Artificial Neural Network for Predicting the Protein Localization Sites in Yeast

Jamlech Iram N. Gojo Cruz

***Abstract* – Protein mislocalization contributes to the development of a variety of human diseases. Thus, automating protein localization became crucial and has been a prominent area of research in bioinformatics since this provides the physiological environment for their function. Among the machine learning algorithms, Artificial Neural Networks (ANNs) shown significant improvement in predicting the classes of the Saccharomyces cerevisiae, or yeast data set, containing 1484 samples. This study developed an ANN. Input with 8 attributes and output variables categorized into 10 classes were encoded. The input layer that corresponds to the attributes of the amino acid sequences, two hidden layers uses the Rectified Linear Unit (ReLU) activation function, and the output layer uses Softmax for multi-class classification. SMOTE is applied to aid the data imbalance among the class. To evaluate the model, accuracy, loss, confusion matrix, precision, recall, and ROC AUC as metrics are used. The ANN model resulted in an accuracy of 81% on training and 51% on the test data set, compared to the initial classification accuracy of 55% of the pioneer study.**

***Index Terms* – artificial neural network, yeast, protein localization site**

I. INTRODUCTION

Eukaryotic cells are organized into membrane-bound compartments, which are characterized by specific sets of proteins and biochemically distinct cellular processes. Determining the appropriate subcellular location of proteins is important because it provides the physiological context for their function. Aberrant localization of the proteins contributes to the development of many human diseases, such as metabolic, cardiovascular, and neurological diseases, even cancer [1]. Thus, automation of protein localization became critical and has been a significant study in bioinformatics. To aid in the prediction of protein cellular localization, various machine learning methods are being applied. Artificial neural networks (ANNs), among these methods, acquired notable improvements in predicting the classes of the *Saccharomyces cerevisiae*, or yeast, the data set [2], [3].

Nakai and Kanehisa [4], [5] proposed the first expert system for deriving the amino acid sequences of proteins to determine their localization sites of Gram-negative bacteria using conditional rules. The architecture of the expert system is divided into three modules, consisting of if-then rules that classify the organism names into one of the four localization sites. Although expert systems provide a rich language for conveying knowledge, it is incapable of self-learning and hence extremely time-consuming to update or adapt to new organisms. This led Horton and Nakai [3] to define a model of classification that integrates

knowledge by experts with probabilistic reasoning. The model resulted in an accuracy of 55% on a dataset of 1,484 yeast proteins with 10 classes. Several classification techniques, including k-nearest neighbors (KNN), binary decision trees, and Bayesian classifiers, were used to improve recognition accuracy. Comparing these algorithms, the KNN achieved the highest classification accuracy [6], with approximately 60% for 10 yeast classes.

Attempts to enhance classification accuracy have been undertaken in the succeeding studies by utilizing neural networks, growing cell structures, genetic algorithms, and expanding range rules. Among the algorithms, the neural network resulted in the highest percentage of correct categorizations (57%) [7]. DeepLoc, a prediction model developed by Armenteros and others [8] employs a recurrent neural network. The model was trained and validated using a different protein dataset collected from one of the most recent UniProt releases, and this model achieved a high level of accuracy. Based on the methods performed by various studies, the advantages of ANN can be utilized and experimented to achieve a higher accuracy in the yeast data set.

This study aims to design an ANN for predicting the protein localization sites in *S. cerevisiae* or yeast dataset. It will classify the amino acid sequences into one of the 10 classes of localization sites and analyze the result of the ANN prediction through categorical accuracy, logarithmic loss, confusion matrix, precision, and recall.

II. MATERIALS AND METHODS

A. Input and Output

1). Input: To aid in classification, eight attributes were used as an input from the amino acid sequences: the McGeoch's method for signal sequence recognition (mcg); the von Heijne's method for signal sequence recognition (gvh); the score of the ALOM membrane-spanning region prediction program (alm); the result of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins (mit); the presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen (erl); the peroxisomal targeting signal in the C-terminus (pox); the score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins (vac); and the result of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins (nuc).

