4. SUPERVISED LEARNING MODELS TO PREDICT ASD CAUSING GENES AND MUTATIONS

Deleterious gene identification and the underlying mutations is an important research problem in biomedical domain. It is complicated to identify a single gene causing ASD as a multitude of genes and their variants lie beneath this disorder. Hence, there is a vital need for efficient approaches to further reveal the genetic basis of ASD which will enable better filtering and specific therapies. This chapter illustrates the development of predictive models to identify causative genes and the triggering mutations causing syndromic and asyndromic ASD using pattern learning algorithms. This chapter also elaborates the multi-dimensional machine learning approach to predict the ASD candidate genes and mutations by classifying them concurrently using the discriminating features. The performance of the models are evaluated using precision, recall, accuracy, F-measure and the result analysis is also presented in this chapter.

4.1 MODEL TO PREDICT THE ASD CAUSATIVE GENES

The genetic ground of a comprehensive developmental disability like ASD is complicated to research and existing methods require further developments to augment perceptions of the genetic cause of the disorder. Development of machine learning models is crucial as they are valuable in a clinical disease risk predictive situation. This work focuses on spotting the diseased genes associated with ASD using machine learning approaches. Supervised machine learning techniques have been effectively used to resolve various important biomedical problems like inference of gene regulatory networks [43], classification of microarray data [44], prediction of drug-target and discovery of gene-gene interaction in disease data [45]. In particular, they have been applied to recognize disease linked genes. The problem of gene identification is formulated as a supervised classification problem wherein the learned classifier is built through knowledge gained from the training data and is then used to forecast the type of gene causing ASD.

Methodology

In this work an ASD causative gene discriminative model is constructed by training the system with genetic blueprints from a labeled set of instances that will offer accurate predictions in unseen cases with similar genetic conditions. The task of identifying ASD gene sequences is modeled as a multi- class classification problem. Disease gene sequences are simulated and used in this multi-class classification problem. The coding regions of diseased gene sequences are utilized as features to train the model. This work utilizes supervised learning techniques such as Decision Tree, Multi Layer Perceptron and Support Vector Machines to build models that predict genes causing ASD. The model includes three components namely dataset creation, model building and performance evaluation of gene classification models and the architecture is depicted in Fig.4.1.

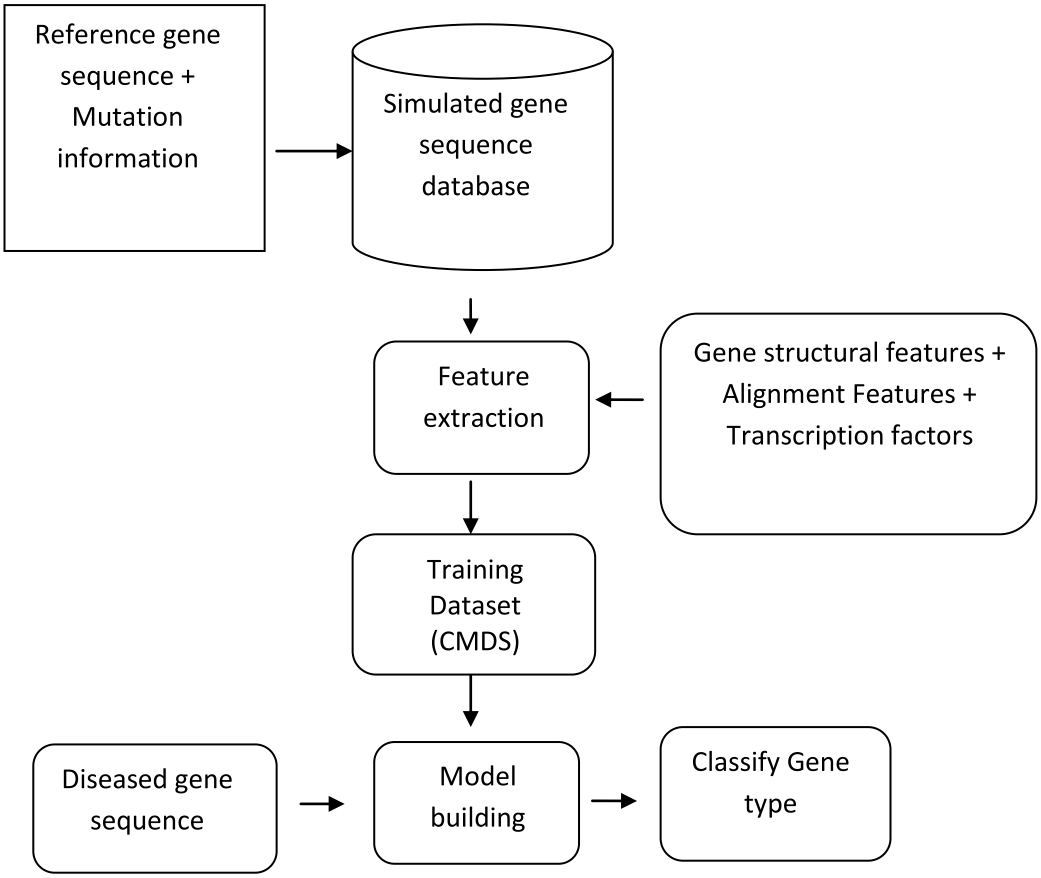
Reference gene sequence + Mutation information
Simulated gene sequence database
Feature extraction
Gene structural features + Alignment Features + Transcription factors
Training Dataset (CMDS)
Diseased gene sequence
Model building
Classify Gene type

Fig. 4.1 Proposed Framework to Identify ASD Causing Genes

The first component deals with the creation of corpus and establishment of the dataset. Ten genes namely FMR1, MECP2, TSC1, CACNA1C, SHANK3, CHD8, FOXP2, CNTNAP2, GABRB3 and HOXA1 have been considered. Four types of mutations namely missense, nonsense, synonymous and frameshift have been considered for generating mutated sequences.

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 corpus is developed as described in Chapter 3 with 1000 mutated gene sequences accounting for ten types of ASD genes and four types of mutations.

The coding measures are dissimilar in different gene families and hence this trait is a well-chosen descriptor for specifying different gene families. To facilitate learning the gene patterns, various features such as nucleotide composition, GC content, Rho values of biwords, Z scores of biwords, Alignment score, Number of exons, Number of donor sites, Number of acceptor sites, CpG percent, ratio of CpG percent / expected are identified and described below.

***Nucleotide composition*:** The number of occurrence of individual nucleotides exhibit noteworthy variations in eukaryotic genes. Each nucleotide in DNA is formed by a nucleobase, a deoxyribose sugar and a phosphate group. The nucleotide bases are generally composed of Adenine, Guanine, Cytosine and Thymine. The nucleotides are bound in a strand by a covalent bond between a sugar of one nucleotide with phosphate group of the next nucleotide. The nucleotide in one strand is attached to a nucleotide from another strand by a hydrogen band to make a double strand DNA. The A, C, G, T nucleotide variations are extracted as features since such dissimilarity result from differential mutational pressures and from the incidence of specific regulatory motifs like transcription sites.

GC Content: GC content is an essential property of a genome sequence, which indicates the portion of the sequence which contains Gs and Cs. The GC content of most species does tend to stay close to 50%. Coding regions of the genome that hold a higher percentage of guanine and cytosine are called GC-rich and those areas of GC content less than 50% are called GC-poor. Thus, just like GC content between species can be used to identify species, GC content of a snippet of DNA from a known species when tested can discriminate if that DNA may belong to a gene. GC content is calculated as given in equation 4.1.

$$\text{GC content} = (\text{count of Gs} + \text{count of Cs})*100 / (\text{genome length}) \qquad (4.1)$$

The variations in GC content inside the genome sequence can provide motivating information like biases in mutation. Hence GC content is considered as an important feature in identifying the gene type.

Rho and Z – scores: The work also investigates DNA words that are two nucleotides long and are over-represented or under-represented. If a DNA word is over-represented in a sequence, probably it occurs many more times in the sequence than expected whereas when it is under-represented in a sequence, it is present less number of times in the sequence than expected. Statistical measures Rho and Z- scores are used to measure over-representation or under-representation of a particular DNA word which also contributes in classifying the genes. For a DNA word that is two nucleotides long, Rho and Z-score is calculated using equations 4.2 and 4.3.

$$\text{Rho}(xy) = f(xy) / (fx*fy) \tag{4.2}$$

$$\text{Z-score}= \mu(xy) / \sigma (xy) \tag{4.3}$$

The Z - score of biwords is computed by finding the difference between the mean divided by the standard deviation. For a single gene sequence, 16 Rho values and 16 Z - score values are obtained.

