Deep Learning Approach for Secondary Structure Protein Prediction based on First Level Features Extraction using a Latent CNN Structure

Adil Al-Azzawi
Electrical Engineering and Computer Science (EECS)
University of Missouri-Columbia
Columbia, MO, 65203, USA

***Abstract*—In Bioinformatics, Protein Secondary Structure Prediction (PSSP) has been considered as one of the main challenging tasks in this field. Today, secondary structure protein prediction approaches have been categorized into three groups (Neighbor-based, model-based, and meta predicator-based model). The main purpose of the model-based approaches is to detect the protein sequence-structure by utilizing machine learning techniques to train and learn a predictive model for that. In this model, different supervised learning approaches have been proposed such as neural networks, hidden Markov chain, and support vector machines have been proposed. In this paper, our proposed approach which is a Latent Deep Learning approach relies on detecting the first level features based on using Stacked Sparse Autoencoder. This approach allows us to detect new features out of the set of training data using the sparse autoencoder which will have used later as convolved filters in the Convolutional Neural Network (CNN) structure. The experimental results show that the highest accuracy of the prediction is 86.719% in the testing set of our approach when the backpropagation framework has been used to pre-trained techniques by relying on the unsupervised fashion where the whole network can be fine-tuned in a supervised learning fashion.**

Keywords*—Secondary structure protein prediction; secondary structure; fine-tuning; Stacked Sparse; Deep Learning; CNN

I. Introduction

Bioinformatics implicates the technology of using the computer aid based system for many reasons such as storage, retrieval, manipulation, and distribution of information. Biological macromolecules such as DNA, RNA, and proteins, are the most related branch of the bioinformatics which is related to the information distribution systematic. The emphasis here is on using the computers aid system to solve these issues since most of the task genomic data analysis are highly repetitive and mathematically complex, computers aid system here is essentially using in mining genomes in terms of information gathering and knowledge building [1]. Although, protein structure prediction methods are classified under the bioinformatics category.

Bioinformatics is a board filed that takes in many other fields and disciplines such as information technology, biology, biochemistry, statistics, and mathematics [2].

Bioinformatics category for protein prediction is depends on the main types of protein structure which are divided into four main types, Primary, Secondary, Tertiary and Quaternary structures. Primary structure is the first type of protein structure which consisting of 20 different types of amino acids. This structure provides foundation information about all the other suture types. The second type of protein structure is the Secondary structure. This type describes and illustrates the arrangement of the connection and attaches within the amino acid groups. It consists of three different structures which are (H, E, and C) [3]. Protein Secondary Structure Prediction (PSSP) Tertiary structure which is the third structure type, provides useful information about protein activity, relationship, and function [3]. That has been done by protein folding which is a prediction of the Tertiary structure. This information can be predicted from linear sequence protein process method which is an unsolved and ubiquitous problem. This approach invites research from many fields of study such as computer science, molecular biology, biochemistry, and physics. The disinfectant information of the Secondary structure use in many proteins folding prediction approaches which is also used in many different area of bioinformatics application [4]. Proteome and gene annotation which is the determination of protein flexibility are the main scientific applications that applied in this area because when searching in a database with peptide mass tags, there is a lack of flexibility in the search programs. In another word, if a single mistake is made during the searching in the assignment of a y- or b-ion which can be possibly happen quite frequently, the amino acid sequence will be incorrect that means the database searching process will bring up irrelevant proteins items. Sub cloning of protein fragments for expression is another application area of this approach which is the assessment of evolutionary trends among organisms [3] [4]. In other hands, Protein Secondary Structure Prediction (PSSP) is an active and significant reach area for many useful applications these days which includes protein integral and analysis [4].

In past years, multi-layer neural networks have been one the popular deep learning approaches. The idea of constructing the network with several levels of nonlinearity to solve more complex problems is not new. [3] However, it is difficult in practice, particularly for deep architectures which have an optimization issue where the expected gain beyond one or two hidden layers is difficult to get [5]. In general, autoencoder is

an unsupervised approach of the neural network that also relies on a back-propagation learning approach [5]. By giving only unlabeled training dataset $\{x_1, x_2, \dots x_n\}$, where $x_i \in R^d$, the autoencoder neural network attempts to learn the identity function of the data samples $f_{W,b}(x) \approx x$ by setting the outputs equal to the inputs, i.e. $y_i = x_i$. If some constraints have been added on the structure of the autoencoder, like limit number of hidden neurons or average rate of firing, the learned identity function will reveal the interesting underlying structure of the data. For example, the activations of the deepest hidden layer can be extracted as new features corresponding to the compressed representation of the input muck like the principal component analysis (PCA) [6].

In this paper, the stacked sparse autoencoder model is introduced and explained after defining the multi-hidden-layer sparse autoencoder model and the stacked Pre-training Method in the second section 2. Then it is followed by section 3 where the CNN model is explained. Finally, in section 5, the summary and discussion of the experimental results.

A. Our Approach Motivation

In this paper, a latent approach using supervised and unsupervised machine learning methods for secondary structure of protein prediction is proposed. A Deep Learning approach using Convolutional Neural Network (CNN) [7] is used as a main structure to build our Latent Deep Learning model for protein prediction. The proposed model relies on an unsupervised learning approach, Stacked Sparse Autoencoder [8] network structure for both 3-state SS first level feature structure detection and prediction using soft-max classifier. Then, compare the results with the latent model by using two training frameworks. The Latent Deep Learning Approach (LDLA) that is proposed for secondary structure of protein prediction relies on using the first level of proteins features that already have been extracted to construct new convolutional filters that will have used in the convolutional layer in the Deep Conditional Neural Network structure (CNN) [7]. Our Latent Deep Learning approach learns not only the complex sequence-structure but also captures the relationship of the models SSlabel correlation through adjacent residues.

