6. DEEP NEURAL NETWORK MODEL TO PREDICT ASD GENES, THEIR SUSCEPTIBILITY AND MUTATIONS

The research is proceeded to the next level for building models with deep learning techniques. A key challenge in ASD pathogenesis is to develop advanced systems that will provide actionable insights from complex, high-dimensional and heterogeneous biomedical data. Biomedical data is rapidly expanding in size which has stimulated the development of novel methods that has led to practical solutions in this domain. 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. Hence it is vital and suitable to address this issue using contemporary deep learning methods.

In this work, a deep neural network is employed as it is competent in capturing the subtle features and their associations based on the training data for efficient discrimination of ASD genes, their susceptibility and mutations without relying on traditional pattern recognition methods. Deep neural networks are outstanding by its deepness attained by the number of hidden layers through which data passes in the network while recognizing patterns [97]. The shallow networks with single input, output and hidden layer are unable to match with the deep network which exhibits better performance in extraction of high level and hidden features. The level of abstraction increases with the hidden layers that learn intricate features differentiated by the nodes, which are summed up and combined with the features from the previous layer. 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. Deep models can be potentially powerful in discriminating ASD genes and mutations as they enable the discovery of high-level features, detect complex interactions among them, increase interpretability and support variable-size data like gene sequences.

This chapter demonstrates the deep learning approach to build the prediction model for ASD causing genes, their susceptibility and driving mutations. The problem modeling with gene, mutation and gene susceptibility datasets using deep network architecture is presented briefly. The development of three independent models trained by implementing DNN is also described.

The performance evaluation of the models with respect to various metrics and their comparison with the results of traditional machine learning models is illustrated in this chapter.

Methodology

Here deep neural architecture is employed for learning feature representations, modeling their sequential dependencies and finally distinguishing the genes, their susceptibility and triggering mutations. Disease gene sequences are simulated and used in this multi-class pattern classification problem. The proposed models are built by training the datasets for gene identification, mutation recognition and gene susceptibility prediction. The methodology includes three building blocks such as datasets creation, model building and performance evaluation and is depicted in Fig.6.1.

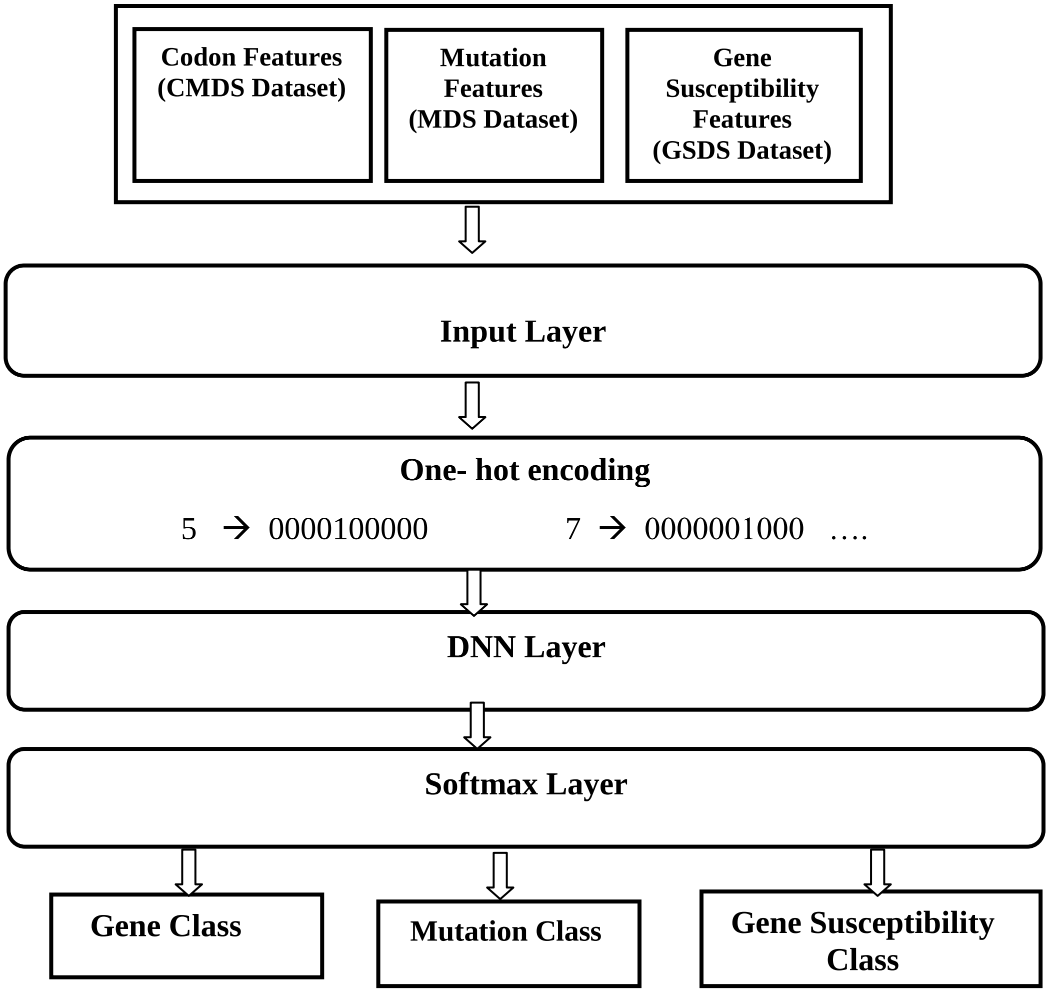
Codon Features (CMDS Dataset)
Mutation Features (MDS Dataset)
Gene Susceptibility Features (GSDS Dataset)
Input Layer
One- hot encoding
5 → 0000100000
7 → 0000001000 ….
DNN Layer
Softmax Layer
Gene Class
Mutation Class
Gene Susceptibility Class