Alignment score: One of the non-consensus properties, alignment score is used to compare the simulated gene sequences with a library of sequences and to spot library sequences that is similar to the query sequence. BLAST (Basic local alignment search tool) algorithm is used to match the biological sequence information, like the amino-acid sequences of proteins or the nucleotides of DNA or RNA sequences [83]. BLAST search is used to compare a query sequence with a library or database of sequences, and discover library sequences that are similar above a certain threshold. The similarity score from BLAST alignment is computed and utilized for building the model.

Exon Count: Exons encode for the amino acid sequence for protein and actually are the coding element of the nucleotide sequence. After post-transcriptional alteration, exons are transcribed and transformed into mature mRNA. They are translated into proteins in the cytoplasm and are the extremely conserved sequence. Their presence in DNA and mature mRNA is well marked. Thus the number of expressed sequences, exons is also employed as descriptors.

Donor site and Acceptor site: The mature mRNA contains only coding sequences and the intronic ones are removed from the transcript during the splicing process. Splicing requires a donor site and an acceptor site within introns. Donor site is the splicing site at the beginning of an intron 5' left end. Acceptor site is the splicing site at the end of an intron 3' right end. Thus, the biological process of removing introns from its 5′ splice site to its 3′ splice site in pre-mRNA and connecting exons to form mRNA plays an important role in gene regulation and expression. Hence the number of donors and acceptors are recognized as essential discriminators.

CpG Island: The incidence of a CpG island is used to help in the prediction and annotation of genes. The CpG sites are DNA portions where a guanine nucleotide occurs after a cytosine nucleotide in the linear sequence of bases along its 5' → 3' direction. The observed-to-expected CpG ratio is computed using equations 4.4 and 4.5.

Observed CpG = No. of CpG (4.4)

Expected CpG = No. of C * No. of G / Length of sequence (4.5)

The summary of the above features are depicted below.

Features	**Count**	**Features**	**Count**
Nucleotide composition	4	Number of donor sites	1
GC content	1	Number of acceptor sites	1
Rho values of biwords	16	CpG percent	1
Z scores of biwords	16	Ratio of CpG percent / expected	1
Alignment score	1	Number of exons	1

Thus a total of 43 features are extracted from each mutated sequence. For the sample mutated sequence given in Fig. 3.4, the features obtained are given below.

1256	2521	2211	1125	0.67	1.04	0.84	1.27	0.78	1.14	1.11
0.68	1.21	1.06	1	1.07	0.79	0.51	0.92	1.28	1.17	0.75
-4.75	7.17	-3.72	4.16	5.27	-13.4	5.8	1.5	0.19	2.54	-5.1
-8.32	-2.15	6.91	2.77	13130	2	15	13	0.149	1.003	

As ten genes are considered for building the ASD causative gene identification model, the class labels are designated as 1 to 10 for all the respective instances. The feature values are all normalized using min-max normalization and finally the dataset with 1000 feature vectors of dimension 43 is developed and named as Codon Measures dataset (CMDS). The sample dataset is shown in Appendix A.

In the second phase of this methodology, three independent gene identification models are built by training the normalized CMDS dataset using pattern recognition algorithms namely Decision Tree, Multi Layer Perceptron and Support Vector Machines.

Finally, 10 - fold cross-validation technique is used to evaluate the performances of the three models using various metrics such as precision, recall, F- measure, accuracy, specificity and ROC area.

Experiment and Results

Experiments have been carried out by implementing standard supervised learning techniques namely Decision tree induction, Multilayer Perceptron and Support Vector Machine (SVM) algorithms with codon measures dataset (CMDS) using the Scikit learn tool. Scikit-learn is an open source machine learning library in Python. It contains a lot of efficient tools for machine learning and statistical modeling. It features various classification, regression and clustering algorithms and is built on top of NumPy, SciPy and Matplotlib libraries. The standard 10- fold cross-validation technique is used to estimate the impact on the predictive performance for unknown samples. The results obtained from the learned classifiers are analyzed through performance measures namely precision, recall, F- measure, accuracy, specificity and ROC area. The results of various measures are tabulated in Table VII.

Table VII Performance Results of ASD Gene Classifiers

Classifier	Multilayer Perceptron	Support Vector Machines	Decision Tree
Precision	0.65	0.68	0.72
Recall	0.72	0.70	0.75
F Measure	0.66	0.69	0.73
Accuracy	68%	72%	75%
Kappa statistic	0.675	0.714	0.767
Mean absolute error	0.4176	0.3233	0.2812
Correctly classified instances	341	362	376
Specificity	0.78	0.80	0.82
Mathew correlation coefficient	0.76	0.75	0.87
ROC Area	0.62	0.71	0.76

The results indicate that decision trees fares well when compared with other techniques. The highest precision and recall of 0.72 and 0.75 respectively was achieved by decision tree classifier. Decision tree has correctly classified 376 instances and its accuracy is 75% whereas MLP and SVM have an accuracy of 68% and 72% respectively. The mean absolute error of Decision tree is 0.2812 which is least when compared to other techniques. SVM attained the mean absolute error of 0.3233 whereas it is 0.4176 for MLP. Kappa statistics of the three classifiers MLP, SVM and Decision tree are 0.675, 0.714 and 0.767 respectively. When evaluating specificity, decision tree gives a prominent score value of 0.82 whereas SVM and MLP have 0.80 and 0.78. Matthews Correlation coefficient of Decision tree is 0.87 and its ROC area is 0.76 which are comparatively higher than the other two classifiers. The performance results of the ASD causative gene identification models with respect to various metrics are depicted from Fig.4.2 to Fig.4.6.

