Deep Learning Based Prediction of Autism Spectrum Disorder using Codon Encoding of Gene Sequences

V. Pream Sudha, Vijaya M S

***Abstract:** The development of computational tools to recognize Autism Spectrum Disorder (ASD) originated by genetic mutations is vital to the development of disease-specific targeted therapies. Identifying genes causing the genetically transmitted ASD is still a challenging task. As genomics data is dependent on domain specific experts for identifying efficient features and extracting hand-crafted attributes involves much time, an alternate effective solution is the need of the hour. The rapid developments in the design of deep architecture models have led to the broad application of these models in a variety of research areas and they have shown considerable success in sequential data processing tasks. The primary goal of this work is to classify the ASD gene sequences by employing a Deep Neural Network based model. This in turn will enable effective genetic diagnoses of this disease and facilitate the targeted genetic testing of individuals. This work utilizes codon encoding and one hot encoding technique to transform the mutated gene sequences which are exploited for self learning the features by deep network. Experiments showed that the performance of the proposed model was better than that of the conventional Multilayer Perceptron with promising accuracy of 77.8%, 80.1% and 81.2% for three different datasets.*

***Keywords:** Deep Neural Networks, Autism Spectrum Disorder, Codon Encoding, Gene Sequences.*

I. INTRODUCTION

A key challenge in transforming health care is to gain knowledge and actionable insights from complex, high-dimensional and heterogeneous biomedical data. Biomedical data is rapidly expanding in size and has stimulated the development of novel deep learning methods that has led to practical solutions in this domain. Deep learning is a rebranding of neural networks which were developed and used in the 1980s. It is also a direct descendant of shallow learning, which can be traced back to the early work on linear regression by Gauss and Legendre. Deep learning networks are structured in layers to create an artificial neural network that can learn and make intelligent decisions on its own. The relatively recent rebranding and expansion of deep learning is driven by trends and progress in data and computing power in the form of clusters of CPUs/GPUs and the cloud. The deep learning expansion has also led to the development of robust, well-maintained deep learning software libraries, such as Theano, Caffe and TensorFlow. These trends have enabled successful applications of deep learning to many areas ranging from computer vision to speech recognition, natural language processing, self driving cars and games. Deep learning in high-throughput biology is used to capture the internal structure of increasingly larger and high-dimensional data sets like DNA sequencing and RNA measurements. In a deep neural network, every layer produces a representation of the observed patterns based on the data it receives as inputs from the layer below, by optimizing a local unsupervised criterion. Deep neural networks process the inputs in a layer-wise nonlinear manner to pre-train the nodes in subsequent hidden layers to learn deep structures and representations that are generalizable. A supervised layer is provided with these representations as input and the entire network is adjusted using the backpropagation algorithm for representations optimized for the specific task. The neural networks used in deep learning are networks of simple computational units connected by synaptic weights. The computational units normally compute a weighted average of their inputs, weighted by the incoming synaptic weights, and then apply a linear or nonlinear function to this weighted average. These computational units can be connected into complex architectures like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Restricted Boltzmann machine (RBM), Auto encoder (AE). CNNs rely on local connections and are tied weights across the units followed by feature pooling to obtain translation invariant descriptors. RNNs are useful to process streams of data and are composed by one network performing the same task for every element of a sequence, with each output value dependent on the previous computations. Long short term memory (LSTM) and Gated Recurrent Unit (GRU) networks addressed the vanishing gradient problem by modeling the hidden state with cells that decide what to keep in memory given the previous state, the current memory and the input value. Restricted Boltzmann Machine discovers a probability distribution over the input space and is a generative stochastic model, whereas an AE is an unsupervised learning model.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

V. Pream Sudha*, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore, India. Email: preamsudha@psgrkcw.ac.in

Dr. M. S. Vijaya, Department. of Computer Science, PSGR Krishnammal College for Women, Coimbatore, India. Email: msvijaya@psgrkcw.ac.in

Retrieval Number: A1817109119/2019©BEIESP
DOI: 10.35940/ijeat.A1817.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

First systematic applications of deep learning methods in computational biology were focused on the prediction of splice sites and coding regions [1,2].

