frontiers in Genetics

ORIGINAL RESEARCH
published: 18 January 2019
doi: 10.3389/fgene.2018.00751

A Novel Protein Subcellular Localization Method With CNN-XGBoost Model for Alzheimer's Disease

Long Pang[1], *Junjie Wang*[2†], *Lingling Zhao*[2*†], *Chunyu Wang*[2] and *Hui Zhan*[3*]

[1] Harbin Nebula Bioinformatics Technology Development Co., Ltd., Harbin, China, [2] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, [3] School of Electronic Engineering, Heilongjiang University, Harbin, China

The disorder distribution of protein in the compartment or organelle leads to many human diseases, including neurodegenerative diseases such as Alzheimer's disease. The prediction of protein subcellular localization play important roles in the understanding of the mechanism of protein function, pathogenes and disease therapy. This paper proposes a novel subcellular localization method by integrating the Convolutional Neural Network (CNN) and eXtreme Gradient Boosting (XGBoost), where CNN acts as a feature extractor to automatically obtain features from the original sequence information and a XGBoost classifier as a recognizer to identify the protein subcellular localization based on the output of the CNN. Experiments are implemented on three protein datasets. The results prove that the CNN-XGBoost method performs better than the general protein subcellular localization methods.

Keywords: protein subcellular localization, deep learning (DL), Conventional Neural Network (CNN), XGBoost, machine learning

Edited by:
Yan Huang,
Harvard Medical School,
United States

Reviewed by:
Yungang Xu,
University of Texas Health Science
Center at Houston, United States
Qiguo Dai,
Dalian Nationalities University, China

***Correspondence:**
Lingling Zhao
zhaoll@hit.edu.cn
Hui Zhan
2000092@hlju.edu.cn

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Neurogenomics, a section of the journal Frontiers in Genetics

Received: *15 September 2018*
Accepted: *31 December 2018*
Published: *18 January 2019*

Citation:
Pang L, Wang J, Zhao L, Wang C and Zhan H (2019) A Novel Protein Subcellular Localization Method With CNN-XGBoost Model for Alzheimer's Disease. Front. Genet. 9:751. doi: 10.3389/fgene.2018.00751

1. INTRODUCTION

The study of neurodegenerative diseases, specifically the Alzheimer's disease(AD) has gained great attention and been addressed widely (Cai et al., 2013; Hu et al., 2017a,b,c). The abnormalities and disorder distribution the compartment or organelle of tau protein and the beta-amyloid protein have been considered to contribute to the pathogenesis of AD. Protein subcellular localization prediction is an essential task in bioinformatics and plays import roles in the further understanding of the relationship among protein locations, their function exhibition, and nosogenesis (Liu et al., 2015; Cheng et al., 2016a, 2018a). Related predictive tools typically use the amino acid sequence information of the protein itself as input to output predicted protein cell sublocalization. It provides information on protein function and gene annotation to aid in the identification of drug targets. The two commonly used methods are: (1) homology-based method and (2) machine learning based method (Wu and Krishnan, 2011; Wu et al., 2014; Zeng et al., 2014; Cheng et al., 2017).

The homology-based method highly depends on the homology of protein sequences, and therefore performs worse for low protein sequence similarity (Wei et al., 2016; Cheng et al., 2018b). The machine learning based methods usually extract some features from the amino acid sequence of the protein (Cheng et al., 2016b; Hu et al., 2018), convert the sequence into a numerical vector, and then use a machine learning model to predict. For example, the most widely used WoLF PSORT software for eukaryotic proteins, characterized by the amino acid composition of the protein, gives the cellular sublocalization of the 32 proteins most similar to the input protein using the k-nearest neighbor algorithm (Horton et al., 2007). There also exist similar methods like BaCelLo (Pierleoni et al., 2006), YLoc (Briesemeister et al., 2010), iLoc-Hum (Chou et al., 2012), and Hum-mPLoc 3.0 (Zhou et al., 2016).

We believe that existing forecasting methods also have some room for improvement. First, the extracted sequence characteristics may not fully reflect the properties of the protein associated with the training task. Second, the current predictions only use information about the protein itself, without considering the interaction between proteins.

In recent years, deep learning has been proven to be a very powerful method by researchers in many fields (LeCun et al., 2015; Xu et al., 2017), like computer vision and natural language processing (Krizhevsky et al., 2012; Mikolov et al., 2013; Sutskever et al., 2014). CNN is an efficient deep learning method due to it can learn high-level features with neural networks. Recently, it also has attracted attentions from researchers and practitioners in bioinformatics. A prediction tool "DeepLoc" (Almagro Armenteros et al., 2017) based on deep learning was proposed with the end-to-end sequence-based model integrated recurrent neural networks (RNNs) with long short-term memory(LSTM) cells, attention models and convolutional neural networks(CNNs), and achieved a better accuracy compared with the traditional machine learning methods. However, the model structure is of high complication, sequentially has too many hyper-parameters to train. Moreover, the proteins in the dataset they constructed have been found to be highly homologous and therefore might provide an overly optimistic model evaluation (Gudenas, 2018). In addition, DeepLoc considers only one possible label for each protein, whereas the protein subcellular location belongs to a multi-label multi-class problem in general.

