Prediction of protein phosphorylation sites using Deep Convolutional and Recurrent Neural Networks

Phuc Khanh Nhi NGUYEN
npkhanhnhi@gmail.com

Abstract

Phosphorylation, a critical post-translational modification, governs diverse cellular processes and is implicated in various diseases when dysregulated. Traditional methods for identifying phosphorylation sites are often labor-intensive and time-consuming, necessitating computational approaches. Inspired by the tremendous progress of Deep Learning and its applications in bioinformatics, in this project, we employ popular Deep Neural Networks (DNNs) such as convolutional and recurrent networks to predict serine/threonine (S/T) and tyrosine (Y) phosphorylation sites. Concretely, we leverage well-known neural network architectures such as VGG, ResNet, and GRU/LSTM to learn discriminant features from the 1D protein sequences. Additionally, along with the baseline data preprocessing method based on one-hot representation, we also propose a proper data augmentation scheme to address limited sample sizes. The models are then evaluated across varying sequence window sizes using metrics like accuracy, and F1 score. Experimental results show that our models achieve better performance on S/T sites ($\sim 80\%$ accuracy) with a large number of training samples while providing lower accuracy ($\sim 70\%$) on Y sites with limited data. More notably, the proposed data augmentation scheme here allows us to reduce overfitting and improve model's performance in general.

1 Introduction

Phosphorylation is a critical post-translational modification carried out by kinases, facilitating the attachment of phosphate groups to specific target amino acids. In human cells, phosphorylation predominantly occurs at serine (S) and threonine (T) residues, with tyrosine (Y) residues being phosphorylated at lower levels [1]. Aberrations in phosphorylation-mediated signaling pathways have been implicated in various diseases, including cancer, contributing to abnormal cell proliferation, migration, metabolic dysregulation, and systemic inflammation. Consequently, analyzing protein phosphorylation sites has become essential for understanding the regulation and dysregulation of signaling networks in both normal and pathological conditions [2]. While traditional experimental techniques for identifying phosphorylation sites are effective, they are often labor-intensive and time-consuming, particularly when dealing with a large number of candidate sites. To overcome these limitations, data-driven learning-based computational methods have gained popularity as efficient alternatives for tackling the challenges posed by experimental approaches [3]. Concretely, it employs DenseNet architecture [4] with three window sizes combined together to extract relevant features from protein sequences. Besides, other convolutional neural networks (CNNs) such as VGG [5] and ResNet [6] models have also demonstrated their potential in handling sequential data, particularly in tasks involving biological sequence analysis such as protein structure and function prediction [7, 8]. Beyond these CNNs, previous works also adopt recurrent neural networks (RNNs) such as Gated Recurrent Units (GRU) or Long Short-Term Memory (LSTM) to learn sequential dependencies within biological sequences [9] [10]. Therefore, in this project, we develop variants of these models for the prediction of S/T and Y phosphorylation sites, combining with a proper data augmentation scheme to improve their generalization ability.

The rest of this report is organized as follows: Section 2 presents the data preprocessing pipeline along with the augmentation scheme. Section 3 describes architectures of convolutional networks such as VGG-like, ResNet-like models, and mixed convolutional-recurrent models with GRU/LSTM. Section 4 presents the training setting, the obtained results, and our comments as well. Finally, the conclusion is given in Section 5.

2 Data preprocessing

2.1 From string sequence to numerical data

The data collection and preprocessing are conducted based on previous work [3] with some modifications. Experimentally verified phosphorylation sites of human proteins are collected from the Phospho.ELM database [11] and all repetitive items are removed. Next, all experimentally verified phosphorylation sites for serine/threonine (S/T) sites and tyrosine (Y) sites are extracted from these protein sequences as positive examples. This results in 39492 sequences of S/T sites and 3369 sequences of Y sites, respectively. A subset of other S/T and Y sites is randomly selected for negative examples to match the number of positive examples. We also investigate the impact of window size L of phosphorylation sites by extracting sequences of different lengths: 15, 33, and 51. The protein subsequences are encoded using a one-hot encoding scheme. Sequences shorter than the window size are padded with a specific symbol. This results in a binary 2D matrix of shape $(L, 21)$ as the input of the model, where 21 is the size of the amino acid symbol dictionary. The label is set to 1 for positive sites and 0 for negative ones. The dataset is then randomly split into independent training, validation, and test subsets with the following ratio $0.64 : 0.16 : 0.2$.