The combination between the Deep Learning and Stacked Sparse Autoencoder produces a new data dimension which has been extracted from the first level of the sparse autoencoder network. Those features are used to build convolutions filters that use later in the convolutional layer in the Deep Learning structure. The proposed system implementation differs from Cheng's method [9] instead of using just a typical Deep Learning network, a Latent Deep Convolutional Network (CNN) after some filters has been learned by applying sparse autoencoder. The implementation of the Latent Deep Learning Approach is done by relying on a convolutional filter that construction by using sparse autoencoder as a preprocessing and feature extraction step, which can capture longer-range sequence information than Cheng's method. Our experimental results show that our implementation has greatly achieves the state-of-the-art, especially on those structures whose types are significant challenging to predict.

TABLE. I. SECONDARY STRUCTURE ASSIGNMENT

3 state classes	Abbreviation	DSSP class
α-helix	H	H
310 helix	G	H
β-sheet (E)	E	E
Isolated β _ ridge	B	E
π − helix	I	C
Turn	T	C
Bend	S	C

B. Background and motivation

Today, secondary structure of protein prediction can be classified to three classes. The classified classes are model-based, neighbor-based, and meta predicator-based [2]. The first approach (neighbor-based) predicts the secondary structure by depending on sequence identifying of similar sequence. The second approach (model-based) implements an advanced machine learning model to learn and build a decent model for sequence structure detection [3]. The third model is the meta predicator-based approach which depends on a combination the results of the neighbor model-based approach because basically this model is a method used to make a prediction by integrating the prediction results of several methods [5]. Obviously, the most useful and successful model-based approach is proposed by PSIPRED [4] which was based on using neural network as a learning model [5] and support vector machine [6] that has been tested and showed a decent performance results [7].

C. Dataset

In this paper, the SCRATCH protein predictor dataset is used an a large scale protein dataset. This dataset consists of primary and secondary structure of protein data (SSpro) with 3 classes. The SSpro data has server homologous protein's secondary structure information. The recent and current accuracy that has been achieved in this data set is about 79% correctly classified, and override about 92% correctly classified [10].

D. Secondary Structure Classes Assignment

Given the 3D atomic coordinate of a protein structure, there are several methods to assign its secondary structures including a dictionary of secondary such as the structure of proteins (DSSP) [6] and Structural Identification (STRIDE) [7]. The secondary structure assignment of each residue is not perfectly well-defined, which means that these methods often disagree on their assignments. For example, DSSP and STRIDE differ on approximately 5% of residues [8]. This inconsistency justifies the need for a certain and standard assignment techniques (methods) that could be used to provide alternate definition of protein amino-acid boundaries. The method was adopted here is DSSP as the standard algorithm and the most frequently used for secondary structure definition method. The Neural Network (NN) is trained to predict a three-category (H, E and C) of the secondary structure assignment which has been reduced from the eight-category assignment which is produced by using DSSP method that has been shown in Table1.

II. Related Works

Little work has been done on secondary structure protein prediction using "SSpro-3 classes" sets [10].

Christophe et al. [11]: This paper presents an approach of training model to predict the secondary structure of protein prediction. This work depends on the distinction of the sequence similarity from the sequence profiles at the input stage and an additional structure based similarity. Multi-class prediction approach has been proposed using SSpro8 and SCCpro20. This work achieved about 79 and 80%. The accuracy of SSpro rises to 92.9% (90% for ACCpro).

Jian Zhou et al [12] in this work, the uniquely architecture of prediction model depends on the low-level labels structured has been proposed. The secondary structure of each amino acid has been trained and tested in this model. This model achieved about 66.4% Q8 accuracy on the dataset.

III. Proposed System

The proposed Latent Deep Learning Approach (LDLA) for secondary structure of protein prediction has is shown in Fig.1. A Latent Deep Learning model relies on the Stacked Sparse autoencoder to detect and extract the first level of proteins features, and the main approach of Deep Learning using (CNN) structure. In this approach, a combination method is proposed between the Sparse Autoencoder to extract the first level of protein features and use those features to construct accurate filters that will have used in the convolutional layer with the original protein data to learn more features than relies just on the random or initialized filters for the convolutional layers in the main Deep Learning Structure. In this case, the Stacked Sparse Autoencoder Approach that is shown in Fig.2 is applied using soft-max classifier for secondary structure protein prediction, and compare the results with our Latent Deep Learning Approach using also sot-max classifier.

In this proposal, two different learning frameworks is used. The first one is without using the fine tuning for the trained data, and the other one is the backpropagation framework which is used to pre-trained the whole network in an unsupervised fashion to fine-tuned the data in a supervised learning fashion.

IV. Stacked Sparse Autoencoder Approach

The predatory layer-wise approach for pre-training the Deep Neural Network works by training each layer in turn. In this section, how autoencoder can be "stacked" in a layer-wise fashion for pre-training which is the initializing of the weights of a Deep Neural Network (DNN) is illustrated and described. Typically, a stacked autoencoder consisting of multiple layers of sparse autoencoder which is the outputs of each layer have been connected to the inputs of the successive layer [13]. A Multi-hidden-layer sparse autoencoder is putting and crooking together many of the simple neurons. In this case, the output of neurons can be represented as input of another layer.