In this work, we propose a new framework for protein subcellular localization prediction by combining CNN and XGBoost. As an outstanding classifier and feature extractor, CNNs have achieved great success, especially in the field of image recognization. For the protein sequence, CNNs have ability to detect short motifs in the input sequence irrespectively of where they occur and automatically extract features from the original protein sequences. Inspired by this advantage, we also exploit CNN as the feature extractor but a new classifier XGBoost to replace the traditional classifiers connected like the soft-max classifier, since they can not well understand the extracted feature by CNN. XGBoost is an efficient implementation of gradient boosted decision trees (GBDT) due to its block structure to support the parallelization of tree construction. In GBDT, gradient boosting refers to a kind of ensemble technique creating new models to predict the residuals or errors of prior models and making the final decision by the summing up the predictions from all models. Meantime, gradient descent algorithm is also exploited to minimize the loss when adding new models.

The main contribution of our work includes the following aspects:

- We propose a new CNN-XGBoost model for prediction of the protein subcellular localization. The high-level features of protein sequence can be learned by a CNN that can be used by XGBoost classifier for prediction the localization of the subcellular of proteins.
- The experiments conducted on four real datasets consisting of protein sequences show that the proposed method achieves competitive performance.

2. METHODS

In this paper, we propose a novel protein subcellular localization method by integrating the CNN and the XGBoost as a new model for possible application in the pathogenes verification of Alzheimer's disease. The general concept of CNN-XGBoost model is to add an XGboost after the feature layer of a CNN and replace the output layer of the CNN. Our CNN-XGBoost model can automatically extract featutue from the protein sequences and provides more precise localization results. **Figure 1** illustrates the whole structure of the CNN-XGBoost model for protein subcellular localization.

2.1. Convolutional Neural Network

In the field of image analysis, the mask (or filter, or kernel) is an important construct. A *convolution* is an operation involving an initial image and the mask. The operation is equivalent to flipping the mask both vertically and horizontally and then visually placing it over each pixel in turn. The output is the sum over a pixel-wise product of the mask and the sub-image. Masks are usually symmetric, so flipping is unnecessary. Recall from signal processing, the *convolution* between two f and g is given by the following equation.

$$(f * g)(t) \triangleq \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau \tag{1}$$

In image processing, a convolution between an image $\mathbf{I}$ and *kernel* $\mathbf{K}$ of size $d \times d$ centered at a given pixel (x, y) is defined as,

$$(\mathbf{I} * \mathbf{K})(x, y) = \sum_{i=1}^{d}\sum_{j=1}^{d} \mathbf{I}(x+i-d/2, y+j-d/2) \times \mathbf{K}(i,j) \tag{2}$$

Convolutional neural networks are a family of neural network architectures having at least one convolutional layer. *LeNet* is the original CNN network architecture bearing the name of Yann Lecun. Its architecture can be written as,

$$\begin{aligned}
\mathbf{H}_1 &= \sigma(\mathbf{X} * \mathbf{K}^{(1)}) \quad \text{(first convolutional layer)}\\
\mathbf{P}_1 &= \text{maxpool}(\mathbf{H}_1) \quad \text{(first pooling layer)}\\
\mathbf{H}_2 &= \sigma(\mathbf{P}_1 * \mathbf{K}^{(2)}) \quad \text{(second convolutional layer)}\\
\mathbf{P}_2 &= \text{maxpool}(\mathbf{H}_2) \quad \text{(second pooling layer)}\\
\mathbf{F}_1 &= \sigma(\mathbf{W}^{(1)}\mathbf{P}_2 + \mathbf{b}^{(1)}) \quad \text{(first fully-connected layer)}\\
\mathbf{F}_2 &= \sigma(\mathbf{W}^{(2)}\mathbf{F}_1 + \mathbf{b}^{(2)}) \quad \text{(second fully-connected layer)}\\
\mathbf{f}(\mathbf{X}) &= \text{softmax}(\mathbf{W}^{(3)}\mathbf{F}_2 + \mathbf{b}^{(3)}) \quad \text{(output layer)}
\end{aligned}$$