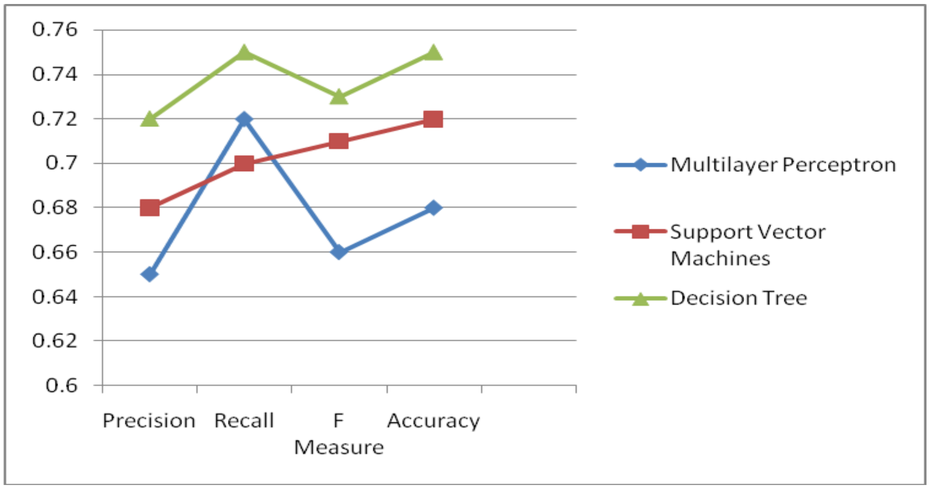

Fig. 4.2 Precision, Recall, F Measure, Accuracy of ASD Gene Classifiers

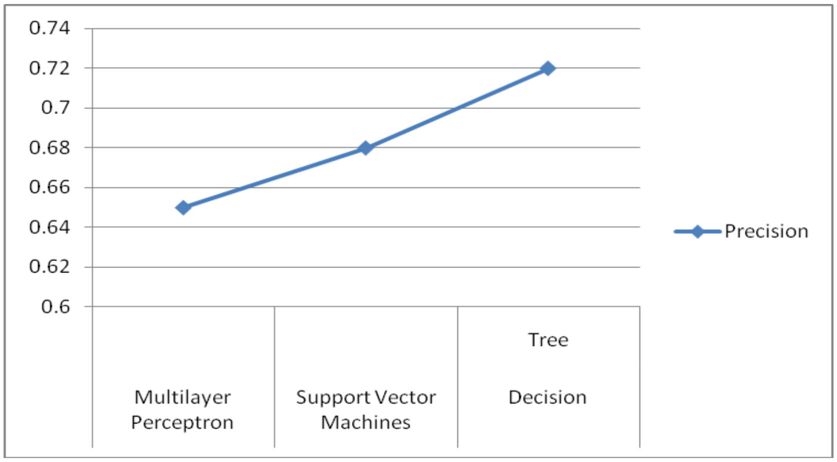

Fig. 4.3 Precision of ASD Gene Classifiers

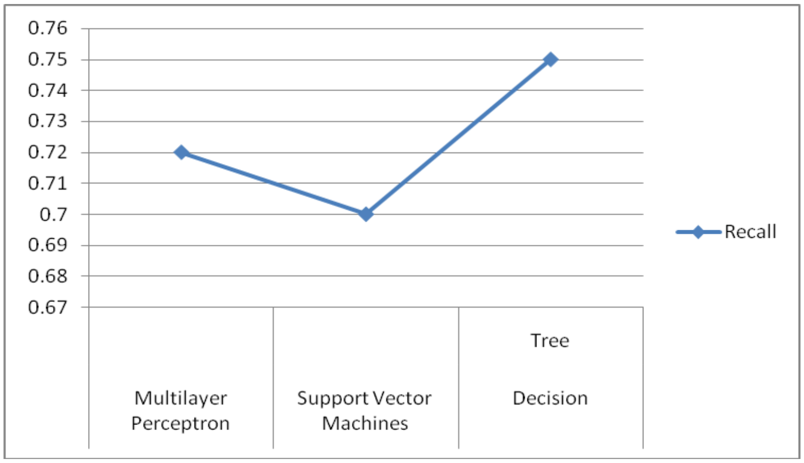

Fig. 4.4 Recall of ASD Gene Classifiers

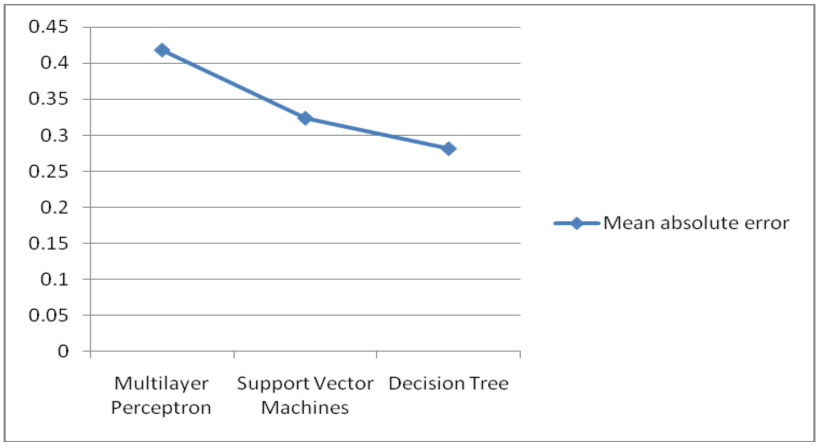

Fig. 4.5 Mean Absolute Error of ASD Gene Classifiers

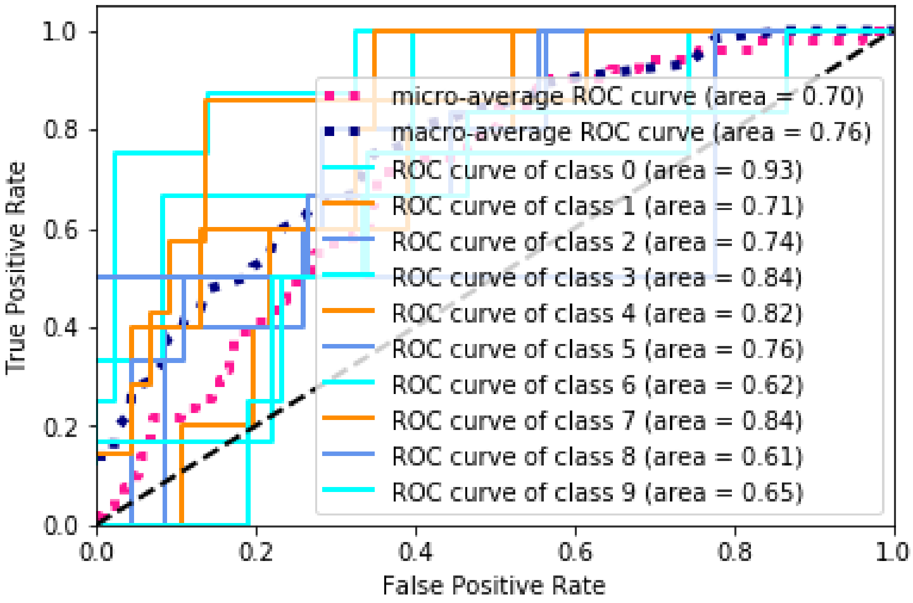

Fig. 4.6 ROC of Decision Tree Classifier

The comparative performance of the classifiers illustrated in Fig.4.2 shows that the decision tree outperforms the other models. The curve of precision as shown in Fig.4.3 is steep for decision tree whereas it is declining for MLP and SVM. The recall of decision tree classifier is 0.75 which is 0.05 higher than that of Multilayer Perceptron as observed in Fig.4.4. Fig.4.5 depicts that the error associated with decision tree classifier is comparatively less than the other two classifiers. ROC curve depicted in Fig.4.6 shows that class 0 has a high area under ROC curve of 0.93 whereas class 8 has it low with 0.61. The macro - average ROC curve area is 0.76 whereas the micro – average area is 0.7.

Findings

The comparative results point out that decision tree based classification model shows better performance when compared to other models and is more appropriate for classifying the ASD causing genes. The features extracted from the gene sequences are highly contributive in discriminating the ASD genes. Given a diseased gene sequence the decision tree model is able to identify one of the ten gene types with high accuracy and precision. This model is capable of identifying the significant features and the pattern of relationships existing in the gene sequences.

The error associated with prediction is much less for the decision tree model and hence it is reliable in a clinical risk prediction environment.

4.2 MODEL TO PREDICT ASD CAUSING MUTATIONS

Mutations are the key molecular players in the cause of ASD and it is essential for developing effective therapeutic strategies that target these mutations. The development of computational tools to recognize ASD causing genetic mutations is vital to help the progress of disease-specific targeted therapies. Mutations cause changes in the genetic code leading to the disorder and therefore it is proposed to build an accurate model to predict the type of mutations by capturing the structural changes and training the model using these mutational features. It is intended to employ supervised machine learning techniques for constructing the ASD triggering mutation recognition model.

Methodology

It is rather a complex task of classifying mutations in ASD genes taking into account the genetic variations due to synonymous and non-synonymous nucleotide polymorphisms underlying the autistic phenotypes. In this work supervised machine learning techniques are employed to learn feature representations, model their sequential dependencies and finally distinguish the triggering mutations. Simulated disease gene sequences are used in this multi-class pattern classification problem. The proposed model consists of the three phases namely corpus and dataset creation, model building, performance evaluation of the model and the architecture is depicted in Fig.4.6.

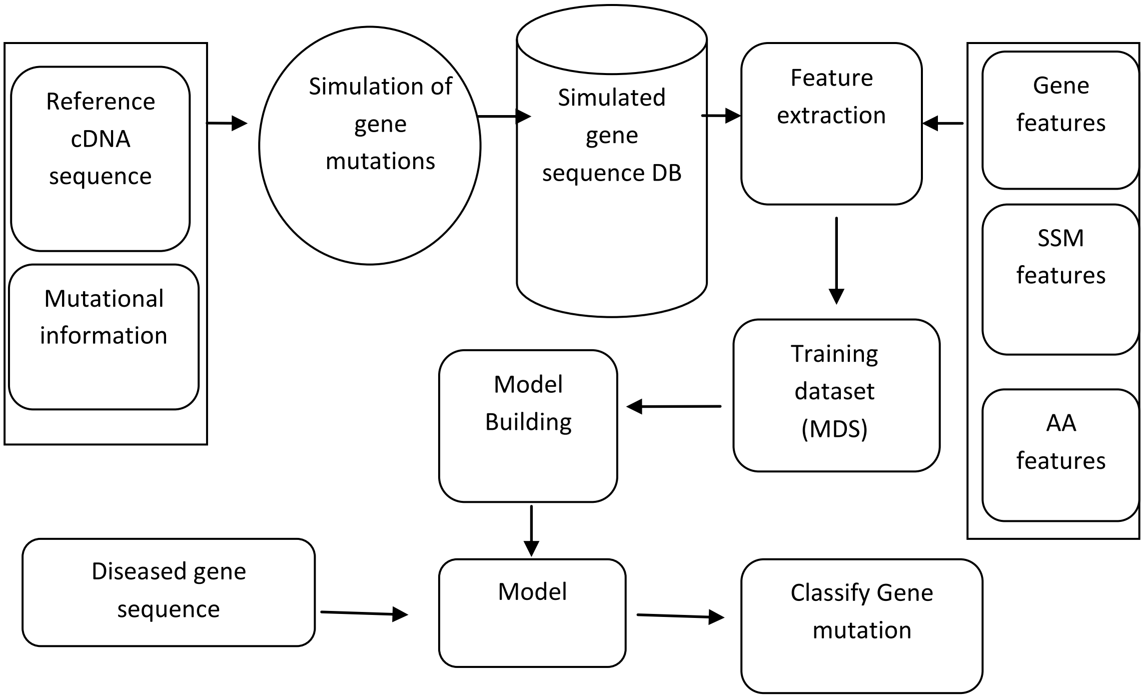

Fig. 4.7 Architecture of ASD Causing Mutation Prediction Model

In the first phase 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 corpus is developed using 1000 mutated gene sequences accounting for ten types of ASD and the four types of mutations, missense, nonsense, synonymous and frameshift mutations as described in Chapter 3.