To train this type of network, it needs to train set of our data samples $(x(i), y(i))$ where$y(i) \in R2$. This type of network is more accurate and useful if there are multiple outputs (multi-class) that are going to classify and predict. Assume that the fixed training number set of the data sample

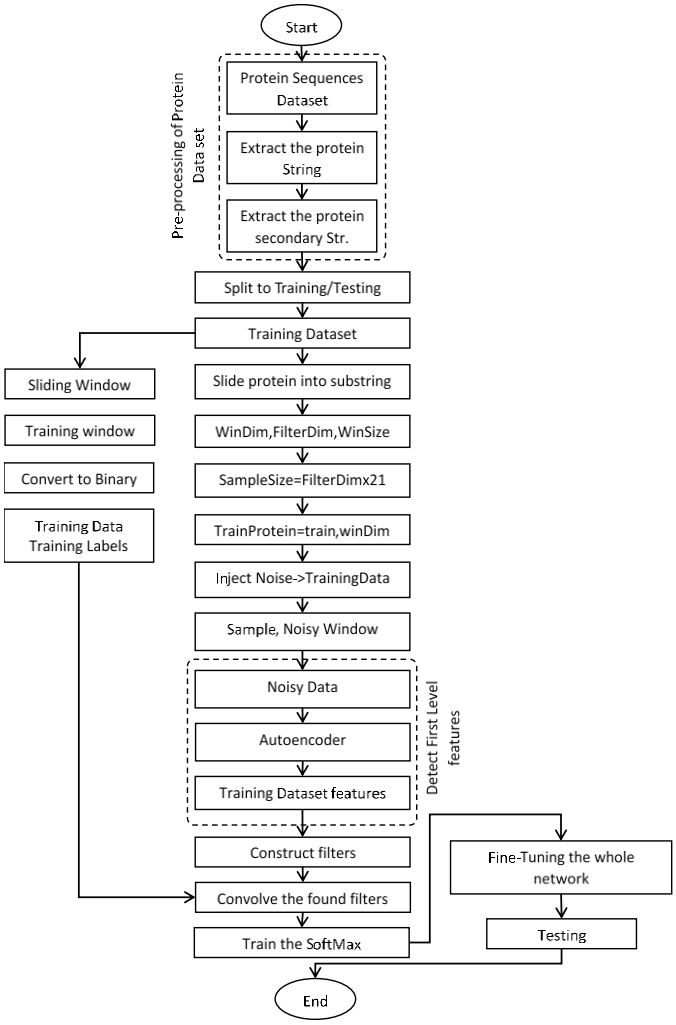

Fig. 1. The proposed Approach Latent Deep Learning Structure for secondary structure of protein prediction using a laten CNN structure

$\{(x(1), y(1)) \ldots (x(m), y(m))\}$ of m training examples, in this case the model can be trained using batch gradient descent approach. In more detail, for a single training data sample (example) (x, y), is proposed the cost function (objective) with respect to multi-hidden layer to be as given in Eqs.(1):

$$(\mathrm{w,b}) = \frac{1}{\mathrm{N}}\left(\sum_{i=1}^{N} J(W_1, b, x_i, \hat{x}_i)\right) + \frac{\lambda}{2}\left(\sum_{l=1}^{m_l-1} \|W_l\|_F^2\right) + \beta\left(\sum_{l=1}^{h_n-1} KL(p||\hat{p}_j)\right) \quad (1)$$

where $J(W_l,b;x_I,\hat{x}_i)=\frac{1}{2}\left\|x_i-\hat{x}_i\right\|_2^2$ is the squared error term and $\hat{x}_I$ is the output of the autoencoder. The second term l_2 is a regularized term that has been used for weight decay term. This term tends to reduce the magnitude of the weights and helps to jump the over-fitting situation where $\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N}\left[\phi(h_j)\right]$ is the average of the activation of hidden unit, and h_j is a sparsity term ρ , which is equal to 0.05 in the experiment. In this model, denotes the desired activation extent of each hidden neuron h_j.

$\mathrm{KL}\left(\rho\|\hat{\rho}_j\right)$ by using the Kullback-Leibler (KL) function. This function divergence between two Bernoulli random parameters with means ρ and $\hat{\rho}_j$ respectively as it given in Eqs (2):

$$\mathrm{KL}(\rho\|\hat{\rho}_j)=\rho\log\frac{\rho}{\hat{\rho}_j}+(\text{“1-}\rho\text{”})\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (2)$$

It is a measurement of how different two distributions are. If the average activation $\hat{\rho}_j$ of hidden unit h_j deviates a lot. from the desired ρ, $\mathrm{KL}\left(\rho\|\hat{\rho}_j\right)$ will add big penalty on the objective function to make $\hat{\rho}_j$ small in the next iteration. By taking partial derivative on objective function $J(W, b)$ with respect to W_l and b_l, the update rule for, the update rule for W_l and b are the following [13]. Multi-hidden-layer Sparse Autoencoder approach process steps are described in the next algorithm (1).