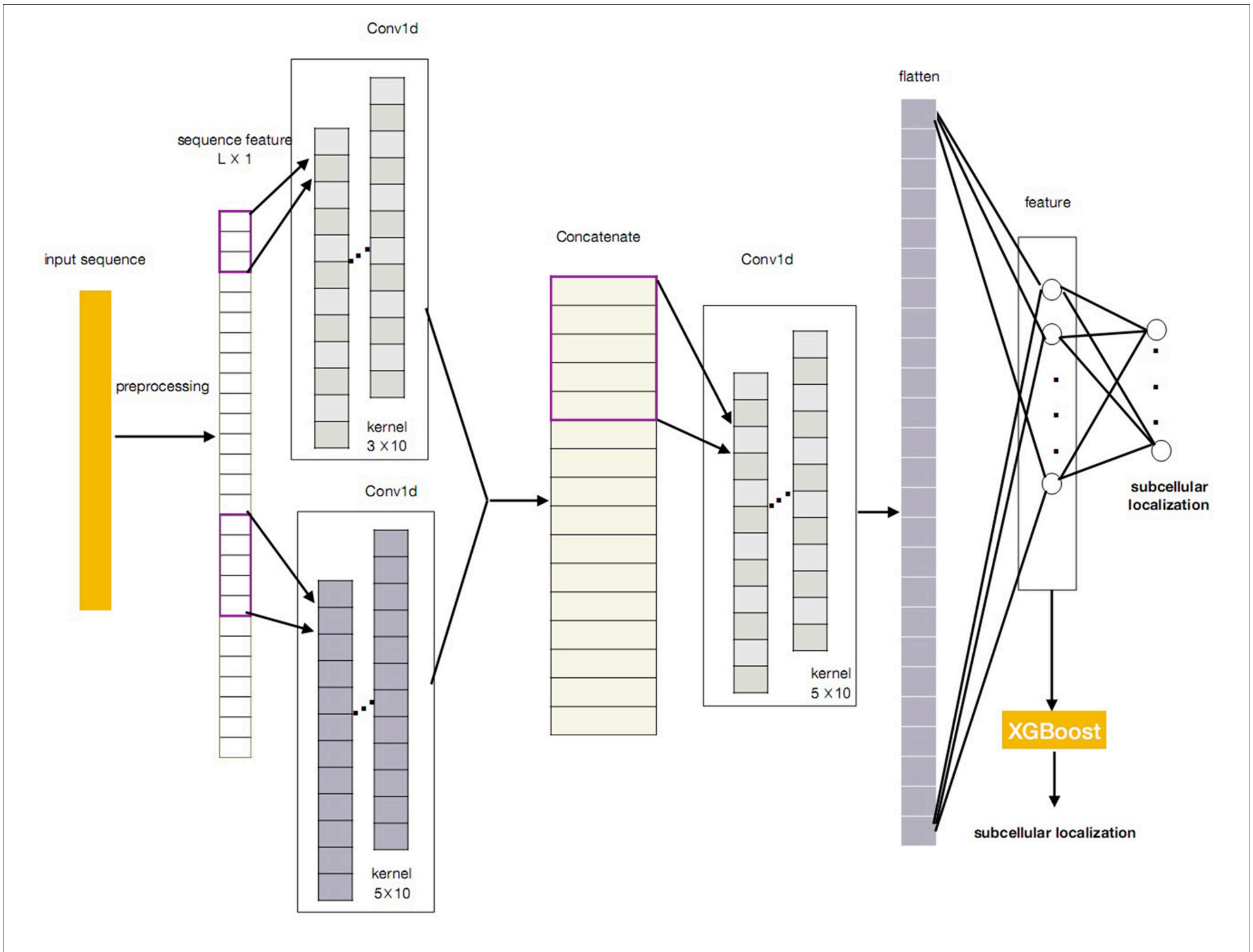

FIGURE 1 | The framework of the CNN-XGBoost based protein subcellular location predictor.

In this architecture, *convolutional layer* is the cornerstone of the CNN, which is a hidden layer where a square grid of weights is convolved with the input, just like an image mask. The output of the convolutional layer is akin to a convolved image. Next, the non-linear activation function, ReLu (REctified Linear Unit), is applied to zero-out any negative values. To reduce the dimension of the feature extracted from the convolutional layer, there is a *pooling* layer emulating *downsampling*. In general, each group of four values or pixels is replaced by the maximum (sometimes the mean) of the four, leaving a single most intense pixel. This pooling method is known as *max pooling*. This sequence of CONV−>RELU−>POOL layers may be repeated multiple times to create a deep architecture. Finally, a few fully-connected layers round off the architecture. Though it seems far more sophisticated than a MLP, it can be shown that a CNN can be represented as a classical fully-connected neural network. For example, a convolutional layer can be represented as a sparse fully-connected layer. Various techniques have been developed for training these vast models, for example momentum optimizers, weight initialization, batch normalization, and dropout.

Convolutional Neural Networks are the current state-of-the-art in many computer vision tasks. In addition to image classification, their great success has attracted wide attention in many fields. It has been found that using a pre-trained CNN as a general-purpose feature extractor for a simple linear model can yield significant improvements over even the most meticulously hand-crafted feature engineering.