The input attributes are preprocessed using standardization. Standardization turns the dataset’s features into a standard Gaussian (or normal) distribution with a mean of zero and a standard deviation of 1 [9]. Considering that networks trained on standardized data produce more accurate outcomes in general [10], the study uses Scikit-learn StandardScaler module to implement data standardization.

2). Output: Yeast proteins were classified into the following ten classes: cytosolic or cytoskeletal (CYT); nuclear (NUC); mitochondrial (MIT); membrane proteins with no N-terminal signal (ME3); membrane proteins with uncleaved signal (ME2); membrane proteins with cleaved signal (ME1); extracellular (EXC); vacuolar (VAC); peroxisomal (POX); and endoplasmic reticulum lumen (ERL).

Since the classes are categorical variables, the labels were encoded using one-hot encoding. The categorical feature variables are converted into an integer matrix. This matrix is sparse, with each column denoting a single possible value for a category. To implement this, the program uses the LabelEncoder module of Scikit-learn to convert a vector of hashable categories (or labels) to an integer representation by encoding each label with a value between 0 and the number of categories minus one. The results will then be undergoing one-hot encoding [9].

The ANN outputs a vector of probabilities for each class. To process the output, we identify the one with the highest probability and change its value to 1, and the rest 0. It will resemble the one-hot encoding applied to the classes.

B. ANN

This study developed an ANN with an input layer, two hidden layers, and an output layer. The input layer has 8 nodes, which corresponds to the attributes of the amino acid sequences. The first hidden layer has 12 nodes while the second layer has 11 nodes. Both layers use the Rectified Linear Unit (ReLU) activation function. The ReLU activation function has been the most extensively utilized activation function for deep learning applications with state-of-the-art results [11]. The ReLU is a faster learning activation function [12], outperforms and generalizes the Sigmoid and Tanh activation functions in deep learning [13], [14]. Because the ReLU represents a nearly linear function, it retains the features of linear models that make them amenable to optimization via gradient-descent methods [15].

Each input element is subjected to a threshold operation using the ReLU activation function, in which values less than zero are set to zero. This gives us the following:

$$f(x) = max(0, x) = \begin{cases} x_i, & if\ x_i \geq 0 \\ 0, & if\ x_i < 0 \end{cases}$$

This function rectifies the values of the inputs that are less than zero, effectively driving them to zero and resolving the vanishing gradient problem associated with the previous types of activation functions.

On the other hand, Softmax was used in the output layer, consisting of 10 nodes that map to the 10 classes of localization sites. Softmax is used to derive the probability distribution of a vector of real

numbers. The Softmax function returns a range of values between 0 and 1, with the probability sum equal to 1. Softmax has the following equation:

$$f(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)}$$

In multi-class models, the Softmax function returns probabilities for each class, with the target class having the highest probability [16].

For the loss function, the ANN uses the Categorical Cross-Entropy loss. It is a measure of dissimilarity between probability distributions in their domain. The Softmax activation is used to convert the network output to this prediction vector, which is interpreted as the log-odds distribution of the probability distribution over the classes. The cross-entropy of the anticipated and true class distributions is then used as a measure of dissimilarity [17].

An optimizer is also used in training the model, specifically the Adam algorithm. Adam is a technique for optimizing stochastic objective functions using first-order gradients, based on adaptive estimations of lower-order moments. The method is simple to implement, computationally efficient, requires minimal memory, is invariant to gradient diagonal rescaling, and is ideally suited for issues with a large amount of data and/or parameters [18]. The following parameters are used in the Adam class: learning rate, α = 0.001; β_1 = 0.9, the exponential decay rate for the first moment estimates; β_2 = 0.999, the exponential decay rate for the second moment estimates; and ε = 1e-7, a small constant for numerical stability. All of these are the default values in the Adam optimizer. Each network weight has its learning rate that is adjusted individually as learning progresses.

