Educational Administration: Theory and Practice
2024, 30(1), 1635-1644
ISSN: 2148-2403
https://kuey.net/

Research Article

NLP Based Protein Sequence Classification Through Convolutional Neural Network

Pooja Sharma[1*], Manish Maheshwari[2]

[1*]PhD Research Scholar, MCNUJC, Bhopal
[2]Professor, MCNUJC, Bhopal

Citation: Pooja Sharma (2024) NLP Based Protein Sequence Classification Through Convolutional Neural Network, *Educational Administration: Theory and Practice*, *30*(1), 1635-1644
Doi: 10.53555/kuey.v30i1.6488

ABSTRACT

Redesigning and modifying proteins is a leading objective in the pharmaceutical industry today. Modern technology has made it possible to efficiently redesign proteins by simulating mutation, natural selection, and amplification in the lab. There are an infinite number of possible mutations for each protein. It would be impossible to synthesise every sequence or even examine every version that could be beneficial. Recently, there has been an increase in the use of machine learning to aid in protein redesign, as prediction models can be used to virtually evaluate a large number of different sequences. Modern machine learning models, notably deep learning models, are poorly understood. In addition, few descriptors of protein sequences have been considered. This paper presents a novel classification method for protein sequences that is propelled by artificial intelligence. Two distinct single-amino-acid descriptors and one structure-based, three-dimensional descriptor are used to create prediction models, and their effectiveness is compared. Several various evaluation metrics were applied to a variety of public and private data sets to determine the accuracy of the predictions. The study's findings indicate that the convolution neural network models constructed using amino acid property descriptors are the most pertinent to protein redesign problems encountered in the pharmaceutical industry.

Keywords- Natural language processing; Deep learning; Protein sequence.

INTRODUCTION

Bioinformatics is a really multidisciplinary field because it draws on concepts from mathematics, computer science, genetics, and molecular biology. Numerous significant and abundant biological subjects are examined via the lens of computational science. The most frequent issues are those pertaining to drawing inferences and identifying patterns from collected data, which are essential for comprehending molecular biological processes. Genetics has advanced significantly in the last many years. This leads to the generation of massive amounts of biological data. When drawing conclusions from this type of data, state-of-the-art computer techniques must be used. Furthermore, effective techniques for streamlining the examination of consecutive data must be created. These techniques can be used to forecast and categorise sequence data. This enables us to summarise the results of several studies pertaining to the life sciences. As the amount and speed of data produced rise, data mining and machine learning techniques are becoming more and more crucial for these applications [1].
With the increasing availability of biological data, bioinformatics has advanced significantly [2-4]. Researchers can identify significant patterns and correlations by sorting through mounds of biological data with the use of data mining. Sequential pattern mining is a branch of data mining where patterns typically form over a few thousand bytes (20 for protein sequences and 4 for DNA sequences) or less. Sequences are just ordered lists that are used in a variety of sectors, such as commerce, science, security, and medicine. On the other hand, sequence data mining offers techniques for locating undiscovered insights inside this data mountain [5]. Protein sequence classification is the process of labelling proteins according to their sequences. The arrangement of a protein's constituent pieces is indicated by its amino acid sequence. Databases maintain a record of every known protein sequence. Proteins are macromolecules composed of lengthy chains of particular amino acid sequences. There are protein molecules that have thousands of amino acids, whereas others might just have a hundred.

Directed evolution techniques are frequently used in the pharmaceutical industry to produce proteins with enhanced properties. Examples include the industrial production of pharmaceuticals and drug precursors, when the natural enzymes may not be able to function optimally. In these cases, enzymes are used as

catalysts. Proteins must undergo repeated experiments that simulate natural selection in order to evolve under control. The best parent sequence that is currently available is used at each junction to create a panel of changed sequences, which are then examined in vitro for desired characteristics such substrate conversion percentage. For example, variants that improve the synthesis of the product under particular circumstances are retained for future development. Directed evolution, to put it simply, is an optimisation issue within the enormous space of potential protein sequences. Finding beneficial mutations with a limited number of tests is still difficult [6]. Finding advantageous mutations could be sped considerably with the use of predictive models. By evaluating sequence data from real-world scenarios, machine learning algorithms can discover relationships between sequences and attributes. This allows algorithms to predict the characteristics of fictitious sequences. Computational methods can guide subsequent experimental iterations to synthesise only the most promising sequences. The interest in using machine learning models to solve this issue has recently increased. (7)

Protein sequence optimisation can benefit from machine learning, but only if the highly informative sequences are given the proper descriptors. The application of protein sequence features such as amino acid content (dipeptide and tripeptide composition), projected secondary structure, and predicted solvent accessibility has proven beneficial for sequence-based protein classification and ligand docking problems. Various approaches have been proposed to accomplish this, including the use of k-Spaced Amino Acid Pairs and Conjoint Triads [8], torsion angle density and amino acid distance density histograms [9], and 3D grid protein-ligand architectures [10].