The protein subcellular localization problem can be viewed as a multi-label multi-class classification task. Unlike the ordinary deep learning methods for multi-classification problems, in our method, we need to change the loss function. The most intuitive way is to extend the cross-entropy loss. The cross-entropy loss function is defined by

$$\min_{\Theta} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L} y_{i,j}\log(\hat{p}_{ij}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j\in y_i^+}\frac{1}{|y_i^+|}log(\hat{p}_{ij}) \quad (3)$$

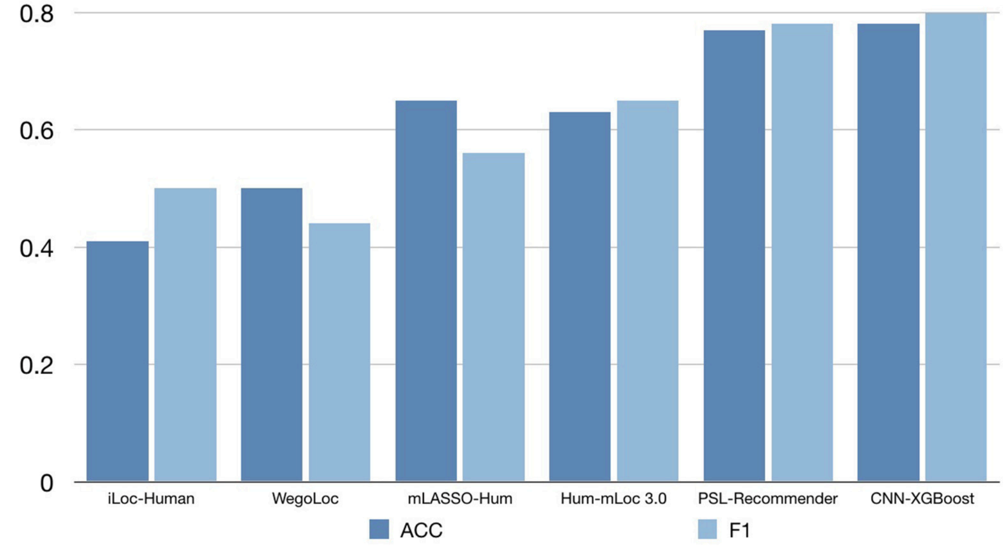

FIGURE 2 | The accuracy comparison on the Hum-mPloc 3.0 data set.

where Θ denotes the parameters of CNN model, y_i^+ is a set that contains the relevant localization of protein i and $\hat{p}_{ij}$ is the result for protein i on localization j, through a softmax activation:

$$\hat{p}_{ij} = \frac{\exp(f_j(x_i))}{\sum_{j'=1}^{L}\exp(f_{j'}(x_i))} \tag{4}$$

Instead of using the cross-entropy loss function, the binary cross-entropy loss (BCE) over sigmoid activation has shown better performance when applied into multi-label task. The binary cross-entropy loss is

$$\min_{\Theta} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L}[y_{i,j}\log(\sigma(f_{ij})) + (1-y_{ij})\log(1-\sigma(f_{ij}))] \tag{5}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$

2.2. Tree Boosting and XGBoost

Tree boosting is a learning method to enhance the classification ability of weak classifiers by iteratively adding new decision trees to the ensembles of decision trees. Let $D = \{(x_i, y_i)\}(|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}^n)$ denotes a dataset with n classes and m feature. Then the prediction of a tree boosting for a (x_i, y_i) is given by

$$\hat{y}_i = g_A(x_i) = \sum_{j=1}^{M} g_j(x_i) \tag{6}$$

where $g_j(x_i) = w_q(x_i)$ is the prediction of the j-th decision tree with leaf weights w_q on a datapoint x_i, and M is the number of members in the ensemble.

It is well-known that the decision tree tends to overfit when the decision tree is fully grown. Thus, the set prediction function of decision trees g_j can be learned by minimizing the objective function

$$\mathcal{C}(x, g_A) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{M}\Omega(g_j) \tag{7}$$

where $l_i(y_i, \hat{y}_i)$ is a term which measures the goodness of the prediction $\hat{y}_i$ and the object y_i. $\Omega(g_j)$ is a regularization term that does not depend on the data.

XGBoost implements a parallel tree boosting in a fast and accurate way. In XGBoost, the regularization function is chosen to be

$$\Omega(g) = \gamma T + \frac{\lambda}{2}\sum_{l=1}^{T} w_l^2 \tag{8}$$

with γ and λ regularization parameters that must be chosen appropriately. Notice this regularization penalizes both large weights on the leaves (similar to L^2-regularization) and has large partitions.

