International Journal on Emerging Technologies **11**(1): 136-141(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Gated Recurrent Neural Network for Autism Spectrum Disorder Gene Prediction

V. Pream Sudha[1] and M. S. Vijaya[2]
[1]*Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women,Tamilnadu, India.*
[2]*Associate Professor, Department of Computer Science, PSGR Krishnammal College for women,Tamilnadu, India.*

(Corresponding author: V. Pream Sudha)
(Received 14 October 2019, Revised 09 December 2019, Accepted 17 December 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Autism Spectrum Disorder (ASD) is the fastest-growing complex disorder and the genetic ground of this comprehensive developmental disability is very difficult to research. Autism diagnosis for an average child is not done till the age of four, though it can be given at the age of 18 months to two years. Hence a computational model that enables the early diagnosis and personalized treatment is the need of the hour. In this research work, a deep learning based approach is proposed for Autism Spectrum Disorder (ASD) gene prediction. There are various contributors for Autism including genes, mutations, chromosomal settings influence of the environment, prenatal influences, family factors and problems during birth. Recurrent Neural Network (RNN) based Gated Recurrent Units (GRU) are implemented to develop a model that predicts ASD genes, mutations and gene susceptibility. GRUs with their internal memory capability are valuable to store and filter information using the update and reset gates. Also GRU offers a powerful tool to handle sequence data. The model is trained using three simulated datasets with features representing genes, mutations and gene susceptibility to ASD. Besides, the proposed approach is compared to original RNN and Long Short Term Memory Units (LSTM) for ASD prediction. The experimental results confirm that the proposed approach is promising with 82.5% accuracy and hence GRU RNN is found to be effective for ASD gene prediction.

Keywords: Autism, Gated Recurrent Units, Genes, Mutations, Prediction, Recurrent Neural Network.

Abbreviations: RNN, Recurrent Neural Network; GRU, Gated Recurrent Units; LSTM, Long Short Term Memory Units; ASD, Autism Spectrum Disorder.

I. INTRODUCTION

In recent times Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which are the notable RNN variants have achieved remarkable results in areas such as sequential data, speech recognition, recommendation systems, image recognition, image segmentation, content editing, image restoration and natural language processing. The major advantages of RNNs are their strong prediction performance and potential to identify long-term temporal dependencies and variable-length observations. But research works have not focused on designing RNN structures for classification of ASD gene sequences. Modeling ASD gene classification by using the influence of customized RNN models is a novel avenue and the main inspiration behind this work.

Yang *et al.,* used BiRen architecture which exploited GRU and used DNA sequences to identify enhancers [1]. The work demonstrated that the model learned representations of enhancers completely from the DNA sequence and exhibited better robustness, accuracy, and generalizability in predicting enhancers when compared to the benchmark enhancer predictor models depending on sequence attributes. Che *et al.,* (2017) proposed a deep model that directly learnt similarity of patients affected by Parkinson's disease from multiple inputs of patient records using a RNN architecture [2]. The model learnt the temporal patterns in patient sequence and was able to find similarity between two patient records using 2D-GRU. The model demonstrated promising performance proving the usefulness and effectiveness of the proposed architecture. Shen *et al.*, (2018) proposed a model KEGRU that used Bidirectional Gated Recurrent Unit (GRU) network with k-mer embedding , to recognize TF binding sites [3]. At first DNA sequences were split into k-mer sequences with particular length and striding window. A pre-trained word representation model was built using word 2 vec algorithm by considering each k-mer as a single word. In order to carry out the task of feature learning and classification, a bidirectional GRU model was then built. It was proved that the proposed method has an upper edge over state-of-the-art methods. Very less work has been done using GRU [4-10] for disease prediction and specifically Autism Spectrum Disorder prediction using DNA sequences is not a well researched area.

