IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 21, 2020, accepted December 27, 2020, date of publication January 5, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049142

RecSNO: Prediction of Protein S-Nitrosylation Sites Using a Recurrent Neural Network

ARSLAN SIRAJ[1], TUVSHINBAYAR CHANTSALNYAM[1], HILAL TAYARA[2], AND KIL TO CHONG[1,3]

[1]Department of Electronics and Information Engineering, Jeonbuk National University, Jeonju 54896, South Korea
[2]School of International Engineering and Science, Jeonbuk National University, Jeonju 54896, South Korea
[3]Advanced Electronics and Information Research Center, Jeonbuk National University, Jeonju 54896, South Korea

Corresponding authors: Kil To Chong (kitchong@jbnu.ac.kr) and Hilal Tayara (hilaltayara@jbnu.ac.kr)

This work was supported in part by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), through the Ministry of Trade, Industry and Energy, Republic of Korea, under Grant 20204010600470, and in part by the Brain Research Program of the National Research Foundation (NRF) funded by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant NRF-2017M3C7A1044816 and Grant 2020R1A2C2005612.

ABSTRACT S-Nitrosylation modification is one of the most important post-translational modifications; it plays a critical role in a vast variety of biological processes and is related to various diseases. Identification of S-Nitrosylation sites in proteins is crucial for understanding and controlling basic biological processes. The conventional experimental identification methods are laborious and cost in-efficient. To overcome these issues, computational biological approaches are under consideration, including use of machine learning and deep learning algorithms. All existing S-Nitrosylation predictors use the handicraft feature extraction method and could be improved upon. We propose an end-to-end deep learning based S-Nitrosylation site predictor with an embedded layer and bidirectional long short-term memory. The proposed method uses protein sequences as inputs without any need for complex features interventions. This sequence-based protein prediction method is associated with a significant improvement in identification of S-Nitrosylation sites. More specifically, the best prediction of the proposed architecture showed an improvement of in MCC 3% on 5-fold cross validation and 5% on an independent test dataset. Finally, the user-friendly publicly available webserver is accessible at http://nsclbio.jbnu.ac.kr/tools/RecSNO/.

INDEX TERMS Post-translational modification, s-nitrosylation, deep learning, BiLSTM.

I. INTRODUCTION

Protein post-translational modifications (PTMs) are important cellular regulatory processes that happen after protein synthesis. PTMs play a very important role in protein mutations, thus altering the regulation of many cellular functions [1] and the physical and chemical properties of proteins. The PTM process begins when a modification group is added to one or more amino acids [2] which alters the properties of a protein. S-Nitrosylation(SNO) is one of the most important and universal ubiquitous PTMs [3]. SNO involves a covalent interaction of nitric oxide (NO) with the thiol group of a cysteine residue [4]–[7]. NO plays an important role in the cardiovascular system [8] and is considered a good source of NO bioactivity [9]. Various studies have proposed that SNO can modify protein stability [9], trafficking [10], [11], and activity [12]; studies have also shown that it plays an important role in a variety of biological processes including transcriptional regulation [9], apoptosis [12], cell death [13], cell signaling [14], redox signaling [15], the immune response [16], and chromatin remodeling [17]. SNO has also been implicated in a wide range of human disease states [5], such as cancer [18], amyotrophic lateral sclerosis (ALS) [19], chronic renal failure [20], cardiovascular disease [21], age-related diseases [22] and neurodegenerative diseases [23] like Alzheimer's [24] and Parkinson's [12], [25]. Recent studies suggest that SNO is a promising target for therapeutic against cancer and some neurodegenerative diseases [18], [26]–[28]. Therefore, protein SNO site prediction is very important and helpful for drug development [28]–[30] and understanding of basic biological processes [9], [12], [14], [17]. Protein SNO sites have been identified by many conventional experimental techniques, including BST

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Salehzadeh-Yazdi.

(biotin switch assay) [31], SNO-RAC (SNO-resin assisted capture) [32], and SNOSID (SNO-Cys site identification) [33], [34]. BST comprises three steps [35] and has been successfully used to predict a large number of S-nitrosylated proteins in different species, such as H.sapiens [36], A. thaliana [37], and M.musculus [38]. SNO-RAC is a BST-based methodology that combines the reduction, labeling, and pull-down steps through of thiol-reactivity [32]; and is used to inspect the process of S-nitrosylation/denitrosylation in intact cells. SNOSID has also been used to determine the locations of SNO sites in MS-derived data [39]. However, these large scale experimental screening techniques for protein SNO site detection are time-consuming, laborious, and economically costly [9], [12]. So, it is necessary to invest in options that can be used to screen proteins for potential PTM SNO sites in a less cost and time effective manner. In recent years, machine learning and deep learning have begun to play a vital role in the computational prediction of protein SNO sites [8], [40]–[43].

Several machine learning based predictors have been developed for predictions of SNO site identifications; some of these are GPS-SNO [8], SNOSite [43], iSNO-PseAAC [42], and preSNO [40]. The GPS-SNO predictor is a group-based prediction (GPS) algorithm developed with a training dataset that consisting of 504 experimentally verified SNO sites in 327 unique proteins [8]. The iSNO-PseAAC predictor is a SVM-based and uses, a training dataset that consists of 731 SNO sites in 438 proteins [42]. The SNOSite predictor,which is also based on SVM, is applied to generate a predictive model for each maximal dependence decomposition (MDD)-clustered motif, and was developed using a training dataset that consisting of 586 sites in 384 unique proteins [43]. Xie *et al.* [41], developed the DeepNitro predictor based on a deep learning methodology; the system applies fully-connected layers on the concatenated features of amino acid pair composition and a position specific scoring matrix (PSSM). DeepNitro was the first time predictor to use a large dataset, which was collected from the scientific literature and, contained 4762 SNO sites in 3113 unique proteins [41]. Before that point, no such dataset was available for good prediction or evaluation with an independent dataset; many samples from older datasets were verified as positive, which were then considered false-negative results in the older methods [40]. Recently, the preSNO predictor was developed, which use the same dataset as DeepNitro, by integrating SVM and random forest (RF) methods [40]. Each of the aforementioned approaches has unique benefits, and each one has played a significant role in the study of protein S-nitrosylation site forecasting. However, these models still have some complications and leave room for improvement. These existing machine learning based tools make predictions using traditional shallow machine learning methods, which fail to learn the basic biological features of protein modification due to a lack of consensus sequences [41]. These existing tools are unable to extract high level features from an input sequence and are reliant on handicraft features. By contrast, a computational architecture based on deep learning is capable of extracting the essential features of a sequence without any human intervention, leading to an accurate and robust computational model. Deep learning based models are associated with extraordinary advancements in the fields of natural language processing [44], speech recognition [45], energy load forecasting [46], image recognition [47] and computational biology [48]–[51]. However Recently, advanced machine learning techniques based on deep forest models were proposed such as DTI-CDF [52] and LMI-DForest [53]. These methods are promising and can be further studied for our task.