Current modern applications of deep learning in genomics are focusing on the analysis of actual DNA or RNA sequences and the inference of functional properties and phenotypic consequences associated with mutations. For instance, CNNs are used in [3] to predict sequence specificities of DNA and RNA-binding proteins. Deep learning in the form of CNNs and recurrent neural networks, specifically bidirectional gated recurrent networks, has also been used to predict the methylation state of CpG dinucleotides at the single-cell level [4]. Fakoor et al.[5] applied deep learning methods to extract key features from gene microarray data in predicting cancers. The work first applied PCA to eliminate the effects of redundant and noisy dimensions, then applied three auto-encoders methods. The stacked auto-encoder with fine-tuning achieved the best accuracy in six datasets with accuracy ranging from 76.67% to 95.15%, while the single-layer sparse auto-encoder performed the best in 5 datasets with ACC ranging from 46.76% to 91.50%. Danaee et al. (2016)[6] used SDAE to transform high dimensional, noisy RNA-seq gene expression data to lower dimensional, meaningful representations, based on which they applied different machine learning methods to classify breast cancer samples from the healthy control. Deep Neural Networks have been used in varied omics research works [7-12] focusing on protein structure prediction, gene expression regulation, protein classification and anomaly classification. [13-14] have used deep networks to detect breast cancer and diabetic retinopathy respectively. However, using deep learning methods to identify genes causing Autism Spectrum Disorder (ASD) is not a well-researched area.

Our previous work investigated the development of machine learning based model for ASD disease gene classification using simulated gene sequences. The coding measures were extracted as features and a model was built by employing supervised machine learning algorithms such as Multi Layer Perceptron, Decision Tree and Support Vector Machines. Deep models can be potentially powerful in discriminating ASD as they enable the discovery of high-level features, detect complex interactions among them, increase interpretability and support variable-size data like gene sequences. The goal of this work is to serve as a starting point to facilitate the application of deep learning in predicting ASD. It will enable the discrimination of ASD gene sequences without relying on feature engineering and domain expertise.

II. METHODOLOGY

This work explores deep learning framework for automatically learning feature representations from ASD gene sequences, modeling their sequential dependencies and finally distinguishing them. The system works directly on the gene sequences with nominal pre-processing, which minimizes feature engineering bias and reduces the need to define features apriori. In a genetic disorder like ASD, mutations completely disable genes that are crucial to early brain development. Given that a candidate ASD gene is affected by various mutations, the classification of genes helps in early diagnosis and hence for targeted therapies. The work is divided into three phases namely datasets creation, model building and evaluation which are elucidated below.

A. Datasets Creation

In this experiment three different datasets namely Codon Measures Dataset (CMDS), Pooled Mutation Dataset (PMDS) and Codon Encoded Dataset (CEDS) were created and used for evaluating the model. The aim is to identify the ASD gene sequences and so the various descriptors like codon measures and mutation features are utilized to create these datasets. As the mutated sequences are not readily available, they are simulated using the following method. In the proposed architecture as depicted in Fig.1, initially CDNA sequences of the ASD genes responsible for syndromic and asyndromic ASD are collected from HGMD database and the mutational information about these genes are collected from SFARI gene database. R coding is used for simulating these mutations with the help of mutation information.

The total dataset comprises of 500 mutated gene sequences causing syndromic and asyndromic ASD. Initially the cDNA sequence and the reference sequence are first stored as text files. Later, R script is used to make nucleotide changes in cDNA sequence against the reference gene sequence and the new mutated gene sequences are generated. Consider the missense mutational information for the SHANK3 gene such as nucleotide change is 612 C>A which indicates in the position 612 the nucleotide changes from C to A alters the protein from Asp to Glu. For example the cDNA sequence of SHANK3 gene before and after the nucleotide change is given below.

```
TCGTGCGCGTCGGCATC…………….GCGGCAGC
```

```
TCGTGCGCGTCGGCATC…………….GCGGCAGA
```

The corpus is built using 500 mutated gene sequences accounting for ten types of ASD genes. The various types of mutations included are missense, nonsense, synonymous and frameshift variants. The creation of three different types of datasets is described below.

Codon Measures Dataset (CMDS): The coding measures are dissimilar in different gene families and hence this trait is a well-chosen descriptor for identifying different gene families. The study investigated a total of 43 attributes in both intrinsic and extrinsic categories which are the contributing features for representing the mutated gene sequences. The training set for the multi-class classification problem included 500 instances with 43 descriptors labelled from 1 to 10 classes. Table I depicts the features considered.

