a sequence identity of 40% as the threshold. We considered the protein with the most neddylation sites in each cluster as representative. To this end, we obtained 1236 neddylation sites of the 715 representatives as positive sites. The remaining 37,464 lysine residues of the same representatives were considered as potential negative sites.

A lysine residue may undergo different types of PTM. For example, the lysine K 774 on protein TIF1 β is annotated with ubiquitination, SUMOylation and neddylation [10], [38], [39]. We reason that lysine sites annotated with any PTM type are more likely to be neddylated than those without any PTM annotation. Thus, the former is inappropriate to consider as a negative sample, as compared to the latter. From potential negative lysine sites, we removed 12,022 lysine sites with PTM annotations according to PLMD [40],

leading to the retention of 25,442 lysine sites. We randomly selected 1236 lysine sites from the retained data as negatives to balance the positives and negatives (Figure 3A). The data were further divided into CV dataset #1 (1124 positives and 1124 negatives) and Independent test dataset #1 (112 positives and 112 negatives) (Figure 3A).

Many computational approaches for predicting PTM sites are generally based on traditional ML algorithms (*e.g.*, RF and SVM) combined with various features encoded from peptide sequences. In this study, we constructed four predictors by integrating two encoding schemes (*i.e.*, EAAC and one-hot) and two ML algorithms (*i.e.*, RF and SVM), respectively. Moreover, deep learning algorithms have recently been applied in the field of PTM site prediction and have demonstrated superior performance [16], [41], [42]. Accordingly, we developed four different DL predictors, integrating CNN (or LSTM) with the encoding approach OH (or WE). We compared the eight models using the ten-fold CV set #1 and independent test #1 (Table 1). The CNN model with the OH encoding (CNN_{OH}) showed the best performance. For example, it had average AUC and MCC values of 0.767 and 0.402 for the CV. These values were similar to those in the independent test (AUC: 0.771; MCC: 0.400) (Table 1 and Figure 5), suggesting the robustness of the CNN_{OH} model.

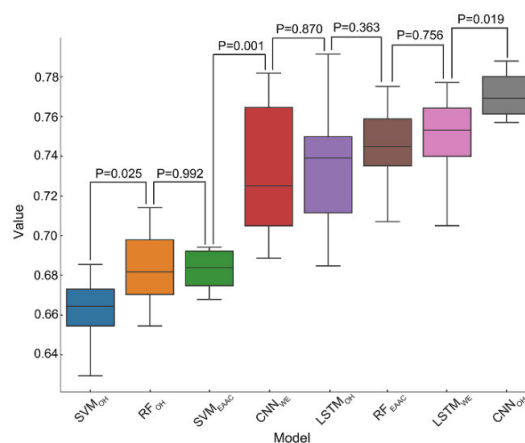


FIGURE 5. Performance comparison of the classifiers constructed using Cross-validation dataset #1 in Independent test #1. A paired student's t-test was calculated to estimate the statistical difference between the neighbouring classifiers.

B. THE DEEP TRANSFER LEARNING STRATEGY IMPROVES PREDICTIVE PERFORMANCE

It is known that a predictor's performance is positively associated with the training data size [22]. The mediocre performance of CNN_{OH} may be due to the limited number of neddylation sites. In contrast with unmodified lysine sites, neddylation sites are similar to lysine sites with other PTM annotations. For example, half of the identified neddylation sites are annotated with other modifications. Because of the availability of a massive number of lysine PTM sites, we attempted to introduce the transfer learning strategy into the CNN_{OH} architecture.

TABLE 1. Performance comparisons of different models for NEDD site prediction*.

	Model	AUC	Sn	Sp	ACC	MCC
Ten-fold Cross-validation #1	SVM _{OH}	0.639±0.035	0.506±0.051	0.687±0.002	0.596±0.029	0.196±0.050
	SVM _{EAAc}	0.673±0.028	0.571±0.053	0.686±0.002	0.628±0.027	0.259±0.053
	RF _{OH}	0.666±0.033	0.558±0.065	0.670±0.016	0.614±0.031	0.229±0.061
	RF _{EAAc}	0.736±0.039	0.666±0.066	0.664±0.015	0.665±0.034	0.330±0.069
	LSTM _{WE}	0.733±0.028	0.645±0.052	0.689±0.002	0.667±0.025	0.334±0.050
	LSTM _{OH}	0.721±0.021	0.621±0.044	0.689±0.001	0.655±0.022	0.311±0.043
	CNN _{WE}	0.742±0.027	0.661±0.065	0.689±0.001	0.675±0.032	0.351±0.064
	CNN _{OH}	0.767±0.024	0.712±0.047	0.689±0.001	0.700±0.023	0.402±0.047
Independent test #1	SVM _{OH}	0.663±0.015	0.549±0.045	0.687±0.000	0.618±0.022	0.238±0.044
	SVM _{EAAc}	0.683±0.009	0.569±0.035	0.687±0.000	0.628±0.017	0.258±0.034
	RF _{OH}	0.683±0.018	0.583±0.041	0.670±0.015	0.627±0.020	0.255±0.040
	RF _{EAAc}	0.746±0.019	0.673±0.047	0.670±0.013	0.671±0.024	0.344±0.049
	LSTM _{WE}	0.749±0.021	0.644±0.039	0.687±0.000	0.666±0.019	0.332±0.038
	LSTM _{OH}	0.735±0.030	0.646±0.061	0.687±0.000	0.666±0.030	0.334±0.060
	CNN _{WE}	0.732±0.032	0.634±0.069	0.687±0.000	0.661±0.034	0.323±0.068
	CNN _{OH}	0.771±0.011	0.712±0.033	0.687±0.000	0.700±0.016	0.400±0.033

