Consecrate Recurrent Neural Network Classifier for Autism Classification

S. Padmapriya, S. Murugan

Abstract: ***Most recent discoveries in Autism Spectrum Disorder (ASD) detection and classification studies reveal that there is a substantial relationship between Autism disorders and gene sequences. This work is indented to classify the autism spectrum disorder groups and sub-groups based on the gene sequences. The gene sequences are large data and perplexed for handling with conventional data mining or classification procedures. The Consecrate Recurrent Neural Network Classifier for Autism Classification (CRNNC-AC) work is introduced in this work to classify autism disorders using gene sequence data. A dedicated Elman [1] type Recurrent Neural Network (RNN) is introduced along with a legacy Long Short-Term Memory (LSTM) [2] in this classifier. The LSTM model is contrived to achieve memory optimization to eliminate memory overflows without affecting the classification accuracy. The classification quality metrics [3] such as Accuracy, Sensitivity, Specificity and F1-Score are concerned for optimization. The processing time of the proposed method is also measured to evaluate the pertinency.***

Keywords: Autism Spectrum Disorder classification, Gene sequence-based autism disorder detection, Recurrent Neural Network., Elman Network, Long Short-Term Memory

I. INTRODUCTION

Autism Spectrum Disorder is a development difficulty that can affect all age groups [4][5]. It is a nerve related problem affects the natural communication and interaction. There are several categories of ASD such as emotional disorders, social disorders, physical disorders, obsessive interests, repetitive behavioral patterns, communicational difficulties and overall cognitive illness. ASD can be easily diagnosed in case of adults than in the babies. Earlier detection of ASD has the higher probability of getting cured by proper treatment [6]. The real challenge is detecting ASD in the earlier stage of babies is more complicated. The development of communication capabilities of babies takes place between 6 to 18 months. Treating them after detecting ASD in a normal way is a time taking process with lower probability of cure.

Recent researches and studies show that there is a connection between gene patterns and ASD [7][8][9]. A particular pattern of gene sequence indicates the chances of having a particular type of ASD[10][11]. By searching several patterns in gene sequences will be useful to detect ASD. Machine Learning and automated classification procedures are used to detect ASD related gene patterns from a gene sequence.

Revised Manuscript Received on October 15, 2019

S. Padmapriya, Assistant Professor, Department of Computer Science, SRM Trichy Arts & Science College, Trichy.

S. Murugan, Associate Professor, Department of Computer Science, Nehru Memorial College, Puthanampatti.

These conventional approach follows a standard pattern matching or sequence pattern detections which are more time depleting and memory starving. The computational complexity befalls because of the size of a human gene sequence. The number of base-pairs in a human gene sequence is about 3312466 which requires 6.31 GB of storage space to store in uncompressed format.

Fortunately, the basic characters are limited to A, C, T and G in a gene sequence. There are several encoding methods are used to compress the gene sequence data[12][13]. The advantage of this compression methods are the reduced memory space and the disadvantages of using these compressions is the processing time. A compression method with a good compression ratio requires more time to compress and extract the gene sequence data. The size to store a gene sequence is comes around 700 MB even-after applying a good compression. Searching for several gene patterns in this huge input data requires more processing time and memory.

Designing a classifier with high accuracy within reasonable computational resource consumption is a perplexed process which is addressed by this proposed CRNNC-AC work. The advantages of Recurrent Neural Networks with Long Short-Term Memory are expended with custom created algorithms to detect Autism Spectrum Disorder more accurately in earlier stages using gene sequence. The proposed procedure is developed to improve the Specificity and Sensitivity which will be a boon to earlier stage ASD detection, classification and treatment.

II. EXISTING METHODS

There are some successful works performed already in ASD classification using gene sequences. Best five methods are taken here for discussion and to state the uniqueness of the proposed method. Key facts about the existing methods along with merits and limitations of the existing gene based ASD classification methods are discussed in this heading. Predicting Autism Spectrum Disorder Using Blood-based Gene Expression Signatures and Machine Learning [PABGESML][14], Heritable genotype contrast mining reveals novel gene associations specific to autism subgroups [HGCM][15], Building a genetic risk model for bipolar disorder from genome-wide association data with random forest algorithm [GRMBGRF][16], Selection and classification of gene expression in autism disorder: Use of a combination of statistical filters and a GBPSO-SVM algorithm [GBPSO-SVM][17] and Improving the classification of neuropsychiatric conditions using gene ontology terms as features [GOTCNC][18] are taken as the existing methods.

1.1. Predicting Autism Spectrum Disorder Using Blood-based Gene Expression Signatures and Machine Learning [PABGESML]

PABGESML process begins with Human Genome Microarray data acquisition followed by data preprocessing and selection of differentially expressed genes. They used supervised and unsupervised learning methods to develop the prediction model. PABGESML uses hierarchical cluster analysis by complete linkage and Euclidean distance for unsupervised learning. Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Linear Discriminant Analysis (LDA) are used by PABGESML for supervised learning. The implementation is performed in R Language with GSE26415 microarray dataset. As per the observations, SVM and KNN achieved the accuracy of 93.8% whereas LDA achieved 68.8%.