As mentioned above, the tree boosting iteratively enlarges the ensemble of decision trees, then the prediction of the t-th iteration can be defined as

$$\hat{y}_i^{(t)} = \sum_{j=1}^{t} g_j(x_i) = \hat{y}_i^{(t-1)} + g_t(x_i) \tag{9}$$

The objective function (7) at step t can be modified as

$$\mathcal{C}_t = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + g_t(x_t)) + \Omega(g_t) \tag{10}$$

Apply a Taylor expansion on the objective function (10) to second order and then the final objective function at step t can be approximated as

$$\mathcal{C}_t \approx \mathcal{C}_{t-1} + \Delta\mathcal{C}_t \tag{11}$$

$$= \mathcal{C}_{t-1} + b_i l(y_i, \hat{y}_i^{(t-1)}) g_t(x_i) + \frac{1}{2} a_i g_t(x_i)^2 + \Omega(g_t) \tag{12}$$

where

$$a_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \tag{13}$$

$$b_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \tag{14}$$

Let $j : I_j = \{i : q_t(x_i) = j\}$ denotes the set of point x_i mapped to leaf, $B_j = \sum_{i \in I_j} b_i$ and $A_j = \sum_{i \in I_j} a_i$. Then we can rewrite the $\Delta\mathcal{C}_t$ as

$$\Delta\mathcal{C}_t = \sum_{j=1}^{T} [B_j w_j + \frac{1}{2}(A_j + \lambda_j) w_j^2] + \lambda T \tag{15}$$

To find the optimal weight w_j of leaf j for a fixed tree structure, $q(x)$ can be obtained by applying the following equation

$$w_j^{opt} = -\frac{B_j}{A_j + \lambda} \tag{16}$$

plugging back into $\Delta\mathcal{C}_t$ gives

$$\Delta\mathcal{C}_t^{opt} = -\frac{1}{2} \sum_{j=1}^{K} \frac{B_j^2}{A_j + \lambda} + \gamma T \tag{17}$$

It is clear that $\Delta\mathcal{C}_t^{opt}$ measures the in-sample performance of g_t and we should find the decision tree that minimizes this value. However, in practice, this is impossible to enumerate all possible trees over the data and find the tree which can minimize $\Delta\mathcal{C}_t^{opt}$. Instead, an approximate greedy algorithm runs to optimize one level of the tree at a time by trying to find optimal splits of the data, leading to a tree with a local minimum of $\Delta\mathcal{C}_t^{opt}$, which is then added to the ensemble.

For the multi-label multi-class classification problem, we utilize XGBoost as classifiers and adopt the binary relevance strategy (Boutell et al., 2004) to construct m binary classifiers.

2.3. CNN-XGBoost Model

Figure 1 gives the overall structure of the CNN-XGBoost model for protein subcellular location prediction. The input of the model is a one-dimensional vector and constructed by the position specific scoring matrices (PSSM) and proteins interaction scoring matrix which are extracted from STRING and GO terms semantic similarities. On this basis, a protein can be expressed as $L \times 1$ vector (L is the number of sequences in training set), analog image data equivalent to a protein is a one-dimensional "image" with 1 channels. So the input is a $L \times 1$ matrix.

After obtaining the proper feature representations by the trained CNN, compared with the classic CNN, our CNN-XGBoost model replaces the soft-max layer of CNN with XGBoost to predict the localization of subcellular of proteins, which enables features automatically obtained from input and provides more precise and efficient classification.

3. RESULTS

3.1. Dataset

To verify the performance of our method, we employ three protein datasets: the Hum-mPloc3.0, the BaCelLo animals, and the Hoglund. **Table 1** gives the details of these datasets. The train set of Hum-mPloc 3.0 consists of 3,122 proteins and 1,023 proteins own more than one label. The test set of Hum-mPloc 3.0 consists of 379 proteins, among which 120 proteins belong to multi-label proteins. Each protein in Hum-mPloc 3.0 is assigned at least one label of 12 subcellular locations (Centrosome, Cytoplasm, Cytoskeleton, Endoplasmic reticulum, Endosome, Extracellular, Golgi apparatus, Lysosome, Mitochondrion, Nucleus, Peroxisome, and Plasma membrane).

For the BaCelLo dataset, there are four subcellular locations: Cytoplasm, Mitochondrion, Nucleus, and Secreted. The size of the training set is set to 2,597 and the testing set consists of 576 proteins. All the proteins of BaCelLo dataset are of a single label. In the Hoglund dataset, the training set includes nine subcellular locations (Nucleus, Cytoplasm, Mitochondrion, Endoplasmic reticulum, Golgi apparatus, Peroxisome, Plasma membrane, Extracellular space, Lysosome, and Vacuole), and the test consists of 158 proteins with six subcellular locations (Endoplasmic reticulum, Golgi apparatus, Peroxisome, Plasma membrane, Extracellular space, and Lysosome).