An innovative deep learning model based on GRU is developed to take advantage of representation of variable length data through masking. GRU is relatively new, and the performance is on par with LSTM, but computationally more efficient. Compared to LSTMs, GRUs train faster and perform better on less training data. Gene sequence data is codon encoded to represent the sequence of strings as numbers. The proposed model learns the observations and their relationships by applying masking to the inputs and uses back-propagation to train all model components. The model is able to identify the dependencies existing in the gene sequences which were long back in the previous time steps and improves the prediction results.

The vanishing gradient problem is addressed by GRU as the hidden nodes in traditional RNN are replaced by GRU node. The primary design of GRUs is that the gradient chains do not fade away due to the length of sequences as values are conceded completely through the cells. Fig. 1 shows the architecture of GRU. GRU consists of a reset gate that decides on merging the new input with the previous memory, and the update gate that identifies the previous memory to be kept around.

Reset Gate: Basically, this gate is employed from the model to fix the amount of the past information to overlook while calculating current information.

Update Gate: It allows the model to resolve the precedent information from previous time steps that is required to be passed to the future. It is really dominant as the model can choose to copy the entire information from the past and remove the threat of vanishing gradient crisis.

The reset gate allows the unit to disregard the previously computed state and the update gate computes the modification done during the activation. The hidden layer is computed using ht that holds information for the present unit and sends it down to the network. The update gate decides the information to be collected from the current memory content and preceding memory content. The model learns to set the vector z_t closer to 1 and stores a bulk of the previous information. During this time 1- z_t will be nearing 0 and hence a majority of the current content will be overlooked.

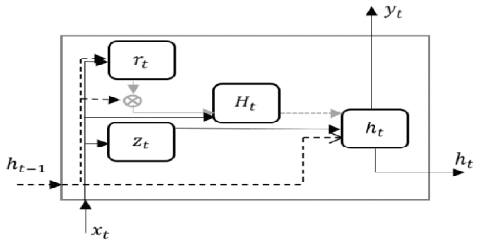

Fig. 1. Gated Recurrent unit.

Hence it is proposed to investigate the use of GRU RNN for recognizing ASD gene sequences, as it is efficient in handling unequal length sequences naturally. In our previous work [11] traditional decision tree was investigated for gene prediction. The GRU model in this work brings down the count of parameters required to be taught as it shares the parameters. Though various variants of RNN [12] are available GRU is applied here, as it is easier to train and also has the capability of retaining information from the past. This paper aims to incorporate the benefits of GRU architecture in the next level of research for the identification of ASD genes, mutations and gene susceptibility

II. MATERIALS AND METHODS

In this work GRU network exploiting the shared feature extraction between user defined layers is employed to distinguish ASD causing genes, their susceptibility and driving mutations. When deep architectures are created using the sequential method there are layers which do not share data and do not allow multiple inputs or outputs. But the functional models have a lot more flexibility and can be easily defined where layers connect to more than just previous and next layers. But when developing a GRU based functional model, instances of layers are initially created and connected directly to each other. The proposed methodology includes three functional parts such as creating datasets, model development and performance assessment as depicted in Fig. 2.

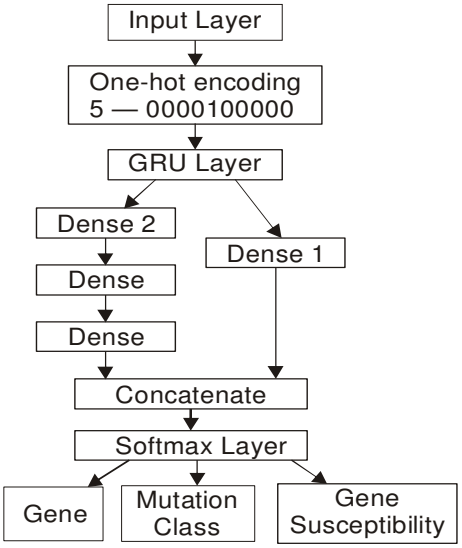

Fig. 2. Proposed GRU based ASD Prediction Model.