In recent years, deep learning (DL) based methods have been used to predict the PTM sites in cellular proteins. Typical applications include DeepSuccinylSite [54], Musit-eDeep [55], DeepRMethylSite [56], and DeepPhos [57]. In DL, a suitable raw vector is given to the architecture and transformed into highly abstract features by propagating through whole model. These approaches are an end-to-end forecasters that never require an additional feature extraction stage. The DL-based method DeepNitro was used to predict SNO sites [41]. DeepNitro used a handicraft feature extraction method for feature extraction and input these features into dense layers, which does not yield the full benefit of DL-based automatic feature extraction. In the current study, we introduce a DL-based predictor that, integrates the advantages of both an embedding layer [58] and bidirectional short-term memory (BiLSTM) [59]. The proposed predictor uses only protein sequences as inputs, resulting in real-time sequenced-based protein prediction. Also, this technique is an end-to-end that does not needs an additional feature-extracted step.The experimental results show that our approach attains better performance than previous works [8], [40]–[43].

II. BENCHMARK DATASET

In this study, we used the dataset from Xie *et al.* [41], which a high quality dataset based on extensive literature research and previously reported datasets. In the dataset, the experimentally confirmed S-nitrosylation sites taken as positive samples and all other sites are taken as negative samples. In many studies [57], [60], [61], construction of a negative dataset as above with erroneous data will affect the prediction performance [40]. We utilized the dataset from Hasan *et al.* [40], which was prepared from the above mentioned dataset containing 3113 unique proteins with 4762 SNO sites. In general, a high degree of homology in the training dataset, can cause overfitting, which may affect the generalization ability of the classifier [60]. To avoid this problem, the protein sequences were filtered with an identity cut-off of 30% using CD-HIT [62]. That is, if more than 30% of the residues in a protein sequence were the same, only one of them was retained and the others were discarded. After removal of redundant proteins, the remaining 2192 protein sequence was truncated with a centered cysteine (C) to create a fragment. The fragment length was calculated as 2Rn + 1, where Rn is equal number of residues for the left and right side.

After many trial value of Rn considered as 20. If the left or right side of the centered residue(C) was less than the Rn, then we used a pseudo-amino acid "–" to fulfill the sequence. We used the same dependent and independent datasets as Hasan *et al.* [40] to retrieve protein sequences from the uniprot database (https://www.uniprot.org/) [63].By this fragments strategy, we obtained 3734 positive and 20548 negative residues. From these fragments randomly 20% selected as the independent dataset contain 351 positive and 3168 negative sites, and for balance training of model, training dataset contains 3383 positive and 3365 negative sites [40]. All fragments used for training and independent testing are available on our web server (http://nsclbio.jbnu.ac.kr/tools/RecSNO/).

III. METHODS

Unlike traditional machine learning methods, our DL-based method reduces the need for manual feature extraction. A prerequisite for this approach is that the sequence data must be encoded in a form that is readable by our DL model. To this end, we utilized an embedding encoding technique [54], [58] and extracted features from this encoded matrix using BiLSTM. To decrease the number of dimensions, we used a max-pooling layer after feature extraction. The extracted features are further feed into the fully connected layer. Finally, SNO sites are predicted through a softmax output layer, considering categorical entropy.

A. EMBEDDING ENCODING

In the natural language processing (NLP) task, dictionary words or phrases in a sentence are mapped in to vectors of real numbers. Like Fu *et al.* [58], we regard each protein as a sentence and the residues in the protein sequence as "words". The amino acid residues and pseudo-amino acid "–" are converted into index base integers ranging from 1 to 23. These integers are then passed as inputs to the embedding layer, which maps these inputs into low dimensional vectors using a lookup table that learns from the data. The embedding weight matrix is initialized with random weights and learns better in subsequent epochs during training. The output dimension and input size are the main arguments of the embedding layer. Embedding encoding is a more beneficial method, as shown in DeepGO [64]; it has inherent advantages over one-hot encoding as it capture the semantic correlations of peptides within the protein sequence.

B. BIDIRECTIONAL LSTM

BiLSTM [65] is a type of recurrent neural network(RNN) based on LSTM [66]. The BiLSTM architecture used to process sequence data relies on two distinct hidden layers, from the forward and backward directions, to trace preceding and succeeding contextual features as shown in Figure 1. The features extracted by BiLSTM can more realistically represent the actual semantics of the text. BiLSTM provides the information gathered from past and future memory concurrently from the data because of the forward and backward hidden layers. The implementation of the forward hidden layer is the same as LSTM; this solved the problem of vanishing gradients due the use of memory blocks, which are self-connected hidden units. LSTM is used to trace long-term dependencies with the help of memory blocks and three gates. As shown in Figure 2, LSTM updates its hidden state (h_t) using historical hidden state features (h_{t-1}) with a forwarding approach and performs nonlinear transformation of the input (x_t) using the candidate cell state (K_t). The result of K_t and cell state at the last timestamp (c_{t-1}) is used to update the cell state (c_t). The LSTM gates include (1) an input gate (IG), which control the importance of new information that, will be saved to memory; (2) a forget gate (FG), which controls the importance of a memory that needs to be forgotten or remembered; and (3) output gate (OG), Which gauges new cell state and provide the output. In these gates, nonlinear activation functions are used to transform the values in the range of 0 to 1; we used a sigmoid activation function for this purpose. Finally, the result of the OG and cell state at the current time is used to update the hidden state, which is the output of LSTM. The backward hidden layer updates its hidden state using future information (h_{t+1}) with a backward approach [67].

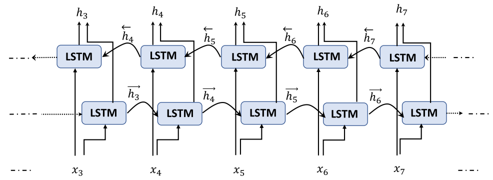


FIGURE 1. BiLSTM.

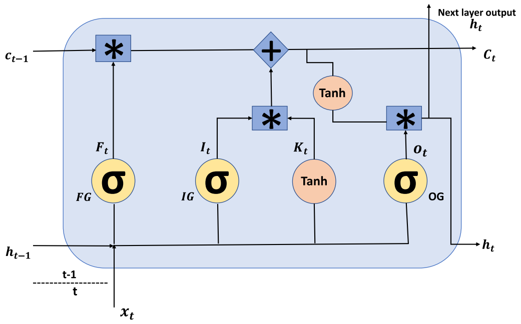


FIGURE 2. LSTM.