The stated advantage is the higher Sensitivity values achieved by SVM and KNN. These the experiment results are based on the small sample size and unable to provide a clear association between gene sequences and ASD classifications – which are the limitations of this work.

1.2. Heritable genotype contrast mining reveals novel gene associations specific to autism subgroups [HGCM]

Imputation of missing genotypes is the preprocessing stage of HGCM. Then opposite subgroup pairs are constructed using existing autism subtype classifications. The Association of Single Nucleotide Polymorphisms (SNP)with each subgroup is verified using genome-wide prioritization procedure to select most significant SNPs. The extended Frequent Pattern Mining procedure of HGCM is used to find the combinations of SNPs and their related autism subgroups. The main modules of HGCM are Missing genotype imputations, Extended Frequent Pattern Mining, Population Division, Genome-wide SNP Prioritization, Contrast Mining and Family based Association statistical Testing. AutDB and PubMed datasets are used to evaluate the metrics of HGCM.

Improved accuracy is achieved by HGCM by its capability of identifying new sensitive genomes. The limitation of HGCM is that the processing time gradually increases while number of associate genomes increase.

1.3. Building a genetic risk model for bipolar disorder from genome-wide association data with random forest algorithm [GRMBGRF]

Genome-wide Association and Random Forest are the keys of GRMBGRF. Genome-wide association data from GAIN dataset are used as the training set. Random Forest method is used for classification and to construct regression trees. The candidate SNPs from the GAIN dataset are used to construct the training model. Molecular Signature Database is used to identify significantly enriched pathways.

The customized Random Forest procedure of GRMBGRF acquires higher accurate results which is the main advantage of this method, whereas, the construction and experiments are designed for Bipolar disorder only – which is the limitation of this procedure.

1.4. Selection and classification of gene expression in autism disorder: Use of a combination of statistical filters and a GBPSO-SVM algorithm [GBPSO-SVM]

In this method the standard SVM procedure is added with Geometric Binary Particle Swarm Optimization to achieve improvements in classification accuracy. High variance is one of the problems in finding autism disorder classification because the gene association with a particular ASD subgroup is in high fluctuation among different people. GBPSO-SVM method uses discriminative motif discovery to resolve the higher variance issue. The classification accuracy is proved by evaluating this method with the dataset from GEO – NCBI. The main phases involved in GBPSO-SVM are Pre-Selection Operations, Selection using statistical filters, Selection using a Wrapper-based GBPSO-SVM algorithm, Dataset reduction, Classifier assignment and Final stage of selection and classification.

Improved classification accuracy is the advantage of GBPSO-SVM and using limited set of key genomes is the observed as the limitation.

1.5. Improving the classification of neuropsychiatric conditions using gene ontology terms as features [GOTCNC]

The authors of this work state that the association between Neuropsychiatric disorders with specific molecular foundations lacks stability and generalizability. To improve the classification performance, they proposed a method that uses annotation-based classifiers. The general hypothesis of this work is that the performance, stability, generalizability and Interpretability can be improved using annotation-based classifiers. Four different classification algorithms are taken to classify six different datasets and the results are presented by the authors. Based on the experimental results it is proved that the Performance, Stability and Interpretability are improved by using Annotation-based feature space. The Generalizability has no significant improvement with Annotation-based feature space.

Improved accuracy is the advantage of this method whereas preparing the Annotation-based feature space requires some Biological Process (BP) consumes more processing time is observed as the limitation.

III. RELATED WORKS

Artificial Neural networks (ANN) are the computer replicates of human brains. ANN has similar components of original brain such as neurons, dendrites and axon. The ANN is the direct simple representation of human decision-making process. ANN has numerous applications in Machine Learning, Classification and Prediction models. There are different types of ANN such as Feed Forward Network, Radial basis function Neural Network, Kohonen self-organizing Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks and Modular Neural Networks. The concept of Recurrent Neural Networks selected in this proposed work because of the classification abilities of RNN in Images [19].A specific type RNN named Elman neural network is used in this proposed system to find and classify the ASD subgroups.

1.6. Recurrent Neural Network

RNN is an extended model of Feedforward Network to materialize some benefits. RNN can be used to classify continuous streaming data which is one of the best advantages over Feedforward networks. RNN has recurrent

hidden state in which the activation state depends on previous iteration.

Therefore, RNN is more dynamic temporal behavior. Let a sequence of data $X = \{x_1, x_2 \dots x_r\}$ where x_i is the data of i^{th} iteration, the Recurrent hidden state h_t of RNN is updated as $h_t = \begin{cases} 0, & if\ t = 0 \\ \varphi(h_{t-1}, x_t) & otherwise \end{cases}$, where φ is a non-linear function. Let the output of the RNN be $Y = (y_1, y_2 \dots y_\gamma)$. The standard RNN update model is $h_t = \varphi(W x_t + \cup h_{t-1})$ where w and U are coefficient matrices of current step t. The probability distribution $p(x_1, x_2 \dots x\gamma)$ can be expanded as $p(x_1) \dots p(x_\gamma | x_1, \dots x_\gamma - 1) = \varphi(h_t)$.