Datasets: In the initial phase R script is used to simulate the mutations in CDNA sequences collected from HGMD database with the help of mutational information from SFARI gene database. The corpus is constructed with 1000 mutation induced gene sequences consisting of ten types of ASD genes and four different types of mutations. The three datasets CMDS, MDS and GSDS are used here in building the GRU based gene prediction, mutation prediction and gene susceptibility prediction model respectively. The dataset CMDS consists of features such as gene features, codon features, alignment features accounting to 43 attributes captured from 1000 gene sequences of 10 gene types. Gene mutation features, amino acid change features and published matrix features are identified and extracted for preparing the MDS dataset comprising 1000 instances related to 4 mutation types with a dimension of 15. The GSDS dataset includes 25 attributes pertaining to gene, mutation, conserved protein domains, gene expression profiles and pathway interactions of 1000 instances. When the gene score is >= 0.8, >0.5 and < 0.8 and <=0.5, the gene susceptibility to ASD is defined as high, medium and low respectively.

Given a gene sequence S consisting of codons which are substrings named s_i the Eqn. (1) defines S as follows

$S= \{s_1,s_2.....s_i\}$, i = 1 to n where n =|S| / 3 (1)

Ex: The gene sequence ACACTGGTTCCA… consists of codons ACA CTG GTT CCA…

A DNA sequence is made up of nucleotide unit consisting of anyone of the four nitrogenous nucleotide bases namely adenine (A) or guanine (G) or cytosine (C) or thymine (T).

Every codon which is a triplet, codes for an amino acid and this codon mapping is mathematically formulated as given in Eqn. (2)

$s_1 \in \{xp\{yq\{zr\}\}\}$, where <xp / yq / zr> = A/C/G/T where p, q, r = 1, 2, 3, 4 (2)

For example, the triplet <x1 y3 z1> represents the Codon AGA.

During protein assembling codons composed of three nucleotide bases specify the amino acids and there exists 64 possible combinations of such codons mapped onto numbers ranging from 1 to 64. The entire gene sequence S is converted into a numerical vector V of codons CO as in Eqn. (3).

$V_j = [CO_i]_j$ where i = 1 to nr, n=4, r=3, j =total number of gene sequences (3)

The length of each record varies and hence masking is done for the maximum codon size of 2582. Masking informs the model about the inputs that are observed and missing. The length of each record varies and hence masking is done for the maximum codon size of 2582. Masking informs the model about the inputs that are observed and missing. Each class value is represented by a binary vector and one hot encoding of the class values is used in all three datasets. For example class 7 is converted into 0000001000 and 2 is converted into 0100000000. The problem is modeled as classification, in which the expected output is a class with 10 possible values. In the final step reshaping of the one hot encoded sequences are done into a format to be used as input to the GRU. There are 1000 samples in the training dataset each with a length 2582 and are assigned class label ranging from one to ten that are one hot encoded.

Model Building: The proposed model applies the concept of shared feature extraction between layers using the Keras functional API. The proposed model is built by creating instances of layers and connecting them to each other in pairs. In a functional API the model is defined with multiple input or output models along with models that share layers. It enables layers to be connected to any other layer more than just the previous and next layers.

In this work, there are two parallel sub-models designed to infer the output of a GRU feature extractor for sequence classification. The input to the model is 2582 time steps of 1 feature. A GRU layer with 10 memory cells is used to interpret this sequence. The foremost interpretation model is a shallow single fully connected layer and next a deep three layer model is designed. The interpretation models output is concatenated into a lengthy vector and is passed to the output layer for classification. There are ten neurons in the output layer which is a fully connected layer for the 10 possible integers that may be output. This layer uses the softmax activation function to enable the network to discover and output the distribution through the possible output values. The GRU model is compared with other RNN variants BRNN and LSTM.

In this model the layers take a more functional form as compared to the sequential model. The inputs to each layer are explicitly specified and the output of each layer is controlled. The layer Input_1 acts as input to GRU layer which in turn acts as input to dense_1 and dense_2. Further Dense_3 receives dense_2 as input and is passed on to dense_4. Now the output from both dense_1 and dense_4 is concatenated and given as input to dense_5. This allows the tensors to be shared with multiple layers. The total parameters used for training is 1220 out of the total 2582 parameters.