Once the ANN is defined and compiled, the backpropagation algorithm is used to train the network, together with the activation functions and loss function provided. In a backpropagation approach, the network is trained on the training dataset for a specific number of epochs. Each epoch is divided into batches, also known as batch-based learning. This specifies the number of patterns to which the network is trained before the update of weights in an epoch. For this study, we applied 100 epochs with a batch size of 5, as a result of the cross-validation.

C. Dataset

The yeast dataset is partitioned into three sets: 70% of each class for training, 15% for validation, and 15% for the test set. To deal with the imbalanced data, the Synthetic Minority Oversampling Technique (SMOTE), a method for building classifiers from unbalanced datasets (Chawla et al., 2002), is applied.

Figure 1 and 2 shows the distribution of the samples from the different classes and the SMOTE applied to the training dataset.

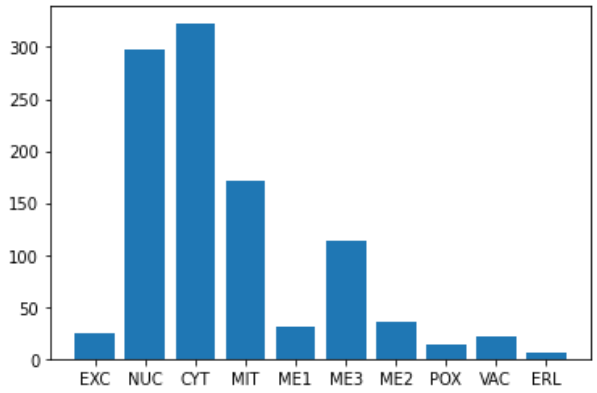

Fig. 1 Original distribution of the dataset

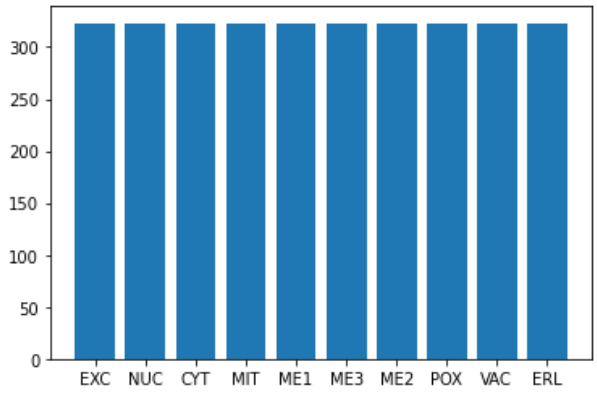

Fig. 2 Applying SMOTE on the dataset

SMOTE works by selecting the minority samples that are adjacent in the feature space, drawing a line between them, and drawing a new sample at a position along that line. Random oversampling is also used in the ERL class with only 5 instances, considering that SMOTE requires the minority class to have at least 6 samples. Table 1 outlines the number of samples per classification that appears in the dataset.

TABLE I. Class distribution in the yeast dataset

Localization Site	Number
CYT (cytosolic or cytoskeletal)	463
NUC (nuclear)	429
MIT (mitochondrial)	244
ME3 (membrane protein, no N-terminal signal)	163
ME2 (membrane protein, uncleaved signal)	51
ME1 (membrane protein, cleaved signal)	44
EXC (extracellular)	35
VAC (vacuolar)	30
POX (peroxisomal)	20
ERL (endoplasmic reticulum lumen)	5

D. Quality Assurance

To evaluate the ANN model, we used the following types of evaluation metrics: accuracy, loss, and confusion matrix, which is interpreted to get the precision, recall, and the area under the curve.

1). Accuracy: Accuracy is the ratio of correct predictions to the total sampled data.

$$Accuracy = \frac{No.of\ correct\ predictions}{Total\ no.of\ predictions}$$

Most of the literature reviewed used accuracy to evaluate their classifiers for the Yeast dataset. However, it works efficiently only if the quantity of samples in each class is equal. Thus, other evaluation methods are considered.