Table – I: Features of CMDS Dataset

Features of CMDS	
Nucleotide composition	Number of donor sites
GC content	Number of acceptor sites
Rho values of biwords	CpG percent

Z scores of biwords	Ratio of CpG percent / expected
Alignment score	Number of exons

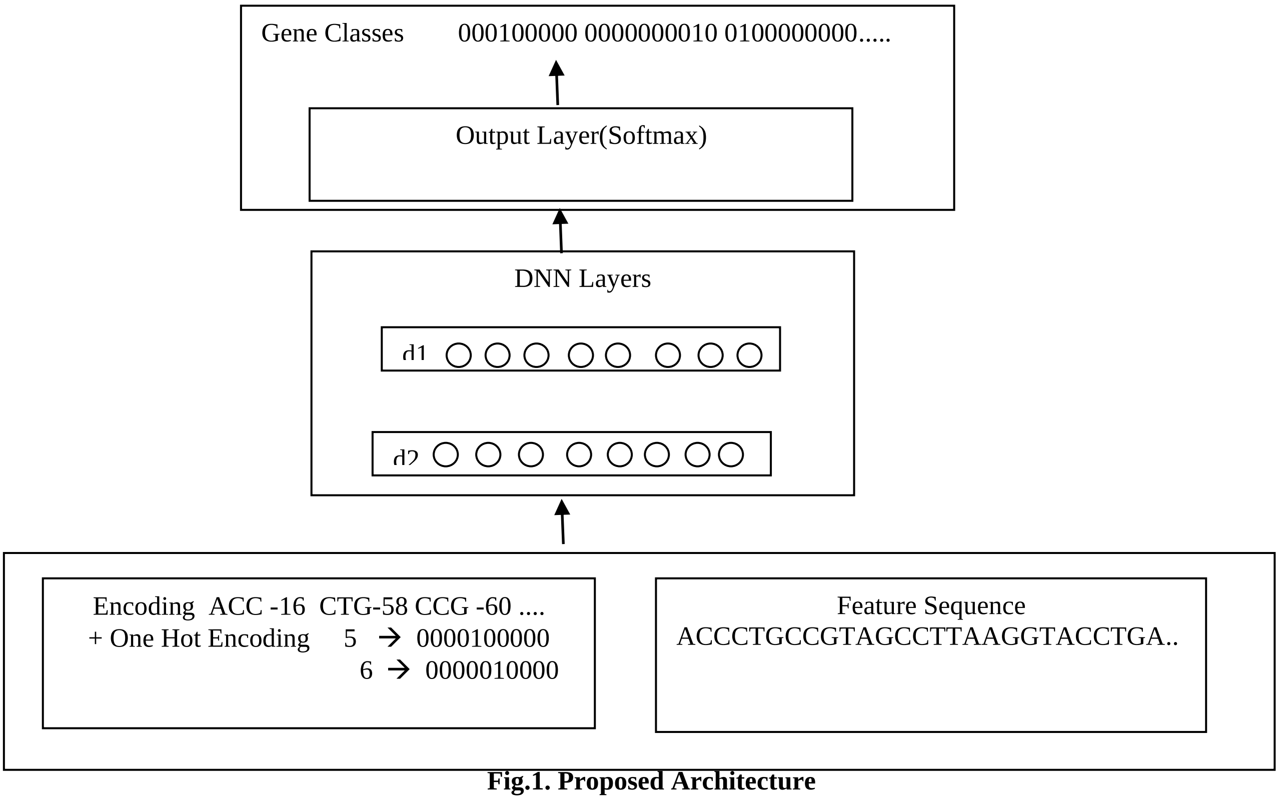

Fig.1. Proposed Architecture

Pooled Mutation Dataset (PMDS) was created initially by pooling the gene specific features (GS), substitution matrix features(SM) and amino acid change residues(AARC) to determine the dissimilarity between the mutations. A total of 15 attributes listed in Table II that describe a mutation on different aspects were investigated. These attributes can be categorized into three groups SM, GS, AARC: 6 features extracted from published substitution scoring matrices, 5 gene specific and 4 features related to amino acid residue changes. The gene specific features for the above mentioned SHANK3 gene sequence will be Mutation start position -612, mutation end -612, mutation length-1,length of CDNA sequence-7113, mutation type-1. This work utilizes the values of 6 scoring matrices namely WAC matrix, Log-odds scoring matrix collected in 6.4-8.7 PAM, BLOSUM80 substitution matrix, PAM-120 matrix, Substitution matrix (VTML160) and Mutation matrix for initial aligning which are collected from the AAIndex database. The Mutation matrix features for the above mentioned SHANK3 gene sequence for the protein alteration Asp-Glu will be 2.7. The training set consists of 500 instances with 15 features and each instance is assigned with a class label ranging from 1 to 4 as four types of mutations are taken for study.