The log loss function was used while training the network that is apt for classification problems involving multiple classes. Log loss minimizes the loss function for the training data. Eqn. (4) defines the loss with respect to predictions and the true labels

$L(y,p) = - 1 / N \sum \sum y_{ij} .\log p_{ij}$ (4)

where number of samples is referred by N, y_{ij} indicates if label j is the correct classification for instance i, and the probability of assigning label j to sample i is given by p_{ij} . The network weights are iteratively updated through Adam optimizer based on training data. The parameter values considered are learning rate as 0.01, beta1 as 0.9, beta2 as 0.999 and epsilon as 1e-08. Adam optimizer uses the average of first moment, and the second moments of the gradients to adapt the parameter learning rates. The optimizer computes an exponential moving average of the gradient and that of the squared gradient. The accuracy metric along with the loss is reported for each training epoch to evaluate the skill of the model. A large batch size of 64 is used to space out weight updates. The learning rate of 0.01 is fixed and when varying dropouts from 0.2 to 0.5 are experimented for these datasets it was found that dropout of 0.3 was optimal. Varying epochs of 50, 100, 150, 200, 250 are experimented and the epoch size of 250 is fixed for the network. Table 1 shows the hyperparameters of the model.

Table 1: Hyperparameters of the proposed model.

Hyperparameters	Values
Optimizer	Adam Optimizer
Learning Rate	0.01
Dropout	30%
Activation function	Softmax
Epochs	250
Batch size	64

Learning rate hints the optimizer about moving the weights in opposite direction of the gradient for a mini-batch. In order to avoid overfitting the training data, regularization is a good approach. During training, the dropout regularization parameter randomly hops neurons, that forces others in the layer to choose the slack. In each weight update cycle nodes to be dropped-out with a given probability are arbitrarily chosen. The model was tested with different dropout percentages varying from 20- 50% and the results were recorded.

III. RESULTS AND DISCUSSION

GRU with shared feature extraction between layers has been implemented using the Keras functional API which is constructive for creating multifarious models such as multi-input or multi-output models, directed acyclic graphs and models with shared layers. The various hyperparameters are configured and the GRU layer is implemented. To improve efficiency, data were partitioned on mini-batches of size 64 during training and testing. The learning rate of 0.01 is used and when dropouts from 0.2 to 0.5 were experimented it was found that dropout of 0.3 was optimal. The Adam optimization algorithm was used and epochs of 50, 100, 150, 200, 250 are experimented.

Three models are built under these parameter settings. The GRU model is trained using these parameter settings with CMDS dataset and the ASD causative gene identification model is built. The MDS dataset is used to train the mutation prediction model and similar experiment is carried out on the GSDS dataset to build the gene susceptibility recognition model. The standard 10 - fold cross-validation technique was applied and the performance of the gene prediction, mutation

identification and gene susceptibility recognition models were evaluated based on prediction accuracy, logarithmic loss, precision, recall and F-measure.

Table 2 depicts the accuracy of the network and it is shown that GRU based ASD gene prediction model has achieved an accuracy of 82.5% at 250 epochs which is higher than that of gene susceptibility identification model which is 80.0%. As depicted in Fig. 3, at 50 epochs the mutation prediction model achieved an accuracy of 80.1%, gradually increased to 81% at 150 epochs and reached 81.8% at 250 epochs. There is an increase of about 2.2% accuracy for the gene susceptibility prediction model from 50 to 250 epochs. The accuracy of gene prediction model at 50 epochs is 78.7%, 81.5% at 150 epochs and attains a maximum of 82.5% at 250 epochs. The experiments prove that accuracy of all three models increases as epochs are increased.

Table 2: Epochwise Accuracy of GRU for Three Models.