2) Loss: Logarithmic Loss is a loss function that penalizes incorrect classifications. It is effective for categorizing multiple classes. After assigning the probabilities to each class for each sample the Log Loss is calculated as follows, given that N samples are divided into M classes:

$$Logarithmic\ Loss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \cdot log(p_{ij})$$

where y_{ij} denotes whether sample *i* is a member of class *j* and p_{ij} indicates the probability of sample *i* being a member of class *j*. Log Loss has no upper limit therefore when the Log Loss is closer to 0, it suggests that the model yields greater accuracy.

3) Confusion Matrix: All other metrics will be associated with the confusion matrix. In this case, we used an N x N confusion matrix, where N = 10 corresponds to the 10 classes of localization sites. After determining the positive and negative classes, true positives, true negatives, false positives, and false negatives are defined. The number of true predictions lies on the main diagonal. Given these, we will be able to calculate other metrics.

4) Precision: Precision gives us the proportion of predicted positives that are true positives. By dividing the true positives by the sum of true positives and false positives, the precision is computed:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

5) Recall: The recall is the proportion of actual positives that are correctly classified. The recall is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

III. RESULTS AND DISCUSSION

The developed ANN achieved an accuracy of 81% on the training data set, and 51% on the test set. Although the loss is 0.51 on the training, the test set resulted in a 1.73. Figure 1 and 2 shows the confusion matrix for the train test set respectively.

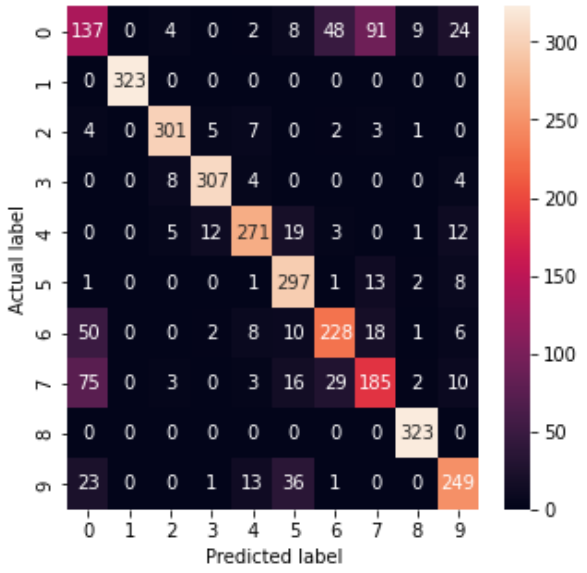

Fig. 3: Train set confusion matrix

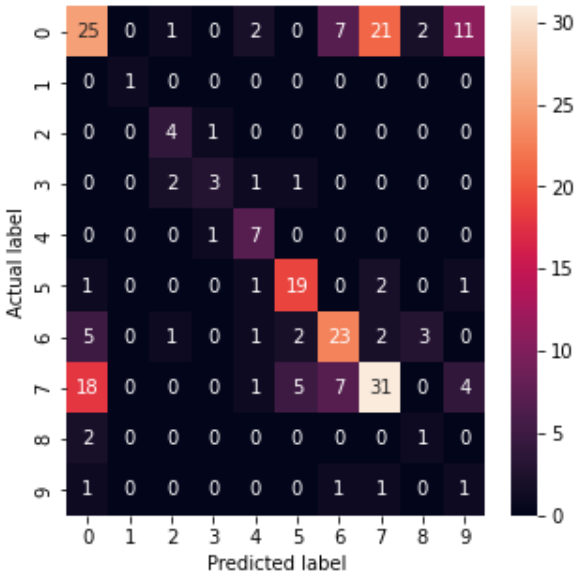

Fig. 4: Test set confusion matrix

Aside from these, the precision and recall are also calculated. The model obtained a precision of 87% and a recall of 73%. On the other hand, a 57% precision and 41% recall are acquired after testing. Table 2 and 3 summarizes the metrics used in analyzing the model. Area Under Curve (AUC) and the Precision Recall Area Under Curve (PRC) has also been included.