Various descriptors that recognizes the pattern of mutations in gene sequences with respect to different aspects including 5 gene specific, 6 features from published substitution scoring matrices (SSM), 4 features related to amino acid (AA) residue changes are incorporated here to facilitate learning the mutation prediction model. These attributes are described below.

Gene Specific Features: The features like mutation start position, mutation end position, length of mutation, length of cDNA sequence, the type of mutational variation i,e Nonsense, Missense, Frameshift, Silent Mutations are characteristics extracted from a gene. Consider the SHANK3 gene sequence

ATGGACGGCCCCGGGGCCAGCGCCGTGG…………….GCGGCAG**C**

ATGGACGGCCCCGGGGCCAGCGCCGTGG…………….GCGGCAG**A**

The gene specific features for the above mentioned gene sequence will be Mutation start position -612, mutation end -612, mutation lengrh-1,length of CDNA sequence-7113, mutation type-2.

Substitution Matrix Features: In a sequence alignment scoring matrices are used to decide the relative score made by matching two characters. They are computed as the log-odds of the probability of two characters that are derivatives of a common ancestral character. Many types of scoring matrices exist for nucleotide sequences, codon sequences and amino acid sequences derived by aligning the known homologous sequences [84]. These alignments are then used to determine the likelihood of one character being at the same position in the sequence as another character.

This work utilizes the values of 6 scoring matrices namely (i) WAC matrix constructed from amino acid comparative profiles, (ii) Log-odds scoring matrix collected in 6.4-8.7 PAM , (iii) BLOSUM80 substitution matrix, (iv) PAM-120 matrix, (v) Substitution matrix obtained by maximum likelihood estimation and (vi) Mutation matrix for initially aligning which are collected from the AAIndex database.

(i) WAC matrix is computed entirely from differences in the observed average environment surrounding amino acids without respect to any multiple alignments. A mutation in which an amino acid is replaced by one with similar physio - chemical properties is more likely to be accepted than one in which the new environment disrupts the protein's conformation. The amino acid micro environment data is used to construct the WAC amino acid similarity matrix.

(ii) Log odd scoring matrix express the probabilities of transformation in what are called log-odds scores. The scores matrix S is defined as given in equation 4.6.

$$S = \log (p_i.\ M_{i,j} / p_i.\ p_j) = \log(M_{i,j} / p_j) = \log (\text{observed frequency / expected frequemcy}) \quad (4.6)$$

where $M_{i,j}$ is the probability that amino acid i transforms into amino acid j, and p_i. p_j are the frequencies of amino acids *i* and *j*.

(iii) BLOSUM (Block Substitution Matrix Block) is a small contiguous interval of multiple aligned sequences. The blocks of conserved sequences found in multiple protein alignments are considered when computing the probabilities used in the matrix calculation. As these conserved sequences are functionally important within related proteins, they are assigned lower substitution rate than less conserved regions. Clustering of segments in a block with a sequence identity above a certain threshold is done to reduce bias from highly related sequences on substitution rates [85]. The threshold was set at 62% for the BLOSUM62 matrix.

(iv) PAM matrices are a common family of score matrices. PAM (Point Accepted Mutation) matrix was developed in the 1970s. PAM units are used to evaluate the amount of evolutionary distance between any two amino acid sequences. Two sequences S1 and S2 are said to be one PAM unit diverged if a series of accepted point mutations has converted S1 to S2 with an average of one accepted point-mutation event per 100 amino acids.PAM matrices contain positive and negative values. If the alignment score is greater than zero, the sequences are considered to be related and if the score is negative, it is assumed that they are not related. Values from PAM 120 matrix is taken in this study.

(v) Substitution matrix obtained by maximum likelihood estimation is given below. VTML matrices are built by iteratively calculating evolutionary distances and substitution rates from a set of pairwise sequence alignments using a maximum likelihood estimator. They offer a more reliable detection of remote homologs. The substitution matrix feature for the protein alteration Asp to Glu represented by letters D and E will be 3.

Substitution matrix (VTML160) obtained by maximum likelihood estimation
M rows = ARNDCQEGHILKMFPSTWYV, cols = ARNDCQEGHILKMFPSTWYV

```
  5
 -2   7
 -1   0   7
- 1  -3   3   7
  1  -3  -3  -5  13
 -1   2   0   1  -4   6
 -1  -1   0   3  -5   2   6
  0  -3   0  -1  -2  -3  -2   8
 -2   1   1   0  -2   2  -1  -3   9
 -1  -4  -4  -6  -1  -4  -5  -7  -4   6
 -2  -3  -4  -6  -4  -2  -4  -6  -3   3   6
 -1   4   0   0  -4   2   1  -2   0  -4  -3   5
 -1  -2  -3  -5  -1  -1  -3  -5  -3   2   4  -2   8
 -3  -5  -5  -7  -4  -4  -6  -6   0   0   2  -5   1   9
  0  -2  -2  -1  -3  -1  -1  -3  -2  -4  -3  -1  -4  -5   9
  1  -1   1   0   1   0   0   0  -1  -3  -3  -1  -3  -3   0   4
  1  -1   0  -1   0  -1  -1  -2  -1  -1  -2  -1  -1  -3  -1   2   5
 -5  -4  -5  -7  -7  -6  -7  -5  -1  -2  -1  -5  -4   3  -5  -4  -6  16
 -3  -3  -2  -5  -1  -4  -3  -5   3  -2  -1  -3  -2   6  -6  -2  -3   4  10
  0  -4  -4  -4   1  -3  -3  -5  -3   4   2  -3   1  -1  -3  -2   0  -5  -3   5
```

(vi) Mutation Matrix : An amino acid mutation matrix is generally 20 × 20 numerical values representing similarity of amino acids numerical values, used for sequence alignments and similarity searches. An important feature of amino acids that can be

Amino Acid Features: The mutated sequences are translated to generate protein sequences which in turn provide the amino acid observed values whereas the amino acid expected value is extracted from SFARI autism database. The 2-gram encoding method extracts different patterns of two consecutive amino acid residues in a protein sequence and count the number of occurrences of the extracted residue pairs. Wang et al. has shown a good enough performance by using 2-gram features alone in a similar research. There are 20^2 combinations of 2-grams which is huge and hence the standard deviation and the mean z-score between the values of the

400 bigrams with respect to the protein sequence are calculated using the formulae in 4.7 and 4.8.

$$\text{Standard deviation}= \surd\sum(i=1)\ \hat{}\ 400\ (x-\mu) / (n-1) \qquad (4.7)$$

$$\text{Mean Z-score} = \surd\sum(i=1)\ \hat{}\ 400\ (\ (x-\mu)\ /\sigma) / n \qquad (4.8)$$

where μ is the mean value of the occurrence of the 2-gram , $1<i<400$, in the dataset and σ is its standard deviation.

The summary of MDS features and their count are depicted below.

Features	**Count**	**Features**	**Count**
Mutation start position	1	Mutation end	1
Mutation length	1	Length of sequence	1
Mutation type	1	WAC matrix	1
Log-odd scoring matrix	1	BLOSUM80 substitution matrix	1
PAM-120 matrix	1	Substitution matrix (VTML160)	1
Mutation matrix	1	Standard deviation of bigrams	1
Mean z-score of bigrams	1	Amino acid observed, expected value	2

Thus a total of 15 features are extracted from each mutated sequence. For the sample mutated sequence given in Fig.3.4, the features obtained are given below.

898	898	1	7113	0	-1	-6	-0.4	-3	-4	-2.2	2
5	7.23	-2.76									

As four mutations are considered for building the ASD causative mutation identification model, the class labels are designated as 1 to 4 for all the respective feature vectors. Min-max normalization is used to normalize the feature values and finally the dataset is developed with 1000 feature vectors of dimension 15 and named as Mutation dataset (MDS). The sample dataset is shown in Appendix A.

The second phase involves the construction of three independent mutation identification models built by training the normalized MDS dataset using supervised machine learning algorithms namely Decision tree, Support Vector Machine and Multilayer Perceptron. The model is able to learn associations between consecutive signals and identify any type of regularity in the input.

In the concluding phase, 10 - fold cross-validation technique is applied and the predictive performance of the three models are evaluated using various metrics such as precision, recall, F - measure, accuracy, specificity and ROC area.