While this accomplishment has been applied to the classification and binding of proteins with ligands, the prediction of protein function—which is frequently assessed on an ongoing basis—remains a major difficulty. Although machine learning-guided protein engineering is a relatively new field of study, only few studies have included a large number of proteins and models. Kimothi et al. [12] and Yang et al. [13] have applied the doc2vec [14] word embedding model to huge protein sequence data sets. These methods use NLP cues to compare sequence fragments to words and protein sequences to documents. Using the embedding, one-hot encoding, mismatch kernel, ProFET, and AAIndex features, Yang et al. [15] tested their prediction models on four public data sets. By utilising a high throughput in silico model, Wu et al. [16] demonstrated how guided directed evolution could assist in identifying better mutants with less labour in the laboratory. Although several supervised learning techniques were applied, input descriptors for protein sequences were not specified.

An introduction of the fundamentals of using machine learning in protein engineering is given by Yang et al. [17]. The authors use two case studies to demonstrate these theories. The fact that a machine-learning sequence-function model for proteins is addressed and supported based on a literature review complicates quantitative data analysis. Protein sequences provide a wealth of descriptors, but predicting biological features from them is difficult due to low signal-to-noise ratio experimental data. The process of choosing or amplifying samples for in vitro protein analysis frequently involves multiple phases. These low-volume, high-throughput studies could yield wildly surprising results. Oftentimes, promising variants receive more research or confirmation. Determining how to extend forecasts to incorporate mutations not seen in the current data set presents another challenge. Building trustworthy prediction models that work well with the actual condition of the data is therefore essential.

This study offers a methodology for using natural language processing (NLP) to the available dataset in order to extract the contextual elements required in order to build a framework for protein sequence classification. This paper describes the steps involved in effectively classifying proteins into their several categories, including data pre-processing, visualisation, feature engineering, modelling, training, and evaluation. CNN models recover high-level properties based on the ordering information of the full protein sequence, as opposed to other approaches that just account for altered sites. A survey of recent presentations by a range of scholars whose work is relevant to the subject at hand is included in the second section of the paper. You will study about CNN's organisational structure in Section III. For the given dataset, Section IV presents a Convolutional Neural Network (CNN) based Protein sequence categorization system. The work is concluded in Part VI. Part V reports the findings of an experimental research that examines and discusses the efficacy of the suggested approach.

RELATED WORK

Drug development is a critical step in the pharmaceutical industry. Computational approaches have dramatically reduced the time and expense of generating new medicines. To deal with challenges of all shapes and sizes, we'll have to use a variety of drug screening and design methodologies. Machine learning and deep learning approaches, which go beyond the constraints of prior studies, are the primary emphasis of this section. Multi-objective evolutionary techniques were created by Wei-Li et al. [18] by combining Rama torsion angle sampling with loop-based resampling, stochastic rank-based selection, loop-based crossover, and near-native sampling. The secondary structural similarity criterion has the potential to address the energy function's inaccuracy. Protein secondary structure prediction was made possible by Zhou et al.'s [19] use of convolutional deep neural networks (CDNN) trained with reinforcement learning. CDNN possesses a robust classification capability on top of the abstraction powers of CNN and the sequence data analysis skills of LSTM. The cross-entropy error between labels for protein secondary structures and dense layer outputs is

used to train the CDNN architecture. Empirical validation on two independent datasets demonstrates the efficacy of the CDNN method. But the projection is still plausible despite the imbalances. This reduces the reliability of future projections.

You et al. [20] created Deep ResNet to predict protein contact/distance and template-free protein folding. Protein-protein interaction and tertiary structure prediction are two areas where deep ResNet has made great strides in recent years. When it comes to making use of inter-residue orientation information, the proposed 3D modelling approach is still less advanced and more fundamental. Since the proposed deep ResNet does not rely on evolutionary information to generate predictions about natural protein folds, it is capable of correctly folding the vast majority of human-created proteins.

Xu et.al. [21] proposed that a computer technique dubbed "deep structural inference" may be used to predict protein residue/residue interactions using a deep-learning algorithm and template-based structural modelling. More than 1,200 single-domain proteins were used for the first time to make a widespread tertiary structure prediction. It appears that the coupling scores derived by CCMPred, which relied on the raw frequency distributions from multiple sequence alignments, cannot fully replace the information gained by statistical co-evolutionary analyses. Du et al. [22] designed a novel recurrent geometric network (RGN) that can predict protein structure from sequences without using any prior knowledge. When orphan and designer proteins don't have enough sequence similarity for multiple sequence alignment to work, this computationally efficient alternative has many advantages. RGN2 does this by employing a simple strategy to describe the geometry of the C backbone. In order to successively recreate the backbone's structure, this method is constrained to considering only local interactions between C atoms (curvature and torsion angles).

By developing a multi-advanced deep belief network-based method, Guo et al. [23] enhanced protein secondary structure prediction. They worked together to improve forecast accuracy by over 80%. Further, the results demonstrated the predictive power of hidden Markov model profiles derived from emission/transition probabilities in identifying secondary structure. However, the network's features will be uneven. By feeding a protein feature vector into a DNN, including the suggested MOS descriptor with AA classification, Wang et al. [24] were able to accurately predict PPIs. The suggested MOS descriptor is able to account for the order connection of the entire AA sequence, unlike earlier protein representations like AC, CT, and LD. After careful deliberation, the network parameters cross entropy cost function, ADAM optimizer, and ReLU AF were chosen for the task. The ideal values for the other parameters, like network depth, network width, and the LR, were determined by computing them for the specific method. The author independently trained the DNN model with AC, CT, and LD to facilitate a comparison with the suggested Work.