3.2. Measurements

A widely-applied method for evaluating a mutli-label multi-class classifier is to compute the ACC and F1 values. ACC is the average of ACC_{x_i} of all proteins in the testing set, calculated for protein x_i is

$$\mathrm{ACC}_{x_i} = \frac{\mathrm{TP}_{x_i}}{\mathrm{TP}_{x_i} + \mathrm{FP}_{x_i} + \mathrm{FN}_{x_i}} \tag{18}$$

where TP, FP, and FN are true positive, false positive, and false negative, respectively. The F1 score considers both the harmonic mean of precision and recall of subcellular location y_j, defined as follows:

$$\begin{aligned}
\mathrm{precision}_{y_j} &= \frac{\sum_{x_i \in P_j} \frac{\mathrm{TP}_{x_i}}{\mathrm{TP}_{x_i} + \mathrm{FP}_{x_i}}}{|P_j|} \\
\mathrm{recall}_{y_j} &= \frac{\sum_{x_i \in T_j} \frac{\mathrm{TP}_{x_i}}{\mathrm{TP}_{x_i} + \mathrm{FN}_{x_i}}}{|T_j|} \\
\mathrm{F1}_{y_j} &= \frac{2 \times \mathrm{precision}_{y_j} \times \mathrm{recall}_{y_j}}{\mathrm{recall}_{y_j} + \mathrm{precision}_{y_j}}
\end{aligned} \tag{19}$$

where T_j and P_j are the set of proteins for true location y_j and the set of proteins for predicted locations y_j respectively.

TABLE 1 | Dataset Summary.

	Hum-mLoc 3.0		BaCelLo		Hoglund	
	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**
No. Proteins	3,126	379	2,597	576	5,959	158
No. Labels	4,229	541	2,597	576	5,959	158
No.Locations	12		4		6	

TABLE 2 | Comparision of CNN-XGBoost on Hum-mPloc 3.0 dataset with other methods.

Location	iLoc-Human			WegoLoc			mLASSO-Hum			Hum-mLoc 3.0			PSL-Recommender			CNN-XGBoost		
	pre	re	F1	pre	re	F1	pre	re	F1	pre	re	F1	pre	re	F1	pre	re	F1
Centrosome	0	0	0	0.75	0.14	0.23	0.59	0.59	0.59	0.75	0.55	0.63	0.94	0.75	**0.83**	0.79	0.50	0.61
Cytoplasm	0.5	0.54	0.52	0.69	0.53	0.60	0.93	0.51	0.66	0.76	0.73	0.74	0.79	0.81	0.80	0.85	0.89	**0.87**
Cytoskeleton	0	0	0	0.32	0.34	0.33	0.9	0.22	0.35	0.8	0.68	0.74	0.93	0.77	0.84	0.89	0.80	**0.85**
ER	0	0	0	0.73	0.2	0.31	0.74	0.49	0.59	0.83	0.37	0.51	0.9	0.7	0.79	0.97	0.71	**0.82**
Endosome	0	0	0	0.25	0.07	0.11	0.38	0.2	0.26	0.58	0.47	**0.52**	0.57	0.37	0.45	0.80	0.27	0.40
Extracellular	0.62	0.62	0.62	0.67	0.77	**0.71**	0.16	0.69	0.26	0.5	0.46	0.48	0.66	0.71	0.68	0.80	0.62	0.70
Golgi apparatus	0.6	0.3	0.4	0.6	0.15	0.24	0.72	0.65	0.68	0.69	0.45	0.55	0.88	0.61	**0.72**	0.80	0.60	0.69
Lysosome	0.5	0.13	0.2	0.2	0.13	0.15	0.55	0.75	0.63	0.71	0.63	0.67	1	0.55	0.71	1.00	0.75	**0.86**
Mitochondrion	0.95	0.33	0.49	0.79	0.73	0.76	0.83	0.88	0.85	0.78	0.75	0.76	0.92	0.88	**0.90**	0.96	0.80	0.87
Nucleus	0.54	0.7	0.61	0.65	0.64	0.64	0.85	0.7	0.76	0.75	0.71	0.73	0.81	0.92	**0.87**	0.83	0.91	**0.87**
Peroxisome	1	0.5	0.67	0.5	1	0.67	0.29	1	0.44	1	1	**1**	1	1	**1**	1	1	**1**
Plasma membrane	0.42	0.33	0.37	0.44	0.53	0.48	0.58	0.56	0.57	0.65	0.44	0.52	0.78	0.74	0.76	0.89	0.75	**0.81**
ACC-mean		0.41			0.50			0.65			0.63			0.77			**0.78**	
F1-mean		0.32			0.44			0.56			0.65			0.78			**0.80**	

The bold marks the first best result and the underline marks the second best result.

TABLE 3 | Comparison of CNN-XGBoost ACC/F1-mean on other proteins datasets with other methods.