Algorithm (1): Stacked Sparse Autoencoder

1. **While**
2. Implemt a feedforward pass approach.
3. Compute the activations for the layers L_2, L_3 until the output layer L_{n_l}, activation function $\phi(\cdot)$ is activation function.
4. **For** the output layer, take $\delta_{n_l} = -(x - \hat{x})\phi'(net_{n_l})$, where net_{n_l} is the net activation energy of each neuron of output layer
5. **For** $l = n_l - 1, n_l - 2, \ldots, 2,$ additional with – $l = \frac{n_l+1}{2}$,
6. **Take** $\delta_l = (w_l^T \delta_{l+1}) \cdot \phi'(net_l)$,
$$\delta_{\frac{n_l+1}{2}}=\left(W^T_{\frac{n_l+1}{2}}\delta_{\frac{n_l+3}{2}}+\beta\left(-\frac{\rho}{\hat{\rho}}+\frac{1-\rho}{1-\hat{\rho}}\right)\right)\cdot\phi'\left(net_{\frac{n_l+1}{2}}\right) \quad (3)$$
where $\frac{n_l+1}{2}$ is the deepest layer whose outputs are the new features it needed.
7. **Compute** the gradients of W_l and b_l,
$$\nabla_{W_l} J(W, b; x, \hat{x}) = \delta_{l+1} \cdot net_l^T + \lambda W_l \quad (4)$$
$$\nabla_{b_l} J(W, b; x, \hat{x}) = \delta_{l+1} \quad (5)$$
8. **Update** the parameters:
$$W_l^{t+1} = W_l^t + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla W_l^i + \lambda W_l \quad (6)$$
$$b_l^{t+1} = b_l^t + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla b_l^i \quad (7)$$
9. **End For**
10. **End For**
11. **Repeat** from 1 until converge

The stacked pre-training method is to train each layer one by one as a one hidden layer autoencoder. First, it trains the first layer on the raw input data to gain parameters and output $W_1^{initial}$ and h_1 as it shown in Fig.3 then trains the second layer using the previous output h_1. Then train of the second layer using the previous output $h_1$as input and desired output of this second hidden layer h_2 to get $W_2^{initial}$ as it shown in Fig.4, which repeat for subsequent layers by using the output of each layer as input for the subsequent layer to initialize $W_l^{initial}$ [14].

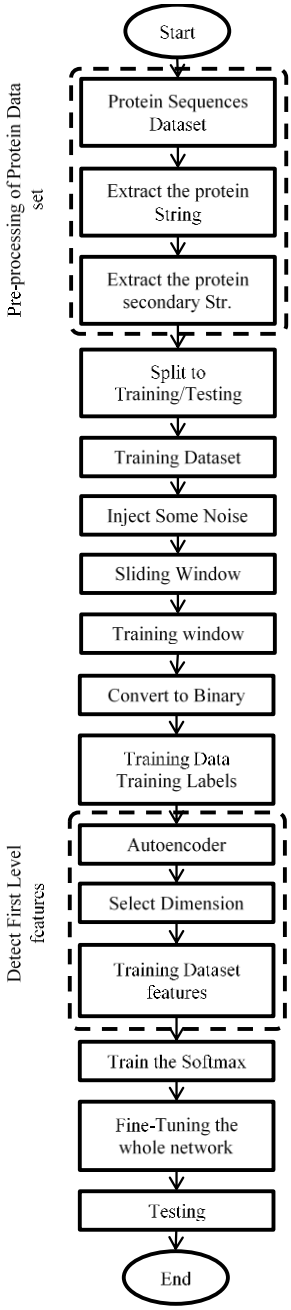

Fig. 2. The proposed Stacked Sparse Autoencoder Network Approach for detect and extract the first level of portions features, using softmax classifier to predict the secondary structure of proteins prdection

V. Deep Learning Approach using Laten CNN Structure

Deep Learning approach using Convolutional Neural Networks (CNN) is a set of biologically-inspired variants of MLPs. This approach has been proposed and produced by Hubel and Wiesel [6]. The main idea of this approach is depending on the visual context which contains significant cells. These cells represent small sub-rejoins of the original visual context. Those cells demonstrate as local filters overcome the input space to extract the strong local relation and correlation in the original space [15]. Moreover, two types

of cells have been significantly used (simple and complex cell). Simple cell is maximally respond to a specific edge-like patterns within their receptive field. In the other hand, the complex cells which have larger receptive fields and are locally invariant to exact the position of the pattern. This kind (complex cells) being the most powerful visual processing system in existence, that seems natural to emulate its behavior [16].

Practicality, the CNNs, local filter hi scanned the whole entire data and replicated across the whole entire visual field. These local filters unit share the same parameters (weight vector and bias) and form a feature map [5]. The CNN, s feature map is acquired by repeating using a function across sub-regions of the entire data. In the other words, this process is done by convolution of the input data with a specific linear filter (line detector as an example) by adding a bias term and then applying a non-linear function. If the k-th is denoated as a feature map at a specific layer as hk, whose filters are determined by the weights wk and bias bk, then the feature map hk is gained as follows as given in the following Eqs (8):

$$h_{ij}^{k} = tanh\left((W^{k} * x)_{ij} + b_{k}\right) \quad (8)$$

By recalling the following definition of convolution process for a 1D signal as given in the following Eqs (9) [16].

$$\begin{aligned} O[n] &= f[n] \times g[n] \\ &= \sum_{u=-\infty}^{\infty} f[u]g[v] = \sum_{u=-\infty}^{\infty} f[n-u]g[u] \end{aligned} \quad (9)$$

which can be extended to the 2D as given in the following Eqs (10) [16]:

$$\begin{aligned} O[m,n] &= f[m,n] \times g[m,n] \\ &= \sum_{-\infty}^{\infty} f[u,v]g[m-u,n-v] \end{aligned} \quad (10)$$

From the 2D form above, each hidden layer is composed of multiple feature maps such as, $h^{(k)}, k = 0..k$. This can be weighted as w of a hidden layer which can be represented as a 4D tensor flow. The 4D tensor consists of combination of destination elements [17]. However, in 4D tensor the feature map, source feature map, source vertical position, and source horizontal position are the common destination elements. Moreover, the biases *b* also can be represented as a vector that containing of one element for every destination feature map [15].