Codon Encoded Dataset (CEDS): Once the gene sequences are simulated, it is required to transform them into a format that is suitable for training an RNN network. The simulated mutated sequences undergo the process of codon encoding. The total count of codons in a DNA sequence is 64. The simulated gene sequences are then converted into records having values ranging from 1 to 64 as there are 64 possible codons. For example ATCGGTCCCAGG is transformed as ATC GGT CCA AGG and hence 14 39 52 11.

Table – II: PMDS features

PMDS Features	
Mutation start position	Mutation end
Mutation length	Length of CDNA sequence
Mutation type	WAC matrix
Log-odds scoring matrix	BLOSUM80 substitution matrix,
PAM-120 matrix	Substitution matrix (VTML160)
Mutation matrix	Standard deviation of bigrams
Mean z-score of bigrams	Amino acid observed , expected value

All sequences are not of the same length, but in order to feed them into the DNN they must be uniform. Hence 0 padding is done to make them equal in length. The length of each record is now uniform with the size of 2582. The problem is framed as multi classification, where the expected output is a class and there are 10 possible class values. One hot encoding of the class values is used where each value is represented by a 10 element binary vector. For example class 5 is converted into 0000100000 and 10 is converted into 0000000001.

The final step is to reshape the one hot encoded sequences into a format that can be used as input to the DNN. The training set consists of 500 instances with a fixed length 2582 and each instance is assigned with one hot encoded class label ranging from 1 to 10.

B. Model Building

The basic structure of the proposed Deep Neural Network (DNN) consists of one input layer, 2 hidden layers with 8 memory units and an output layer. Once the input dataset comprising of 500 encoded gene sequences are given to the DNNs, output values are computed sequentially along the layers of the network. At each layer, the weights are adjusted suitably by multiplying the output values of each unit in the layer below by the weight vector of each unit in the current layer to generate the weighted sum. The hidden layer employs a rectifier activation function which is applied to the weighted sum to compute the output values of the layer. The representations in the layers below are modified by layer wise computations into more abstract representations. The output layer is a fully connected dense layer with 10 neurons for the 10 possible integers that may be output. As one-hot encoding is used, the output layer must create 10 output values, one for each class. The output value that is highest will be considered as the class prediction given by the model. The output layer consists of a softmax activation function that allows the network to learn and output the distribution over the possible ten output values. The network used the log loss function while training, suitable for multiclass classification problems and the efficient Adam optimization algorithm.

III. RESULTS AND DISCUSSIONS

In the deep learning based approach to detect ASD genes, three datasets namely CMDS, PMDS and CEDS were created and used for evaluating the model. In this section, the performance results of the proposed method is presented and compared with the baseline Multilayer Perceptron model. The performance of the models was evaluated based on prediction accuracy, logarithmic loss, precision, recall and F-measure. The standard 10 - fold cross-validation technique was applied to split the data and to estimate their impact on the model's prediction performance for unknown samples. When training and testing, data were segmented on mini-batches of size of 32 data segments. Varying epochs of 50,100,150,200, 250 and dropouts of 0.2 to 0.5 were experimented and the results were reported.

In this work, when the model was trained and tested with the CEDS dataset, it was able to find more than 81% of the gene sequences correctly. This is attributed to the fact that the network by itself has learned the intricacies of gene sequences, contributive features and learned representations of these sequences from different layers of the network. When trained and tested with CMDS and PMDS datasets, the model showed an accuracy of 77.8% and 80.1% which is lesser to that of CEDS dataset. The specificity of the model for CEDS dataset is better when compared to other two datasets. Identifying true negative gene sequences is equally important as that of true positives and the system is found to be performing well in this regard. The precision, recall and f-measure of the model is uniformly good for all three datasets which is a promising sign that the proposed model is excellent in automatic identification of codon encoded ASD gene sequences. It is expected that when variations of deep neural networks are applied to these datasets, it may increase the performance of the model greatly.