Epochs	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
50	78.7%	80.1%	77.8%
100	79.2%	80.6%	78.1%
150	81.5%	81.0%	78.7%
200	82.0%	81.5%	79.2%
250	82.5%	81.8%	80%

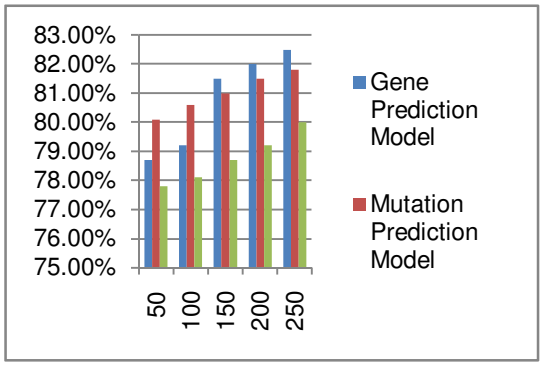

Fig. 3. Accuracy of GRU model at different epochs.

Log Loss calculates the accuracy of a classifier by punishing misclassifications. Table 3 illustrates the loss of the network and as seen in Fig. 4, the log loss keeps reducing as epochs are increasing. The models have achieved enhanced performance by reducing its misclassifications. Initially at 50 epochs the three models had log loss of 0.9845, 0.953 and 1.0849 for predicting genes, mutations and susceptibility respectively. This gets reduced with a difference of 0.2683, 0.1339 and 0.2874 respectively for the above three models at the end of 250 epochs. The log loss associated with classifying the ASD causative genes is 0.7162 which is comparatively less when compared to that of 0.8184 for mutation classification and 0.7615 for recognizing the predisposition of genes to the disorder.

The accuracy and training loss was found to be optimum at 250 epochs. The performance of the proposed network was compared with the prediction models for gene, mutation and gene susceptibility that were configured with two Dense layers and a masking layer. The learning rate and dropout was fixed as 0.01 and 0.03 respectively. The performance comparison of these three models is shown in Table 4.

Table 3: Epochwise Log loss of GRU for Three Models.

Epochs	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
50	0.9845	0.9523	1.0489
100	0.9661	0.9515	0.9961
150	0.8015	0.8893	0.9215
200	0.7965	0.8417	0.8268
250	0.7162	0.8184	0.7615

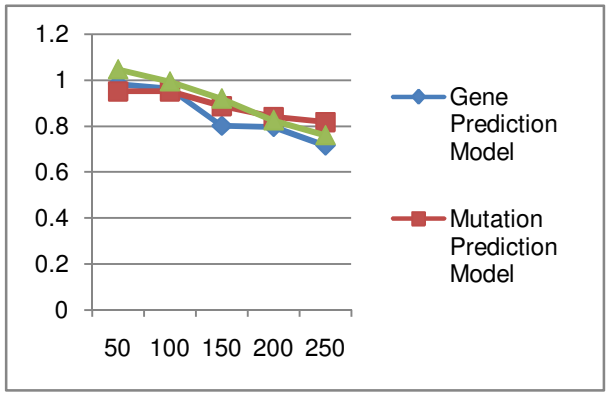

Fig. 4. Epochwise Log loss of GRU model.

Table 4: Comparative Performance of GRU based Models.

Metrics	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
Precision	0.83	0.81	0.79
Recall	0.81	0.82	0.81
F- Measure	0.82	0.81	0.8
Accuracy	82.5%	81.8%	80.00%
Correctly classified instances	414	409	401
Incorrectly classified instances	86	91	99
Sensitivity	83.5%	85.2%	82.6%
Specificity	78.3%	80.3%	77.4%

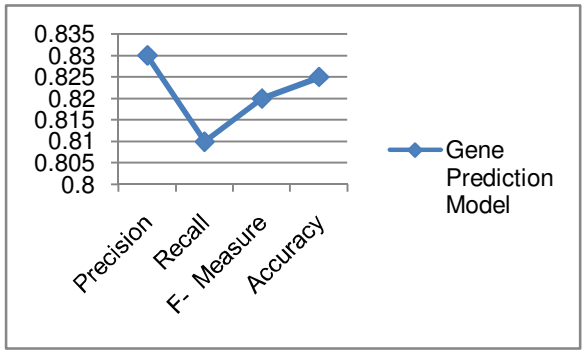

Fig. 5. Performance of GRU Based Gene Prediction Model.