2.2 Data augmentation (DA) scheme

Since the resulting training sets are relatively small-sized, specifically the Y sites, the models may likely exhibit overfitting during training. To overcome this issue and improve model's generalizability, we propose a simple data augmentation scheme for the training sets. In detail, noise with a uniform distribution in the range of $[-0.1, 0.1]$ is added to each input sequence. For the original ground-truth label y, random uniform noise is applied as follows to obtain the augmented label y_{aug}:

$$y_{aug} = \begin{cases} p & \text{if } y = 0 \\ 1 - p & \text{if } y = 1 \end{cases} \tag{1}$$

with $p \sim \mathcal{U}(0., 0.4)$. Adding small variations into input may increase model's stability during training, while softening the ground-truth label makes it better aligned with the sigmoid probability outcome of the model, therefore offering better tolerance and inherently facilitating the learning process.

3 Model architectures

Since the phosphorylation data have a sequential structure, it is highly relevant to adopt convolutional and/or recurrent computations to model the local dependencies and extract discriminant features from the whole sequence. In this project, we consider two types of model architectures: CNN with VGG-like and ResNet-like models, and mixed CNN-RNN involving some blocks of VGG-like model on top followed by GRU/LSTM layers at the bottom.

3.1 Simple plain VGG-like network

Figure 1 depicts the network architecture of our VGG-like model (originally introduced in [5]). It basically consists of 6 convolutional layers of kernel size 3. Each convolution is followed by a batch normalization and a ReLU activation. After each two convolutional layers, we make use of a Max Pooling of kernel size 2 to reduce the length dimension by half. The number of output channels is controlled by the parameter F and it is doubled after every two convolutions. At the end of the convolutional part, we vectorize the feature tensor before projecting it into the output probability space via a fully connected (Dense) layer of 1 unit only, followed by sigmoid activation. To reduce overfitting, we also insert dropout after each convolution with a rate of 0.1. As we set $F = 32$, the model contains approximately $100k$ parameters.

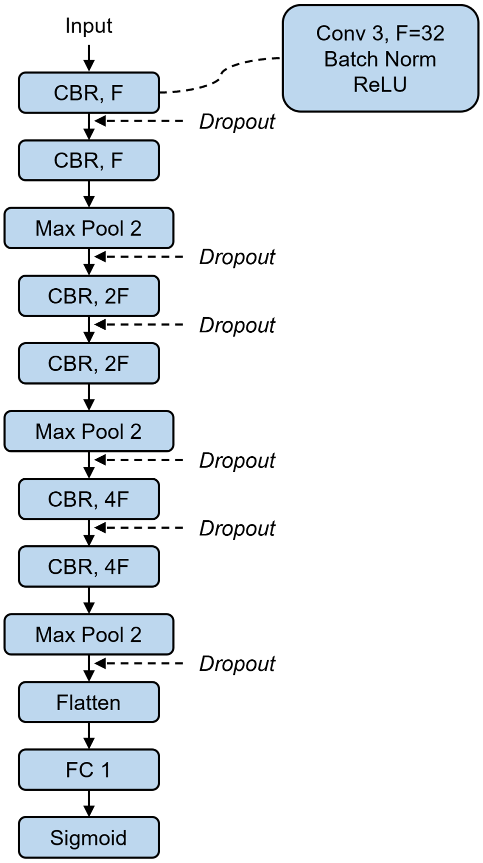


Figure 1: VGG-like model.

3.2 ResNet-like model

Figure 2 depicts the architecture of our ResNet-like model. It mainly contains 6 residual blocks that were originally proposed in [6]. In each residual block, there are two consecutive convolution layers of kernel size 3 and an addition shortcut connection to combine the result of the convolutions with the input. In particular, when we reduce the length dimension by setting the stride $s = 2$ for the first convolution layer, we also transform the shortcut with a point-wise convolution of kernel size 1. Similarly to the setting in VGG-like model, the channel dimension is also doubled while the length is halved after every two residual blocks. Here we set $F = 16$ resulting in the model of $70k$ parameters.