The validity of the model is tested using sensitivity and specificity. The sensitivity of model is the ability of a model to correctly recognize genes that are actually positive. Sensitivity is calculated by dividing true positives by the sum of true positives and false negatives. True positives are data points classified as positive by the model that are actually positive. False negatives are data points the model identifies as negative that are actually positives. The specificity of a test is the ability of a model to recognize correctly genes that are actually negative. Specificity is calculated by dividing the number of genes that are true negatives by the number of genes that are true negative and false positives. The sensitivity and specificity of the model when trained and tested with CEDS corpus is approximately equal 82.89% and 78.57 % respectively which shows that the model predicts both true positives and true negatives exactly. The DNN model has been found to show improved performance over various epochs and has reached a maximum at 250 epochs. The error rate of the model decreases over epochs as the log loss is found to be decreasing over the epochs. The prediction accuracy values of DNN model for ASD gene prediction using CEDS corpus for different dropout rates over epochs is presented in Table III. DNN achieves an accuracy of 0.72 at 250 epochs at 0.3 dropout. As it is illustrated in Table IV, the proposed method has the least log loss at 0.2 dropout over 200 epochs. Fig 2 depicts the log loss of the DNN for CEDS corpus in various epochs and it is evident that in the early epochs the log loss is the same and drastically comes down in 0.2 dropout. This is attributed to the rationale that the model learns by minimizing the false classifications as the epochs increase.

Table - III: Accuracy of DNN model for CEDS dataset

Epochs	DNN			
	0.2	**0.3**	**0.4**	**0.5**
50	0.61	0.65	0.66	0.64
100	0.63	0.68	0.65	0.65
150	0.63	0.73	0.68	0.65
200	0.64	0.76	0.69	0.66
250	0.65	0.81	0.70	0.68

Table - IV: Epochwise Log Loss of DNN model for CEDS dataset

Epochs	DNN			
	0.2	**0.3**	**0.4**	**0.5**
50	2.0812	0.9934	1.4294	1.9954
100	1.1155	1.4112	1.7846	1.4477
150	0.7922	1.0848	0.9509	0.9810
200	0.6822	0.8981	1.2164	0.9438
250	1.0399	0.87911	1.0715	0.9759

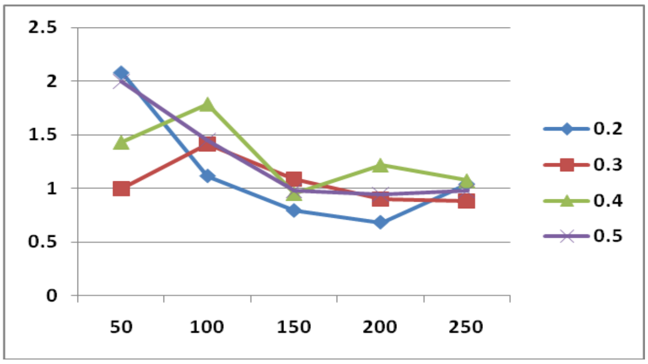

Fig.2. Log Loss of DNN Model

The model performs modestly well in finding all relevant instances in the appropriate datasets. The results depict that precision of the classifier keeps increasing with the epochs and reaches its maximum of 0.7974 at 250 epochs. At the early epochs the recall values of the classifiers do not show major difference and it reaches 0.8410 at 250 epochs. Precision, Recall and F-measure values of the DNN classifier at various epochs are depicted in Table V. As shown in Fig.3 the F-measure reaches the highest of 0.8151 at 250 epochs.

Table - V: Performance Evaluation of DNN for CEDS over epochs

Epochs	DNN		
	Precision	Recall	F-measure
50	0.6712	0.6566	0.6638
100	0.692	0.7023	0.6971
150	0.7106	0.7475	0.7331
200	0.7481	0.7984	0.7895
250	0.7974	0.8410	0.8151