C. PROPOSED ARCHITECTURE

Here, we develop a DL-based classifier for SNO prediction using a combined word embedding and BiLSTM approach. This classifier contains six layers, as shown in Figure 3. These layers include: (1) an input layer, in which a residue fragment of length 41 (including the pseudo-amino acid '–') is converted via integer encoding; (2) an embedding layer, which is used to represent properties in the form of a word vector such that, every peptide in the sequence is converted into a 64-dimensional word vector; (3) two consecutive BiLSTM layers, one with 32-memory units and the other with 24. The first BiLSTM layer takes n-dimensional

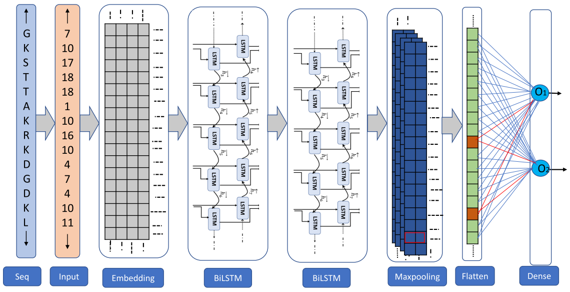


FIGURE 3. Proposed model architecture.

word vectors as input and extracts the features of those inputs. The result of the first BiLSTM layer is passed as input to the second BiLSTM layer, which extracts the features more deeply; (4) a max-pooling layer, which reduces the dimensions to half. The max-pooling layer preserves the features with maximum values in pool size; and (5) a prediction layer, which contains two neurons activated by the 'softmax' activation function and, provides a probability score for each class. We use dropout layers with different probabilities. After finding the best hyper-parameters for each layer with grid search, the hyper-parameter setting information for each layer is defined (shown in Table 1), except the given hyper-parameters values for each layer set as default. The details of the grid search hyper-parameters are given in the supplementary file (Section A).

TABLE 1. Proposed model layer details.

Layers	hyperparameter Settings	Output shape
Embedding	Input dim = 24 Output dim = 64 Input shape = (41,)	(41,64)
BiLSTM	LSTM units = 32 Kernal reg = L2($1e^{-4}$) Recurrent reg = L2($1e^{-4}$) Bias reg = L2($1e^{-4}$)	(41,64)
Dropout	Rate = 0.1	(41,64)
BiLSTM	LSTM units = 24 Kernal reg = L2($1e^{-2}$) Recurrent reg = L2($1e^{-2}$) Bias reg = L2($1e^{-2}$)	(41,48)
Dropout	Rate = 0.2	(41,48)
Max Pooling	Pool size = 2	(20,48)
Flatten	Just flatten the matrix	(960)
Dropout	Rate = 0.2	(960)

Dense	Activation = softmax
	Units = 2
		(2)

In our proposed model, We used batch size of 12 and applied Adam optimizer to our framework, which merges the dividend of both the adaptive gradient algorithm and root mean square propagation, resulting in effective training [68]. We also used early stopping to monitor validation loss with a patience of 5 for stop training because further training would increase the variance of the model and lead to overfitting. We also used a learning rate schedular after 20 epochs, which decreased the learning rate by multiplying it by (e^{-1}). The architecture was implemented using the Keras (https://keras.io/) deep learning library. Since we used softmax-based prediction, a categorical cross-entropy function was used as the loss function and the results were obtained by applying a threshold of 0.5.

D. MODEL EVALUATION AND PERFORMANCE METRICS

The present study uses stratified k-fold cross validation, the folds are generally formed in such a way as to be consisted of almost the same proportion of predictor labels as original dataset. Studies have shown that stratified cross validation generates comparative upshots with lower bias and lower variance when compared to regular cross validation [69]. we used 5-fold strategy, in which the data are divided into 5 equal bunches by which one part is used for validation and the remaining four parts are used for training. The technique persists untill each fold is sorted out as validation data and assesses the performance of the model using different types of matrices, including a confusion matrix, matthew's correlation coefficient (MCC), receiver operating characteristics (ROC) curve and precision-recall curve (PRC). A confusion matrix is one of the basic matrix used to assess the quality of the classification predictor. A confusion matrix envisages the results in the form of a matrix where each column constitutes the predicted result and each row indicates the actual class of the sample. A confusion matrix relies on four values, the number of true positives (Tp), the number of true negatives (Tn), the number of false-positive (Fp), and the number of false negatives (Fn). Another performance matrices used confusion matrix as.

$$\begin{cases} Sensitivity = \dfrac{T_p}{T_p + F_n} \\ Specificity = \dfrac{T_n}{T_n + F_p} \\ Accuracy = \dfrac{T_p + T_n}{T_p + T_n + F_p + F_n} \\ MCC = \dfrac{(T_p)(T_n) - (F_p)(F_n)}{\sqrt{(T_p + F_p)(T_p + F_n)(T_n + F_n)(T_n + F_p)}} \end{cases} \tag{1}$$

Sensitivity (SN) is a measure of the accurate positive rate and Specificity (SP) represents the true negative rate of the classifier. Accuracy (ACC) is the proportion of all accurately predicted samples, both positive and negative. MCC is a balanced measure in which true and false negatives are both used in the evaluation.The area under the ROC curve is used to indicate the degree of quality and separability of the classification models. The PRC is the tradeoff between precision and recall using different threshold. The higher area under the curve is the representation of both the high recall and the high precision. As the high value of precision is due to a low false positive rate,while the high recall is due to low false negative rate.

IV. RESULTS AND DISCUSSION

A. OPTIMAL WINDOW SIZE AND ENCODING SCHEME

The length of the amino acid sequence given to the learning construction model is also an important hyper-parameter.

TABLE 2. Comparison of encoding techniques on different fragment length.

Fragment	Embedding (64 dimension)				onehot + PCA (25 dimension)			
	SN	SP	ACC	MCC	SN	SP	ACC	MCC
21	0.59	0.80	0.67	0.35	0.79	0.51	0.65	0.32
23	0.62	0.75	0.68	0.37	0.77	0.54	0.65	0.31
25	0.67	0.69	0.68	0.37	0.79	0.54	0.67	0.34
27	0.60	0.78	0.69	0.38	0.79	0.53	0.66	0.32
29	0.61	0.78	0.69	0.39	0.80	0.56	0.68	0.38
31	0.62	0.79	0.70	0.42	0.83	0.51	0.67	0.35
33	0.63	0.78	0.71	0.42	0.82	0.55	0.69	0.39
35	0.65	0.77	0.71	0.42	0.79	0.58	0.69	0.38
37	0.59	0.81	0.70	0.41	0.84	0.54	0.69	0.41
39	0.62	0.80	0.71	0.43	0.80	0.56	0.68	0.37
41	0.79	0.66	0.72	0.45	0.81	0.62	0.71	0.43
43	0.61	0.80	0.71	0.43	0.81	0.58	0.69	0.40
45	0.64	0.78	0.71	0.43	0.84	0.55	0.70	0.41