Another fascinating and original piece of work was conducted by Jha and Saha [25], who used an LSTM-based classifier that included properties supplied by two separate protein modalities, namely sequence-based and structure-based information. Using the structural representation of the proteins, we first generated three distinct protein representations based on three different characteristics, and then we got corresponding feature sets using a ResNet50 model. Li et al. [26] released the first work on sequence-based PPI prediction using DNs that relied solely on auto-feature engineering, i.e., without the use of manually derived features. The NN architecture can only acquire knowledge from numerical input. The author modified the protein sequence by randomly assigning natural numbers to each amino acid.

PPIs prediction with RNNs and embedding systems was also done by Gonzalez-Lopez et al. [27] without the need for feature engineering. Each sequence triplet was assigned a token (an integer) as part of the tokenization procedure so that the sequence could be represented numerically. Each protein's pair representation in the NN was fed to and analysed by two similar-looking branches. The design's FC layer served a unique purpose, as did the embedding and recurrent layers. To avoid over-fitting and ensure consistent input, we also made use of Dropout and Branch normalization.

ARCHITECTURAL FRAMEWORK OF CNN

CNNs are among the greatest learning algorithms for comprehending visual content, and they show extraordinary performance in tasks relating to picture segmentation, classification, detection, and retrieval. The academic community isn't the only one interested in CNNs anymore. Google, Microsoft, AT&T, NEC, and Facebook are just a few of the major tech companies with active research groups investigating novel CNN designs for commercial usage. At present, deep convolutional neural network (CNN)-based models perform the best in image processing and computer vision (CV) contests.

CNN's appeal comes from its capacity to make use of temporal or geographical correlation in data. There are several stages of learning embodied in a CNN's structure, including convolutional layers, non-linear processing units, and subsampling layers. A convolutional neural network (CNN) is defined by LeCun et al. [28] as a feedforward multilayered hierarchical network in which each layer uses a pool of convolutional kernels to perform different transformations. Significant characteristics can be extracted from locally connected data points with the help of the convolution method. The activation function takes the output of the convolutional kernels and incorporates non-linearity into the feature space; this has dual benefits for learning abstractions. This non-linearity generates unique activation patterns for unique responses, which aids in learning the semantic differences between images. Subsampling is typically applied to the output of a non-linear activation function in order to generalise the results and prevent geometric distortions in the input. Because CNN can extract features automatically, a specialised feature extractor is unnecessary. Thus,

CNN can obtain a suitable internal representation from raw pixels with no further processing required. Hierarchical learning, automatic feature extraction, parallel processing, and weight sharing are just a few of CNN's other distinguishing features.

During training, CNN uses the backpropagation technique to learn by controlling the weight rebalancing to get the desired outcome. Similar to how the brain acquires knowledge through experience, a backpropagation algorithm optimises an objective function. Because of its multi-level, hierarchical design, Deep CNN can efficiently process data at varying levels of complexity. The combination of lower and middle-level features yields the more abstract high-level features. Like the human brain's Neocortex, CNN can dynamically learn properties from the raw data by extracting them in a hierarchical fashion. CNN's success is largely attributable to the method's ability to extract features in a hierarchical structure. [29]

Deep designs often outperform shallow structures when dealing with severe learning difficulties. By stacking several linear and non-linear processing units, it becomes able to learn sophisticated representations at various levels of abstraction. In recognition tests including hundreds of image categories, deep CNNs significantly outperformed traditional vision-based models. CNNs have been increasingly popular for use in image classification and segmentation applications once it was discovered that deep architectures can increase a CNN's representational capacity. To create deep CNNs, significant technological progress and vast volumes of data had to be made available. [30]

Today, CNN is among the most widely used machine learning techniques, especially in visual contexts. Modern ML applications benefit greatly from CNN's capacity to learn representations from grid-like input. Due to its superior feature creation and discriminating capabilities, CNN is frequently used in ML systems for both feature production and classification.

PROPOSED DEEP CNN BASED PROTEIN SEQUENCE CLASSIFICATION

This paper describes approaches such as data pre-processing, visualisation, feature engineering, modelling, training, and evaluation to efficiently categorise various proteins into different categories. Figure 1 illustrates how the entire project is primarily divided into four parts, which are outlined as follows:

a) Dataset definition

The dataset from kaggle.com known as the Structural Protein Sequences is used in this specific experiment. Protein meta data, covering subjects like protein classification and extraction techniques, makes up the first section of the collection. Protein structural sequences make up the collection's second section. The "structureID" feature of the proteins that serve as the foundation for both databases' organisation. The first data set has 1,41,000 rows and 14 columns, while the second data set has 4,67,000 rows but only five columns. The proteins utilised in this work were sourced from the Protein Data Bank (PDB) at the RCSB's Research Collaboratory for Structural Bioinformatics.

b) Data Preprocessing

Using the "structureID" property, we combine the two data sets into a single one in the first phase. Rows without appropriate names or sequences are eliminated after merging. Next, we use the "macromoleculeType_x" feature to filter out all macromolecule kinds other than proteins because the dataset contains many different types of macromolecules. The data set includes representations of many different kinds of macromolecules that are relevant to biology. The majority of the files contain data regarding proteins. Since RNA is converted into proteins by DNA and because DNA is the building block of RNA, proteins are the biomolecules that directly interact in biological pathways and cycles. A protein may be able to carry out one or two particular functions, depending on their family. For example, a protein belonging to the Hydrolase group works to catalyse hydrolysis, which is the process of dissolving bonds by adding water, in order to help break down protein chains and other compounds. A transporter protein, which facilitates the passage of other molecules into and out of cells, including water and sugars like fructose and sucrose, is another example. Furthermore, based on the total number of rows, only the top 10 protein classes are used.