The LSTM unit activation is triggered using the equation $h_t = O_t\ tanh(c_t)$ where O_t if the output gate of t^{th} iteration i.e.. $O_t = \sigma(w_{oi} x_t + w_{oh} h_{t-1} + w_{oc} c_t)$, where w_{oi} and w_{oc} are the input-output weight matrix and memory output weight matrix respectively. The memory cell update c_t is performed as $i_t \circ \bar{c}_i + f_t \circ c_{t-1}$ where $\bar{c}_t = tanh(w_{ci} x_t + w_{ch} h_{t-1})$, f_t is the forget gate.

The input gate i_t and forget gate f_t are calculated using the following equations (1) & (2)

$$i_t = \sigma(W_i i x_t + W_{ih} h_{t-1} + W_{ic} c_{t-1})$$
- Equation (1)

$$f_t = \sigma(W_f i x_t + W_{fh} h_{t-1} + W_{fc} c_{t-1})$$
- Equation (2)

1.7. Elman network

Elman Network is introduced by Jeffrey L. Elman to make the RNN learn from time-varying pattern input sequences. Elman network has four types of layers, they are Input Layers, Hidden Layers, Context Layers and Output Layers. The network architecture of Elman network is given in Figure 1.

Elman network can remember certain level of previous training outcomes that can influence the current decision-making policy of the RNN. A set of standard error calculation procedures followed in Elman Networks for training and fine tuning the Elman model.

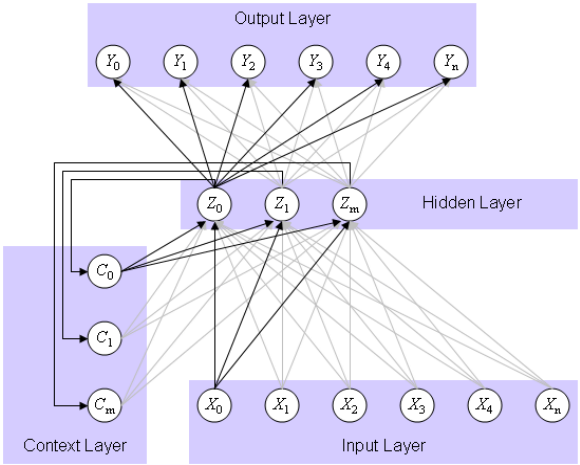

Figure 1: Elman Network

$$\text{Mean Absolute Error} = \frac{1}{N} \sum_{t=1}^{N} |X(t) - \hat{X}(t)|$$
- Equation (3)

$$\text{Mean Absolute Percentage Error} = \frac{1}{M} \sum_{t=1}^{N} \left| \frac{X(t) - \hat{X}(t)}{\hat{X}(t)} \right|$$
- Equation (4)

$$\text{Root Mean Square Error} = \sqrt{\frac{1}{N-1} \sum_{t=1}^{N} [X(t) - \hat{X}(t)]^2}$$
- Equation (5)

IV. PROPOSED METHOD

CRNNC-AC is built based on four major modules. They are CRNNC Elman network, Altered LSTM for better classification, Associative genome sequence heuristic parallel parser and Integrated Classifier. These modules are build using some existing methods along with legacy procedures to achieve the best Autism Spectrum Classification results with higher speed.

1.8. CRNNC Elman Network

A standard Elman network contains four different types of layers. The Input layer, hidden layers and output layers are very similar to the Artificial Neural Network environment. Context layer is added in the Elman network to produce more relevant results by which previous iteration results are permitted to control the current outputs to a certain degree.

In the process of finding ASD classifications from Genome sequences, a lengthy stream of human gene sequence has to be verified for several suspected occurrences of predefined genome patterns. A new layer named participant polypeptide is introduced in CRNNC Elman network, in which a set of polypeptide/proteins involved in ASD findings. The number of participant polypeptides can reach any value of integer $q$based on the correlation between the gene sequences and ASD subgroup classifications. This layer is developed during the training phase to ensure the identification of desired gene sequences. The sigmoid function is also biased to adopt new participant polypeptides in this layer. Each node in the context layer gets an influence from each other node in the participant polypeptide layer which shifts-up the sigmoid function to fire the connection in detection of a ASD gene sequence occurrence. Proposed CRNNC Elman network is illustrated in Figure 2.