The Deep Learning design requires two main operations. The main one is the convolution operator which is the main workhorse for implementing a convolutional layer in the CNN structure [15]. According to the mini-batches of (training sample) of input data, the shape of the tensor is constructed. In other words, mini-batch size, several input feature maps, image height, and data width are main category of the tensor design in the CNN structure. A 4D tensor is corresponding to the weight matrix W which is a significant technique of the tensor to determine the number of feature maps at layer *m*. and the number of the feature maps at layer m-1, filter height, filter width [17].

The second operation of the CNN structure is the Max-pooling. This operation takes the input data (sub-region) into a set of non-overlapping regions. For each sub-region, the outputs are the maximum value. The main reason of using the

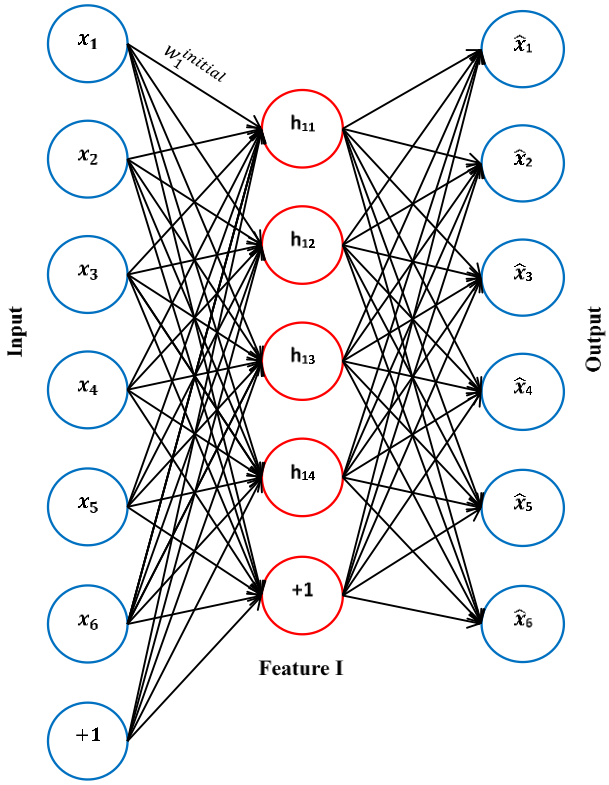

Fig. 3. Stacked sparse autoencoder network weights initialization example of the first level weights w^1

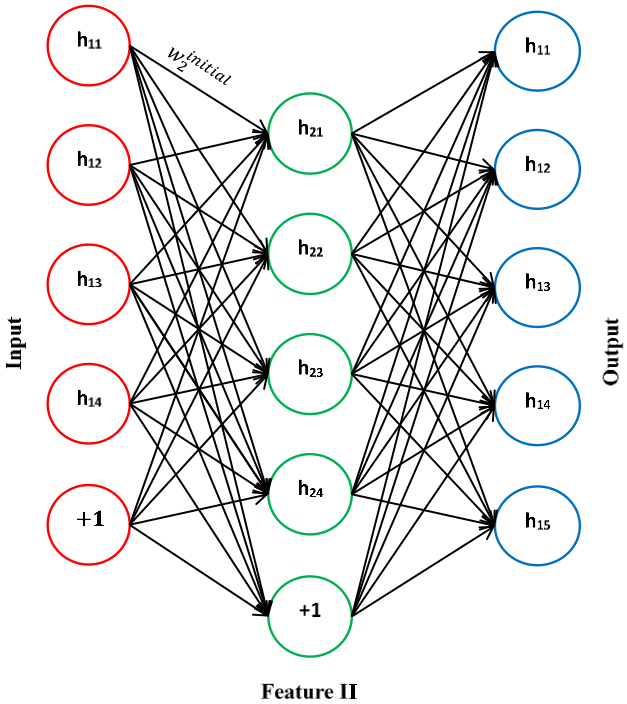

Fig. 4. An example of the first level of the stacked sparse autoencoder structure

max-pooling is to it reduces computation for upper layers on the architecture. Also, it provides a form of translation invariance. In the max-pooling layer, there are 8 directions in

which one can translate the input image by a single pixel [16]. For example, if the max-pooling i s done over by a 2x2 region, 3 out of these 8 possible configurations will produce the same output at the convolutional layer [18]. In our design, the Full model of the Deep Learning consists of two convolution layer and two max-pooling layers with one fully connected layer. The lower layers are collected by alternating convolution and max-pooling layers and the upper layers are fully-connected which corresponds to a traditional MLP (hidden layer with logistic regression). The fully connected layer operates on 4D tensors technique which can be flattened to a 2D matrix of the feature maps, to be convenient with the MLP main implementation.