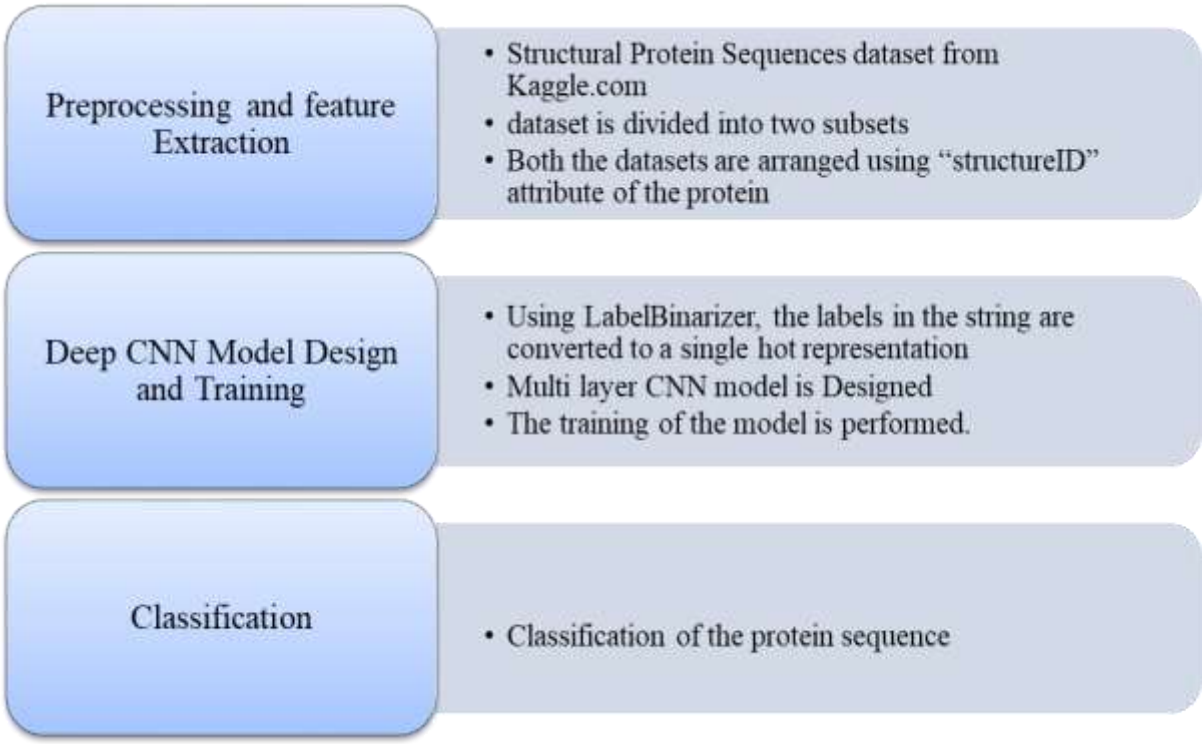


Fig.1 Proposed Framework

c) Feature Extraction

For All ten labels are found to be categorical values. These categorical data must be translated into binary or numeric form because machine learning algorithms can only read numeric values. Using LabelBinarizer, the labels in the string are converted to a single hot representation for this. In a single hot representation, values are given a 1 if they are present, else a 0. The Tokenizer method from the Keras library is used to further pre-process sequences by turning each character in the sequence into a number. Additionally, each sequence's length is uniformized for exact processing. Here, a character limit of 256 is applied.

Term Frequency - Inverse Document Frequency (*tfidf*) [31], which is widely used in the field of NLP, is one of the main features that are now derived from the segmented data in this stage of the work. The *word2vec* approach is then used to extract the more detailed attributes, using the *tfidf* as a point of reference. The two statistics that are utilized in the *tfidf* to identify the significance of a word in a document are the frequency of a word's occurrence in a document (referred to as *tf*) and the rare or frequent appearance of the word in a document (referred to as *idf*). The parameter *idf* is defined as follows: where *nd* represents the total number of documents, and *df(d,w)* represents the number of documents that include the word *w*.: [32]

$$idf(w) = log\frac{nd}{df(d,w)} + 1$$

Tokenizing the training text allows for the use of the statistical distribution of tf across the dataset in order to compute the *tfidf*. The *tfidf* can then be determined. After that, the appropriate *idf* was applied to each word on the list. Word vectors can be generated with the help of the *word2vec* model by feeding it a tokenized corpus. To locate the words that are close to the supplied one and extract the context, we employed an architecture called Continuous Bag of Words (CBOW). Its structure is very similar to that of a neural network, and its inputs are projections. In an effort to streamline the time series, this method omits the standard non-linear hidden layer often displayed in the output. Furthermore, the projection layer information is consistent across all words, and the context of the word is used as an input.

d) CNN Model Training and Testing

Several iterative rounds of testing are commonly used in protein engineering to expand the sequence space. New mutations can be introduced into previously unidentified and known genetic areas. In order to replicate the conditions of an actual application, the data is split into training and testing sets according to when they were collected. The convolutional neural network (CNN) family of neural networks is widely used for image-based applications. Each hidden node in a convolutional neural network (CNN) model receives its input from a condensed region of the layer above it. Convolutional layers are better than fully-linked ones because they can take use of these local connections to extract significant high-level properties and manage spatial dependency in images. Some data sources may benefit more from a 1D CNN, but most can benefit from a 2D CNN model's ability to take in numerous channels (such as the RGB colour channels).