Fig. 6.1 Architecture of DNN Models for Identifying Genes, Susceptibility and Mutations

In the first phase, the corpus developed using 1000 mutated gene sequences accounting for ten types of ASD genes and four types of mutations described in Chapter 3 is used. The three datasets CMDS, MDS and GSDS described in Chapter 4 and 5 are used to build the models for classification of genes, mutations and susceptibility prediction. One hot encoding of the class values is used where each value is represented by a binary vector. For example class 5 is converted into 0000100000 and 6 is converted into 0000010000. The feature vectors are reshaped into a format that can be used as input to the DNN.

The second phase consists of building a deep neural network wherein every layer produces a representation of the observed patterns existing in the data it receives as inputs from the layer below, by optimizing a local unsupervised criterion. 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 problem is framed as multi classification, where the expected output is a class label.

The basic structure of the proposed DNN consists of one input layer, 2 hidden layers with 8 memory units and an output layer. Once the input dataset is given to the DNN, 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 as many neurons for the possible integers that may be output. In the ASD gene classification problem, the model consists of 10 neurons in the output layer whereas for mutation classification the number of neurons is four. The output layer consists of three neurons for predicting gene susceptibility to ASD. The output value that is highest will be considered as the class prediction given by the model. The output layer is defined with softmax activation function that allows the network to learn and output the probable class values.

To improve the accuracy and efficiency of the 2 - layered DNN architecture, various hyperparameters such as batch size, epochs, dropout, learning rate and optimizer are considered. Mini - batch gradient descent is used in this network to update the parameters of the network.

The epochs mean the number of times the DNN will work through the entire training dataset. Dropout technique enables to randomly ignore selected neurons during training. The learning rate parameter is used to decide to what extent a model replaces the concepts it has learned with the new ones. The optimization algorithm in a network enables to minimize the error function and Adam optimizer is the algorithm used in this model. The above mentioned hyperparameters are fine tuned to achieve the best configuration for the network. The network is trained using three different datasets and gene identification, mutation recognition, gene susceptibility prediction models have been built.

In the final phase, 10 - fold cross-validation technique is used to evaluate the predictive performance of models using various metrics such as precision, recall, F- measure, accuracy, logloss and specificity.

Experiment and Results

In this work, experiments have been carried out by implementing deep neural network using Keras which is a high-level API for neural networks. It is coded in Python and is able to run on top of state-of-art deep learning libraries. TensorFlow deep learning library is used in this work. The primary reason for using Keras is its ability to implement the deep learning concepts with higher levels of abstraction with a simple approach. Keras modular design is another important feature. Keras is extensible and easy to learn as it works with Python. The core idea of Keras is seamlessly connected layers and simple APIs, reducing the number of user actions required for common use cases and providing understandable and actionable feedback for user error.