*The abbreviations (e.g., CNN, OH and Sn) are described in the “Materials and Methods” section. For example, the CNN classifiers with the one-hot encoding approach are named CNN_{OH}. Ten models were constructed using ten different validation data sets in Ten-fold Cross-validation #1. The ten models' average performance and standard deviation were calculated for Cross-validation dataset #1 and for Independent test dataset #1, respectively.

We constructed the source model using the human lysine modification sites from the PLMD database [40]. We collected 80,027 lysine modification sites from 9381 human proteins after CD-HIT clustering with a sequence identity of 40% as the threshold. A total of 79,369 modification sites were retained after removing the neddylation datasets. Similarly, an exact number (79,369) of lysine non-modification sites was randomly selected from the same proteins as the negatives. Both lysine modification and non-modification sites constituted our PTM dataset (Figure 3B). Note that this dataset did not contain any data from the neddylation-related datasets (including the CV and independent test datasets), thereby avoiding potential interference in the model evaluation. The lysine PTM dataset was randomly separated into a training dataset with 141,754 samples (70,877 modified/unmodified samples) and a test dataset with 14,174 samples (7087 modified/unmodified samples). We built and tested the CNN_{OH}-based source model using the training and test datasets, respectively. The source model had an AUC value of 0.729. We further examined the performance of the source model for predicting neddylation sites using Independent test dataset #1 (Figure 3B). Its AUC value (0.768) is larger than 0.729, suggesting that neddylation may have distinctive characteristics compared with other lysine PTM types. This AUC value is similar to that (0.771) of the CNN_{OH} model constructed using CV neddylation dataset #1 (Table 2). To some extent, it indicates the resemblance between neddylation and other PTM types.

In the fine-tuning step, the weights (knowledge) learned by the source model were transferred to the target model that was later optimized using newly labelled data. The target

models were trained and evaluated in this study using the neddylation CV dataset #1. We conducted four target models with various combinations of frozen and unfrozen layers (Figure 6). The weights of the frozen layers in the source model were transferred to the target model without change whereas the weights of the unfrozen layers could be modified. The first target model (T1) contained the frozen convolution and unfrozen fully connected layers (Figure 6A). From the second T2 to the fourth T4, the convolutional sublayers were unfrozen sequentially in reverse order, along with the constant unfrozen fully connected layer (Figure 6B–6D). Their AUC values increased with the number of unfrozen layers, from T1 as 0.804 to T4 as 0.818 (Table 2). A similar observation was also made using Independent test dataset #1 (Table 2) for model evaluation. Therefore, T4, with unfrozen, fully connected and convolution layers, had the best performance.

C. CONSTRUCTION OF SPECIFIC MODELS FOR RECOGNITION OF NEDDYLATION FROM LYSINE PTM SITES

Because half of the reported neddylation sites engage in *in situ* crosstalk with other PTM types, we reason that many of the known lysine PTM sites may engage in crosstalk with neddylation. As a large number (>100,000) of lysine PTM sites has been experimentally identified [40], it is necessary to differentiate the neddylation sites from lysine PTM sites. Accordingly, we constructed the related independent test dataset (#2), where 112 neddylation peptides were collected from Independent test dataset #1 and 112 lysine PTM peptides were randomly selected from the test dataset of the lysine PTM dataset (Figure 3B).