The result analysis indicates that GRU model shows promising performance for the ASD gene prediction model. It is effective in predicting the ASD genes with a precision of 0.83, recall of 0.81 and F-measure of 0.82. The GRU based gene susceptibility prediction model has correctly identified 401 instances and has achieved sensitivity of 82.6%. GRU classifier for the mutation prediction has achieved 85.2 % sensitivity, 0.81 precision and a recall of 0.82. When evaluating the specificity, GRU gives a prominent score value of 80.3% for identifying the gene mutations whereas it is 78.3% and 77.4% for ASD gene and susceptibility prediction models. The performance of the GRU based ASD gene prediction model is shown in Fig. 5.

The effectiveness of the GRU classifier is compared with other deep learning models developed using DNN, BRNN and LSTM in the previous experiments. The performance measures like precision, recall, accuracy and F-measures are used to compare GRU based models with the other deep network architectures. The comparative results of accuracy and log loss of GRU based ASD gene prediction model with other deep models is reported in Table 5 and 6 and the same is depicted in Fig. 6 and 7 respectively.

Table 5: Comparative Results of GRU based ASD Gene Prediction Model with other deep Models.

Metrics	BRNN	LSTM	GRU
Precision	0.80	0.81	0.83
Recall	0.77	0.78	0.81
F- Measure	0.78	0.79	0.82
Accuracy	81.3%	81.9%	82.5%

Table 6: Comparative Results of Log Loss of GRU Based Prediction Model with other deep Models.

Model	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
BRNN	0.8010	0.9615	0.9459
LSTM	0.8284	0.9562	0.7811
GRU	0.7162	0.8184	0.7615

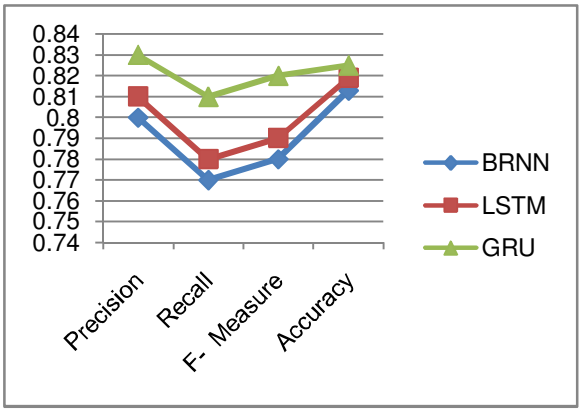

Fig. 6. Comparative performance of BRNN, LSTM, GRU models for ASD gene prediction.

It is proved that GRU model outperforms the other three deep models to identify the ASD causing genes. The LSTM model achieved precision of 0.81 and recall of 0.78 whereas GRU has achieved 0.83 and 0.82 for the same task. GRU model has comparatively less misclassifications for all three models and hence evidenced reduced logarithmic loss of 0.7162, 0.8184 and 0.7615 respectively. Both the layers have learned 1220 parameters which contributed significantly in the classification task. No rigid preprocessing procedures are required which can simplify the diagnosis procedure and save the computation costs.

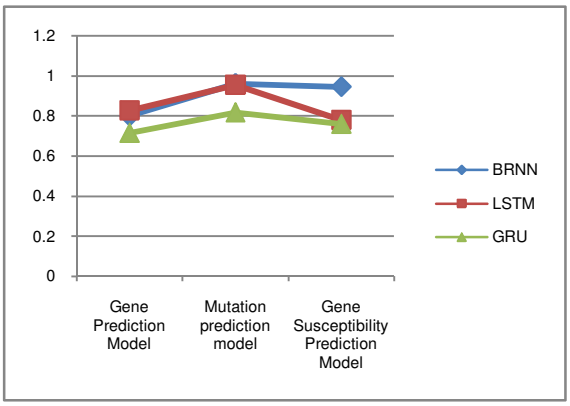

Fig. 7. Comparative Results of Log Loss of GRU based Prediction Model with other deep Models.