3.3 Mixed CNN-RNN model

Figure 3 demonstrates the architecture of our mixed CNN-RNN model, which is also adopted for other tasks such as sound classification [12]. It basically consists of the first two blocks of a VGG-like model. The number of output channels is also controlled by the parameter F. At the end of the convolutional blocks, the extracted feature vectors are fed into two consecutive recurrent layers, chosen either GRU (denoted as CNN-GRU) or LSTM (denoted as CNN-LSTM). The first recurrent layer has the output unit numbers equal to the output channels of the last convolutional layer, while the second only outputs a single value for each position. A Max Pooling operation along the sequence length axis enables capturing only the most crucial features, followed by a sigmoid activation to provide the predicted probability. Similar to VGG-like model, we also introduce dropout after each convolution with a rate of 0.05 to mitigate overfitting. With $F = 32$, the CNN-GRU model has $50k$ parameters, while the CNN-LSTM model has $60k$.

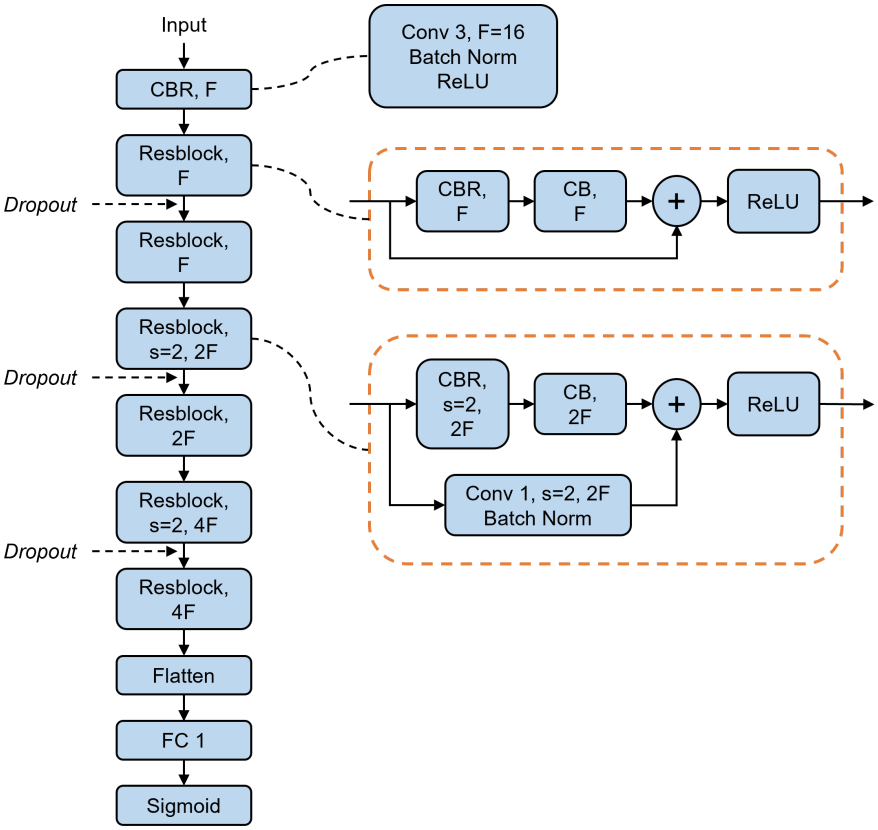


Figure 2: ResNet-like model.

4 Experiments

4.1 Training configurations

To verify the impact of our DA scheme, each model is trained two times: one without and one with the proposed random noises added. Adam optimizer is used to minimize the binary cross-entropy loss. The initial learning rate is set to 10^{-3}, and it is gradually reduced with a cosine decay scheduler to reach the final value of 10^{-8}. To handle the softened labels introduced by the random noise, a custom accuracy metric is defined, which rounds the true and predicted values and then compares them together. To avoid overfitting, an Early Stopping callback is leveraged to stop training if the validation loss does not improve for 20 consecutive epochs. At the end of training, the model which has the smallest validation loss will be restored. The number of epochs is set to 100 and 60 for Y site and S/T site prediction, respectively.

4.2 Evaluation metrics

To evaluate the performance of phosphorylation site prediction, this study utilized several widely used statistical metrics, including accuracy (Acc) and F1 score. Numerically, for each of these metrics, a value approaching 100% indicates superior performance.

4.3 Results

4.3.1 Evaluation on S/T site prediction

Table 1 presents the experimental result of models on S/T site dataset. It is clearly shown that increasing the input sequence length L enables higher prediction accuracy, *i.e.*, from nearly 76.5% to 79% for all model architectures. In particular, VGG model achieves higher accuracy than ResNet ($\sim 1.0\%$ with $L = 51$ and DA), and this may be simply explained by its higher number of parameters.