DNN is implemented by setting the various hyperparameters as mentioned in Table XVII. When training and testing, data were segmented on mini-batches of size 64 data segments. 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. The network used the log loss function while training, suitable for multiclass classification problems and the efficient Adam optimization algorithm. Varying epochs of 50, 100, 150, 200, 250 are experimented and the epoch size of 250 is fixed for the network.

Table XVII Hyperparameter Setting for DNN Training

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

The DNN is trained with the above parameter settings using CMDS dataset and the ASD causative gene identification model is built whereas the MDS dataset is used to train the mutation prediction model. Similar experiment is carried out on the GSDS dataset and the gene susceptibility recognition model is built.

The effectiveness of the three independent models have been evaluated using different evaluation measures such as prediction accuracy, logarithmic loss, precision, recall and F-measure to explore the reliability of the method. The standard 10 - fold cross-validation technique is applied to estimate their impact on the model's prediction performance for unknown samples. The results of the experiments are tabulated from Table XVIII to Table XX.

Table XVIII Accuracy of DNN for Varying Epochs

Epochs	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
50	73.2%	75.5%	77.9%
100	74.4%	75.7%	76.5%
150	74.1%	78.2%	78.3%
200	75.6%	78.4%	80.3%
250	80.8%	78.1%	80.4%

The experiment was carried out for different epochs and the results showed that DNN based ASD gene prediction model has achieved an accuracy of 80.8% at 250 epochs. The network demonstrated an accuracy of 80.4% for the gene susceptibility identification model whereas the mutation identification model attained an accuracy of 78.1% which is fairly good. It is found that as epochs are increased from 50 to 250 the accuracy also increases to a considerable extent in all the three models.

Table XIX Epochwise Log loss of DNN Models

Epochs	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
50	2.0812	1.4294	1.9954
100	1.1155	1.7846	1.4477
150	0.9922	1.0709	0.9810
200	0.9722	0.9164	0.9438
250	0.8641	0.8415	0.8159

The experimental results elucidate that in the early epochs the log loss is almost the same and drastically comes down as epochs escalate. The log loss function for classifiers quantifies the price paid for the inaccuracy of predictions. The DNN method has the least log loss of 0.8159 at 250 epochs for ASD gene susceptibility identification model whereas it is 0.8641 and 0.8415 for gene prediction model and mutation prediction models respectively. This is attributed to the rationale that the model learns effectively by minimizing the false classifications in identifying gene susceptibility as the epochs increase. Also, the error rate of all the three models decrease over epochs as the log loss is found to be decreasing over the epochs.

Table XX Performance Results of DNN Models

Metrics	Gene Prediction Model	Mutation Prediction Model	Gene Susceptibility Prediction Model
Precision	0.79	0.77	0.79
Recall	0.81	0.81	0.80
F- Measure	0.80	0.79	0.80
Accuracy	80.8%	78.1%	80.4%
Correctly classified instances	405	390	401
Incorrectly classified instances	95	110	99
Specificity	72.1%	73.07%	74.2%

The result analysis indicates that DNN model performs fairly well for all the three models but has an upper edge for the ASD gene prediction model. It is effective in predicting the ASD genes with a precision of 0.79, recall of 0.81 and F-measure of 0.80. The DNN classifier performs well with precision of 0.77 and recall of 0.81 for mutation prediction model. The DNN based gene susceptibility prediction model has correctly identified 401 instances and 405 instances for gene prediction. When evaluating the specificity, DNN gives a prominent score value of 74.2% for identifying the gene susceptibility whereas it is 72.1% and 73.07 % for ASD gene and mutation prediction models. The following charts from Fig.6.2 to Fig.6.7 depict the experimental results with respect to various parameters.