	BaCelLo	Hoglund
MultiLoc2-LowRes	0.73/0.76	–
MultiLoc2-HighRes	0.68/0.71	0.57/0.41
BaCelLo	0.64/0.66	–
Hum-mPloc 3.0	0.86/0.84	0.64/0.59
PSL-Recommender	0.94/0.92	0.92/0.90
CNN-XGBoost	**0.94/0.94**	**0.94/0.92**

The bold marks the first best result and the underline marks the second best result.

3.3. Results and Discussions

To verify the performance of our approach, some typical protein subcellular location tools including Hum-mPLoc 3.0 (Zhou et al., 2016), YLoc+ (Briesemeister et al., 2010), iLoc-Hum (Chou et al., 2012) , WegoLoc (Chi and Nam, 2012), mLASSO-Hum (Wan et al., 2015), and PSL-Recommender (Jamali et al., 2018) were compared to our method. The F1 score and ACC for each subcellular localization are summarized in **Table 2** and **Figure 2** for Hum-mploc 3.0 dataset. As seen in **Table 2** and **Figure 2**, the CNN-XGBoost outperforms the mean value of F1 score and ACC of all other methods. Also, in 7 out of 12 subcellular locations, CNN-XGBoost has the best performance among all the methods while in the other three locations it has the second best performance. It is only in centrosome and endosome that CNN-XGBoost shows unsatisfactory results. As seen in **Table 3**, the CNN-XGBoost slightly outperforms the second best method by both mean F1 score and ACC.

In addition, we also evaluated our method on the DeepLoc dataset, compared to the DeepLoc, our method provides slightly better prediction with significantly lighter model, meanwhile, it is known that DeepLoc can not handle multilabel multiclass problem, whereas our method still shows outstanding performance.

4. CONCLUSIONS

In order to make balance of the classification performance and the complexity when training the model for the protein subcellular location in Alzheimer's disease, this paper proposes a prediction framework integrating CNN and XGBoost, taking advantage of the outstanding ability of feature expression of CNN, and the good classification performance of XGBoost. Experiments are implemented on the Hum-mPloc3.0, the BaCelLo animals, and the Hoglund database, and the results demonstrate that the new method outperforms the typical machine learning based tools. Further work will focus on the verification of our model on more datasets, especially the datasets related to Alzheimer's disease, and the optimization of the structure of CNN utilized in the model.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (NSFC, Grant no. 61305013 and 61872114). The authors would like to thank the reviewers for their valuable comments.

REFERENCES

Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. (2017). DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics* 33, 3387–3395. doi: 10.1093/bioinformatics/btx431

Boutell, M. R., Luo, J., Shen, X., and Brown, C. (2004). M. Learning multi-label scene classification. *Pattern Recogn.* 37, 1757–1771. doi: 10.1016/j.patcog.2004.03.009

Briesemeister, S., Rahnenführer, J., and Kohlbacher, O. (2010). YLoc–an interpretable web server for predicting subcellular localization. *Nucleic Acids Res.* 38, W497–W502. doi: 10.1093/nar/gkq477

Cai, S., Yang, S., Zheng, F., Lu, M., Wu, Y., and Krishnan, S. (2013). Knee joint vibration signal analysis with matching pursuit decomposition and dynamic weighted classifier fusion. *Comput. Math. Methods Med.* 2013:904267. doi: 10.1155/2013/904267

Cheng, L., Hu, Y., Sun, J., Zhou, M., and Jiang, Q. (2018a). DincRNA: a comprehensive web-based bioinformatics toolkit for exploring disease associations and ncRNA function. *Bioinformatics* 34, 1953–1956. doi: 10.1093/bioinformatics/bty002

Cheng, L., Jiang, Y., Ju, H., Sun, J., Peng, J., Zhou, M., et al. (2018b). InfAcrOnt: calculating cross-ontology term similarities using information flow by a random walk. *BMC Genomics* 19:919. doi: 10.1186/s12864-017-4338-6

Cheng, L., Jiang, Y., Wang, Z., Shi, H., Sun, J., Yang, H., et al. (2016a). DisSim: an online system for exploring significant similar diseases and exhibiting potential therapeutic drugs. *Sci. Rep.* 6:30024. doi: 10.1038/srep30024

Cheng, L., Sun, J., Xu, W., Dong, L., Hu, Y., and Zhou, M. (2016b). OAHG: an integrated resource for annotating human genes with multi-level ontologies. *Sci. Rep*. 10:34820. doi: 10.1038/srep34820

Cheng, L., Yang, H., Zhao, H., Pei, X., Shi, H., Sun, J., et al. (2017). MetSigDis: a manually curated resource for the metabolic signatures of diseases. *Brief. Bioinformatics*. doi: 10.1093/bib/bbx103. [Epub ahead of print].