Experiment and Results

In this experiment, the training dataset MDS comprising of 1000 instances of ten types of ASD genes involving four types of genetic mutations has been used to build the classifiers. The standard supervised classification algorithms namely Decision tree induction, Multilayer Perceptron, SVM were used to build the models using Scikit learn. The validation technique namely 10 - fold cross-validation was used to estimate their predictive performance. The results obtained from the classifiers were analysed through precision, recall, F- measure, accuracy, specificity and ROC area which is tabulated in Table VIII and Table IX.

Table VIII Performance Results of the Mutation Classifiers

Classifier	Precision	Recall	F-Measure	Accuracy	Specificity	ROC area
SVM	0.72	0.73	0.73	0.73	0.78	0.79
MLP	0.65	0.74	0.68	0.69	0.75	0.64
Decision Tree	0.75	0.78	0.76	0.77	0.85	0.82

Table IX Classwise Performance Results of Decision Tree Classifier

Statistics by Class:	Class: 1	Class: 2	Class: 3	Class: 4
Sensitivity	0.7600	0.7143	0.8444	0.8371
Specificity	0.8711	0.8288	0.8412	0.8530
Positive Predictive Value	0.7104	0.7712	0.7535	0.8020
Negative Predictive Value	0.8800	0.7977	0.7800	0.8836

The comparative analysis shows that decision tree achieves high accuracy of 0.77 than the other classifiers SVM and MLP with 0.73 and 0.69 respectively. The precision and recall values of decision tree are 0.75 and 0.78 clearly outperforming the other two classifiers. Receiver Operating Characteristic (ROC) curve can be used to estimate classifier performance. The ROC area of decision tree is 0.82 which is comparatively higher than MLP with 0.64 and SVM with

0.79. The classwise performance analysis of decision tree classifier is depicted in Table XIII. The sensitivity value for Class 3 is 0.8444 which shows that most of the relevant instances of this class have been retrieved over total relevant instances. The specificity is high for classes 1 and 4. The performance analysis of the models with respect to various measures is illustrated in Fig.4.8 to Fig.4.11.

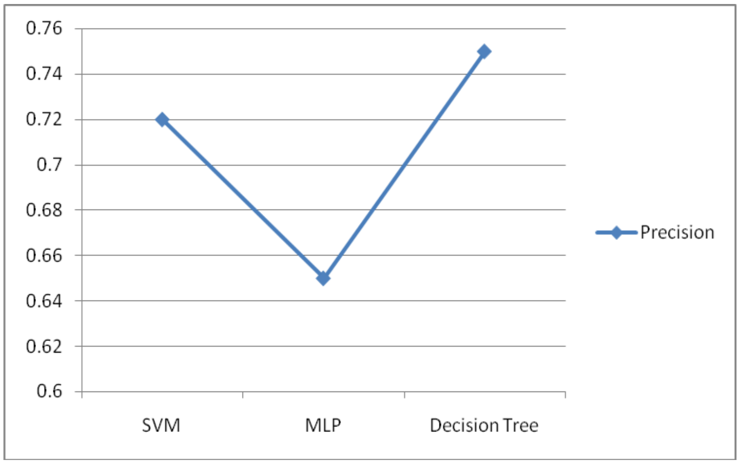

Fig. 4.8 Precision of Mutation Classifiers

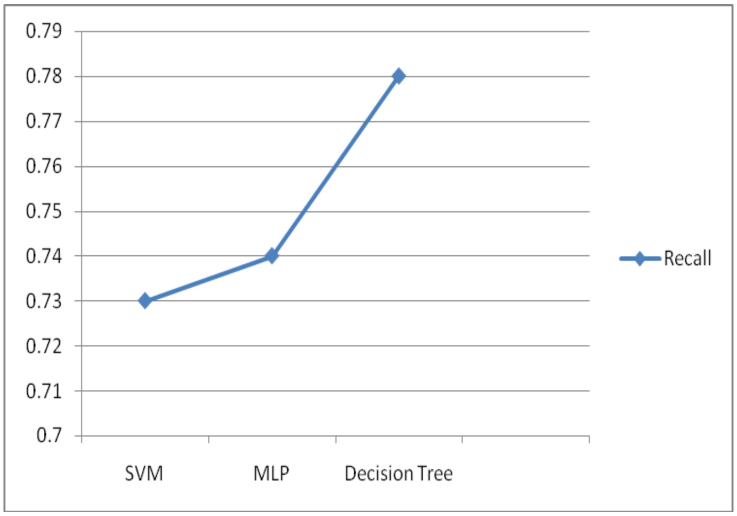

Fig. 4.9 Recall of Mutation Classifiers

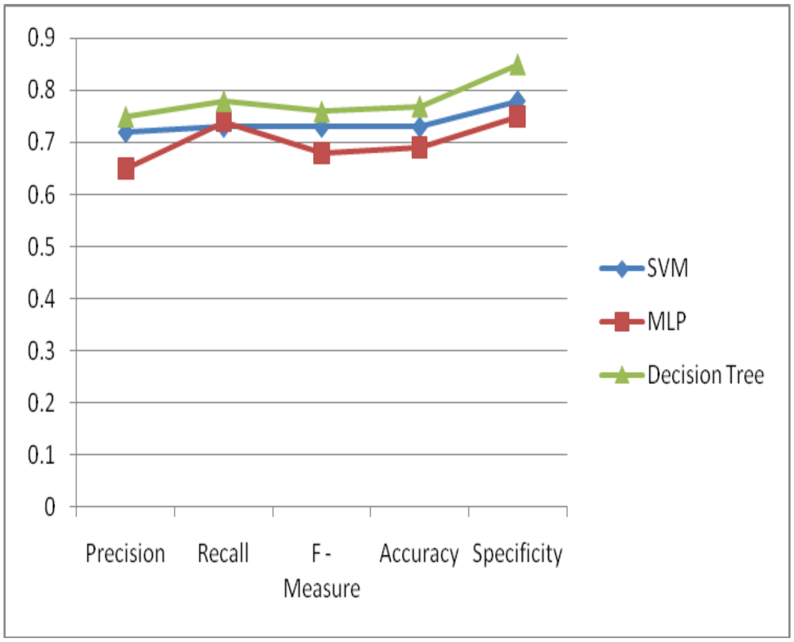

Fig. 4.10 Performance Comparison of Mutation Classifiers

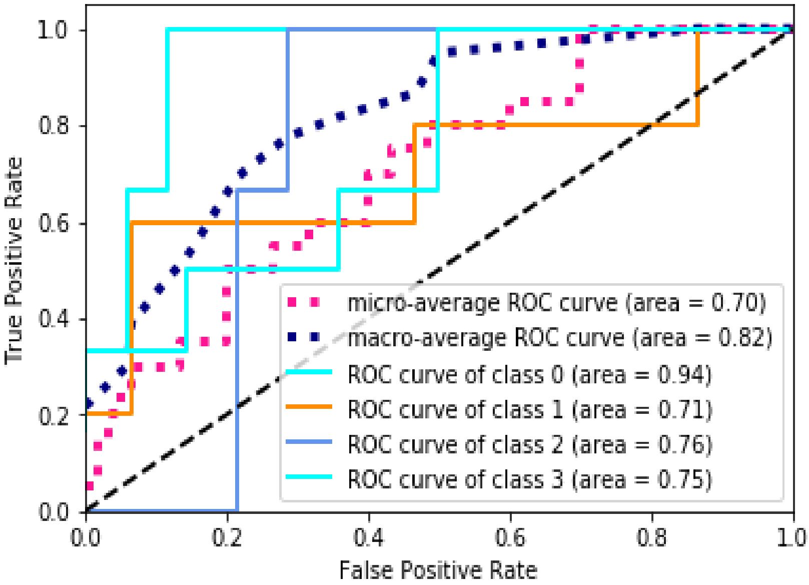

Fig. 4.11 ROC Curve of Decision Tree Classifier

As Fig.4.8 depicts, decision tree model achieves high precision while MLP has a dip in its precision value. Fig.4.9 clearly portrays that, SVM and MLP classifiers have almost equal recall of 0.73 and 0.74 but decision tree has a better value of 0.78. The comparative performance of classifiers depicted in Fig.4.10 shows that decision tree clearly outperforms other two models in terms of precision, recall, accuracy, F-measure and specificity. The SVM model competes well with the decision tree but has less specificity and accuracy values. ROC curve depicted in Fig.4.11 shows that class 0 has a high area under ROC curve of 0.94 whereas class 1 has it low with 0.71. The macro - average ROC curve area is 0.82 whereas the micro – average area is 0.7.