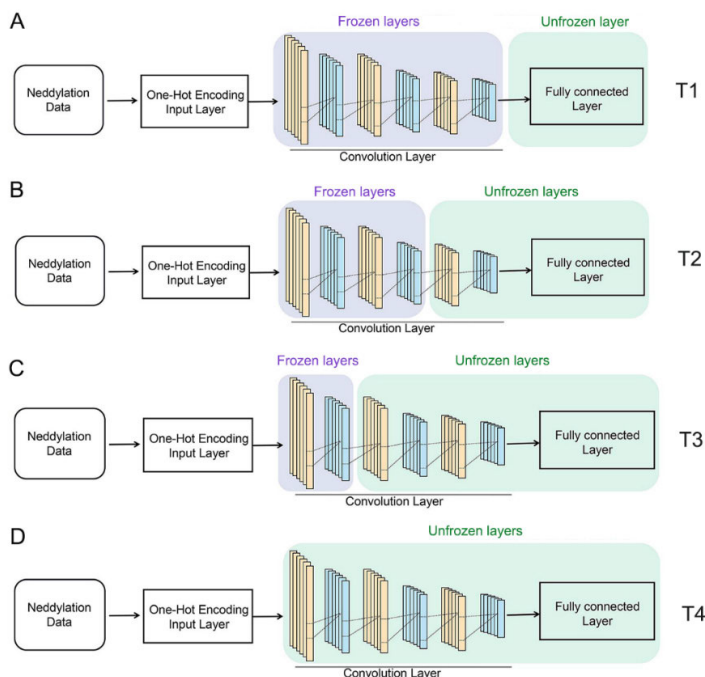


FIGURE 6. The architectures of four target models with combinations of various frozen and unfrozen layers: (A) The target model with a fully connected layer unfrozen; (B) The model with a fully connected layer and the last convolutional sublayer unfrozen; (C) The model with a fully connected layer and the last two convolutional sublayers unfrozen; (D) The model with convolutional and fully connected layers unfrozen.

TABLE 2. The performance (AUC values) of the CNN_{OH} model, the CNN_{OH}-based source model and target models with various combinations of frozen and unfrozen layers*.

Dataset	CNN _{OH}	Source model	Target models			
			All convolutional layers frozen	The first two convolutional layers frozen	The first convolution layer frozen	All convolutional layers unfrozen
Ten-fold Cross-validation #1	0.767±0.024	0.790±0.020	0.804±0.034	0.806±0.020	0.812±0.032	0.818±0.020
Independent test dataset #1	0.771±0.011	0.768	0.800±0.010	0.805±0.011	0.816±0.012	0.818±0.009
Independent test dataset #2	0.652±0.014	0.471	0.588±0.009	0.633±0.008	0.650±0.013	0.652±0.011

* CNN_{OH} was constructed using Cross-validation dataset #1. The source model was built using the training dataset of the lysine PTM dataset. The four target models were fine-tuned using Cross-validation dataset #1.

We examined the performance of the CNN_{OH} model, the source model and the four target models built using Independent test dataset #2. Out of the six target models, the CNN_{OH} model and T4 had the greatest AUC value of 0.652 (Table 2). Because of their mediocre performance, we attempted to build new models with superior performance. Accordingly, we constructed CV dataset #2 for model training and assessment, which included the 1124 neddylation sites from CV dataset #1 and 1124 PTM sites from the test dataset of the lysine PTM dataset (Figure 3).

We constructed and examined the CNN_{OH} model using CV dataset #2. We also constructed and evaluated four target models (T1–T4) based on the source model developed

above (Figure 6). The AUC value of CNN_{OH} was 0.659 and increased gradually from 0.669 to 0.698 for T1 to T4 in CV dataset #2 (Table 3). Therefore, T4 had the best performance among these five models. T4 also showed superior performance in Independent test #2 (Table 3). In summary, T4 could effectively recognize neddylation from lysine PTM sites.

Furthermore, we compared CNN_{OH} and T4 using four more metrics (*i.e.*, ACC, Sn, Sp and MCC). Since these metrics are related, we set the Sp value as the same (around 0.689) and used the remaining metrics to compare prediction performance (Table 4). T4 had larger Sn, ACC and MCC values than did CNN_{OH} in terms of the ten-fold cross-validation datasets (#1 and #2) and the independent test datasets (#1 and #2).

TABLE 3. The performance (AUC values) of the source model CNN_{OH} and target models with various combinations of frozen and unfrozen layers*.

Dataset	CNN _{OH}	Source model	Target models			
			All convolutional layers frozen	The first two convolutional layers frozen	The first convolution layer frozen	All convolutional layers unfrozen
Ten-fold Cross-validation #2	0.659±0.026	0.545±0.035	0.669±0.025	0.680±0.029	0.688±0.025	0.698±0.033
Independent test dataset #2	0.649±0.025	0.471	0.688±0.006	0.671±0.012	0.682±0.017	0.687±0.007
Independent test dataset #1	0.697±0.027	0.768	0.743±0.016	0.750±0.013	0.755±0.016	0.697±0.027

* CNN_{OH} was constructed using Cross-validation dataset #2. The source model was built using the training dataset of the lysine PTM dataset. The four target models were fine-tuned using Cross-validation dataset #2.

TABLE 4. Performance comparison of CNN_{OH} and target model T4.