For this investigation, a one-dimensional convolutional neural network (CNN) model on protein sequence data is suggested. High-level information can be extracted from surrounding sites using a 1D convolution filter by operating along the amino acid sequence dimension (columns) in Figure 1 (top). Either a big number of unique amino acid properties or a high number of input channels may be indicated by the columns. In Figure 2, a typical CNN architecture is displayed.

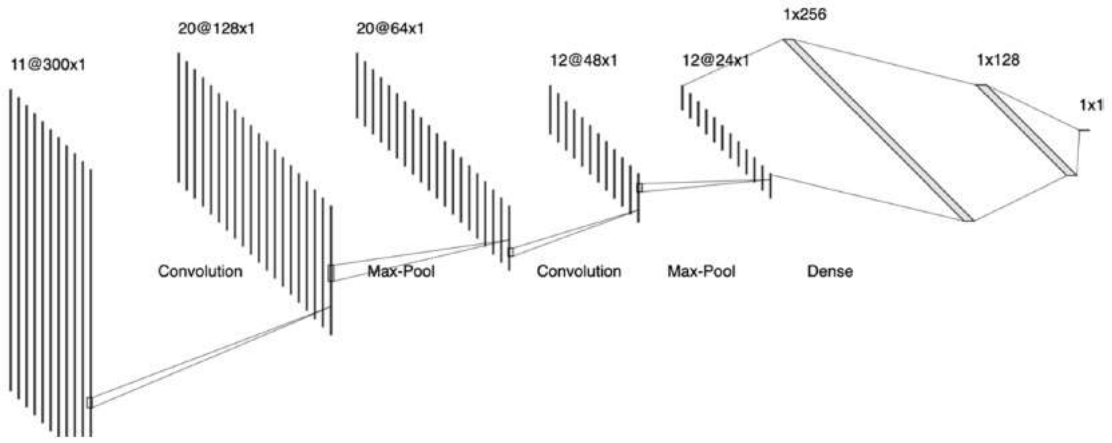


Figure 2 Common layout for a single-dimensional CNN [33]

In this case, when the input protein sequence is 300 amino acids long, we use the 11-dimensional PCscores descriptor as a single amino acid feature. The first block of Figure 2 depicts the 11 vertical lines of an 11x300 matrix, which is one input characteristic for one protein. To reduce the number of features recovered, the following max-pooling layer downsamples along the sequence dimension after each convolutional layer receives input from positions in its immediate neighbourhood. Each convolutional layer contains a collection of filters that, like a sliding window, extract a different set of features from the underlying layer's data channels. Figure 2 shows an example of a convolution layer that uses 20 filters to generate a 20-channel output feature (the 20 vertical lines) for the layer that follows it. The first max-pooling layer then takes the maximum value for each of the two elements in the feature sequence to shorten the sequence by 2. Multiple convolution and maxpooling layers can be applied to the input feature matrix to extract features. The final step in processing the high-level features and performing the regression task is to use a "fingerprint vector" derived from the output of the convolutional layers. After that, a number of layers, each of which is fully connected, will be used.
The design of the CNN architecture must also be predetermined, much like MLP. The model's architecture can be tailored in a variety of ways by adjusting parameters like the number of convolution and max-pooling layers, the number of filters in each convolution layer, the filter size and stride, and the number of fully connected layers that come after the flattening of the CNN's output features. The hyperparameter settings in 2 for things like minibatch size and learning rate were used in a gridsearch.

e) Classification

Most proprietary protein engineering efforts are focused on identifying the protein sequences that allow for the highest possible substrate conversion. Thus, machine learning models are required to differentiate between protein sequences that are predicted to have high conversion and those that are not. As a result, it is better to build a positive or negative label based on the actual measured conversion and evaluate the predicted model's performance using a binary classification problem.

IMPLEMENTATION AND RESULTS

In order to compare the efficacy of various techniques and descriptors for predicting protein attributes, prediction models were trained using every possible combination of descriptor available for each data set. Since CNN takes a 2D matrix as input, there are 44 possible combinations of technique and descriptor. Each set of prediction models was trained using the other set as testing data. The following are the modifications made to the hyperparameters in Table 1:

Table 1. Hyperparameters [33]

Parameters	Description/Values		Parameters	Description/Values
Learning rates	0.001, 0.005, 0.01		Momentum	0.9
Optimization technique	Nesterov_momentum		Objective	mse
Kernel	GeLU		Maximum training epochs	500
Number of convolution layers	4		Initialization	HeNormal
Number of dense layers	2		Size of evaluation batch	16
Batch size	20, 40, 60		Size of embedding	128

The aforementioned strategy has been successfully applied to both public and private datasets. The publicly available dataset contains various experimentally observed properties that can be predicted and a wide range of protein classifications (membrane, globular).