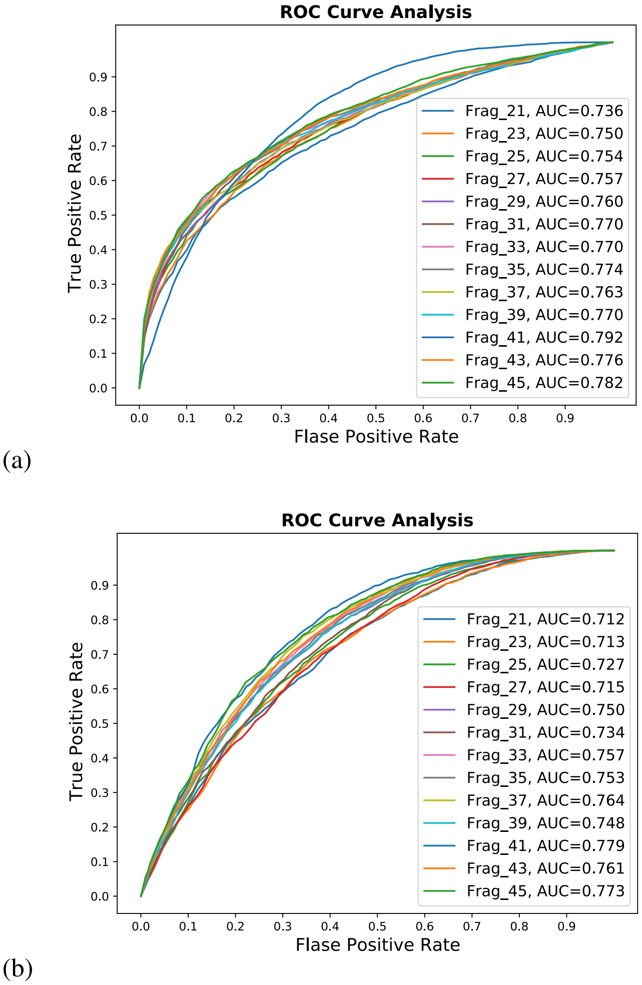


FIGURE 4. 5-fold cross validation ROC-AUC comparisons of different fragments length. (a) Embedding Encoding. (b) Combine one-hot and PCA encoding.

The general range for protein chain length is (21-41) for prediction of PTM sites. We previously performed experiments on fragment of different lengths (21 to 45) and using different encoding schemes, including word embedding and combined one-hot and PCA encoding. A PCA vector is a 5 dimensional quantitative representation of 20 amino acids and is derived from multi-dimensional scaling of 237 physicochemical properties [70]. As shown in Table 2 and Figure 4, we identified 41 as the optimal length; this length was also used in previous SNO studies [40], [41]. Details of the performance of other encoding schemes and models are given in the supplementary file (Sections A and B).

B. EXPERIMENTS ON DIFFERENT DEEP LEARNING ARCHITECTURES

We looked at other deep learning architectures with encoding schemes including embedding, ProtVec [71] and one-hot, and other architectures including convolution neural network (CNN), LSTM, and BiLSTM. ProtVec is type of word embedding similar to word2vec [72] using 3-mer residue to construct a hundred dimension vector. We also used a combination of one-hot and PCA, and a combination of one-hot and embedding as inputs for a CNN-based architecture. The details of these methods are given in the supplementary file, sections B and C, respectively, while the results of 5-fold cross-validation and independent data testing are listed in Table 3 and a comparison of the area under ROC curves is shown in Fig 5. RecSNO provides better results than the other architectures.

TABLE 3. Comparison of RecSNO with other deep learning architectures.

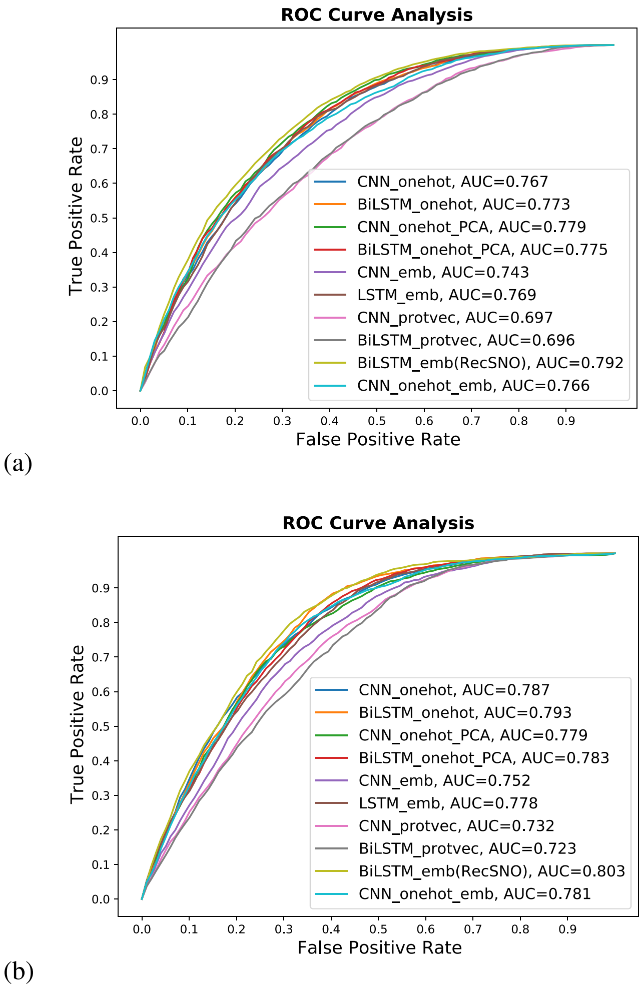
Models	5-fold cross validation				Independent			
	SN	SP	ACC	MCC	SN	SP	ACC	MCC
CNN-onehot	0.83	0.58	0.70	0.42	0.80	0.63	0.65	0.27
BiLSTM-onehot	0.79	0.63	0.71	0.42	0.78	0.68	0.9	0.28
CNN-onehot-PCA	0.81	0.62	0.71	0.43	0.78	0.66	0.67	0.27
BiLSTM-onehot-PCA	0.78	0.65	0.71	0.42	0.72	0.70	0.70	0.27
CNN-emb	0.76	0.59	0.68	0.36	0.74	0.64	0.65	0.23
LSTM-emb	0.77	0.64	0.71	0.42	0.72	0.69	0.69	0.26
CNN-protvec	0.71	0.57	0.64	0.29	0.70	0.64	0.65	0.21
BiLSTM-protvec	0.70	0.59	0.64	0.28	0.67	0.64	0.64	0.19
BiLSTM-emb (RecSNO)	0.79	0.66	0.72	0.45	0.77	0.71	0.71	0.30
CNN-onehot-emb	0.77	0.62	0.70	0.40	0.77	0.68	0.69	0.28

C. CROSS-VALIDATION PERFORMANCE

Our ultimate predictor, RecSNO, makes use of embedding with window and dimension sizes of 41 and 64, respectively. We employed 5-fold cross-validation to test the results. For an accurate comparison, we used the same training and testing dataset as were used for the preSNO model. The outcomes are shown in Table 4. RecSNO exhibits robustness; the performance metrics were as follows: sensitivity, 0.79; specificity, 0.66; accuracy, 0.72; MCC, 0.45; AUC, 0.79 and PRC, 0.75. as sown in Fig 6 and Fig 7 respectively. In terms of sensitivity, accuracy, and MCC, our predictor is superior to preSNO. Considering that the MCC is often used as a substitute for overall model performance, it was preferred slightly over the other evaluation parameters [73]. The preSNO predictor is biased towards the negative class, but our proposed model overcomes this problem and gives more balanced results.