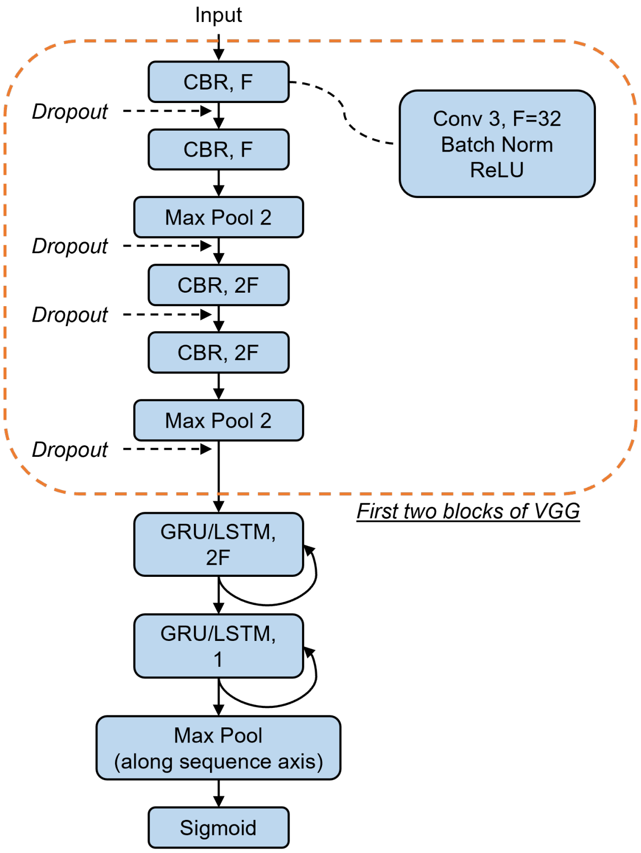
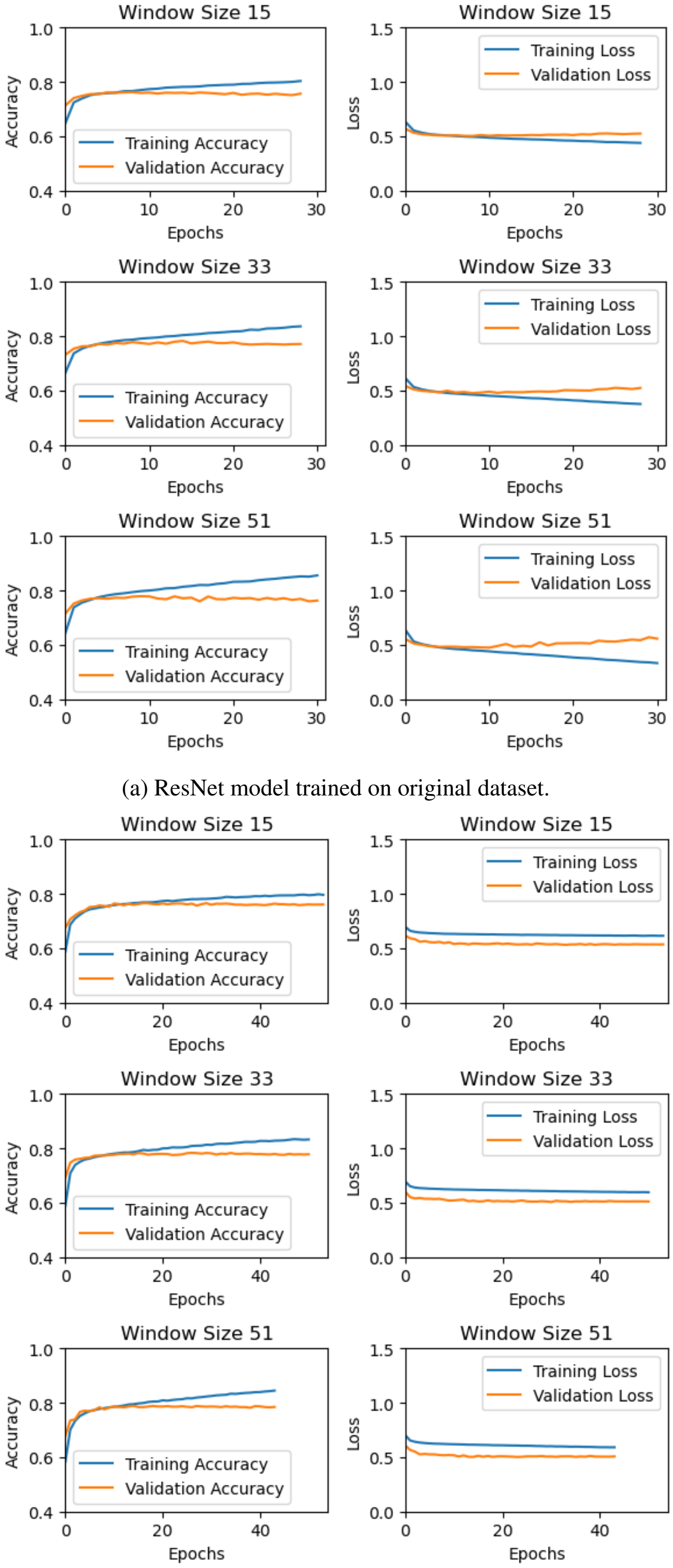