VI. Experimental Results

In terms of measuring the performance of the prediction model, the statistical method of k-fold cross validation is used in this approach. In supervised learning, a certain amount of labeled data is available for training the prediction model. The performance of a prediction model depends on its efficiency on detecting the labels of unlabeled data. To estimate performance, one can set aside some of the labeled data for testing, making sure that the test data is not also used for training. Where the available data is limited, then the process of training on part of the labeled data and testing on the remaining part can be repeated to improve the estimate of accuracy.

A. Evaluation Criteria

The evaluating performance of Protein secondary structure prediction system is calculated by using Q3 measurement, which is defined as the ratio between the numbers of correct recognition decision to the total number of attempts.as it is given in equation (11).

$$Accuracy = \frac{Number\ of\ correct\ attempts}{Total\ number\ of\ attempts} \qquad (11)$$

B. Stacked Sparse Autoencoder prediction Results

Fig.5 shows the sparse autoencoder prediction on the testing dataset. It's clear to see that the sparse autoencoder has predicted about (67%) 'C' as a true positive (TP) which is correctly predicted, about (43 %) for correct prediction for class 'E', and it has been satisfied about (65 %) on class 'H'.

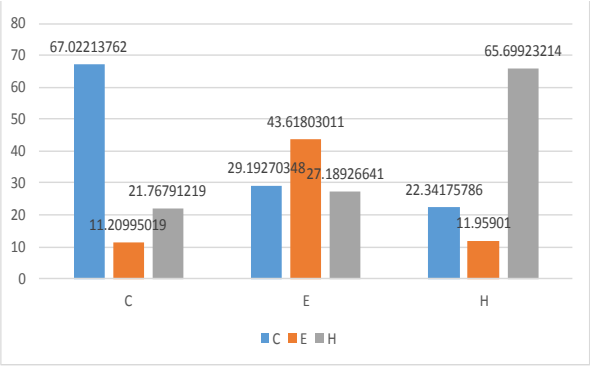

Fig. 5. Stacked Sparse Autoencoder approach performance results

Although, Fig.6 shows the different accuracy result when the fine-tuning approach has been applied in this approach that has been shown in Table.2. it's clear enough to notice that this approach has been increased about (7.047%) on training set, and (2.283%) on testing set.

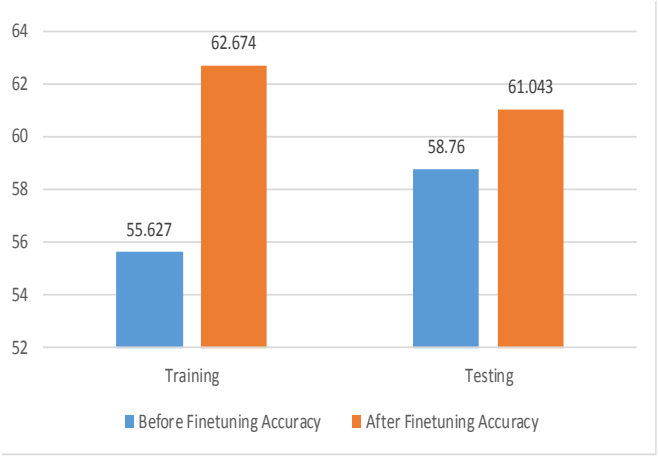

Fig. 6. Sparse Autoencoder performance results with/without fine-tuning

TABLE. II. Stacked Sparse Autoencoder Approach Performance Results

Approach	Training	Testing
Before Finetuning Accuracy	55.627	58.76
After Finetuning Accuracy	62.674	61.043

The overall performance results of Stacked Sparse Autoencoder for secondary structure of protein prediction is illustrated in Fig.7 which illustrates the difference in the prediction accuracy result using fine tuning is better that using the same approach without tuning the trained data.

C. Latent Deep Learning using Prediction Results

In this section, it explores the performance of our Latent Deep Learning approach depends on constructed filter from the previous approach (Stacked Sparse Autoencoder for first level features detection and extraction) to convolve the new features with the original proteins data using Convolutional Neural Networks (CNN) structure. In this approach, two training frameworks are used. The first one is Deep Learning approach without a fine tuning, and the second one is with fine tuning. Matlab program language is used to design and implementation of those two structures.

1) Our Deep Learning Approach-without Fine Tuning

Fig.7 shows the Latent Deep Learning approach prediction result using forward pass approach (without fine-tuning) on the Training dataset. It's clear to notice that the Deep Learning has predicted about (66.073%) 'C' as a true positive (TP) which is correctly predicted, about (26.972 %) for uncorrected prediction for class 'E', and (20.104 %) uncorrected prediction for class 'H', so it has been correctly predicted about (41.981 %) for class 'E', and (70.671 %) for class 'H".

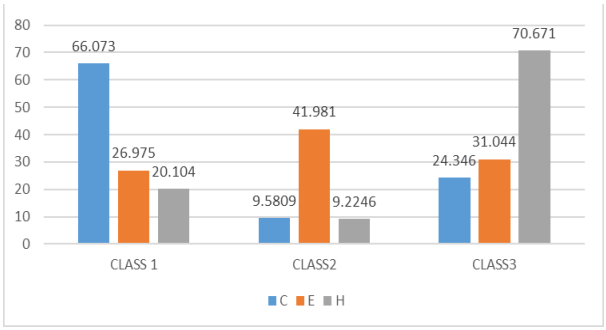

Fig. 7. Our Deep Learning Approach without fine-tuning performance results for the testing dataset

Fig.8 shows the Latent Deep Learning prediction result on the testing dataset. It's clear to notice that the Deep Learning approach has predicted about (67.37 %) 'C' as a true positive (TP) which is correctly predicted, about (28.638%) for uncorrected prediction for class 'E', and (21.898%) uncorrected prediction for class 'H'. Although, it has been correctly predicted about (43.496 %) for class 'E', and (67.14 %) for class 'H".