D. INDEPENDENT DATASET COMPARISON WITH EXISTING PREDICTORS

The performance of RecSNO was compared with that of existing SNO site predictors using an independent test set for reasons that were highlighted earlier, in the Benchmark



(a)

(b)

FIGURE 5. ROC-AUC comparisons of different deep learning architectures. (a) 5-fold cross validation. (b) Independent data results.

TABLE 4. Comparison of RecSNO with recent existing predictor.

Predictors	Sensitivity	Specificity	Accuracy	MCC	AUC
PreSNO	0.54	**0.86**	0.70	0.42	**0.84**
RecSNO	**0.79**	0.66	**0.72**	**0.45**	0.79

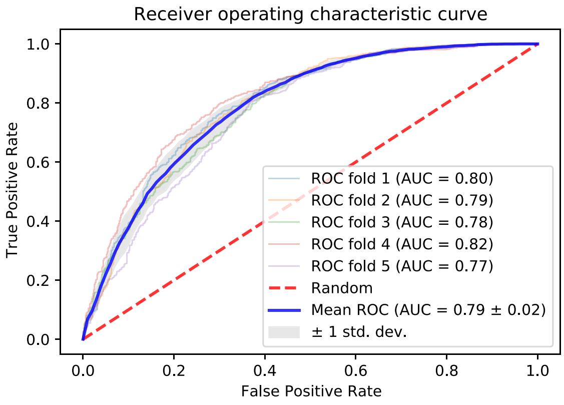


FIGURE 6. AUCs of 5 folds cross validation.

Dataset section. We examined five existing publicly available SNO predictors, including GPS-SNO [8], SNOSite [43], iSNOPseAAC [42], DeepNitro [41], and PreSNO [40].

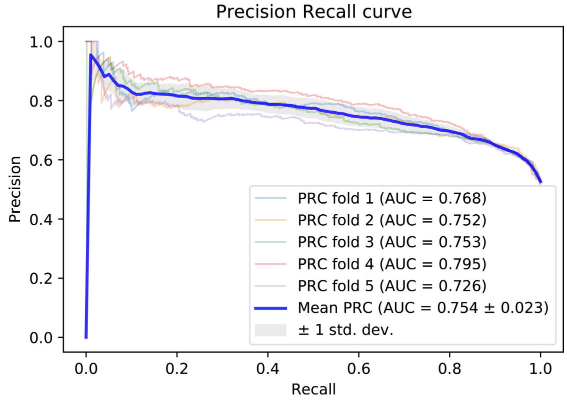


FIGURE 7. AUPRCs of 5 folds cross validation.

TABLE 5. Independent dataset comparison of RecSNO with existing predictors.

Predictors	Sensitivity	Specificity	Accuracy	MCC	AUC
GPS-SNO	0.28	0.74	0.69	0.01	0.52
iSNOPseAAC	0.29	0.76	0.71	0.03	NA
SNOSite	0.67	0.45	0.47	0.07	NA
DeepNitro	0.58	0.76	0.73	0.22	0.73
PreSNO	0.60	**0.77**	**0.75**	0.25	0.76
RecSNO	**0.77**	0.71	0.71	**0.30**	**0.80**

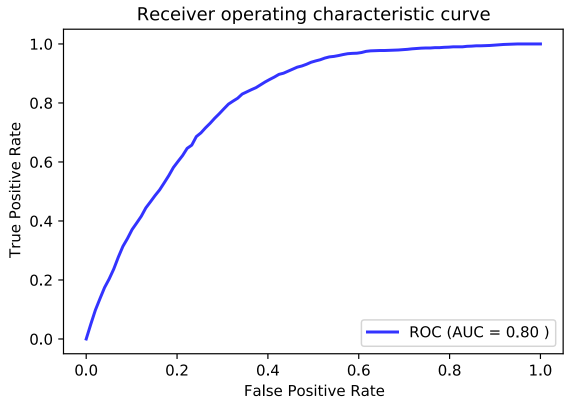
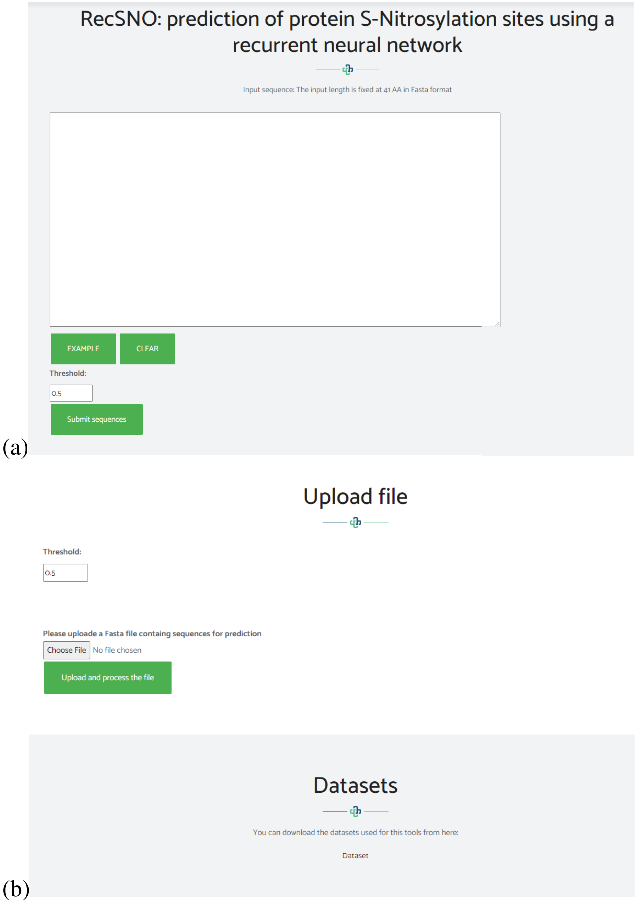


FIGURE 8. Independent AUC-ROC result.

We evaluated the independent sample prediction results in terms of specificity, sensitivity, accuracy, MCC, and AUC. As shown in Table 5, our deep learning predictor, RecSNO, had a sensitivity of 0.77, specificity of 0.71, accuracy of 0.71, MCC of 0.30, and ROC-AUC of 0.80 as shown in Fig 8. These findings reveal that our model offers the best output in terms of sensitivity, MCC and AUC. Other predictors show large differences between sensitivity and specificity because they are prejudiced towards one or the other. The proposed model gives balanced results to solve this problem and achieves a high true positive rate. The proposed RecSNO model gives reliable forecasts compared to existing computational SNO methods.