TABLE II. Metrics used in analyzing the model from training

Metric	Value
Accuracy	0.8115
Loss	0.5142
Precision	0.8662
Recall	0.7319
AUC	0.9848
PRC	0.9013

TABLE III. Metrics used in analyzing the model from testing

Metric	Value
Accuracy	0.5133
Loss	1.7344
Precision	0.5749
Recall	0.4107
AUC	0.9058
PRC	0.5524

We can infer that the model performs significantly well on the training dataset, but the accuracy obtained from the test set has no significant difference from the classifiers used in the literature cited. Overfitting is also a challenge in the yeast dataset. We tried higher values of epochs during training, although it resulted in a high accuracy, the loss function of the validation set increased notably and the accuracy did not improve, making the model subjected to overfitting. The imbalance data from the test set may have also contributed to the obtained results. Although having a low accuracy and high loss values on the testing phase, the model is reasonable enough to generalize the prediction of localization sites.

IV. CONCLUSION AND RECOMMENDATION

In this study, we designed an ANN for predicting the protein localization sites in *S. cerevisiae* or yeast dataset and was able to classify the sequences into one of the 10 classes of localization sites. We analyzed the result of the ANN prediction through categorical accuracy, logarithmic loss, confusion matrix, precision, and recall. Although the model provided a high accuracy and low loss value during training, the lack of representation on some of the classes in the yeast dataset and the issues of imbalanced data on the test set contributed to a low performance of the model.

The ANN model can be further improved by applying efficient techniques on the preprocessing of data to solve the issue of imbalanced dataset. The solutions to solve the challenges on overfitting should also be considered to improve the performance of the model.

REFERENCES

[1] M. C. Hung and W. Link, "Protein localization in disease and therapy," *Journal of Cell Science*, vol. 124, no. 20, 2011, doi: 10.1242/jcs.089110.

[2] A. D. Anastasiadis and G. D. Magoulas, "Analysing the localisation sites of proteins through neural networks ensembles," *Neural Computing and Applications*, vol. 15, no. 3–4, 2006, doi: 10.1007/s00521-006-0029-y.

[3] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins.," *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, vol. 4, 1996.

[4] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Bioinformatics*, vol. 11, no. 2, 1991, doi: 10.1002/prot.340110203.

[5] K. Nakai and M. Kanehisa, "A knowledge base for predicting protein localization sites in eukaryotic cells," *Genomics*, vol. 14, no. 4, 1992, doi: 10.1016/S0888-7543(05)80111-9.

[6] P. Horton and K. Nakai, "Better prediction of protein cellular localization sites with the k nearest neighbors classifier.," *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, vol. 5, 1997.

[7] P. Cairns, C. Huyck, I. Mitchell, and W. X. Wu, "A comparison of categorisation algorithms for predicting the cellular localization sites of proteins," in *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, 2001, vol. 2001-January. doi: 10.1109/DEXA.2001.953078.

[8] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, "DeepLoc: prediction of protein subcellular localization using deep learning," *Bioinformatics (Oxford, England)*, vol. 33, no. 21, 2017, doi: 10.1093/bioinformatics/btx431.

[9] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. 2019. doi: 10.1007/978-1-4842-4470-8.

[10] M. S. Shanker, M. Y. Hu, and M. S. Hung, "Effect of data standardization on neural network training," *Omega*, vol. 24, no. 4, 1996, doi: 10.1016/0305-0483(96)00010-2.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," 2010.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, 2015.

[13] M. D. Zeiler *et al.*, "On rectified linear units for speech processing," 2013. doi: 10.1109/ICASSP.2013.6638312.

[14] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," 2013. doi: 10.1109/ICASSP.2013.6639346.

[15] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," *Nature*, vol. 26, no. 7553, 2016.

[16] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.03378

[17] B. Barz and J. Denzler, “Deep learning on small datasets without pre-training using cosine loss,” 2020. doi: 10.1109/WACV45572.2020.9093286.
[18] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 2015.

APPENDIX I

GITHUB REPOSITORY: https://github.com/jngojocruz/gojocruz-cmsc191-termproject