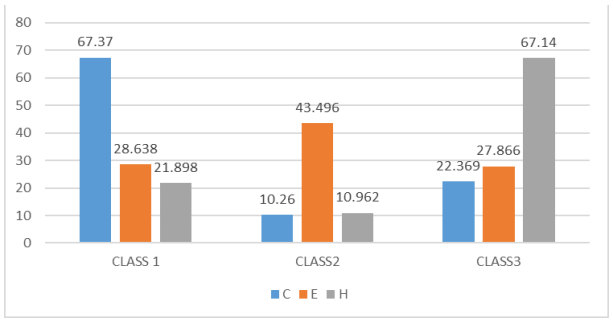

Fig. 8. Our Latent Deep Learning Approach without Fine-tuning performance results for the testing dataset

2) Our Deep Learning Approach-with Fine Tuning

Fig. 9 shows the Deep Learning with fine tuning results (backpropagation approach) on the training dataset.

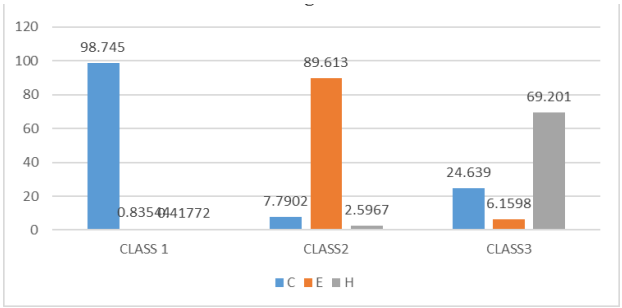

Fig. 9. Our Deep Learning with Fine-tuning performance results for the training dataset

It's clear to see that this approach has predicted about (98.745 %) 'C' as a true positive (TP) which is correctly predicted, about (0.835%) for uncorrected prediction for class 'E', and (0.41772%) uncorrected prediction for class 'H'. Although, it has been correctly predicted about (89.613 %) for class 'E', and (69.201%) for class 'H".

Finally, Fig.10 shows the Deep Learning approach with fine tuning prediction result on the testing dataset. This approach has predicted about (99.923 %) 'C' as a true positive (TP) which is correctly predicted, about (0.05159%) for uncorrected prediction for class 'E', and (0.27%) uncorrected prediction for class 'H'. Although, it has been correctly predicted about (99.876 %) for class 'E', and (71.139%) for class 'H".

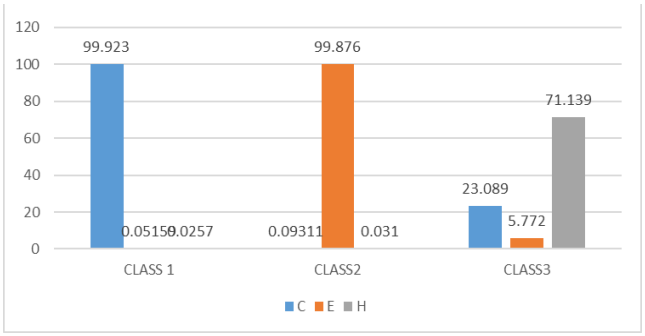

Fig. 10. Our Deep Learning (CNN) with Fine-tuning performance results for the testing dataset

A. Latent Deep Learning Model Comparison with Other Approaches

In term of evaluate our Latent Deep Learning approach for secondary structure of protein prediction against the "state-of-the-art" and other approaches that have been discussed in (Section. II), Table 3 shows that we achieved 90.3126% Q8 accuracy on the test set sequences using SCRATCH protein predictor dataset which consists of primary and secondary structure of protein data (SSpro) with 3 classes. As it shown in Table 3, The machine learning and structural similarity methodology that has been proposed by Christophe [11] achieved (84.51%) Q8 accuracy. This approach provides a sequence-based structural similarity methods which systematically combining the protein profile in such grows dataset using machine learning methods and sequence-based. in this case, the structural similarity seems to be the best strategy, and this is one of the reasons why it has been chosen because this approach provides separate modules for each one of these three tasks [11]. In contrast, since protein structures are more conserved than protein sequences this model has small improvements since this approach capable of detecting remote structural similarity, not readily visible in the sequences alone. However, Table 3 shows that the second approach that use Deep Supervised and Convolutional Generative Stochastic Network for protein secondary structure prediction, which is proposed by Jian Zhou et al [12] has achieved 72.1 ± 0.6% Q8 accuracy on the same dataset. This approach proposed a Deep features extractor model by using a 3-layer convolutional structure. Starting with {80 × conv5} − {pool5 − 80 × conv5} − {pool5−80×conv4}. This model suggests the combination of convolutional and supervised generative stochastic network which is applied well suited for low-level structured prediction that is sensitive to local information, while being informed of high-level and distant features. In contrast, one limitation of this approach is that the current architecture may not be optimal to capture the spatial organization of protein sequence

in some cases, especially for structures that formed by long-range interactions.