V. WEB SERVER

The web server for proposed model is freely accessible at http://nsclbio.jbnu.ac.kr/tools/RecSNO/. Where single



RecSNO: prediction of protein S-Nitrosylation sites using a recurrent neural network

Input sequence: The input length is fixed at 41 AA in Fasta format

Threshold: 0.5

Upload file

Threshold: 0.5

Please uploade a Fasta file containg sequences for prediction

No file chosen

Datasets

You can download the datasets used for this tools from here:

Dataset

FIGURE 9. The home page of the web server. (a) Finding a s-nitrosylation site for a single sequence. (b) Finding s-nitrosylation site in a fast file and downloading the dataset for training and independent.

protein sequence can be used as the input or the file containing sequences in fasta format upload. Figure 9 shows the home page of the web server where dataset is also available.The webserver is constructed using Flask library in Python.

VI. CONCLUSION

In the current study, we construct a computational tool, RecSNO, to identify SNO sites. Although RecSNO gives accurate and better predictions than other published predictors in aspects of 5-fold cross-validation and independent tests, it still has some issues that should be considered in future work. The structural preferences of S-nitrosylation sites should be considered in greater detail because tertiary structure is a key feature for the occurrence of protein nitrosylation sites, and was not taken into account in this study [61]. Notably, we did use the RNN-based method BiLSTM to measure contextual dependencies in nitrosylation sequences. Compared to previous works that used handicraft features, the proposed method uncovers high-level features and has enhanced prediction capability for protein SNO sites. RecSNO may prove to be very useful for biologists attempting to identify SNO sites and understand their role in diseases. The web application of our tool is provided for public use.

REFERENCES

[1] I. Gusarov and E. Nudler, "Protein S-nitrosylation: Enzymatically controlled, but intrinsically unstable, post-translational modification," *Mol. Cell*, vol. 69, no. 3, pp. 351–353, Feb. 2018.

[2] M. Mann and O. N. Jensen, "Proteomic analysis of post-translational modifications," *Nature Biotechnol.*, vol. 21, no. 3, pp. 255–261, Mar. 2003.

[3] C. M. Padgett and A. R. Whorton, "S-nitrosoglutathione reversibly inhibits GAPDH by S-nitrosylation," *Amer. J. Physiol.-Cell Physiol.*, vol. 269, no. 3, pp. C739–C749, Sep. 1995.

[4] A. Stern, P. Costa, H. Monteiro, and A. C. A. Reis, "Nitric oxide: Protein tyrosine phosphorylation and protein S-nitrosylation in cancer," *Biomed. J.*, vol. 38, no. 5, p. 380, 2015.

[5] M. W. Foster, D. T. Hess, and J. S. Stamler, "Protein S-nitrosylation in health and disease: A current perspective," *Trends Mol. Med.*, vol. 15, no. 9, pp. 391–404, Sep. 2009.

[6] B. Derakhshan, G. Hao, and S. Gross, "Balancing reactivity against selectivity: The evolution of protein S-nitrosylation as an effector of cell signaling by nitric oxide," *Cardiovascular Res.*, vol. 75, no. 2, pp. 210–219, Jul. 2007.

[7] D. T. Hess, A. Matsumoto, S.-O. Kim, H. E. Marshall, and J. S. Stamler, "Protein S-nitrosylation: Purview and parameters," *Nature Rev. Mol. Cell Biol.*, vol. 6, no. 2, pp. 150–166, Feb. 2005.

[8] Y. Xue, Z. Liu, X. Gao, C. Jin, L. Wen, X. Yao, and J. Ren, "GPS-SNO: Computational prediction of protein S-nitrosylation sites with a modified GPS algorithm," *PLoS ONE*, vol. 5, no. 6, Jun. 2010, Art. no. e11290.

[9] F. Li, P. Sonveaux, Z. N. Rabbani, S. Liu, B. Yan, Q. Huang, Z. Vujaskovic, M. W. Dewhirst, and C.-Y. Li, "Regulation of HIF-1α stability through S-nitrosylation," *Mol. Cell*, vol. 26, no. 1, pp. 63–74, Apr. 2007.

[10] E. Hernlund, O. Kutuk, H. Basaga, S. Linder, T. Panaretakis, and M. Shoshan, "Cisplatin-induced nitrosylation of p53 prevents its mitochondrial translocation," *Free Radical Biol. Med.*, vol. 46, no. 12, pp. 1607–1613, Jun. 2009.

[11] K. Ozawa, E. J. Whalen, C. D. Nelson, Y. Mu, D. T. Hess, R. J. Lefkowitz, and J. S. Stamler, "S-nitrosylation of β-arrestin regulates β-adrenergic receptor trafficking," *Mol. Cell*, vol. 31, no. 3, pp. 395–405, Aug. 2008.

[12] A. H. K. Tsang, Y.-I. Lee, H. S. Ko, J. M. Savitt, O. Pletnikova, J. C. Troncoso, V. L. Dawson, T. M. Dawson, and K. K. K. Chung, "S-nitrosylation of XIAP compromises neuronal survival in Parkinson's disease," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 12, pp. 4900–4905, Mar. 2009.

[13] D. Kasten, A. Mithöfer, E. Georgii, H. Lang, J. Durner, and F. Gaupels, "Nitrite is the driver, phytohormones are modulators while NO and H_2O_2 act as promoters of NO_2-induced cell death," *J. Experim. Botany*, vol. 67, no. 22, pp. 6337–6349, Dec. 2016.

[14] E. J. Whalen, M. W. Foster, A. Matsumoto, K. Ozawa, J. D. Violin, L. G. Que, C. D. Nelson, M. Benhar, J. R. Keys, H. A. Rockman, and W. J. Koch, "Regulation of β-adrenergic receptor signaling by S-nitrosylation of G-protein-coupled receptor kinase 2," *Cell*, vol. 129, no. 3, pp. 511–522, 2007.

[15] V. Fernando, X. Zheng, Y. Walia, V. Sharma, J. Letson, and S. Furuta, "S-nitrosylation: An emerging paradigm of redox signaling," *Antioxidants*, vol. 8, no. 9, p. 404, Sep. 2019.

[16] B. Bonavida and H. Garban, "Nitric oxide-mediated sensitization of resistant tumor cells to apoptosis by chemo-immunotherapeutics," *Redox Biol.*, vol. 6, pp. 486–494, Dec. 2015.

[17] A. Nott, P. M. Watson, J. D. Robinson, L. Crepaldi, and A. Riccio, "S-nitrosylation of histone deacetylase 2 induces chromatin remodelling in neurons," *Nature*, vol. 455, no. 7211, pp. 411–415, Sep. 2008.