This is helpful since it allows you to evaluate the broader applicability of the principles outlined here. Particular enzymes, starting with "Enzyme A" and going through "Enzyme D," are given unique names to highlight their significant involvement in the confidential information. Even though the specific chemistry at play in each case is unique, all of the patented enzymes are manufactured using the same precise protocol. This is because the end goal of all patented enzymes is substrate conversion (how quickly an enzyme turns a substrate into a product). This phenomenon has developed consequently. Normalisation and conversion of the raw experimental results are performed to facilitate quantitative modelling.

The 11-dimensional PCscores descriptor is used as a single amino acid feature to characterise each of the 300 amino acids in the input protein sequence. In the top block of Figure 2, 11 vertical lines represent the input characteristic for a single protein, which is a matrix of size 11 by 300. When the input from the convolutional layers in the near vicinity has been processed, the next layer does a downsampling along the sequence dimension using a max-pooling layer. This is done to restrict the amount of previously retrieved features. Each convolutional layer comprises a collection of filters that, in a manner akin to that of a sliding window, extract a unique set of properties from the data channels of the layer behind it. Each convolutional layer comprises a collection of filters that, in a manner akin to that of a sliding window, extract a unique set of properties from the data channels of the layer behind it.

It is essential to plan out CNN's structure in great detail in advance. Changing the number of convolution and max-pooling layers, the number of filters in each convolution layer, the filter size and stride, and the number of fully connected layers that come after the flattening of the CNN's output features are all ways to alter the model's architecture. All of these settings can be found in the model's configuration file.

Several model configurations were available as options during the tuning procedure. The hyperparameters, including the minibatch size and the learning rate, were used in a gridsearch. The GPU processing was handled by Theano, while the Lasagne module was used to build the structure of the Python CNN model. The MXNet Python library's features were implemented using the MLP model's GPU mode. While the RF model was built in Python with the Scikit-learn module, the remaining machine learning techniques relied on the R packages glmnet, kernlab, xgboost, and caret. Experiment hardware consisted of a 2.40 GHz Intel(R) Xeon(R) CPU E5-2640 v4 and a single NVIDIA TITAN X (Pascal) GPU card. Figure 3 displays the optimal tuning of the proposed model, which indicates superior training performance in terms of the cost function. Figure 4 depicts the variation of estimating error and the learning features is reflected from converging nature of the error.