Dataset	Model	Sn	Sp	ACC	MCC
Ten-fold Cross-validation #1	CNN _{OH}	0.712±0.047	0.689±0.001	0.700±0.023	0.402±0.047
	T4	0.775±0.044	0.689±0.001	0.732±0.022	0.467±0.047
Independent test dataset #1	CNN _{OH}	0.712±0.033	0.687±0.000	0.700±0.016	0.400±0.033
	T4	0.788±0.031	0.687±0.000	0.737±0.015	0.478±0.033
Ten-fold Cross-validation #2	CNN _{OH}	0.541±0.049	0.689±0.001	0.615±0.024	0.233±0.047
	T4	0.580±0.068	0.689±0.001	0.634±0.034	0.270±0.066
Independent test dataset #2	CNN _{OH}	0.495±0.052	0.687±0.000	0.591±0.026	0.185±0.051
	T4	0.538±0.023	0.687±0.000	0.612±0.011	0.228±0.023

For example, in Independent test dataset #1, the ACC value of CNN_{OH} is 0.712, and that of T4 is 0.788 (p -value=0.0020, student's t-test). In summary, T4 had a performance superior to that of CNN_{OH}.

To comprehensively evaluate our models, we changed the ratio of the cross-validation and the independent test datasets from 10:1 to 5:1, reconstructed the DTL model and assessed its performance. We found that this change did not affect the model's performance. For instance, the AUC values, before and after the change, are similar for CV dataset #1 (p -value = 0.495, student's t-test) and for Independent test dataset #1 (p -value = 0.200, student's t-test). Therefore, our DTL model is robust.

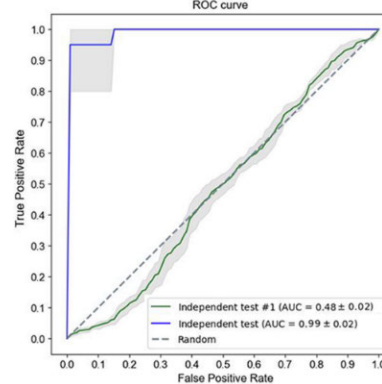
D. MODEL EVALUATION BY COMPARISON OR PRACTICAL APPLICATION

We compared our model with two reported classifiers (*i.e.*, NeddPred and NedlyPreddy). The two classifiers were based on the small dataset (*i.e.*, 51 neddylation sites). Because the characteristics of the neddylation sites from this small dataset are different from those of the neddylation sites from our dataset (Figure 1A), we were suspicious of the performances of these models for predicting the newly identified neddylation sites. As we failed to reproduce the classifiers, we constructed a model to represent them. The representative model was SVM_{EAAAC}, which performed similarly to the previous models based on the small dataset (Figure 7A). The performance (AUC=0.48) of SVM_{EAAAC} using Independent test #1 was similar to that of random prediction (Figure 7B),

A

Method	AUC	Sn	Sp	ACC
Ten-fold NedlyPreddy	0.95	0.76	0.91	0.91
Cross-validation NeddPred	0.97	0.79	0.97	0.97
SVM _{EAAAC}	0.94±0.10	0.77±0.40	0.96±0.01	0.95±0.01
Independent test NedlyPreddy	0.80	0.64	0.91	0.90
NeddPred	1	1	0.95	0.95
SVM _{EAAAC}	0.99±0.02	0.95±0.15	0.95±0.01	0.95±0.01

B

**FIGURE 7.** Evaluation of the reported classifiers: (A) Performance comparison of NedlyPreddy, NeddPred and SVM_{EAAAC} using the previous small dataset; (B) Prediction performance of SVM_{EAAAC} using Independent test #1 of this study and an independent test collected in the previous small dataset.

suggesting that the reported models fail to predict newly identified neddylation sites. Furthermore, as the representative SVM_{EAAAC} performed more poorly than CNN_{OH}, developed using our dataset (Table 1), the previously reported models might not perform well either.

Apriori is an algorithm for frequent item set mining and association rule learning over relational databases [43], [44].

We used the Apriori algorithm to explore the relationship between elements in the neddylation samples of Dataset #1. We set a support level of 25% (or 45%) and a confidence level of 90% (or 99%) [43], [44], but we did not find any association rules. This indicates that there is no strong association within the dataset. When we reduced the support level to 2.5% and the confidence level to 15%, we found four association rules: {L@P-1, G@P1}, {K@P3, K@P5}, {K@P-11, K@P3} and {L@P-1, E@P16} (See Table S1 for details). In Dataset #1, 7% (=86/1236) of the positives and 4.4% (=55/1236) negatives meet one of the four rules.