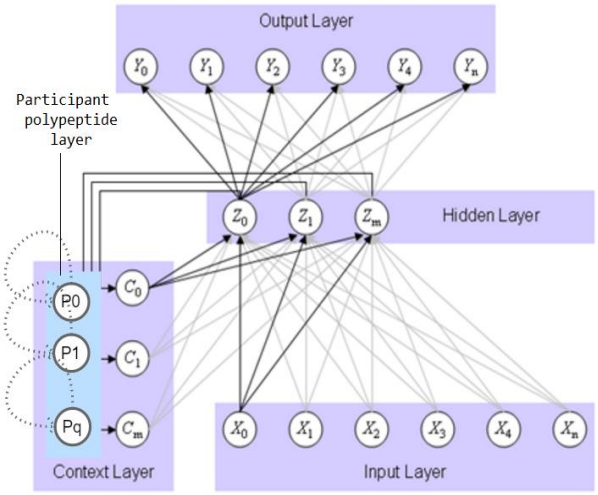

Figure 2: CRNNC Elman Network

The RNN sigmoid function $\sigma(x) = \frac{2}{1+e^{-2x}} - 1$ is changed as $\sigma(x) = \frac{2}{1+e^{-2f(x)}} - 1$

where $f(x) = \begin{cases} 0 : x < 0 \\ x : x \geq 0 \end{cases}$

These changes are introduced in proposed CRNNC Elman Network model to ensure the earliest detection of a particular gene sequence based on its correlation to the ASD classification. The detection time is inversely proportional to the correlation of the gene sequence and ASD subgroup. The activation function curve of proposed CRNNC Elman network is given in Figure 3.

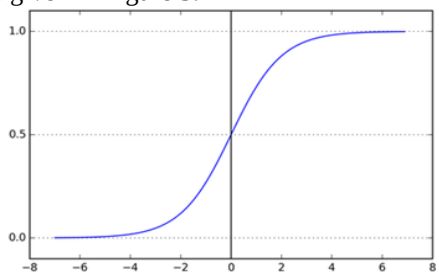

Figure 3: Activation function curve

1.9. Altered LSTM

Long Short-Term Memory is the key process of Recurrent Neural Networks in deep learning. The Long Short-Term Memory used in proposed CRNNC in two different levels. The first level LSTM is used to identify the ASD associated gene sequences and the second level LSTM used to identify a particular ASD subgroup based on the detected gene sequences. These two LSTM procedures are explained here with an Example. A gene sequence associated with Asperger's Syndrome is given below.

TCATAGACTCATTCAGCGTACGTACCTAAAGCTCA
GGATGGAGTTTCGAATTGATCGAAGGCAACCTGAA
TTCCTTTCAGCTATGCGATTTTTGATGGTCTGTGTCT
TCTTCCCAACTCACGACGGGGGAGCCGGACACGGA
AGGATAATATATTCCCAACGGGTTCGGTAGCGAAG
ATTCTGAAAACCCCGTTCAACCATTTAGCGATCGG
GCAACATACTATCTATAGCCTCCCAACAATGCTCCC
CGAACCGATGTGCCTGTTATACTTAACTGAAGGCT
CATACTATTGGAGTGGGGGTATGACCTGTATGCTA
CCCTGGGCTAGTTCGACCGTCGCCAGTCCCAAACT
CTTCTTTCGATCCTCTTCTTGCACGTTCTTACAAAA
GGCTAGAG

This sequence is given an ID as S0000, where ever this sequence occurred the first level LSTM has to detect the sequence and label the sequence with corresponding IDs. Another sequence associated with the same Asperger's syndrome is given below, the sequence is labeled as S0001 in training sessions. Occurrence of both the sequence in a single input ensures the presence of Asperger's Syndrome.

AACGGAACTCGCGTCCTCGGCAGACCCACGGCTAA
AAGTGGTAGCTGGGCACTGCAGCGTTCGTGGCTGA
TCAATGGCGCTCGACCCTCTTGGAGAGTCAGAGGT
GAGCCATACCACACCCCCTAGTCCAGGTCTACGTA
GCAACGTAATAGCCCACTAGGCAGGCTCCCCCACG
ACACCCTACGTCCCCTTGCGGCGATTACATCCCACT
GGACGCATCTAAAGAAATGGGAAATATACTCTCAA
ACCAGGGAAGCGATATGGGCCGTTCCGAAATGATC
CCAGACATCGAATCGTGCTGGTATGAATGCTTTTA
AAATTCGGGCAGGTGACATACAGAATAGATCTAAG
TCTGGGTCTCGACTCTTGCCTCGCGGTAGCATAACA
GCCGCGC

The Level 2 LSTM hovers about the Level 1 LSTM where finding the gene sequence pattern is the work of level 1 LSTM whereas mapping the associated disorder is the task if level 2 LSTM. This process is explained in Figure 4.

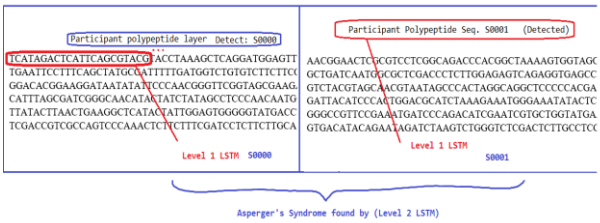

Figure 4: Level 1 & Level 2 LSTM

The input gate and forget gate are designed using equations 1 and 2 respectively. For first level LSTM, the input vector has the amino-acids of gene sequences. The output the of the level 1 LSTM then assigned as the input vector for level 2 LSTM. All the other methods are similar for both the processes. The dataflow is explained in Figure 5.