Findings

The idea of combining gene specific features and amino acid changes along with substitution matrix features designed for building the classifier is found to be decisive in identifying the mutations. The comparative performance analysis shows that the decision tree classifier with MDS dataset yields high accuracy in distinguishing the mutations. Decision tree has an upper edge over the other pattern recognition methods of SVM and MLP. The SVM model finds the optimized hyperplane involving substantial computations The decision tree model performs implicit feature selection and is able to identify the contributive features by pruning unwanted features without the need of much computations.

4.3 MULTI - DIMENSIONAL MODEL TO PREDICT ASD GENES AND MUTATIONS

Multi-dimensional approach has its advantages with regard to predictive accuracy and time taken to build the model. The search for candidate genes of ASD is complicated as it involves significant interactions among mutations in several genes. Hence this work is aimed at employing multi-dimensional machine learning approach to predict the ASD candidate genes and mutations by classifying them concurrently based on the contributing features. The insightful association between genes and mutations is modeled as a multi-label problem since the dependencies between them can be captured.

Multidimensional Classification

In machine learning, single label classification is a learning problem where the goal is to learn from a set of instances, each associated with a unique class label from a set of disjoint class labels L. The problem can be identified as binary classification (when $L_j = 2$) or multi-class classification (when $L_j > 2$) problem based on the total number of disjoint classes in L. A multi-label classification is the learning task in which the instances are associated with more than one class. The objective in multi-label classification is to learn from a set of instances where each instance belongs to one or more classes in L. Multi dimensional classification (MDC) or multi-target classification is a generalization of Multi-label classification. The applications of Multi dimensional classification are varied in real-world domains like medical diagnosis, bioinformatics, robotics, text, image and audio processing.

Conventional binary and multi-class problems both can be considered as specific cases of multi-label problem. But the generality of multi-label problems makes it more difficult than the others. MDC is concerned with learning from examples, where each data instance is associated with multiple target variables, and each variable takes multiple values. In MDC, each class variable $|Y_j| = K_j$, where j = 1, 2……d and d is the number of class variables, K is the number of values each of these variables may take.

Multi-label classification algorithms are often grouped into two categories namely problem transformation and algorithm adaptation. Problem transformation transposes the problem to multiple single-label classification tasks and by training a set of any single-label classifiers the problem can be solved. Some of the problem transformation methods are binary relevance, label powerset, Rakel, pairwise methods etc. Algorithm adaptation is based on single-label classification approaches that are adapted to handle multi- label data directly and thereby address the full problem. Boosting, Decision trees, k-NN, Bayesian, Probabilistic Classifier Chain are some of the Algorithm adaptation methods. Multi-label classification approach solves classification problems in which instances described by a number of features are assigned to multiple classes simultaneously. The three problem transformation techniques Bayesian classifier chains (BCC), Nearest Set Replacement (NSR), class relevance (CR) and algorithm adaptation method Ensemble of classifier chains (ECC) incorporated in multi label modeling are stated below.

Bayesian classifier chains (BCC)

Given a multidimensional classification problem with *d* classes, a Bayesian Chain Classifier (BCC) uses *d* classifiers, one per class, linked in a chain. The objective of this problem can be posed as finding a joint distribution of the classes C = $(C_1, C_2, \dots, C_d)$ given the attributes x = $(x_1, x_2, \dots, x_l)$ as given in equation 4.9.

$$P(C \mid x) \quad = \quad \Pi\, P(C_i \mid pa(C_i), x) \tag{4.9}$$

where pa(C_i) represents the parents of class C_i and i=1 to d.

Nearest Set Replacement (NSR)

NSR is a multi-target version of Pruned Set. The nearest sets are used to replace outliers, rather than subsets. Pruned set is a multi-label method that treats each label set as a single-label, prunes away the infrequent sets and decomposes these sets into frequent sets. NSR reduces the number of values associated with each class in the training data.

Class relevance (CR)

It is also known as one versus rest and a straightforward problem transformation method. CR decomposes the problem to a set of q classification tasks, that is one for each different label in L. This is done by transforming the original training set into *q* data sets where each set contains all the instances from the original set, labeled according to the class based on whether the label is set or not. When classifying a new instance CR will output the union of the labels that are predicted by the q classifiers.

Ensemble of classifier chains (ECC)

ECC combines output vectors from each classifier by selecting the output value for each output that maximizes the scores given by each model to each possible value for that output. It takes votes using the confidence outputs of the base classifier. It maximizes each output separately, so the resulting output vector may not have been predicted by any sub-model. It also employs bagging with every sub-model in the ensemble.

Few research works that have been carried out by researchers using multi dimensional classification approach are outlined below. Blessing Ojme[86] reported the findings of a study based on simultaneous identification of depression and physical illness using multidimensional Bayesian approach. Johan Brodin[87] has evaluated 17 configurations of different multi-label classifiers for classifying a music track into a set of core values. Results showed that problem transformation algorithm Label Powerset together with Sequential minimal optimization outperformed the other configurations. Julia Zaragoza [88] initiated a method for chaining binary Bayesian classifiers that united the strengths of classifier chains and Bayesian networks for multi target classification. A chain of naive Bayes classifiers were tested on different benchmark multidimensional datasets and the new approach outperformed other state-of-the-art methods. Several researchers [89 -92] have worked on multi-dimensional Bayesian classifiers and also on choosing the efficient classifier for multiple labels [93 – 94]. An ensemble of pruned sets for multi-dimensional classification was attempted by J. Read [95]. The above research works motivate that the recognition of candidate ASD genes co-existing with mutations which can be carried out using multi dimensional approach.

Methodology

In a genetic disorder like ASD, mutations completely disable genes crucial to early brain development. Mutations causes change in the genetic code leading to genetic variation and the potential to develop a disease. Identifying a gene and mutation can be modeled as two different problems but the dependencies between them will be ignored. As there may be mutations involved in different sets of genes in different autistic individuals, it is essential to model the dependencies among them in order to learn what is probable. In this context MDC is appropriate as its primary task is to model the dependencies between multiple classes and to tackle the computational complexity involved in it. The aim is to learn a function that assigns to each instance represented by its feature vector, the most probable assignment of the class variables i,e. gene type and mutation. This large number of possible combination increases the complexity of MDC problems, making them more difficult to solve than single label classification problems.

This work investigates the existing multi-dimensional approaches to identify the ASD genes and the co-occurring mutations which exhibit dependency among them. Given that a candidate ASD gene is affected by various mutations, the identification of gene - mutation from a set of features is modeled as a multi dimensional classification task. A multi-dimensional predictive model is constructed using problem transformation and algorithm adaptation to find the ASD causing genes and the triggering mutations. Here, the attributes of genetic sequences and the mutations which increase the risk of ASD are used to train the model. The proposed architecture of multi dimensional model for ASD gene - mutation prediction comprises of the three building blocks: feature engineering and dataset creation, model building, evaluation and is depicted in Fig.4.12.

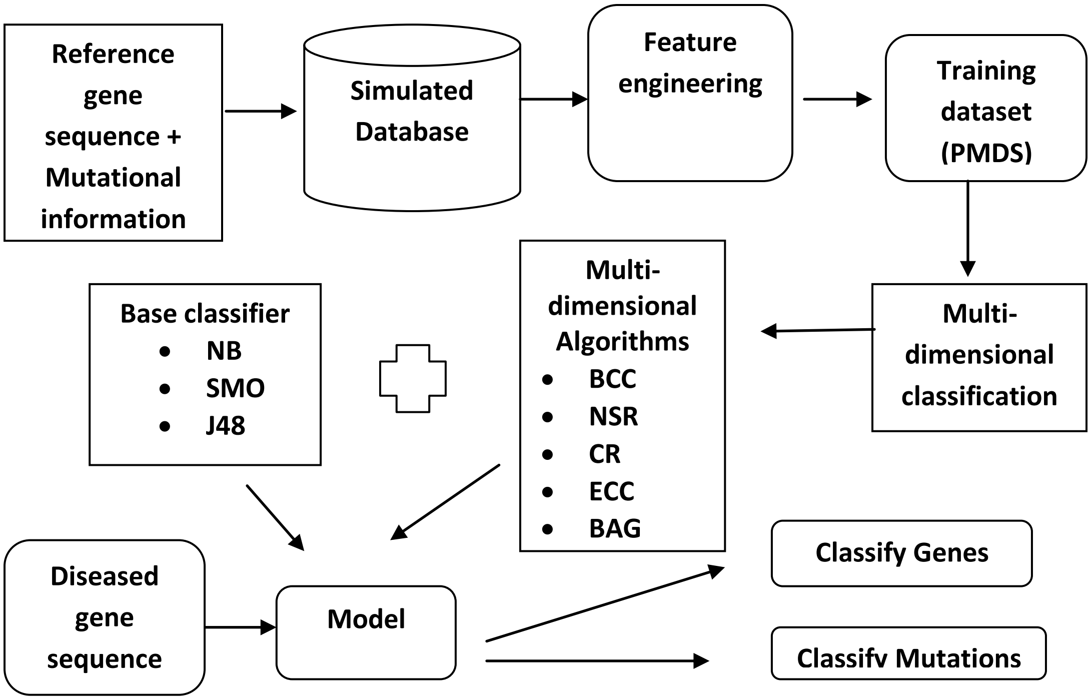

Fig. 4.12 Proposed Framework for Multi Dimensional Classification of Genes and Mutations

In the initial phase the corpus described in Chapter 3 is used and the gene specific features like nucleotide composition, Rho values of biwords, Z scores of biwords, Alignment score, count of exons, donor sites, acceptor sites are pooled with the mutation features such as mutation features (GS), substitution matrix features (SM) and amino acid change residues (AARC) and 1000 instances of dimension 58 are created. The summary of the features are depicted below.