GRU model in this work involves two sub models which are parallelly processed to learn and share the parameters thereby reducing the total number of parameters needed to learn. Compared to other deep models BRNN and LSTM, GRU has the advantage of less parameters and easier training. The proposed method also outperforms BRNN, LSTM due to less number of updations in update gate z and reset gate r. The model has extracted gene characteristics automatically from CMDS dataset through self learning and has exhibited superior performance. The loss associated with misclassifications is also reduced for GRU based prediction model.

The exhaustive experiments of GRU architecture on three datasets ascertain that the prediction of ASD

causative genes and mutations can be done efficiently using this model

IV. CONCLUSION

In this paper, ASD gene classifier based on GRU RNN was implemented and the performance was evaluated. For training and testing phase, three datasets CMDS, MDS and GSDS were generated. By implementing various RNN models, it is confirmed that GRU approach outperforms our previous researches in this field. The GRU-based model proposed in this work, recognizes the variable length sequence information by retaining memory from previous activations. The work proves that the proposed model is promising and has an upper edge over other RNN methods on synthetic datasets. Although this paper is focused on ASD gene sequences, it is expected that this approaches will be extensively helpful for a range of gene sequence prediction tasks that will arise in healthcare.

V. FUTURE SCOPE

Empirical experiments on simulated datasets show that the GRU model outperforms deep learning models like LSTM and BRNN. Also it is proved that the proposed method is well suited for a variety of gene sequence classification problems, and is relevant to the predictive tasks in emerging health care applications. Further, the proposed method provides constructive insights into broad research challenges of gene sequence data including

(1) A common deep learning framework to deal gene sequence data.

(2) An efficient solution to exemplify the variable length gene sequence data with masking and codon encoding.

The work can be extended to the next level with the implementation of Bidirectional GRU or a combination of RNN architectures to probe the potential of embedded layers.

Conflict of Interest. There is no conflict of interest.

REFERENCES

[1]. Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., & Shu, W. (2017). BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics*, *33*(13), 1930-1936.

[2]. Che, C., Xiao, C., Liang, J., Jin, B., Zho, J., & Wang, F. (2017). An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease. In *Proceedings of the 2017 SIAM International Conference on Data Mining* (pp. 198-206). Society for Industrial and Applied Mathematics.

[3]. Shen, Z., Bao, W., & Huang, D. S. (2018). Recurrent Neural Network for Predicting Transcription Factor Binding Sites. *Scientific reports*, *8*(1), 15270.

[4]. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[5]. Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[6]. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015, June). Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, *37,* 2067-2075.

[7]. Krishnan, K. G., Vanathi, P. T., Raj, S. S., Nancy, M. & Parveene, S. S. R. (2019). Image classification using deep learning technique. *International Journal on Emerging Technologies*, *10*(2), 577-590.

[8]. Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (pp. 324-328). IEEE.

[9]. Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., & Wang, J. (2017). Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, *65*(2), 1539-1548.

[10]. Ha, J. W., Kim, A., Kim, D., Kim, J., Kim, J. W., Park, J. J., & Ryu, B. (2017). Predicting high-risk prognosis from diagnostic histories of adult disease patients via deep recurrent neural networks. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 394-399). IEEE.

[11]. Sudha, V. P., & Vijaya, M. S. (2017). Decision Tree Based Model for the Classification of Pathogenic Gene Sequences Causing ASD. In *International Conference on Smart Trends for Information Technology and Computer Communications* (pp. 201-212). Springer, Singapore.

[12]. Dey, R., & Salemt, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* (pp. 1597-1600).

How to cite this article: Pream Sudha, V. and Vijaya, M. S. (2020). Gated Recurrent Neural Network for Autism Spectrum Disorder Gene Prediction. *International Journal on Emerging Technologies*, *11*(1): 136–141.