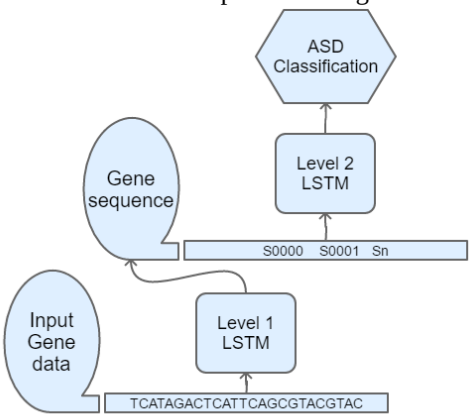

Figure 5: LSTM dataflow

1.10. Associative genome sequence heuristic parallel parser

Processing entire genome sequence for different ASD associated polypeptides in a sequential manner will consume more time. A genome sequence parallel parser is introduced in this proposed CRNNC-AC to reduce the processing time. The heuristics approach is used to take a parallel diversion while continuing the sequence run through the regular RNN. If the parallel parser confirms a presence of a particular ASD associated gene, then It triggers Level 2 LSTM to with the gene sequence ID. While introducing heuristic parallel searches, the memory requirement will be high but in this case it is balanced by the memory efficiency of RNN over the conventional ANN procedures.

Whenever the gene sequence is identified to match a particular ASD associated sequence with one tenth of the portion, then a heuristic parallel search is initialized by the associative genome sequence parallel parser. Every single heuristic genome sequence parallel parser is indented to validate a sequence with a particular sequence ID. The sequence is pre-loaded into memory and a batch X-OR operation ($\oplus$) is performed to find the exact or relational match of the sequence. If a sequence match scores more than 90% of 0s, then the match with that particular ASD associated sequence will be triggered. The Associative genome sequence parallel

parser is explained in Figure 6.

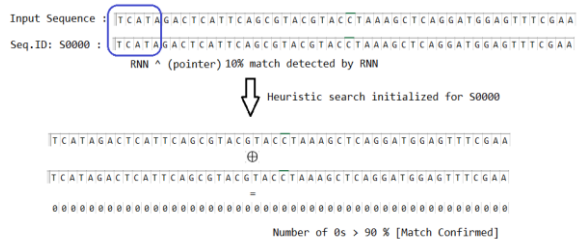

Figure 6: Associative genome sequence parallel parser

Since the parallel batch $\oplus$ process activated immediately after getting 10% of initial match by RNN, there is a reduction of 90% computational cost saved by introducing the parallel heuristic search. Let the expected processing time of the entire gene sequence be $\rho_\varepsilon = \rho_0 + \rho_1 + \cdots \rho_n$ where n is the size of the sequence, then the approximate computational time can be calculated as $\rho_a = \frac{\sum_1^n \rho_i}{0.8\eta\Psi}$ where η is the maximum number of parallel heuristic searches permitted by the computational environment and Ψ is the number of sequences associated with ASD subgroups. The modern computational devices are loaded with ample of memory these days tangibly reduces the processing time of proposed CRNNC-AC procedure.

The associative genome sequence parallel parser can trigger any number of parallel heuristic searches between $0 \rightarrow \eta$ based on the input data and available computational environment. This parallel heuristic process is applicable only for Level 1 LSTM in this proposed method. Since Level 2 LSTM involved in computing the association sequences with ASD classifications, applying heuristic method is excluded here to preserve the Accuracy, Sensitivity and Specificity.

1.11. Integrated CRNNC-AC

The modified CRNNC Elman Network is used to reduce the memory usage of conventional Artificial Neural Networks which makes space to implement the associative genome sequence parallel parser in RNN. Introduction of participant Polypeptide layer is used to improve the classification accuracy. First level and second level LSTM are used to detect the ASD associated gene sequences and to classify them into ASD subgroups. Parallel Heuristic Search parsers are used to find fast pattern matching to conserve computational resources wisely. The proposed CRNNC-AC reads gene sequences as input and produce ASD classifications as Outputs – illustrated in Figure 7.

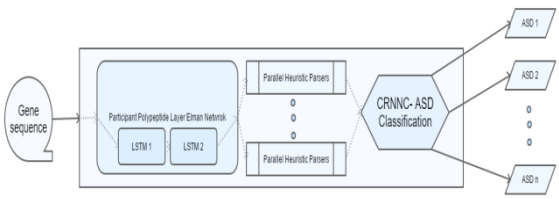

Figure 7: CRNNC-AC

V. EXPERIMENTAL SETUP

A vast collection of data is acquired from the National Center for Biotechnology Information (NCBI) website. The Autism Genome Project (AGP) Consortium – is a complete genome association stage I and II study over 1500 offspring Trios [20]. This dataset is authorized by Department of Health and Human Services – National Institute of Health. The software frameworks to access the dataset are provided by the NCBI itself. Each genome record contains around 800000 lines of genome sequences which contain a minimum of 65000000 polypeptides collected to find autism spectrum disorders associated gene sequences. CoreLib software development kit (SDK) and Library is used to access the basic functionalities provided by hundreds of researchers during the last few decades. The portable Core Library (CoreLib) [21]is accessed through Visual Studio Integrated Development Environment [22][23] to make use of easy User Interface (UI) design and to visualize results.