<u>Gene Specific features</u>

Nucleotide composition	4	Number of donor sites	1
GC content	1	Number of acceptor sites	1
Rho values of biwords	16	CpG percent	1
Z scores of biwords	16	Ratio of CpG percent / expected	1
Alignment score	1	Number of exons	1

Mutation features

Mutation start position	1	Mutation end	1
Mutation length	1	Length of CDNA sequence	1
Mutation type	1	WAC matrix	1
Log-odds scoring matrix	1	BLOSUM80 substitution matrix	1
PAM-120 matrix	1	Substitution matrix	1
Mutation matrix	1	Standard deviation of bigrams	1
Mean z-score of bigrams	1	Amino acid observed, expected value	2

For the sample mutated sequence given in Fig.3.4 the pooled instance obtained is given below.

1256	2521	2211	1125	0.6652	1.04	0.84	1.27	0.78	1.14	1.11	0.68
1.21	1.06	1	1.07	0.79	0.51	0.92	1.28	1.17	0.75	-4.75	7.17
-3.72	4.16	5.27	-13.42	5.8	1.5	0.19	2.54	-5.1	-8.32	-2.15	6.91
2.77	13130	2	15	13	0.149	1.003	898	898	1	7113	0
-1	-6	-0.4	-3	-4	-2.2	2	5	7.23	-2.76		

There are two class labels assigned for this dataset namely the gene class and mutation class. The gene class has ten class labels as there are ten genes taken for the study. There are four types of mutations considered for the study and hence the dataset consists of four class labels for mutation class. Min-max normalization is used to normalize the feature values and finally the dataset with 1000 feature vectors of dimension 58 is developed and named as Pooled Mutation dataset (PMDS). The sample dataset is shown in Appendix A.

The second phase consists of building multi-dimensional models for classifying the genes and mutations simultaneously. The dataset comprising both gene specific and mutation specific descriptors are run through different types of configurations focusing on finding an efficient and accurate way of building the model. The multi-dimensional model is built using three problem transformation techniques Bayesian classifier chains (BCC), NSR which is a multitarget version of pruned sets, class relevance (CR) two algorithm adaptation methods Ensemble of classifier chains (ECC) and Bagging(BAG) with three base classifiers Naive Bayes(NB), SMO and J48. In order to make the experiments reproducible, the algorithms are implemented using default parameters of the Explorer panel in the graphical user interface. The seed value, pruning value p and subsampling value n are set to default value of 0.

In the last phase, 10- fold cross-validation technique is used to evaluate the models for their predictive performance using various metrics like Hamming loss, zero one loss, Hamming score, exact match and accuracy.

Hamming Loss - Hamming loss gives the percentage of data predicted incorrectly on average and when hamming loss is equal to 0, best performance is reached. It measures accuracy by calculating the fraction of wrong labels with the total number of labels.

Exact match - Exact match which is also called global accuracy gives the percentage of test dataset predicted exactly same as in the training dataset. Exact match is one of the metrics that overlooks the fact that multi-label prediction has a notation of being partially correct and count only exact matches. 0 / 1 loss has its optimal value at 0 and since exact match only is its inverse, it has its optimal value at 1.

Hamming score - Hamming score is given by the ratio of set of correct classes to union of predicted and correct classes.

Zero one loss - It is one of the strictest metrics since it completely ignores the fact that multi-label prediction has a notation of being partially correct and count only exact matches where all labels in the set for an instance is correctly predicted. The optimal value of this measure is at 0.

Experiment and Results

This experiment is conducted by learning the PMDS dataset with five different multi-dimensional machine learning techniques BCC, NSR, CR, ECC and BAG along with three different base classifiers namely Naïve Bayes, SMO and J48 using MEKA. It is a popular Java library for multi-target classification. It is an open source ML framework which provides an extensible support for developing, running and evaluating multi-target classifiers. Since MEKA contains a more extensive library, it is the natural choice for this work which aims to compare several multi-label classification algorithms. The performance of the trained models was evaluated using 10-fold cross validation for its predictive accuracy. To investigate the effectiveness of gene- mutation prediction models empirical evaluation was carried out using the evaluation measures namely Hamming loss, Zero One loss, Hamming score, Exact match and Accuracy and the results obtained with respect to these measures are tabulated in Table X.

Table X Performance of Multi Dimensional Gene – Mutation Classifiers Based On Loss Measures

Method	Base Classifier	Hamming score	Exact Match	Hamming loss	Zero one loss
BCC	J48	0.742	0.448	0.258	0.552
ECC		0.642	0.388	0.358	0.612
BAG		0.746	0.488	0.254	0.512
NSR		0.668	0.288	0.332	0.712
CR		0.752	0.488	0.248	0.512
BCC	SMO	0.724	0.748	0.076	0.152
ECC		0.623	0.744	0.078	0.156
BAG		0.788	0.756	0.072	0.144
NSR		0.712	0.744	0.078	0.156
CR		0.624	0.728	0.076	0.152
BCC	NB	0.688	0.736	0.082	0.164
ECC		0.754	0.728	0.086	0.172
BAG		0.718	0.636	0.182	0.364
NSR		0.794	0.810	0.066	0.120
CR		0.610	0.736	0.090	0.164

The above comparative results show that the combination of Nearest Set Replacement (NSR) and Naïve Bayes(NB) had the lowest Hamming Loss (0.066) and Zero one loss(0.120). Similarly the BAG and SMO method is also found to be promising with Hamming Loss of 0.072 and Zero one loss of 0.144. J48 with any combination has high levels of Hamming loss and Zero one loss. J48 and NSR have the maximum zero one loss of 0.712. NSR and NB attain an exact match of 0.81 whereas NSR and J48 have the worst match of 0.288. The configuration of SMO and BAG attains an exact match of 0.756 and its Hamming score is 0.788. Among all combinations NSR and NB achieves the highest Hamming score of 0.794. The performance of classifiers based on training time, test time and accuracy for the various configurations built is depicted in Table XI.

Table XI Training Time, Test Time, Accuracy for Each Configuration

Classifier	Base Classifier	Training Time	Test Time	Accuracy for Gene class	Accuracy for mutation class	Average accuracy
BCC	J48	0.019	0.002	0.788	0.806	0.797
ECC		0.058	0.001	0.445	0.806	0.625
BAG		0.059	0.002	0.681	0.806	0.743
NSR		0.011	0.002	0.484	0.818	0.651
CR.J48		0.006	0.001	0.588	0.802	0.695
BCC	SMO	0.239	0.001	0.782	0.815	0.798
ECC		2.212	0.008	0.771	0.795	0.782
BAG		1.983	0.005	0.725	0.781	0.753
NSR		0.101	0.002	0.672	0.772	0.722
CR		0.147	0.001	0.727	0.766	0.746
BCC.	NB	0.011	0.006	0.792	0.764	0.778
ECC		0.036	0.034	0.819	0.756	0.787
BAG		0.03	0.031	0.801	0.764	0.782
NSR		0.003	0.006	0.807	0.826	0.816
CR		0.006	0.002	0.811	0.784	0.797

The results show that the accuracy of gene class is high for ECC and NB configuration with 0.819 whereas NSR with NB shows excellent accuracy of 0.826 for mutation class. ECC and Decision tree J48 has the least average accuracy of 0.625.On an average Nearest Set Replacement and Naïve Bayes outperformed other models with an average predictive accuracy of 0.816. The training time taken for this configuration is 0.003 which is comparatively less. The performance of individual classifiers based on average accuracy, training and testing time of the classifiers are presented in Table XII.