[18] Z. Wang, "Protein S-nitrosylation and cancer," *Cancer Lett.*, vol. 320, no. 2, pp. 123–129, Jul. 2012.

[19] C. M. Schonhoff, M. Matsuoka, H. Tummala, M. A. Johnson, A. G. Estevéz, R. Wu, A. Kamaid, K. C. Ricart, Y. Hashimoto, B. Gaston, and T. L. Macdonald, "S-nitrosothiol depletion in amyotrophic lateral sclerosis," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 7, pp. 2404–2409, 2006.

[20] M. Piroddi, A. Palmese, F. Pilolli, A. Amoresano, P. Pucci, C. Ronco, and F. Galli, "Plasma nitroproteome of kidney disease patients," *Amino Acids*, vol. 40, no. 2, pp. 653–667, Feb. 2011.

[21] I. V. Turko, "Protein nitration in cardiovascular diseases," *Pharmacol. Rev.*, vol. 54, no. 4, pp. 619–634, Dec. 2002.

[22] C. Montagna, C. Cirotti, S. Rizza, and G. Filomeni, "When S-nitrosylation gets to mitochondria: From signaling to age-related diseases," *Antioxidants Redox Signaling*, vol. 32, no. 12, pp. 884–905, Apr. 2020.

[23] T. Nakamura, O. A. Prikhodko, E. Pirie, S. Nagar, M. W. Akhtar, C.-K. Oh, S. R. McKercher, R. Ambasudhan, S.-I. Okamoto, and S. A. Lipton, "Aberrant protein S-nitrosylation contributes to the pathophysiology of neurodegenerative diseases," *Neurobiol. Disease*, vol. 84, pp. 99–108, Dec. 2015.

[24] D.-H. Cho, T. Nakamura, J. Fang, P. Cieplak, A. Godzik, Z. Gu, and S. A. Lipton, "S-nitrosylation of Drp1 mediates β-amyloid-related mitochondrial fission and neuronal injury," *Science*, vol. 324, no. 5923, pp. 102–105, 2009.

[25] T. Uehara, T. Nakamura, D. Yao, Z.-Q. Shi, Z. Gu, Y. Ma, E. Masliah, Y. Nomura, and S. A. Lipton, "S-nitrosylated protein-disulphide isomerase links protein misfolding to neurodegeneration," *Nature*, vol. 441, no. 7092, pp. 513–517, May 2006.

[26] S. Ben-Lulu, T. Ziv, P. Weisman-Shomer, and M. Benhar, "Nitrosothiol-trapping-based proteomic analysis of S-nitrosylation in human lung carcinoma cells," *PLoS ONE*, vol. 12, no. 1, Jan. 2017, Art. no. e0169862.

[27] T. Nakamura and S. A. Lipton, "Protein S-nitrosylation as a therapeutic target for neurodegenerative diseases," *Trends Pharmacol. Sci.*, vol. 37, no. 1, pp. 73–84, Jan. 2016.

[28] G. Huang, J. Li, and C. Zhao, "Computational prediction and analysis of associations between small molecules and binding-associated S-nitrosylation sites," *Molecules*, vol. 23, no. 4, p. 954, Apr. 2018.

[29] E. Bignon, M. F. Allega, M. Lucchetta, M. Tiberti, and E. Papaleo, "Computational structural biology of S-nitrosylation of cancer targets," *Frontiers Oncol.*, vol. 8, p. 272, Aug. 2018.

[30] T. Nakamura, S. Tu, M. W. Akhtar, C. R. Sunico, S.-I. Okamoto, and S. A. Lipton, "Aberrant protein S-Nitrosylation in neurodegenerative diseases," *Neuron*, vol. 78, no. 4, pp. 596–614, May 2013.

[31] J. R. Burgoyne and P. Eaton, "A rapid approach for the detection, quantification, and discovery of novel sulfenic acid or S-nitrosothiol modified proteins using a biotin-switch method," in *Methods in Enzymology*, vol. 473. Amsterdam, The Netherlands: Elsevier, 2010, pp. 281–303.

[32] M. T. Forrester, J. W. Thompson, M. W. Foster, L. Nogueira, M. A. Moseley, and J. S. Stamler, "Proteomic analysis of S-nitrosylation and denitrosylation by resin-assisted capture," *Nature Biotechnol.*, vol. 27, no. 6, pp. 557–559, Jun. 2009.

[33] G. Hao, B. Derakhshan, L. Shi, F. Campagne, and S. S. Gross, "SNOSID, a proteomic method for identification of cysteine S-nitrosylation sites in complex protein mixtures," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 4, pp. 1012–1017, Jan. 2006.

[34] B. Derakhshan, P. C. Wille, and S. S. Gross, "Unbiased identification of cysteine S-nitrosylation sites on proteins," *Nature protocols*, vol. 2, no. 7, p. 1685, 2007.

[35] S. R. Jaffrey and S. H. Snyder, "The biotin switch method for the detection of S-nitrosylated proteins," *Sci. Signaling*, vol. 2001, no. 86, p. pl1, Jun. 2001.

[36] B. Huang, S. C. Chen, and D. L. Wang, "Shear flow increases S-nitrosylation of proteins in endothelial cells," *Cardiovascular Res.*, vol. 83, no. 3, pp. 536–546, Aug. 2009.

[37] C. Lindermayr, G. Saalbach, and J. Durner, "Proteomic identification of S-nitrosylated proteins in arabidopsis," *Plant Physiol.*, vol. 137, no. 3, pp. 921–930, Mar. 2005.

[38] T. Kuncewicz, E. A. Sheta, I. L. Goldknopf, and B. C. Kone, "Proteomic analysis of S-nitrosylated proteins in mesangial cells," *Mol. Cellular Proteomics*, vol. 2, no. 3, pp. 156–163, Mar. 2003.

[39] J. S. Paige, G. Xu, B. Stancevic, and S. R. Jaffrey, "Nitrosothiol reactivity profiling identifies S-nitrosylated proteins with unexpected stability," *Chem. Biol.*, vol. 15, no. 12, pp. 1307–1316, Dec. 2008.

[40] M. M. Hasan, B. Manavalan, M. S. Khatun, and H. Kurata, "Prediction of S-nitrosylation sites by integrating support vector machines and random forest," *Mol. Omics*, vol. 15, no. 6, pp. 451–458, 2019.

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

[42] Y. Xu, J. Ding, L.-Y. Wu, and K.-C. Chou, "ISNO-PseAAC: Predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition," *PLoS ONE*, vol. 8, no. 2, Feb. 2013, Art. no. e55844.

[43] T.-Y. Lee, Y.-J. Chen, T.-C. Lu, H.-D. Huang, and Y.-J. Chen, "SNOSite: Exploiting maximal dependence decomposition to identify cysteine S-nitrosylation with substrate site specificity," *PLoS ONE*, vol. 6, no. 7, Jul. 2011, Art. no. e21849.