A computer with Intel Core i5-7200 processor running at 2.7 GHz and equipped with 8 GB RAM is used to perform the experiments. The processing time is based on the 64-bit Windows 10 Operating System with dedicated process threads to get complete utilization of CPU cores to train and to test the methods. The Operating system and the hardware controls are controlled by the dedicated User Interface Application to measure the evaluation metrics of the existing and proposed methods.

VI. RESULTS AND ANALYSIS

The experiments are conducted by splitting the entire dataset into 10 different chunks to evaluate the intermediate performances of existing methods and proposed method. Accuracy, Sensitivity, Specificity, F1-Score and processing time are measured for all methods in each time chunk to get the complete analysis.

1.12. Accuracy

Accuracy is one of the prime evaluating factors of any classification algorithm. Accuracy is calculated as $\frac{(TP+TN)}{(TP+TN+FP+FN)}$, where TP is True Positive, TN is True Negative, $FP$is False Positive and FN is False Negative. Accuracy is directly proportional to the quality of a classification algorithm. The measured values of accuracy for existing methods and proposed method are given in Table 1.

Table 1: Accuracy (%)

Accuracy (%)						
Data Chunk	PABGESML	HGCM	GRMBGRF	GBPSO_SVM	GOTCNC	CRNNC_AC
1	84.37	86.28	91.04	88.81	90.22	94.54
2	85.35	86.06	91.93	89.63	89.84	94.79
3	84.93	84.91	92.73	89.71	90.52	95.43
4	84.18	85.76	93.85	89.93	89.68	96.11
5	84.04	88.41	92.95	86.82	89.36	96.01
6	84.02	85.5	93.31	89.29	88.93	96.28
7	84.07	85.54	92.35	89.48	89.29	97.41
8	84.69	87.24	92.45	90.84	90.39	95.49
9	86.1	87.51	92.93	89.99	89.75	94.43
10	83.64	85.48	92.67	90.16	90.35	95.1
Avg	84.53	86.269	92.621	89.46	89.833	95.55

Proposed CRNNC-AC scored the highest Accuracy average of 95.56% and secured the first place in ranking. GRMBGRF has the Accuracy average of 92.62% and in the second place. GOTCNC,

GBPSO_SVM, HGCM and PABGESML are succeeding in order with the accuracy average values of 89.83%, 89.47%, 86.27% and 84.54% respectively. The highest achieved Accuracy value is 97.41% while processing the 7th data chunk by proposed CRNNC-AC method.

The measured accuracy values are plotted as graph and given in Figure 8 for visual comparison.

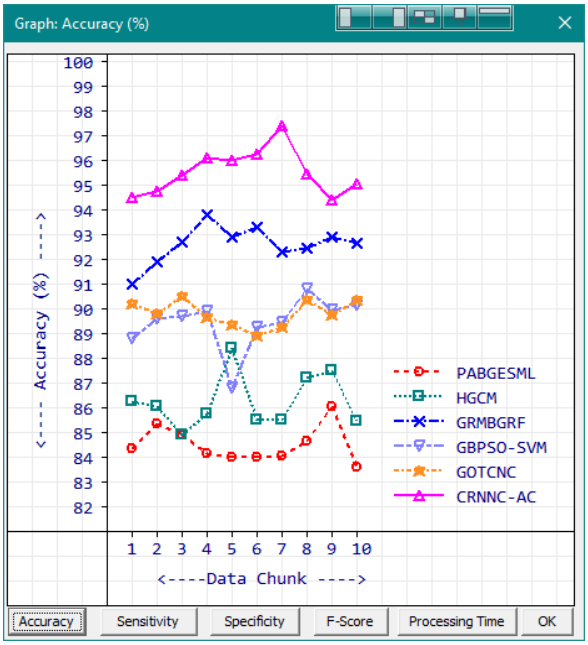

Figure 8: Accuracy (%)

1.13. Sensitivity

Sensitivity or Recall also called as True Positive rate reflects the quality of a classification algorithm. The higher values of sensitivity refer the higher quality of the algorithm. Sensitivity is calculated using the formula $\left(\frac{TP}{TP+FN}\right)$. Calculated Sensitivity values of the participated methods are given in Table 2.