TABLE. III. STACKED SPARSE AUTOENCODER APPROACH PERFORMANCE RESULTS

Approach	Q8 Accuracy
Christophe [11]	84.51%
Jian Zhou [12]	72.1%
State-of-the-art [19]	0.649%
Our Approach	**90.3126%**

Latent Deep Learning approach proposes a combination between the Deep Learning and Stacked Sparse Autoencoder. This combination produces a new data dimension which has been extracted from the first level of the sparse autoencoder network. Those features are used to build convolutions filters that use later in the convolutional layer in the Deep Learning structure which is a powerful complement to classical machine learning tools and other analysis strategies. In this approach, a new technique of learning the low-level features is produce to build the convolutional kernel that are used later in the convolutional layer inside the Latent Deep Learning structure. This method jump out the other limitation on the previous approaches by using a powerful structure that capable of detecting the remote structural similarity of the protein sequence depending on the structure feature extraction where the readily protein features are visible in the sequences alone. Although, this approach has more capability to capture the spatial organization of protein sequence since it uses the original low-level features itself (constructed kernels) to extract the portion sequence features in the latent Deep Learning Approach.

VII. CONCLUSION

First level features detection is the main contribution and a new approach that has been used to predict the protein secondary structure. In this approach, two machine learning approaches and a combination between them have been proposed and used in this paper. The first one is the unsupervised learning approach based on using Sparse Auto-encoder network structure, and the semi-supervised learning approach based on using Deep Learning neural network structure. The first approach using sparse autoencoder has been achieved about (65.627%) in training set, and about (72.674%) in the testing set without using the fine-tuning approach. The highest accuracy of the same approach is used in the fine-tuning approach which is 86.760% in training set and 71.043% in the testing set. The highest accuracy is (86.719%) in the testing set of the Deep Learning approach, and (85.853%) on the training set when the fine-tuning approach uses, but without that the Deep Learning approach has been satisfied (70.575%) in training set and (79.33%) in the testing set. In the conclusion, the Deep learning methods are a powerful complement to classical machine learning tools and other analysis strategies. However, this paper presents a proposed system that implements a combined structure between semi-supervised and convolutional architecture to learn hierarchical representation on full-sized data. Finally, the fine-tuning approach of the whole network gives a better result, which brings the network's hidden weights W and biases b to a descent area of the parameter space to comprise a better startup point of the weight than random initialization.

For further works and development of the protein sequence and structure relies on the Latent Deep Learning structure. An adaptive and dynamic architecture will be proposed to better model the long-range interactions in a protein. This adaptive will changes the connectivity adaptively based on input of the low-level of the portion features which may further improve the quality of representation in of the Q8 accuracy in the future.

REFERENCES

[1] Xiong, J., "Essential bioinformatics, Cambridge University Press",2006.

[2] Safaai Bin Defis, "Protein Secondary Structure Prediction using Artificail Inteligence Techniqye", Technology University, Malaysia 2007.

[3] Lipontseng Cecilia Tsilo ,"Protein Secondary Structure Prediction using Neural Network and Support Vector Machine", 2008.

[4] Gianluca Pollastri, "Accurate prediction of protein secondary structure and solvent Accessibility by consensus combiners of sequence and structure information", 2007.

[5] Rost, B. and Sander, C.,"Prediction Of Protein Secondary Structure", Journal of Molecular Biology. 232. 584-599, 1993.

[6] I. Joliffe, "Principal Component Analysis", New York: Springer-Verlag, 1986.

[7] Peng, J., Bo, L. & Xu, J.,"Conditional neural fields", Annual Conference on Neural Information Processing Systems,Vancouver, British Columbia, Canada. 7-10 December, 1419-1427.

[8] Lee, H., Grosse, R., Ranganath, R. & Ng, A. Y,"Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations, 26th Annual International Conference on Machine Learning, ICML, Montreal, Quebec, Canada, June 14-18, 2009.

[9] Spencer, M., Eickholt, J. & Cheng, J. A.,"Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction", IEEE/ACM Trans. Comput. Biol. Bioinform. 12, 103–112 (2015).

[10] Cheng J1, Randall AZ, Sweredoski MJ, Baldi P," SCRATCH: a protein structure and structural feature prediction server", Nucleic Acids Res. 2005 Jul 1;33(Web Server issue):W72-6.

[11] Christophe N. Magnan and Pierre Baldi,"Perfect Prediction of Protein Secondary Structure and Relative Solvent Accessibility using Profiles", machine learning and structural similarity", Vol. 30 no. 18 2014.

[12] Jian Zhou ,Olga G. Troyanskaya,"Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction", Princeton University Princeton, NJ 08540 USA.

[13] Ng A.," Sparse autoencoder", CS294A Lecture notes, 2011: 72.

[14] Yoshua Bengio,"Learning Deep Architectures for AI. Foundations and Trends R in Machine Learning",1-127, 2009.

[15] Hubel, D. and Wiesel, T..,"Receptive Fields and Functional Architecture of Monkey Striate Cortex",Journal of Physiology (London), 195, 215–243.

[16] Ukushima, K.,"Recognition: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", Biological Cybernetics, 36, 193–202.

[17] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.,"Gradient-based Learning Applied to Document Recognition", Proceedings of the IEEE, 86(11), 2278–2324.

[18] Lee, C. Ekanadham, and A.Y. Ng., "Sparse Deep Beliefnet Model for Visual Area V2",Advances in Neural Information Processing Systems (NIPS) 20, 2008.

[19] Wang, Zhiyong, Zhao, Feng, Peng, Jian, and Xu, Jinbo. Protein 8class secondary structure prediction using conditional neural fields. Proteomics, 11(19):3786–3792, 2011. ISSN 1615-98