[44] Y. Wang, S. Liu, N. Afzal, M. Rastegar-Mojarad, L. Wang, F. Shen, P. Kingsbury, and H. Liu, "A comparison of word embeddings for the biomedical natural language processing," *J. Biomed. Informat.*, vol. 87, pp. 12–20, Nov. 2018.

[45] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, P.-Y. Chen, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition," 2020, *arXiv:2010.13309*. [Online]. Available: http://arxiv.org/abs/2010.13309

[46] F. Mohammad and Y.-C. Kim, "Energy load forecasting model based on deep neural networks for smart grids," *Int. J. Syst. Assurance Eng. Manage.*, vol. 11, no. 4, pp. 824–834, Aug. 2020.

[47] A. Khan, T. Ilyas, M. Umraiz, Z. I. Mannan, and H. Kim, "CED-net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture," *Electronics*, vol. 9, no. 10, p. 1602, Oct. 2020.

[48] W. Alam, S. D. Ali, H. Tayara, and K. T. Chong, "A CNN-based RNA N6-methyladenosine site predictor for multiple species using heterogeneous features representation," *IEEE Access*, vol. 8, pp. 138203–138209, 2020.

[49] J. Khanal, I. Nazari, H. Tayara, and K. T. Chong, "4mCCNN: Identification of N4-methylcytosine sites in prokaryotes using convolutional neural network," *IEEE Access*, vol. 7, pp. 145455–145461, 2019.

[50] D. Wang, Y. Liang, and D. Xu, "Capsule network for protein post-translational modification site prediction," *Bioinformatics*, vol. 35, no. 14, pp. 2386–2394, Jul. 2019.

[51] S. D. Ali, W. Alam, H. Tayara, and K. Chong, "Identification of functional piRNAs using a convolutional neural network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Oct. 29, 2020, doi: 10.1109/TCBB.2020.3034313.

[52] Y. Chu, A. C. Kaushik, X. Wang, W. Wang, Y. Zhang, X. Shan, D. R. Salahub, Y. Xiong, and D.-Q. Wei, "DTI-CDF: A cascade deep forest model towards the prediction of drug-target interactions based on hybrid features," *Briefings Bioinf.*, Dec. 2019, Art. no. bbz152.

[53] W. Wang, X. Guan, M. T. Khan, Y. Xiong, and D.-Q. Wei, "LMI-DForest: A deep forest model towards the prediction of lncRNA-miRNA interactions," *Comput. Biol. Chem.*, vol. 89, Dec. 2020, Art. no. 107406.

[54] N. Thapa, M. Chaudhari, S. Mcmanus, K. Roy, R. H. Newman, H. Saigo, and D. B. Kc, "DeepSuccinylSite: A deep learning based approach for protein succinylation site prediction," *BMC Bioinf.*, vol. 21, no. S3, pp. 1–10, Apr. 2020.

[55] D. Wang, S. Zeng, C. Xu, W. Qiu, Y. Liang, T. Joshi, and D. Xu, "MusiteDeep: A deep-learning framework for general and kinase-specific phosphorylation site prediction," *Bioinformatics*, vol. 33, no. 24, pp. 3909–3916, Dec. 2017.

[56] M. Chaudhari, N. Thapa, K. Roy, R. H. Newman, H. Saigo, and B. K. C. Dukka, "DeepRMethylSite: A deep learning based approach for prediction of arginine methylation sites in proteins," *Mol. Omics*, vol. 16, no. 5, pp. 448–454, 2020.

[57] F. Luo, M. Wang, Y. Liu, X.-M. Zhao, and A. Li, "DeepPhos: Prediction of protein phosphorylation sites with deep learning," *Bioinformatics*, vol. 35, no. 16, pp. 2766–2773, Aug. 2019.

[58] H. Fu, Y. Yang, X. Wang, H. Wang, and Y. Xu, "DeepUbi: A deep learning framework for prediction of ubiquitination sites in proteins," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–10, Dec. 2019.

[59] I. Nazari, H. Tayara, and K. T. Chong, "Branch point selection in RNA splicing using deep learning," *IEEE Access*, vol. 7, pp. 1800–1807, 2019.

[60] M. Wu, Y. Yang, H. Wang, and Y. Xu, "A deep learning method to more accurately recall known lysine acetylation sites," *BMC Bioinf.*, vol. 20, no. 1, p. 49, Dec. 2019.

[61] K.-Y. Huang, J. B.-K. Hsu, and T.-Y. Lee, "Characterization and identification of lysine succinylation sites based on deep learning method," *Sci. Rep.*, vol. 9, no. 1, pp. 1–15, Dec. 2019.

[62] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: Accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, Dec. 2012.

[63] UniProt Consortium, "UniProt: A worldwide hub of protein knowledge," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D506–D515, Jan. 2019.

[64] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018.

[65] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[66] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[67] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," 2017, *arXiv:1801.01078*. [Online]. Available: http://arxiv.org/abs/1801.01078

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[69] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. IJCAI*, Montreal, QC, Canada, 1995, vol. 14, no. 2, pp. 1137–1145.

[70] W. Braun and M. S. Venkatarajan, "New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical–Chemical properties," *J. Mol. Model.*, vol. 7, no. 12, pp. 445–453, Dec. 2001.

[71] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, Nov. 2015, Art. no. e0141287.

[72] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[73] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020.

ARSLAN SIRAJ received the M.Sc. degree in information technology from the Institute of Information Technology, Quaid-i-Azam University, Islamabad, Pakistan, in 2019. He is currently pursuing the master's degree in electronics and information engineering with Jeonbuk National University, Jeonju, South Korea. His research interests include bioinformatics, machine learning, and image processing.

TUVSHINBAYAR CHANTSALNYAM received the B.Sc. degree in electronics engineering and information technology from the Mongolian University of Science and Technology and the M.S. degree from Jeonbuk National University, Jeonju, South Korea. He is currently a Graduate Student with the School of Electronics and Information Engineering, Jeonbuk National University. He is working on network system control, time-delay systems, neural networks, deep learning, and bioinformatics.

HILAL TAYARA received the B.Sc. degree in computer engineering from Aleppo University, Aleppo, Syria, in 2008, and the M.S. and Ph.D. degrees in electronics and information engineering from Jeonbuk National University, Jeonju, South Korea, in 2015 and 2019, respectively. He is currently an Assistant Professor with the Department of International Science and Engineering, Jeonbuk National University. His research interests include bioinformatics, machine learning, and image processing.

KIL TO CHONG received the Ph.D. degree in mechanical engineering from Texas A&M University, in 1993. He is currently a Professor with the School of Electronics and Information Engineering, Jeonbuk National University, Jeonju, South Korea, the Head of the Advanced Information and Electronics Research Center, Jeonbuk National University, and the President of the Korean Electronics Engineering Society, Systems and Control. His research interests include bioinformatics, artificial intelligence, brain disease, and new drug discovery.