Table 2: Sensitivity (%)

Sensitivity (%)						
Data Chunk	PABGESML	HGCM	GRMBGRF	GBPSO_SVM	GOTCNC	CRNNC_AC
1	85.94	87.86	91.71	90.87	92.26	94.89
2	86.15	87.31	92.85	89.34	89.99	95.45
3	84.25	84.6	93.83	89.54	89.73	95.56
4	83.42	85.58	94.23	91.44	90.78	97.68
5	82.94	88.58	91.5	87.29	90.29	97.38
6	83.74	85.44	93.52	90.15	88.98	96.82
7	84.22	86.52	93.31	88.62	90.23	98.77
8	85.25	88.33	91.12	91.54	89.57	97.18
9	86.41	87.91	93.47	90.05	88.98	94.7
10	84.7	86.07	93.93	90.23	88.96	96.36
Average	84.702	86.82	92.947	89.907	89.977	96.479

Based on the calculated results, it is observed that the classification method CRNNC-AC secured the highest Sensitivity Value of 98.77% with the average Sensitivity Value of 96.48%. It is also observed that the minimum Sensitivity value of CRNNC-AC is not less than 94.7%. GRMBGRF secured the next highest average sensitivity value 92.95%.

The measured Sensitivity values are given as comparison graph in Figure 9.

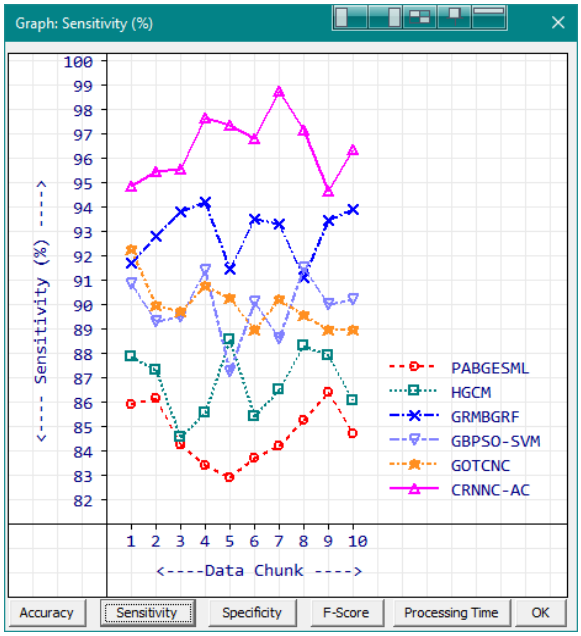

Figure 9: Sensitivity (%)

1.14. Specificity

Specificity which is also known as True Negative Rate refers the classification algorithms ability in identifying the negative results. Specificity has the equal priority to the sensitivity in classification and data mining algorithms. Specificity is calculated using the formula $\left(\frac{TN}{TN+FP}\right)$. Measured Specificity values are tabulated and given in Table 3.

Table 3: Specificity (%)

Specificity (%)						
Data Chunk	PABGESML	HGCM	GRMBGRF	GBPSO_SVM	GOTCNC	CRNNC_AC
1	82.93	84.84	90.39	86.94	88.37	94.19
2	84.59	84.89	91.05	89.93	89.69	94.15
3	85.64	85.23	91.69	89.88	91.36	95.3
4	84.97	85.95	93.47	88.54	88.64	94.64
5	85.21	88.25	94.51	86.37	88.47	94.71
6	84.3	85.57	93.09	88.47	88.88	95.76
7	83.92	84.61	91.42	90.37	88.38	96.12
8	84.14	86.21	93.87	90.17	91.25	93.91
9	85.79	87.13	92.4	89.94	90.55	94.16
10	82.63	84.91	91.47	90.1	91.86	93.9
Average	84.412	85.759	92.336	89.071	89.745	94.684

As per the observed results, CRNNC-AC classification method scored the highest Specificity Value of 96.12%. The highest Specificity average 94.68% is also secured by CRNNC-AC. The lowest observed value of CRNNC-AC is 93.9% which is also higher than the highest values of other methods.

The comparison graph is plotted with the table values and given in Figure 10.