We used the Apriori algorithm to predict neddylation sites, where the samples meeting these rules were predicted as positive and the rest were considered negative. As a result, the Specificity value was 0.96, Sensitivity was 0.07 and Accuracy was 0.51. We evaluated the performance of the T4 model with a fixed Specificity of 0.96. Its Sensitivity (0.29) was larger than that (0.07) of the Apriori algorithm, and its Accuracy (0.62) was larger than that (0.51) of the Apriori algorithm as well. Therefore, the T4 model compares favourably to the Apriori algorithm.

We further attempted to evaluate the T4 model's performance in a real-world situation. As we did not find new neddylation data in the literature, we could not use them as a benchmarking dataset. Nevertheless, we examined T4 by separating our dataset into two sub-datasets based on their different origins: a large one with 1101 neddylation sites identified from the HCT-116 cell line [11] and a small one with 607 neddylation sites reported from the HET-293 cell line [10]. Because the datasets were generated from different cell lines by different research groups, they were truly independent. We used the large dataset to reconstruct our model and evaluated its performance using the small dataset. Its AUC value was 0.735, which is significantly larger than random prediction (AUC=0.5), indicating that our model captured the general characteristics of neddylation and has good predictive ability. Meanwhile, we admit that data from different experiments are subject to variation, and more data are required to improve the model performance.

IV. DISCUSSION

In this study, we collected newly identified human neddylation sites (>1000) and constructed several models based on different algorithms and distinct encoding features. The CNN_{OH} model showed the best performance. Nevertheless, its performance was lower than expected due to data limitations. To increase its performance, we introduced information on a massive number of lysine PTM sites using the DTL strategy. Specifically, the source model was built using the lysine PTM sites and then fine-tuned as target models using the known neddylation sites. The DTL model T4 showed superior performance, in which all the layers were unfrozen during the fine-tuning step. This phenomenon was also discovered in DTL-DephosSite, which was developed for

TABLE 5. Performance (AUC) of the target models with different source models trained using various lysine PTM types.

PTM types used for source model training*	Ten-fold Cross-validation #1	Independent test #1
A mixture of PTM types	0.793±0.028	0.798±0.014
Ubiquitination	0.795±0.032	0.804±0.011
Acetylation	0.780±0.026	0.778±0.015
SUMOylation	0.779±0.029	0.792±0.010

* In each training dataset, the number of PTM sites was fixed as 5674.

predicting dephosphorylation sites [45]. These two models differ from general DTL models with good performance where frozen and unfrozen layers are combined [46], [47]. Whether this phenomenon is common in the predictors of PTM sites remains to be investigated.

We compared DTL model T4 and the CNN_{OH} model to determine why the former performed better. The differences between them are the weights of the convolution and fully connected layers before training. The initial weights in the CNN_{OH} model are random, whereas those in T4 are pre-trained from the source model. In other words, the pre-trained weights captured the information of lysine PTMs that contain neddylation characteristics and boosted the prediction performance through knowledge transfer. Therefore, the pre-trained weights demonstrated advantages over the random weights.

The massive amount of human lysine PTM data was essential for constructing the DTL model for neddylation site prediction. The data were mainly contributed by three PTM types: SUMOylation (5674 sites), ubiquitination (23,413 sites) and acetylation (60,342 sites). We compared the three PTM types to explore which would be best for constructing the source model. To fairly compare them, we fixed the number of modified sites as 5674 (the number of reported SUMOylation sites in PLMD). Accordingly, we randomly selected 5674 lysine PTM sites, SUMOylation sites, ubiquitination sites and acetylation sites as positive samples, respectively. Negative samples were selected using the same procedure as listed above. Based on each PTM type, we constructed the base model and target model T4. The four target models performed similarly in ten-fold CV and independent tests (Table 5). Moreover, Their AUC values were smaller than that (0.818) of the target model, where the massive lysine PTM training dataset (including 70,877 PTM sites) was used for constructing the source model (p -value < 0.030, student's t -test). Therefore, the performance of the target model is positively related to the number of PTM sites used for source model construction.

Our developed DTL model demonstrated superior performance in predicting neddylation sites. This observation shows the advantages of the DTL algorithm for modelling based on small data sets. It is also consistent with the application of the DTL strategy that reuses a previously learned model on a new project with a small amount of data. The DTL strategy may be widely used in the PTM field as many PTM

types are identified with limited modification data, such as lysine and arginine phosphorylation [48].

V. CONCLUSION

This study demonstrates the advantage of the DTL strategy for constructing a neddylation prediction model. Although the identified neddylation sites are limited, integrating a massive number of lysine PTM sites improved the prediction performance. The DTL strategy is particularly suitable for PTM types with limited identified sites. Therefore, we would like to see it widely used on the newly discovered modifier types, such as lysine and arginine phosphorylation.