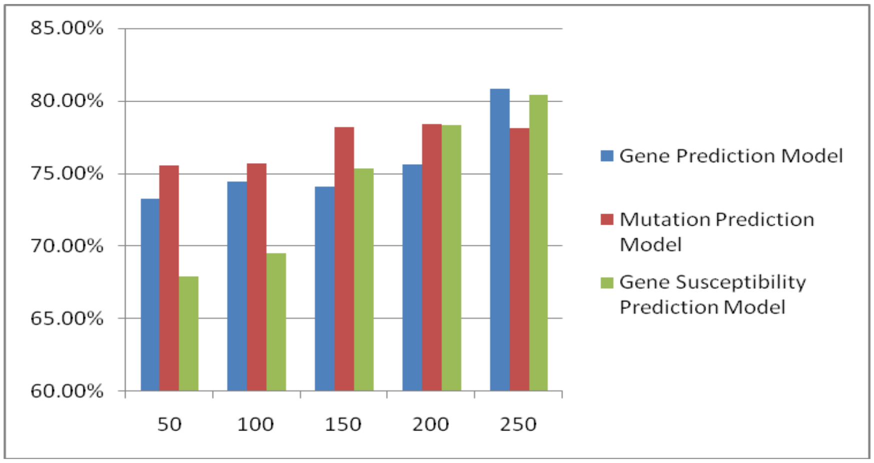

Fig. 6.2 Epochwise Accuracy of DNN Models

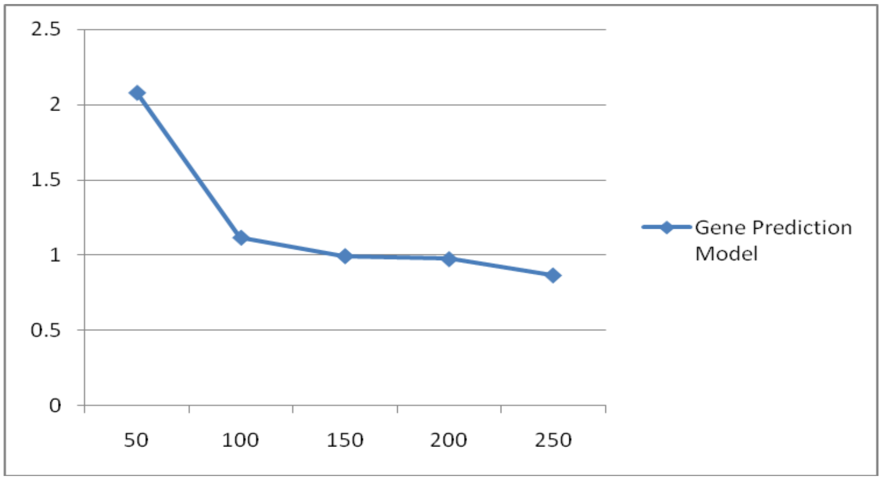

Fig. 6.3 Epochwise Log Loss of DNN Based Gene Identification Model

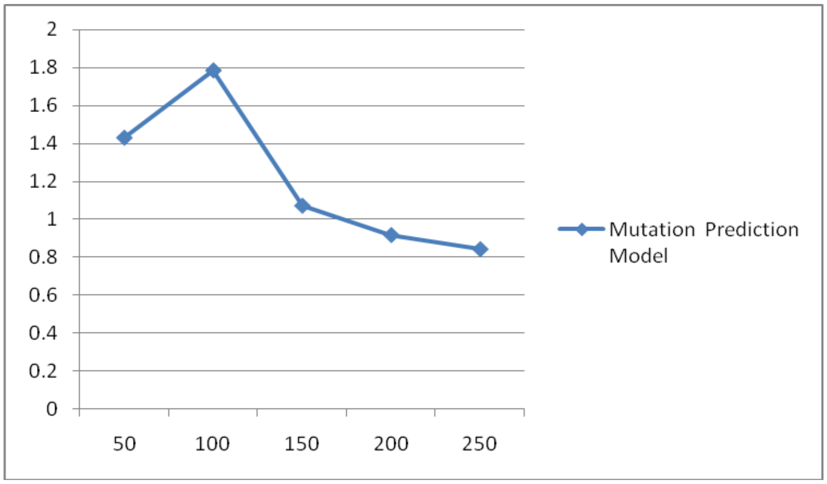

Fig. 6.4 Epochwise Log Loss of DNN Based Mutation Prediction Model

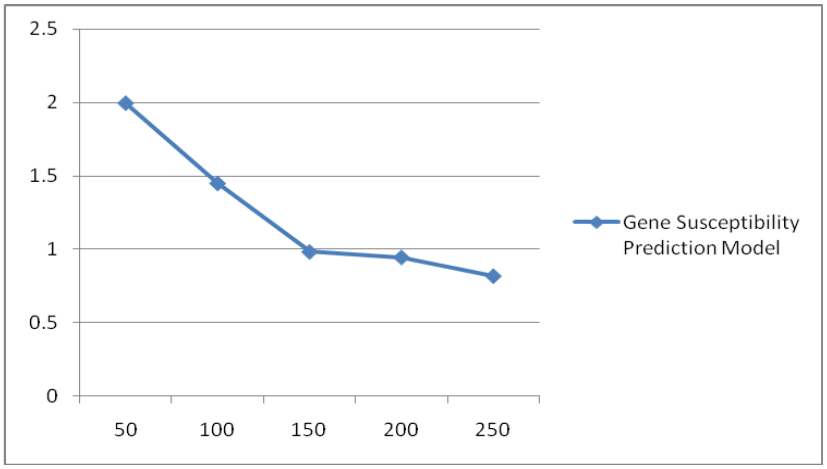

Fig. 6.5 Epochwise Log Loss of DNN Based Gene Susceptibility Prediction Model

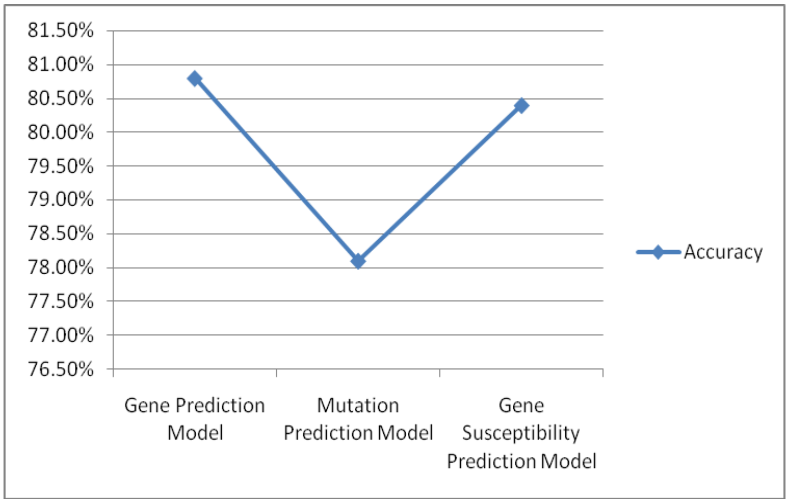

Fig. 6.6 Accuracy of DNN Models

Fig.6.2 shows that the epochwise accuracy values are higher for DNN based gene recognition model. The logloss reduces with every epoch for the gene prediction model which is illustrated in Fig.6.3. In the case of Mutation Prediction Model the logloss elevates at 100 epochs and then reduces as shown in Fig.6.4. The gene susceptibility prediction model has a logloss of 0.8159 at 250 epochs which is depicted in Fig.6.5. The accuracy of the DNN models depicted in Fig.6.6 indicates that DNN based gene prediction model performs well than the mutation classification and the gene susceptibility prediction models.

Comparison of Deep Learning with Shallow Models

The effectiveness of the DNN classifier in predicting gene type, mutation category and gene susceptibility class is compared with shallow models developed in the previous experiments described in chapter 4. The supervised machine learning models Decision tree and SVM have shown better