Figure 3: Mixed CNN-RNN model.

Considering mixed convolutional-recurrent models, we can see that CNN-LSTM achieves on-par accuracy with VGG model, while CNN-GRU obtains slightly lower accuracy due to the simpler architecture of the GRU cells. More importantly, our proposed DA scheme offers accuracy gain in all configurations, with the biggest gain of approximately 1% for VGG at $L = 15$. Figure 4 shows the learning curves of ResNet model with and without the proposed DA scheme. Concretely, without DA, the model converges rapidly after only some epochs, then the training loss continues to descend while the validation loss re-increases, resulting in a large generalization gap. The model therefore exhibits strong overfitting. When using DA, the learning procedure is more stable with both training and validation losses steadily decreasing over epochs, resulting in a much smaller generalization gap. This demonstrates that the DA successfully mitigates overfitting and improves the model's performance.

4.3.2 Evaluation on Y site prediction

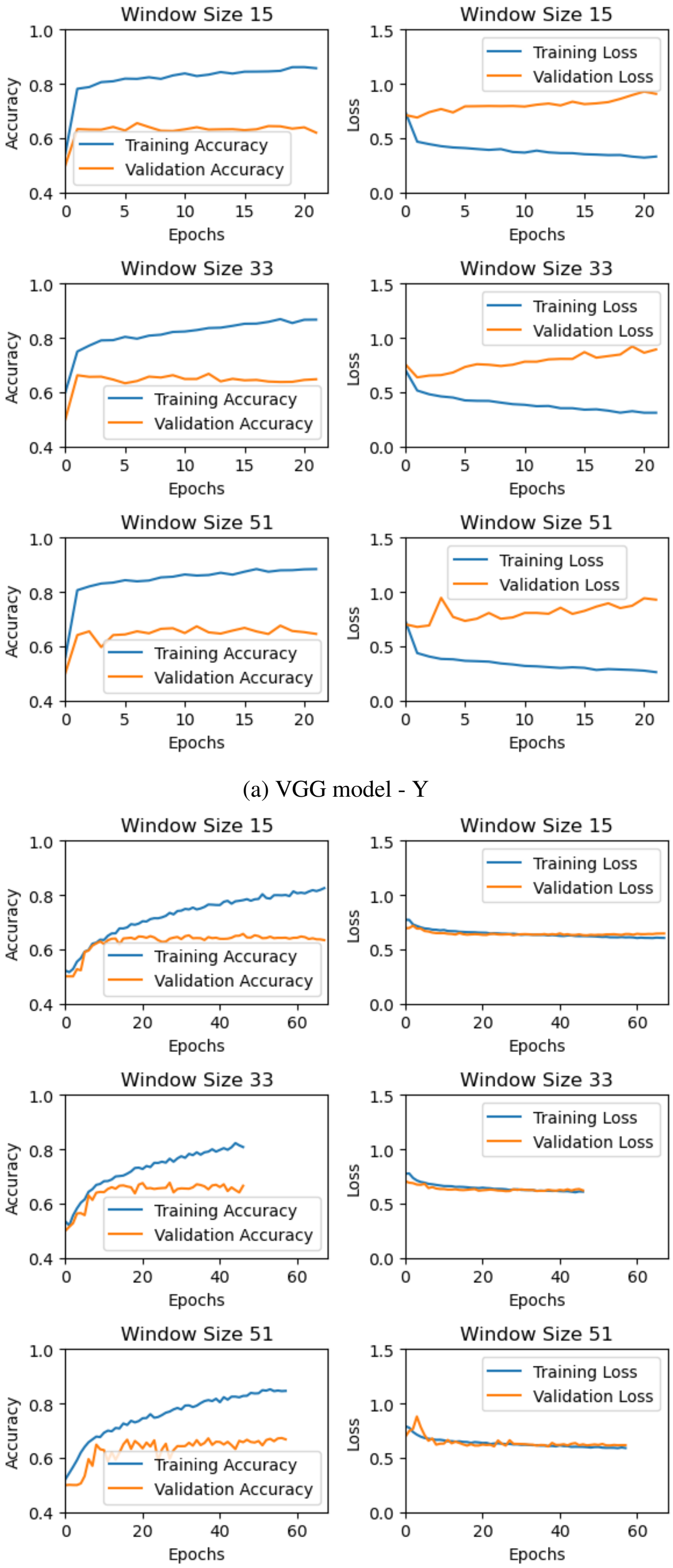
As the Y site dataset contains much fewer samples, all models now achieve lower accuracy compared to their performance on S/T dataset. It is now more complex to assess the impact of input sequence length L. Regarding pure CNN models such as VGG and ResNet, increasing the sequence length does not coherently improve the performance, as VGG achieves highest accuracy ($\sim 68\%$) at $L = 33$ while ResNet even achieves highest accuracy ($\sim 64.50\%$) at $L = 15$. On the contrary, mixed convolutional-recurrent exhibits much better performance when increasing L. In detail, CNN-LSTM achieves 68.69% accuracy when using DA at $L = 51$, corresponding to a gain of nearly 2.7% compared to VGG at the same configuration. This result suggests that compared to pure CNN, the mixed model with recurrent models is more robust in learning sequential dependencies within longer sequences of Y site dataset. Besides, numerical results generally show good impacts of the DA on improving the performance, although in some cases, it may reduce the prediction quality. Figure 5 depicts the learning curves of VGG model on the Y site dataset. We can see that with this small-sized