AUTHOR CONTRIBUTIONS

Lei Li conceived of this project; Deli Xu and Yafei Zhu constructed the algorithms under the supervision of Yu Chen, Yuhai Liu, Yang Zou, and Lei Li; Yafei Zhu generated the online website; Deli Xu, Yang Zou, and Lei Li wrote the manuscript. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

- [1] R. I. Enchev, B. A. Schulman, and M. Peter, "Protein neddylation: Beyond cullin-RING ligases," *Nature Rev. Mol. Cell Biol.*, vol. 16, no. 1, pp. 30–44, 2015.
- [2] Y. Chen, R. L. Neve, and H. Liu, "Neddylation dysfunction in Alzheimer's disease," *J. Cellular Mol. Med.*, vol. 16, no. 11, pp. 91–2583, Nov. 2012.
- [3] Y. S. Choo, G. Vogler, D. Wang, S. Kalvakuri, A. Iliuk, W. A. Tao, R. Bodmer, and Z. Zhang, "Regulation of parkin and PINK1 by neddylation," *Hum. Mol. Genet.*, vol. 21, no. 11, pp. 2514–2523, Jun. 2012.
- [4] Y. Wang, Z. Luo, Y. Pan, W. Wang, X. Zhou, L. S. Jeong, Y. Chu, J. Liu, and L. Jia, "Targeting protein neddylation with an NEDD8-activating enzyme inhibitor MLN4924 induced apoptosis or senescence in human lymphoma cells," *Cancer Biol. Therapy*, vol. 16, no. 3, pp. 420–429, Mar. 2015.
- [5] L. Zhou, W. Zhang, Y. Sun, and L. Jia, "Protein neddylation and its alterations in human cancers for targeted therapy," *Cellular Signalling*, vol. 44, pp. 92–102, Apr. 2018.
- [6] Y. Cui, Z. Chen, B. Pan, T. Chen, H. Ding, Q. Li, L. Wan, G. Luo, L. Sun, C. Ding, J. Yang, X. Tong, and J. Zhao, "Neddylation pattern indicates tumor microenvironment characterization and predicts prognosis in lung adenocarcinoma," *Frontiers Cell Develop. Biol.*, vol. 10, Sep. 2022, Art. no. 979262.
- [7] X. He, A. Zhu, J. Feng, and X. Wang, "Role of neddylation in neurological development and diseases," *Biotechnol. Appl. Biochemistry*, vol. 69, no. 1, pp. 330–341, Feb. 2022.
- [8] P. Olaizola, P. Y. Lee-Law, M. G. Fernandez-Barena, L. Alvarez, M. Cadamuro, M. Azkargorta, C. J. O'Rourke, F. J. Caballero-Camino, I. Olaizola, R. I. Macias, and J. J. Marin, "Targeting NAE1-mediated protein hyper-neddylation halts cholangiocarcinogenesis and impacts on tumor-stroma crosstalk in experimental models," *J. Hepatology*, vol. 77, no. 1, pp. 177–190, Jul. 2022.
- [9] J. Xu, Z. Li, Q. Zhuo, Z. Ye, G. Fan, H. Gao, S. Ji, X. Yu, X. Xu, W. Liu, and W. Xu, "Pevonedistat suppresses pancreatic cancer growth via inactivation of the neddylation pathway," *Frontiers Oncol.*, vol. 12, Jan. 2022, Art. no. 822039.
- [10] A. M. Vogl, L. Phu, R. Becerra, S. A. Giusti, E. Verschuere, T. B. Hinkle, M. D. Bordenave, M. Adrian, A. Heidersbach, P. Yankilevich, F. D. Stefani, W. Wurst, C. C. Hoogenraad, D. S. Kirkpatrick, D. Refojo, and M. Sheng, "Global site-specific neddylation profiling reveals that neddylated cofilin regulates actin dynamics," *Nature Struct. Mol. Biol.*, vol. 27, no. 2, pp. 210–220, Feb. 2020.
- [11] S. Lobato-Gil, J. B. Heidelberger, C. Maghames, A. Bailly, L. Brunello, M. S. Rodriguez, P. Beli, and D. P. Xirodimas, "Proteome-wide identification of NEDD8 modification sites reveals distinct proteomes for canonical and atypical neddylation," *Cell Rep.*, vol. 34, no. 3, Jan. 2021, Art. no. 108635.
- [12] A. S. Yavuz, N. B. Sözer, and O. U. Sezerman, "Prediction of neddylation sites from protein sequences and sequence-derived properties," *BMC Bioinf.*, vol. 16, no. 18, pp. 1–11, Dec. 2015.
- [13] Z. Ju and S.-Y. Wang, "Identify lysine neddylation sites using bi-profile Bayes feature extraction via the Chou's 5-steps rule and general pseudo components," *Current Genomics*, vol. 20, no. 8, pp. 592–601, Jan. 2020.
- [14] V. Vacic, L. M. Iakoucheva, and P. Radivojac, "Two Sample Logo: A graphical representation of the differences between two sets of sequence alignments," *Bioinformatics*, vol. 22, no. 12, pp. 1536–1537, Jun. 2006.
- [15] S. Zhang, J. Zhou, H. Hu, H. Gong, L. Chen, C. Cheng, and J. Zeng, "A deep learning framework for modeling structural features of RNA-binding protein targets," *Nucleic Acids Res.*, vol. 44, no. 4, p. 32, Feb. 2016.
- [16] X. Lyu, S. Li, C. Jiang, N. He, Z. Chen, Y. Zou, and L. Li, "DeepCSO: A deep-learning network approach to predicting cysteine S-Sulphenylation sites," *Frontiers Cell Develop. Biol.*, vol. 8, Dec. 2020, Art. no. 594587.
- [17] L. Zhang, Y. Zou, N. He, Y. Chen, Z. Chen, and L. Li, "DeepKhib: A deep-learning framework for lysine 2-Hydroxyisobutyrylation sites prediction," *Frontiers Cell Develop. Biol.*, vol. 8, Sep. 2020, Art. no. 580217.
- [18] X. Wei, Y. Sha, Y. Zhao, N. He, and L. Li, "DeepKcrot: A deep-learning architecture for general and species-specific lysine crotonylation site prediction," *IEEE Access*, vol. 9, pp. 49504–49513, 2021.
- [19] Y. Sha, C. Ma, X. Wei, Y. Liu, Y. Chen, and L. Li, "DeepSADPr: A hybrid-learning architecture for serine ADP-ribosylation site prediction," *Methods*, vol. 203, pp. 575–583, Jul. 2022.
- [20] W. Li and A. Godzik, "CD-HIT: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.
- [21] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT suite: A web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, Mar. 2010.
- [22] Z. Chen, N. He, Y. Huang, W. T. Qin, X. Liu, and L. Li, "Integration of a deep learning classifier with a random forest approach for predicting malonylation sites," *Genomics, Proteomics Bioinf.*, vol. 16, no. 6, pp. 451–459, Dec. 2018.
- [23] G. Zou, Y. Zou, C. Ma, J. Zhao, and L. Li, "Development of an experiment-split method for benchmarking the generalization of a PTM site predictor: Lysine methylome as an example," *PLOS Comput. Biol.*, vol. 17, no. 12, Dec. 2021, Art. no. e1009682.
- [24] Z. Chen, P. Zhao, F. Li, A. Leier, T. T. Marquez-Lago, Y. Wang, G. I. Webb, A. I. Smith, R. J. Daly, K.-C. Chou, and J. Song, "IFeature: A Python package and web server for features extraction and selection from protein and peptide sequences," *Bioinformatics*, vol. 34, no. 14, pp. 2499–2502, Jul. 2018.
- [25] L. Ge and T. Moh, "Improving text classification with word embedding," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1796–1805.
- [26] D.-L. Wang, S. Zeng, C.-H. Xu, W.-R. Qiu, Y.-C. 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, Aug. 2017.
- [27] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Special Topics*, vol. 228, no. 10, pp. 2313–2324, Oct. 2019.
- [28] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
- [29] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, USA, Jun. 2007, pp. 193–200.
- [30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, Dec. 2014, pp. 3320–3328.
- [31] K. Weiss, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.
- [32] M. Bernico, Y. Li, and D. Zhang, "Investigating the impact of data, volume and domain similarity on transfer learning applications," in *Proc. Future Technol. Conf.*, 2019, pp. 53–62.
- [33] F. Yu, X. Xiu, and Y. Li, "A survey on deep transfer learning and beyond," *Mathematics*, vol. 10, no. 19, p. 3619, Oct. 2022.