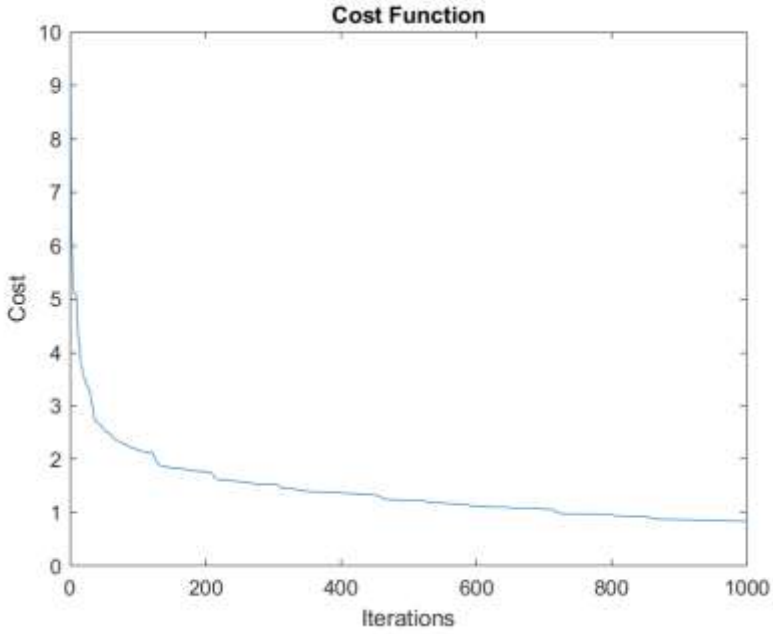


Figure 3 Cost function v/s time

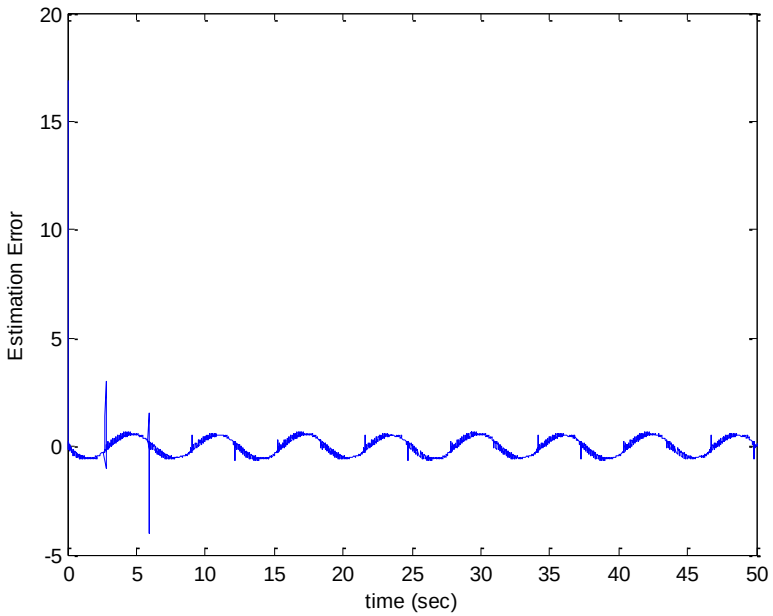


Figure 4 Estimation error v/s time

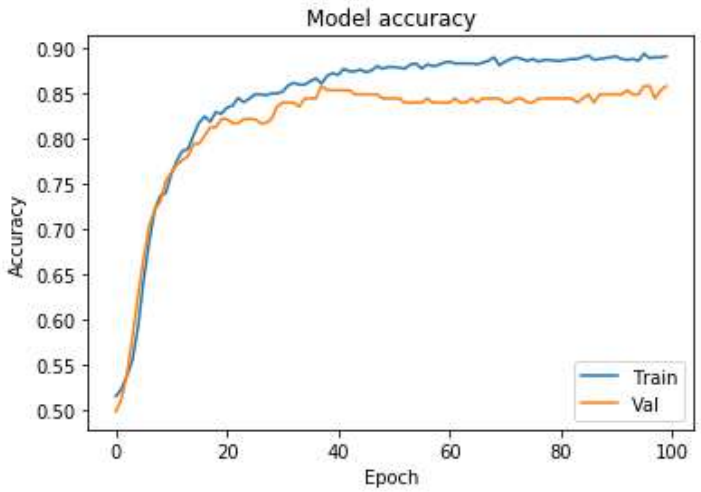
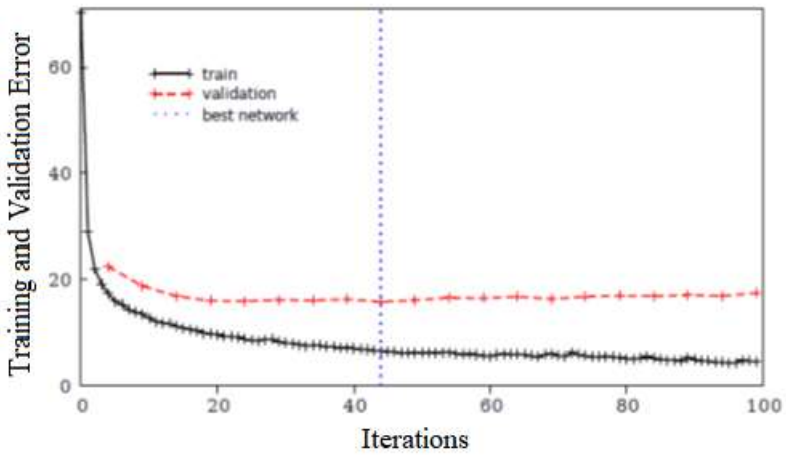


Figure 5 Modeling Accuracy v/s number of iterations



b

Figure 6 Training and Validation Error wrt number of iterations

Figure 5 and 6 show the modelling accuracy and error during the training and validation over time for the training run on the online database.

CONCLUSION

In this study, we present a novel AI-based method for classifying protein sequences. Two single-amino-acid descriptors and one structure-based, three-dimensional descriptor are employed to construct prediction models, and their efficacy is compared. Several evaluation metrics were applied to both public and private data sources to determine the accuracy of the predictions. The study found that convolution neural network models constructed from descriptions of amino acid properties are most effective in addressing protein redesign issues in the pharmaceutical industry. Even though CNN models have a more complex model structure and a large number of hyperparameters compared to other machine learning techniques, the recommended model structures and hyperparameter sets that have been optimized for the data set may serve as a good starting point for researchers pursuing a machine learning approach to protein engineering. In contrast to other methods, which only take into account changed sites, the CNN models recover high-level features based on the ordering information of the entire protein sequence.

REFERENCES

[1] Tillquist Richard C. Low-dimensional representation of biological sequence data. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. BCB '19, New York, NY, USA: Association for Computing Machinery; 2019, p. 555.

[2] Villmann Thomas, Schleif Frank-Michael, Kostrzewa Markus, Walch Axel, Hammer Barbara. Classification of mass-spectrometric data in clinical proteomics using learning vector quantization methods. Briefings Bioinform 2008;9(2):129–43.

[3] Schleif Frank-Michael, Villmann Thomas, Hammer Barbara. Prototype based fuzzy classification in clinical proteomics. Internat J Approx Reason 2008;47(1):4–16.

[4] Alley Ethan C, Khimulya Grigory, Biswas Surojit, AlQuraishi Mohammed, Church George M. Unified rational protein engineering with sequence-based deep representation learning. Nature Methods 2019;16(12):1315–22.

[5] A. E. W. Johnson, T. J. Pollard, L. Shen, H. Lehman, L. Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi and R. G Mark, "Mimic-III, a freely accessible critical care database, Scientific data, 3:160035, 2016.

[6] Nambiar Ananthan, Heflin Maeve, Liu Simon, Maslov Sergei, Hopkins Mark, Ritz Anna. Transforming the language of life: transformer neural networks for protein prediction tasks. In: Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2020; P. 1–8.