(a) ResNet model trained on original dataset.

(b) ResNet model trained with data augmentation.

Figure 4: Accuracy and loss curves of ResNet models of S/T sites dataset.



(a) VGG model - Y

(b) VGG model - Y_noise

Figure 5: Accuracy and loss curve of VGG models of Y site dataset.

Table 1: Performance of models on S/T sites dataset

Model	Data config.	Acc (%)	F1 (%)
VGG	$L = 15$	76.69	76.78
(100k params)	$L = 15$, DA	77.61	78.08
	$L = 33$	78.43	78.03
	$L = 33$, DA	79.08	79.63
	$L = 51$	78.98	79.31
	$L = 51$, DA	79.43	79.31
ResNet	$L = 15$	76.30	76.37
(70k params)	$L = 15$, DA	76.44	76.80
	$L = 33$	77.84	77.90
	$L = 33$, DA	78.03	78.21
	$L = 51$	78.10	78.10
	$L = 51$, DA	78.36	78.77
CNN-LSTM	$L = 15$	76.53	76.54
(60k params)	$L = 15$, DA	76.75	77.43
	$L = 33$	78.43	78.68
	$L = 33$, DA	78.74	79.58
	$L = 51$	79.24	79.69
	$L = 51$, DA	79.36	79.75
CNN-GRU	$L = 15$	76.29	76.22
(50k params)	$L = 15$, DA	76.61	76.73
	$L = 33$	78.36	78.55
	$L = 33$, DA	78.88	79.08
	$L = 51$	78.77	78.99
	$L = 51$, DA	79.05	79.26

Table 2: Performance of models on Y sites dataset

Model	Data config.	Acc (%)	F1 (%)
VGG	$L = 15$	62.76	65.09
(100k params)	$L = 15$, DA	65.13	65.08
	$L = 33$	67.28	65.41
	$L = 33$, DA	67.95	68.83
	$L = 51$	65.36	58.41
	$L = 51$, DA	65.95	65.72
ResNet	$L = 15$	64.39	63.96
(70k params)	$L = 15$, DA	63.58	64.70
	$L = 33$	62.09	62.06
	$L = 33$, DA	62.09	63.94
	$L = 51$	64.32	63.97
	$L = 51$, DA	63.35	59.11
CNN-LSTM	$L = 15$	64.02	62.43
(60k params)	$L = 15$, DA	65.06	63.80
	$L = 33$	67.36	69.90
	$L = 33$, DA	66.47	66.91
	$L = 51$	67.73	66.92
	$L = 51$, DA	68.69	70.24
CNN-GRU	$L = 15$	62.39	61.15
(50k params)	$L = 15$, DA	63.43	62.96
	$L = 33$	65.80	63.33
	$L = 33$, DA	65.95	66.32
	$L = 51$	68.03	68.88
	$L = 51$, DA	67.95	66.56