performance in the previous experiments for identifying the ASD genes and their susceptibility respectively. The performance measures like precision, recall, accuracy and F-measures are used to compare DNN with the traditional pattern recognition techniques and the comparative performance is reported in Table XXI.

Table XXI Comparative Performance of DNN with Shallow Methods

	Gene Prediction Model		**Mutation Prediction Model**		**Gene Susceptibility Prediction Model**	
Metrics	**DNN**	**Decision tree**	**DNN**	**Decision tree**	**DNN**	**SVM**
Precision	0.79	0.72	0.77	0.75	0.80	0.79
Recall	0.81	0.75	0.81	0.78	0.81	0.80
F- Measure	0.80	0.73	0.79	0.76	0.80	0.79
Accuracy	80.8%	75%	78.1%	77%	82.4%	81.5%

The results prove that DNN based gene prediction model outperforms the shallow model of Decision tree to identify the ASD causing genes with approximately 0.07 difference in their precision. The decision tree model achieved accuracy of 75 % whereas DNN has achieved 80.8% for the same. The DNN based mutation prediction model performs comparatively better than decision tree to identify the mutations with precision of 0.77 and recall of 0.81. In the case of gene susceptibility prediction model DNN shows better performance than SVM with precision of 0.80 and recall 0.81. The comparative performance of DNN model with traditional machine learning methods for various measures of Precision, Recall, Accuracy, F-measure is depicted in Fig.6.7.