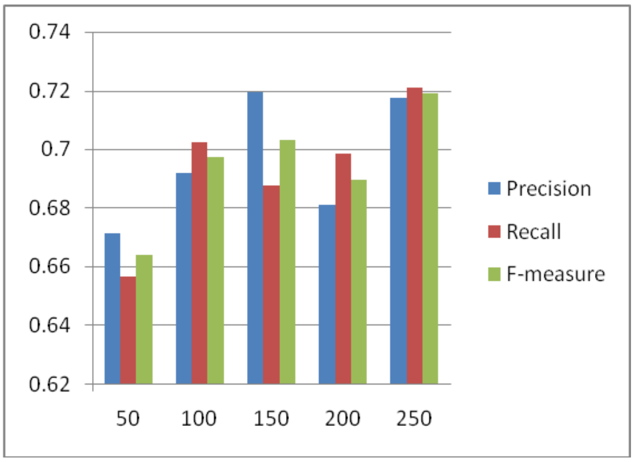

Fig.3. Performance measures of DNN

The DNN model is found to be superior at learning high-level features from the gene sequences, reducing the task of developing new feature extractor and generalizing the learning. Hence it is concluded that it is more suitable for predicting ASD genes than conventional MLP. Results clearly indicate that DNN performs comparatively better than MLP with Precision of 0.71, Recall of 0.72 and F-Measure of 0.71.. It is apparent that DNN has an upper edge over MLP while working with high dimensional biological data like gene sequences. The performance evaluation of the shallow MLP and the Deep Neural Network based on Precision, Recall and F-Measure is summarized in Table VI.

Table –VI : Evaluation Measures of DNN Vs MLP

Metrics	DNN			MLP		
	CMDS	PMDS	CEDS	CMDS	PMDS	CEDS
Precision	0.77	0.78	0.79	0.74	0.75	0.68
Recall	0.82	0.84	0.84	0.81	0.68	0.70
F-Measure	0.79	0.80	0.81	0.77	0.71	0.69

The sensitivity, specificity and accuracy rate of the DNN model for CEDS dataset is found to be convincingly high. From the observations, it is suggested that DNN can be employed to predict the ASD gene sequences that will lead to the development of new set of therapies and personalized approach to the treatments. It will reduce time involved in feature engineering and will eliminate manual errors in defining the features. Table VII shows the correctly classified instances and incorrectly classified instances of the data set for the classifiers. Fig. 4 shows that accuracy and specificity of DNN for CEDS corpus is high compared to other datasets. As illustrated in Fig. 5, DNN predicts better with an accuracy of 81.4% which is a slight edge over MLP with 67% accuracy.

Table –VII: Prediction Accuracy of DNN Vs MLP

Metrics	DNN			MLP		
	CMDS	PMDS	CEDS	CMDS	PMDS	CEDS
Correctly classified instances	388	395	406	376	370	335
Incorrectly classified instances	112	105	94	124	130	165
Sensitivity	83.7%	82.8%	82.8%	82.6	79.1%	70%
Specificity	62.1%	73.07%	78.5%	61.3%	60.7%	60.5%
Accuracy	77.8%	80.1%	81.2%	75.2%	74%	67%