[7] Heinzinger Michael, Elnaggar Ahmed, Wang Yu, Dallago Christian, Nechaev Dmitrii, Matthes Florian, Rost Burkhard. Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinformatics 2019;20(1):1–17.

[8] Madani Ali, McCann Bryan, Naik Nikhil, Keskar Nitish Shirish, Anand Namrata, Eguchi Raphael R, Huang Po-Ssu, Socher Richard. Progen: Language modeling for protein generation. 2020, arXiv preprint arXiv:2004.03497.

[9] Elnaggar Ahmed, Heinzinger Michael, Dallago Christian, Rehawi Ghalia, Wang Yu, Jones Llion, Gibbs Tom, Feher Tamas, Angerer Christoph, Steinegger Martin, et al. ProtTrans: towards cracking the language of life's code through self-supervised learning. 2021, BioRxiv, 2020-2007.

[10] Rives Alexander, Meier Joshua, Sercu Tom, Goyal Siddharth, Lin Zeming, Liu Jason, Guo Demi, Ott Myle, Zitnick C Lawrence, Ma Jerry, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci 2021;118(15).
[11] Rao Roshan, Bhattacharya Nicholas, Thomas Neil, Duan Yan, Chen Xi, Canny John, Abbeel Pieter, Song Yun S. Evaluating protein transfer learning with TAPE. In: Advances in Neural Information Processing Systems. 2019.
[12] Kimothi, D.; Soni, A.; Biyani, P.; Hogan, J. M. Distributed Representations for Biological Sequence Analysis. 2016, arXiv preprint arXiv:1608.05949.
[13] Yang, K. K.; Wu, Z.; Bedbrook, C. N.; Arnold, F. H. Learned Protein Embeddings for Machine Learning. Bioinformatics 2018, 34, 2642–2648.
[14] Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. International conference on machine learning 2014, 1188–1196.
[15] Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-Learning-Guided Directed Evolution for Protein Engineering. Nat.Methods 2019, 16, 687.
[16] Wu, Z.; Kan, S. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine Learning-Assisted Directed Protein Evolution with Combinatorial Libraries. Proc. Natl. Acad. Sci. U. S. A. 2019, 116, 8852–8858.
[17] Yang, K. K.; Wu, Z.; Arnold, F. H. Machine Learning in Protein Engineering. 2018,arXiv preprint arXiv:1811.10775.
[18] Wei L, Xing P, Zeng J, Chen J, Su R, Guo F. Improved pre-diction of protein–protein interactions using novel negative samples, features, and an ensemble classifier. Artif Intell Med. 2017;83:67–74.
[19] Zhou YZ, Gao Y, Zheng YY. Prediction of protein-protein inter- actions using local description of amino acid sequence. In: Zhou M, Tan H, editors. Advances in computer science and education applications. Berlin: Springer; 2011. p. 254–62.
[20] You ZH, Zhu L, Zheng CH, Yu HJ, Deng SP, Ji Z. Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. BMC Bioinform. 2014;15(15):1–9.
[21] Xu H, Xu D, Zhang N, Zhang Y, Gao R. Protein-protein interaction prediction based on spectral radius and general regression neural network. J Proteome Res. 2021;20(3):1657–65.
[22] Du X, Sun S, Hu C, Yao Y, Yan Y, Zhang Y. DeepPPI: boosting prediction of protein–protein interactions with deep neural networks. J Chem Inf Model. 2017;57(6):1499–510.
[23] Guo Y, Chen X. A deep learning framework for improving protein interaction prediction using sequence properties. bioRxiv, 843755; 2019
[24] Wang X, Wu Y, Wang R, Wei Y, Gui Y. A novel matrix of sequence descriptors for predicting protein-protein interactions from amino acid sequences. PLoS ONE. 2019;14(6): e0217312.
[25] Jha K, Saha S. Amalgamation of 3D structure and sequence information for protein–protein interaction prediction. Sci Rep. 2020;10(1):1–14.
[26] Li H, Gong XJ, Yu H, Zhou C. Deep neural network based predictions of protein interactions using primary sequences. Molecules. 2018;23(8):1923.
[27] Gonzalez-Lopez F, Morales-Cordovilla JA, Villegas-Morcillo A, Gomez AM, Sanchez V. End-to-end prediction of protein-protein interaction based on embedding and recurrent neural networks. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE; 2018. p. 2344–2350.
[28] LeCun Y, Kavukcuoglu K, Farabet CC, others (2010) Convolutional networks and applications in vision. In: ISCAS. IEEE, pp 253–256.
[29] Lee C-Y, Gallagher PW, Tu Z (2016) Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In: Artificial Intelligence and Statistics. pp 464–472.
[30] Li S, Liu Z-Q, Chan AB (2014) Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp 488–495.
[31] Nwankpa C, Ijomah W, Gachagan A, Marshall S (2018) Activation Functions: Comparison of trends in Practice and Research for Deep Learning. arXiv Prepr arXiv181103378.
[32] Qureshi AS, Khan A (2018) Adaptive Transfer Learning in Deep Neural Networks: Wind Power Prediction using Knowledge Transfer from Region to Region and Between Different Task Domains. arXiv Prepr arXiv181012611.
[33] Shin H-CC, Roth HR, Gao M, et al (2016) Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Trans Med Imaging 35:1285–1298.