Chi, S.-M., and Nam, D. (2012). Wegoloc: accurate prediction of protein subcellular localization using weighted gene ontology terms. *Bioinformatics* 28, 1028-1030. doi: 10.1093/bioinformatics/bts062

Chou, K.-C., Wu, Z.-C., and Xiao, X. (2012). iloc-hum: using the accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites. *Mol. Biosyst.* 8, 629-641.

Gudenas, B. L. (2018). *Genomic Data Mining for Functional Annotation of Human Long Noncoding RNAs*. Avaialble online at: https://tigerprints.clemson.edu/all_dissertations/2146

Horton, P., Park, K. J., Obayashi, T., Fujita, N., Harada, H., Adams-Collier, C. J., et al. (2007). WoLF PSORT: protein localization predictor. *Nucleic Acids Res.* 35(Suppl. 2), W585–W587. doi: 10.1093/nar/gkm259

Hu, Y., Cheng, L., Zhang, Y., Bai, W., Wang, T., Han, Z. et al. (2017a). Rs4878104 contributes to Alzheimer's disease risk and regulates DAPK1 gene expression. *Neurol. Sci*. 38, 1255–1262. doi: 10.1007/s10072-017-2959-9

Hu, Y., Zhao, T., Zhang, N., Zang, T., Zhang, J., and Cheng, L. (2018) Identifying diseases-related metabolites using random walk. *BMC Bioinformatics* 19(Suppl. 5):116. doi: 10.1186/s12859-018-2098-1

Hu, Y., Zheng, L., Cheng, L., Bai, W., Wang, T., Han, Z., et al. (2017b). GAB2 rs2373115 variant contributes to Alzheimer's disease risk specifically in European population. *J. Neurol. Sci.* 375, 18–22. doi: 10.1016/j.jns.2017.01.030

Hu, Y., Zhou, M., Shi, H., Ju, H., Jiang, Q., and Cheng, L. (2017c). Measuring disease similarity and predicting disease-related ncRNAs by a novel method. *BMC Med. Genomics* 10(Suppl. 5):71. doi: 10.1186/s12920-017-0315-9

Jamali, R., Eslahchi, C., and Jahangiri-Tazehkand, S. (2018). Psl- recommender: protein subcellular localization prediction using recommender system. *bioRxiv* 462812. doi: 10.1101/462812

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 1097–1105.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521:436. doi: 10.1038/nature14539

Liu, X., Li, Z., Liu, J., Liu, L., and Zeng, X. (2015). Implementation of arithmetic operations with time-free spiking neural P systems. *IEEE Trans. Nanobiosci.* 14, 617–624. doi: 10.1109/TNB.2015.2438257

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 3111–3119.

Pierleoni, A., Martelli, P. L., Fariselli, P., and Casadio, R. (2006). BaCelLo: a balanced subcellular localization predictor. *Bioinformatics* 22, e408–e416. doi: 10.1093/bioinformatics/btl222

Sutskever, I., Vinyals, O., and Le, Q. (2014). V. "Sequence to sequence learning with neural networks, " in *Advances in Neural Information Processing Systems* (Montreal, QC), 3104–3112.

Wan, S., Mak, M.-W., and Kung, S.-Y. (2015). mlasso-hum: a lasso-based interpretable human-protein subcellular localization predictor. *J. Theor. Biol.* 382, 223-234.

Wei, L., Liao, M., Gao, X., Wang, J., and Lin, W. (2016). mGOF-loc: a novel ensemble learning method for human protein subcellular localization prediction. *Neurocomputing* 217, 73-82. doi: 10.1016/j.neucom.2015.09.137

Wu, Y., and Krishnan, S. (2011). Combining least-squares support vector machines for classification of biomedical signals: a case study with knee-joint vibroarthrographic signals. *J. Exp. Theor. Artif. Intell.* 23, 63–77. doi: 10.1080/0952813X.2010.506288

Wu, Y., Luo, X., Zheng, F., Yang, S., Cai, S., and Ng, S. C. (2014). Adaptive linear and normalized combination of radial basis function networks for function approximation and regression. *Math. Probl. Eng.* 2014:913897. doi: 10.1155/2014/913897

Xu, Y., Wang, Y., Luo, J., Zhao, W., and Zhou, X. (2017). Deep learning of the splicing(epi) genetic code reveals a novel candidate mechanism linking histone modifications to ESC fate decision. *Nucleic Acids Res.* 45, 12100–12112. doi: 10.1093/nar/gkx870

Zeng, X., Zhang, X., Song, T., and Pan, L. (2014). Spiking neural P systems with thresholds. *Neural Comput.* 26, 1340–1361. doi: 10.1162/NECO_a_00605

Zhou, H., Yang, Y., and Shen, H.-B. (2016). Hum-mploc 3.0: prediction enhancement of human protein subcellular localization through mod- eling the hidden correlations of gene ontology and functional domain features. *Bioinformatics* 33, 843-853.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.