- [34] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.
- [35] H. K. Jeon, S. Kim, J. Edwin, and C. S. Yang, "Sea fog identification from GOCI images using CNN transfer learning models," *Electronics*, vol. 9, no. 2, pp. 311–320, 2020.
- [36] J.-S. Han and K.-C. Kwak, "Image classification using convolutional neural network and extreme learning machine classifier based on ReLU function," *J. Korean Inst. Inf. Technol.*, vol. 15, no. 2, pp. 15–23, Feb. 2017.
- [37] S. De, A. Mukherjee, and E. Ullah, "Convergence guarantees for RMSProp and ADAM in non-convex optimization and an empirical comparison to Nesterov acceleration," 2018, *arXiv:1807.06766*.
- [38] S. A. Wagner, P. Beli, B. T. Weinert, M. L. Nielsen, J. Cox, M. Mann, and C. Choudhary, "A proteome-wide, quantitative survey of in vivo ubiquitylation sites reveals widespread regulatory roles," *Mol. Cellular Proteomics*, vol. 10, no. 10, Oct. 2011, Art. no. M111.013284.
- [39] I. A. Hendriks, R. C. D'Souza, B. Yang, M. V.-D. Vries, M. Mann, and A. C. Vertegaal, "Uncovering global SUMOylation signaling networks in a site-specific manner," *Nature Struct. Mol. Biol.*, vol. 21, no. 10, pp. 927–937, Oct. 2014.
- [40] H. Xu, J. Zhou, S. Lin, W. Deng, Y. Zhang, and Y. Xue, "PLMD: An updated data resource of protein lysine modifications," *J. Genet. Genomics*, vol. 44, no. 5, pp. 243–250, May 2017.
- [41] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8614–8618.
- [42] M. Tahir, H. Tayara, and K. T. Chong, "IPseU-CNN: Identifying RNA pseudouridine sites using convolutional neural networks," *Mol. Therapy-Nucleic Acids*, vol. 16, pp. 463–470, Jun. 2019.
- [43] M. M. Hassan, A. Karim, S. Mollick, S. Azam, E. Ignatious, and F. A. Haque, "An apriori algorithm-based association rule analysis to detect human suicidal behaviour," *Proc. Comput. Sci.*, vol. 219, pp. 1279–1288, 2023.
- [44] M. M. Hassan, S. Zaman, S. Mollick, M. M. Hassan, M. Raihan, C. Kaushal, and R. Bhardwaj, "An efficient apriori algorithm for frequent pattern in human intoxication data," *Innov. Syst. Softw. Eng.*, vol. 19, no. 1, pp. 61–69, Mar. 2023.
- [45] M. Chaudhari, N. Thapa, H. Ismail, S. Chopade, D. Caragea, M. Köhn, R. H. Newman, and D. B. Kc, "DTL-DephosSite: Deep transfer learning based approach to predict dephosphorylation sites," *Frontiers Cell Develop. Biol.*, vol. 9, Jun. 2021, Art. no. 662983.
- [46] E. Wu, L. M. Hadjiiski, R. K. Samala, H.-P. Chan, K. H. Cha, C. Richter, R. H. Cohan, E. M. Caoili, C. Paramagul, A. Alva, and A. Z. Weizer, "Deep learning approach for assessment of bladder cancer treatment response," *Tomography*, vol. 5, no. 1, pp. 201–208, Mar. 2019.
- [47] A. Mehmood, S. Yang, Z. Feng, M. Wang, A. S. Ahmad, R. Khan, M. Maqsood, and M. Yaqub, "A transfer learning approach for early diagnosis of Alzheimer's disease on MRI images," *Neuroscience*, vol. 460, pp. 43–52, Apr. 2021.
- [48] Y. Hu, B. Jiang, Y. Weng, Z. Sui, B. Zhao, Y. Chen, L. Liu, Q. Wu, Z. Liang, L. Zhang, and Y. Zhang, "Bis(zinc(II)-dipicolylamine)-functionalized sub-2 μm core-shell microspheres for the analysis of N-phosphoproteome," *Nature Commun.*, vol. 11, no. 1, p. 6226, Dec. 2020.