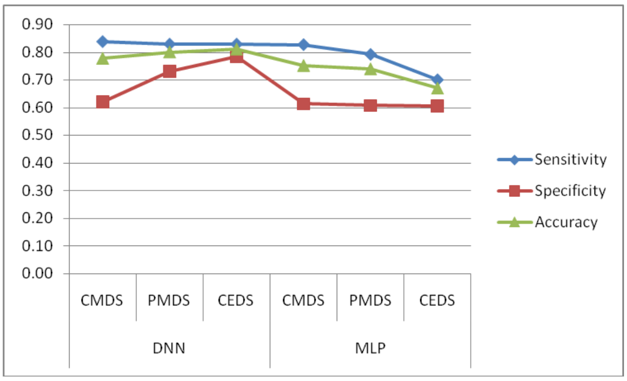

Fig.4. Comparison of classifiers on three datasets

DOI: 10.35940/ijeat.A1817.109119

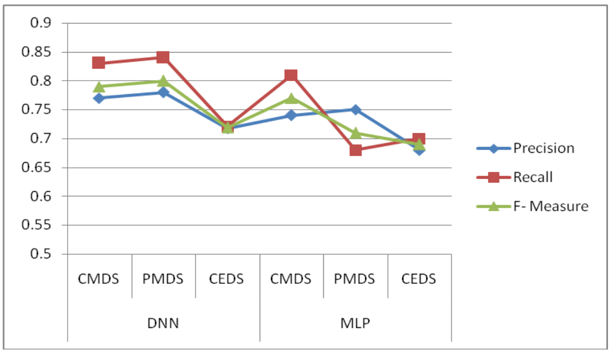

Fig.5. Performance Comparison of DNN Vs MLP

IV. CONCLUSION

Identification of ASD causative genes is still a challenging task. This work is aimed at exploring the application of deep learning based model to discriminate ASD causing genes. Deep learning can open the way toward next generation of predictive health care systems that can scale to include many millions of patient. The proposed model puts forth a codon encoded approach combined with DNN to classify ASD genes. The model was trained and tested with three different datasets namely CMDS, PMDS and CEDS. The results of the deep learning based model and the shallow MLP were compared. The effectiveness of these models to identify potential ASD causing genes was evaluated using different evaluation measures to explore the reliability of the method. It is verified that DNN-based network achieved superior results than the shallow learning method. The predictive accuracy of DNN in discriminating ASD genes is 77.8, 80.1, and 81.2 for the CMDS, PMDS and CEDS dataset which is comparatively higher than that of the shallow method. These comparisons of results provide a baseline for future research and it is expected that it can give better results when using variants of Deep Neural Networks.

REFERENCES

1. Brunak S, Engelbrecht J, Knudsen S. 1991. "Prediction of human mRNA donor and acceptor sites from the DNA sequence". J. Mol. Biol. 220(1):49–65
2. Hebsgaard SM, Korning PG, Tolstrup N, Engelbrecht J, Rouz´e P, Brunak S. 1996. 'Splice site prediction in Arabidopsis thaliana pre-mRNA by combining local and global sequence information". Nucleic Acids Res.24(17):3439–52
3. Alipanahi B, Delong A, Weirauch MT, Frey BJ. 2015. "Predicting the sequence specificities of DNA and RNA-binding proteins by deep learning". Nat. Biotechnol. 33(8):831–38
4. Angermueller C, Lee HJ, Reik W, Stegle O. 2017. "DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning". Genome Biol. 18(1):67
5. Fakoor R, Ladhak F, Nazi A et al. "Using deep learning to enhance cancer diagnosis and classification". In: Proceedings of the International Conference on Machine Learning. 2013.
6. Danaee, P., R. Ghaeini, and D.A. Hendrix, "A deep learning approach for cancer detection and relevant gene identification". Pac Symp,Biocomput, 2016. 22: p. 219-229.
7. Spencer M, Eickholt J, Cheng J. "A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction". Computational Biology and Bioinformatics, IEEE/ACM Transactions on 2015;12(1):103-12.
8. Nguyen SP, Shang Y, Xu D. 'DL-PRO: A novel deep learning method for protein model quality assessment". In: Neural Networks (IJCNN), 2014 International Joint Conference on. 2014. p. 2071-8. IEEE.
9. Zhang S, Zhou J, Hu H et al. "A deep learning framework for modeling structural features of RNA-binding protein targets". Nucleic acids research 2015:gkv1025.
10. Zhou J, Troyanskaya OG. "Predicting effects of noncoding variants with deep learning-based sequence model. Nature methods 2015;12(10):931-4.
11. Lee B, Baek J, Park S et al. "deepTarget: End-to-end Learning Framework for microRNA Target Prediction using Deep Recurrent Neural Networks". arXiv preprint arXiv:1603.09123 2016.
12. Asgari E, Mofrad MR. "Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics". PloS one 2015;10(11):e0141287
13. Wang D, Khosla A, Gargeya R, Irshad H, Beck AH. 2016. "Deep learning for identifying metastatic breast cancer". arXiv:1606.05718 [q-bio.QM]
14. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, et al. 2016. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs". JAMA 316(22):2402–10

AUTHORS PROFILE

Mrs.V. Pream Sudha is currently working as Assistant Professor in the department of Computer Science at PSGR Krishnammal College. Her areas of specialization include Machine Learning, Data mining and Bioinformatics. She has published 12 research papers in reputed national and international journals and has authored 2 ebooks.

Dr. M. S. Vijaya is currently working as Associate Professor and Head of Computer Science department at PSGR Krishnammal College. Her areas of specialization include Data Mining, Machine Learning, Support Vector Machine, Pattern Recognition and Bioinformatics. She has produced 5 Phds and 35 M.Phils. Currently she is guiding 5 PhD scholars and 2 M.Phil scholars. She has published 58 research papers in reputed national and international journals and has authored 3 books. She is a recipient of 2 state awards.