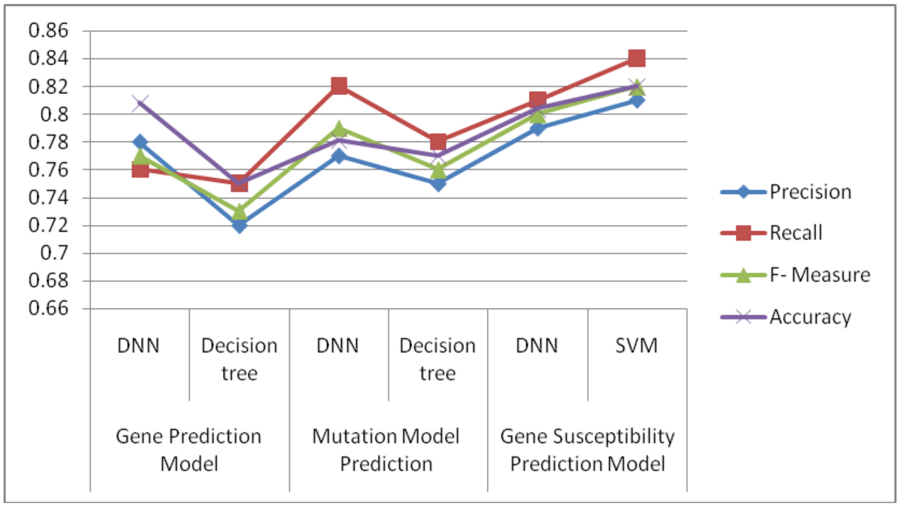

Fig. 6.7 Comparative Performance of DNN with Shallow Models

The performance evaluation of the shallow and the Deep Neural Network models based on precision, recall and F-Measure depicted in charts above show that DNN performs convincingly better than traditional pattern recognition models. The accuracy of DNN is high for predicting the gene susceptibility. The accuracy of DNN is 5.8% more than decision tree which is considerably high in predicting the ASD genes as illustrated in Fig.6.8. DNN shows high precision of 0.80, Recall of 0.81 and accuracy of 82.4% for predicting the gene susceptibility which is comparatively higher than that of SVM.

Findings

The empirical results confirm that DNN can efficiently handle the complex problem of gene, susceptibility and mutation identification for ASD. This is attributed to the fact that the network by itself has learned the intricacies of contributive features and learned representations of these features from different layers of the network. It is evident that the suitable choice of the hyperparameters has improved the prediction performance of the models.

It is apparent that DNN has an upper edge over traditional pattern recognition methods. The shallow SVM builds model by finding the optimized hyperplane which requires more computation whereas DNN is effective in learning complex, high level features and training by backpropagation without the need of substantial computations. The DNN model is proficient in learning the hidden relationships that exist in the training data but the decision tree model is incompetent in discovering the relationships between the predictors and the response variable. Also decision tree model does not perform well when the size of the dataset is small whereas DNN can learn the model with less number of instances. These factors contribute to the potential power of DNN architecture in multi- class classification.

The precision, recall and F - measure of the DNN based models is high for gene susceptibility prediction whereas it is fairly good for other two tasks which is a promising sign that the proposed model is excellent in identification of susceptibility of ASD genes. Hence it is concluded that it is more suitable for predicting ASD genes and their predisposition to ASD than conventional machine learning methods.

SUMMARY

Deep learning can open the way towards next generation of predictive health care systems that can scale to include complex data. This chapter elaborated the methodology for the development of deep learning models to classify ASD gene sequences, their predisposition to ASD and driving mutations. The models were trained and tested with three different datasets and the implementation using DNN was also demonstrated. The results of the experiments were illustrated in this chapter using tables and charts. The experimental results of DNN based predictive models were compared with shallow learning methods and the comparative analysis was presented. Various interpretations of the experimental results were also summarized here. The work is proceeded to the next level with the implementation of different architectures of deep learning namely Recurrent Neural Networks, its variants Long Short Term Memory Units and Gated Recurrent Networks which will be elaborated in the next chapter.