YAFEI ZHU received the B.Eng. degree from the School of Computer Science and Engineering, Linyi University, China, in 2019. He is currently pursuing the master's degree with the School of Data Science and Software Engineering, Qingdao University, China. His research interests include bioinformatics and deep learning.



QIANG XU received the master's degree from Qingdao University, in 2015. He is currently an Attending Physician with the Department of Geriatrics, The Affiliated Hospital of Qingdao University. His research interests include the diagnosis and treatment of common geriatric diseases and electrophysiology.



YUHAI LIU received the Ph.D. degree from the Ocean University of China. He is currently an Engineer with Dawning International Information Industry Company Ltd. His research interest includes artificial intelligence.



YU CHEN received the Ph.D. degree from Inha University, South Korea. He is currently the Director of the Department of Software Engineering, College of Computer Science and Technology, Qingdao University. His research interests include deep learning and bioinformatics.



YANG ZOU received the Ph.D. degree from East China Normal University. He is currently a Lecturer with the Faculty of Biomedical and Rehabilitation Engineering, University of Health and Rehabilitation Sciences, Qingdao, China. His research interests include deep learning, bioinformatics, and proteomics.



DELIXU received the B.Eng. degree from the College of Medical Information Engineering, Shandong First Medical University, China, in 2019. She is currently pursuing the master's degree with the School of Data Science and Software Engineering, Qingdao University, China. Her research interests include bioinformatics and deep learning.



LEI LI received the Ph.D. degree from Nanyang Technological University, Singapore, in 2006. He is currently a Professor with the Faculty of Biomedical and Rehabilitation Engineering, University of Health and Rehabilitation Sciences, Qingdao, China. His research interests include deep learning, bioinformatics, systems biology, and proteomics.