dataset, it is more difficult to train the model without using DA. The generalization gap is notably enlarged, with training loss steadily decreases while validation loss resurges rapidly after only 3 epochs. This phenomenon is successfully mitigated by applying the proposed DA scheme, and we can hardly see the gap between train and validation. We believe that by tuning more carefully the dropout rate combined with the DA, we might improve the performance of models and achieve more stable results.

5 Conclusion

This project highlights the effectiveness of different types of deep learning models in predicting phosphorylation sites. Two main model types have been developed: CNNs with VGG and ResNet, and mixed CNN-RNN with LSTM/GRU layers. Besides, a specific data augmentation scheme is proposed to deal with small-sized datasets. Experimental results show that VGG and CNN-LSTM achieve the best performance on the larger-sized S/T site dataset, offering around 79% accuracy. On the other hand, mixed models achieve much better performance on the smaller-sized Y site dataset, with 68.70% as the highest accuracy delivered by CNN-LSTM. Specifically, the proposed data augmentation scheme successfully alleviates overfitting while stabilizing the learning and thus offering performance gain in almost every configuration. Overall, this suggests the possibility of applying different deep neural network architectures combined with proper data processing and augmentation scheme to learn discriminant features from biological sequence data, offering a valuable alternative to traditional experimental methods for protein function analysis. Extended works may be focusing on the use of generative learning (*e.g.*, generative adversarial networks) to synthesize more protein sequences and investigate model's performance when using this additional source of data.

References

[1] M. C. Annunziata, M. Parisi, G. Esposito, G. Fabbrocini, R. Ammendola, and F. Cattaneo, "Phosphorylation sites in protein kinases and phosphatases regulated by formyl peptide receptor 2 signaling," *International Journal of Molecular Sciences*, vol. 21, no. 11, 2020. [Online]. Available: https://www.mdpi.com/1422-0067/21/11/3818

[2] J. S. Gerritsen and F. M. White, "Phosphoproteomics: a valuable tool for uncovering molecular signaling in cancer cells," *Expert Review of Proteomics*, vol. 18, no. 8, pp. 661–674, 2021, pMID: 34468274.

[3] 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, 01 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty1051

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14124313

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594692

[7] Z. Tao, Z. Yang, B. Chen, W. Bao, and H. Cheng, "Protein sequence classification with letnet-5 and vgg16," in *Intelligent Computing Theories and Application*, D.-S. Huang, K.-H. Jo, J. Jing, P. Premaratne, V. Bevilacqua, and A. Hussain, Eds. Cham: Springer International Publishing, 2022, pp. 687–696.

[8] N. Watanabe, M. Yamamoto, M. Murata, Y. Kuriya, and M. Araki, "Enzymenet: residual neural networks model for enzyme commission number prediction," *Bioinformatics Advances*, vol. 3, no. 1, p. vbad173, 11 2023. [Online]. Available: https://doi.org/10.1093/bioadv/vbad173

[9] N. Auslander, A. B. Gussow, S. Benler, Y. I. Wolf, and E. V. Koonin, "Seeker: alignment-free identification of bacteriophage genomes by deep learning," *Nucleic Acids Research*, vol. 48, no. 21, pp. e121–e121, 10 2020. [Online]. Available: https://doi.org/10.1093/nar/gkaa856

[10] X. L. Liu, "Deep recurrent neural network for protein function prediction from sequence," *bioRxiv*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:10483625

[11] H. Dinkel, C. Chica, A. Via, C. M. Gould, L. J. Jensen, T. J. Gibson, and F. Diella, "Phospho.elm: a database of phosphorylation sites—update 2011," *Nucleic Acids Research*, vol. 39, no. suppl_1, pp. D261–D267, 11 2010. [Online]. Available: https://doi.org/10.1093/nar/gkq1104

[12] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 1291–1303, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:8708461