Table XII Average Accuracy, Training and Testing Time of the Gene-Mutation Classifiers

Classifers	Accuracy	Avg. Training Time	Avg. Test Time
Multi dimensional classifiers			
BCC	0.791	0.089	0.003
ECC	0.731	0.769	0.014
BAG	0.759	0.691	0.013
NSR	0.729	0.038	0.003
CR	0.746	0.053	0.001
Base Classifiers			
NB	0.792	0.017	0.016
SMO	0.760	0.936	0.003
J48	0.702	0.031	0.002

The results prove that among multi dimensional classifiers, Bayesian Classifier chain combined with any base classifier had generally achieved good accuracy of 0.791 and the time taken for training and testing is 0.089 and 0.003. Among the base classifiers Naive Bayes combined with any problem transformation has accuracy of 0.792 which is better than SMO and J48. Naive Bayes utilizes the probability of a feature based on prior knowledge of conditions related to that feature which attributes to its highest accuracy. This probabilistic classier converges quickly than other classifiers and hence the time taken for training is comparatively less. SMO has a reasonable training time and accuracy whereas J48 has least accuracy of 0.702. The experimental results of multi-dimensional classifiers and base classifiers are illustrated in Fig 4.13 to Fig.4.16.

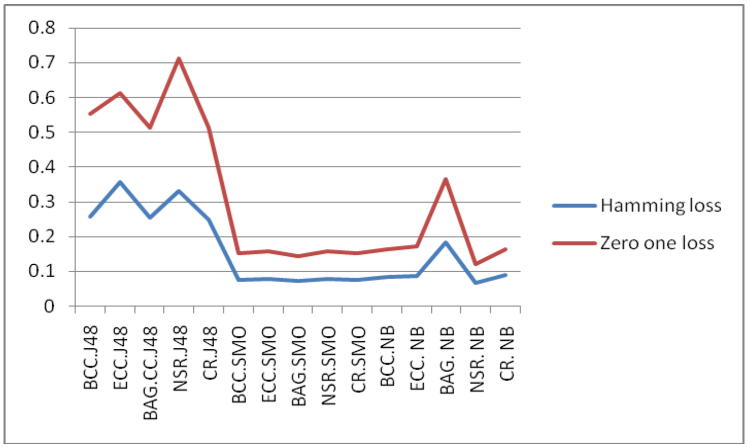

Fig. 4.13 Hamming Loss and Zero-One Loss of Gene-Mutation Classifiers

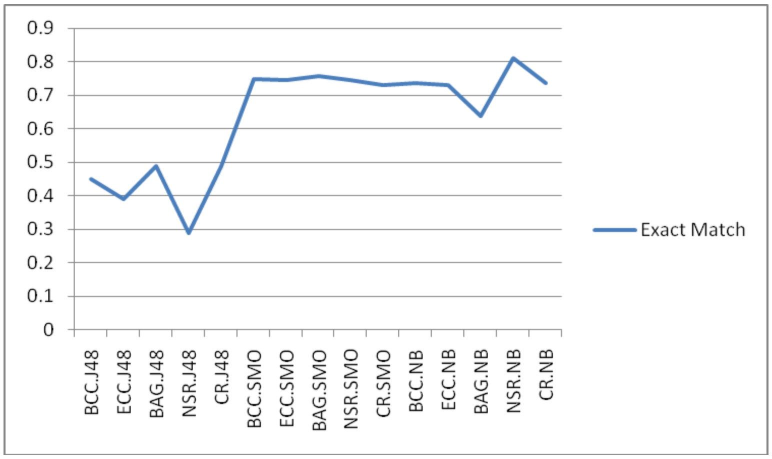

Fig. 4.14 Exact Match of Multidimensional Gene-Mutation Classifiers

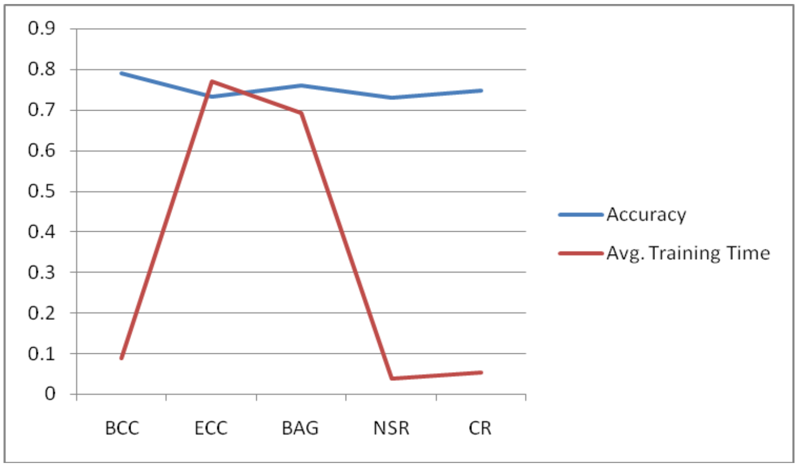

Fig. 4.15 Accuracy and Average Training Time of Multi-Dimensional Classifiers

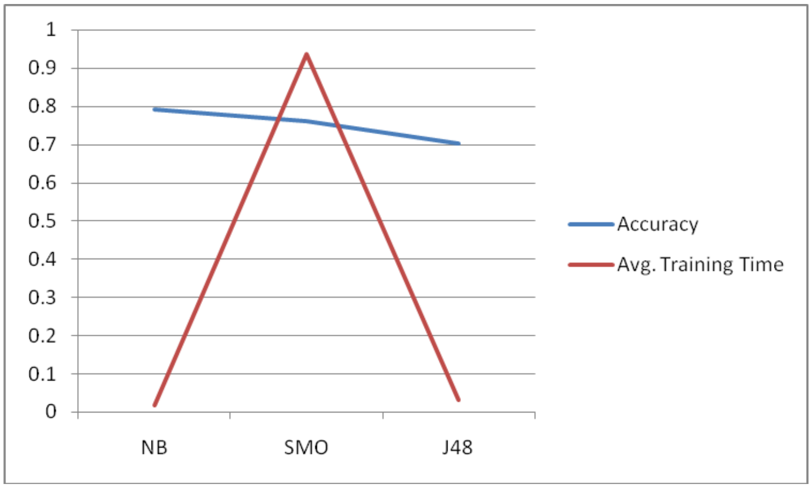

Fig. 4.16 Accuracy and Average Training Time of Base Classifiers

From the comparative results it is noticed that both Hamming loss and zero one loss is minimum for the combination of NSR and Naive Bayes as seen in Fig.4.13. The exact match of multi-dimensional classifiers depicted in Fig.4.14 shows that it is minimum for NSR and J48 whereas it is maximum for NSR and Naïve Bayes. Among the multi-dimensional classifiers BCC has the highest accuracy whereas NSR has the lowest training time. The accuracy of NB is maximum and average training time is minimum for both J48 and NB.

Findings

The comparative performance of the multi-dimensional classifiers shows that the combination of NSR and NB has outperformed other configurations with high classification accuracy of 81.6%. This combination of classifiers is able to learn the gene and mutation patterns from the pooled feature set in a faster and efficient manner. In addition, this approach affords fast and highly scalable model building. Naive Bayes has few tunable parameters and is useful as a baseline for the classification of ASD gene – mutation problem. This advantage is effectively exploited when combining NSR with Naive Bayes than any other multi-dimensional classifiers. The pooled mutation dataset also plays a major role in enhancing the model performance.

SUMMARY

This chapter demonstrated the modeling of identifying ASD causing genes and mutations as two different multi class classification problems. The implementation of supervised machine learning techniques for identifying the ASD causing genes and the triggering mutations based on gene specific and mutation specific features have been described in detail. Three independent models were built for these tasks based on CMDS and MDS datasets. The results proved that decision tree based model is promising in both the cases when compared to other methods. Further the multi-targeted machine learning approach for gene - mutation concurrent classification through multi dimensional modeling has also been described. The experimental results of 15 different configurations were reported and the comparative analysis was presented. Various interpretations of the experimental results were summarized in this chapter. The development of machine learning model to identify the gene susceptibility to ASD will be discussed in the following chapter.

Remarks

- Paper titled “Decision Tree Based Model for the Classification of Pathogenic Gene Sequences Causing ASD”, published in Communications in Computer and Information Science, Vol. 876. pp 201-212, Springer, Singapore **(Scopus indexed)**
- Paper titled “Machine Learning-Based Model for Identification of Syndromic Autism Spectrum Disorder”, published in Studies in Computational Intelligence, Vol. 771. pp. 141-148, Springer, Singapore **(Scopus indexed)**
- Paper titled “Identification of Autism Spectrum Disorder using a Multi-Label Approach”, published in Journal of Advanced Research and Dynamical Control Systems, Vol. 11, Issue – 2, pp. 134-141, 2019, **(Scopus indexed)**