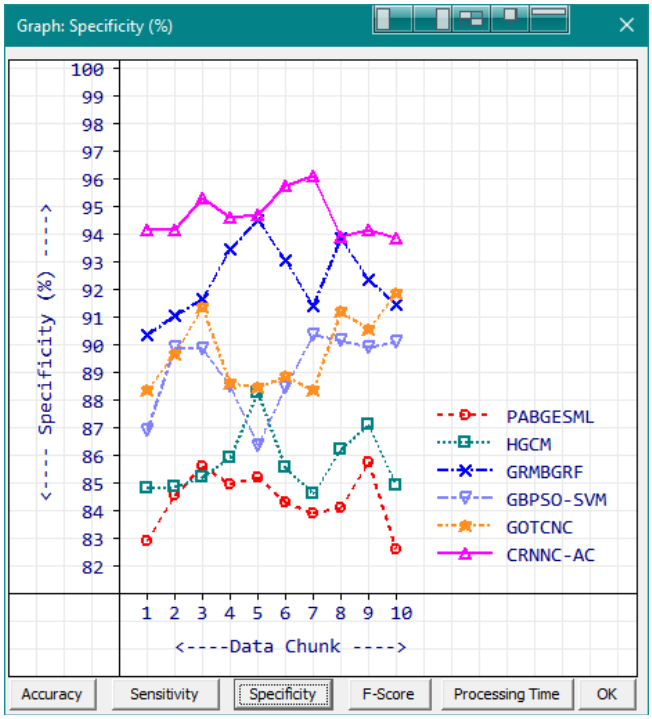

Figure 10. Specificity

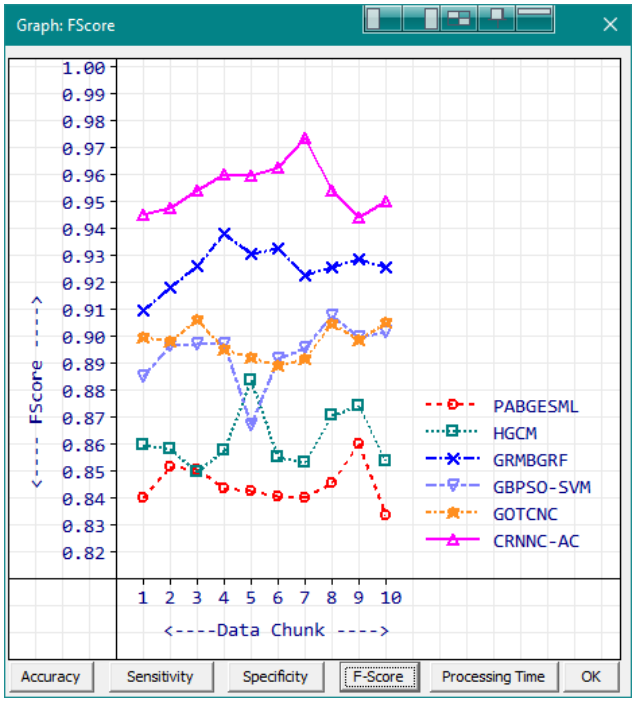

Figure 11: F1-Score

1.15. F1-Score

F1-Score is the hormonic mean of Precision and Sensitivity of a classification algorithm. F1 Score is calculated using the formula $\left(\frac{Recall^{-1}+Precision^{-1}}{2}\right)^{-1}$. The F1-Score for existing and proposed methods are given in Table 4.

Table 4: F1-Score

F1-Score						
Data Chunk	PABGESML	HGCM	GRMBGRF	GBPSO_SVM	GOTCNC	CRNNC_AC
1	0.84	0.86	0.91	0.89	0.9	0.95
2	0.85	0.86	0.92	0.9	0.9	0.95
3	0.85	0.85	0.93	0.9	0.91	0.95
4	0.84	0.86	0.94	0.9	0.9	0.96
5	0.84	0.88	0.93	0.87	0.89	0.96
6	0.84	0.86	0.93	0.89	0.89	0.96
7	0.84	0.85	0.92	0.9	0.89	0.97
8	0.85	0.87	0.93	0.91	0.9	0.95
9	0.86	0.87	0.93	0.9	0.9	0.94
10	0.83	0.85	0.93	0.9	0.91	0.95
Average	0.844	0.861	0.927	0.896	0.899	0.954

While calculating the F1-Score for the methods, CRNNC-AC has the highest value 0.97. The lowest value of CRNNC-AC is 0.94. Therefore, it is realized that the F1-Score of CRNNC-AC is better than any other existing method compared here.

The F1-Score comparison graph is provided below as Figure 11.

1.16. Processing Time

Processing time is one of the important factors in measuring the quality of the classification algorithms. It is inversely proportional to the quality of the algorithm. That is the best classification algorithm should consume the least processing time. A standard measurement with the experimental setup is used to calculate the processing time. The processing time is calculated using the formula $T_p = T_s - T_e$ where T_p is the overall processing time, T_s is the process starting time and T_e is the process ending time. The average processing time is calculated as $\frac{T_P}{\eta}$ where η is the number of records processed. The measured average processing times are given in Table 5.

Table 5: Average Processing Time (mS)

Average Processing Time (ms)						
Data Chunk	PABGESML	HGCM	GRMBGRF	GBPSO_SVM	GOTCNC	CRNNC_AC
1	1832	2380	2670	2144	2469	1372
2	1860	2367	2660	2137	2463	1382
3	1844	2320	2658	2103	2475	1341
4	1840	2318	2625	2150	2448	1329
5	1857	2362	2618	2093	2418	1325
6	1873	2389	2613	2133	2469	1347
7	1833	2372	2664	2097	2422	1337
8	1833	2337	2627	2143	2467	1330
9	1818	2353	2632	2094	2484	1372
10	1874	2323	2669	2165	2434	1373
Average	1846.4	2352.1	2643.6	2125.9	2454.9	1350.8

As per the observed results, proposed CRNNC-AC has the least processing times in its data chunk processing sequence. The average processing time of CRNNC-AC is 1350.8 mS which is lesser than the other methods involved in comparison.

The processing time comparison chart is given as Figure 12 – given below.

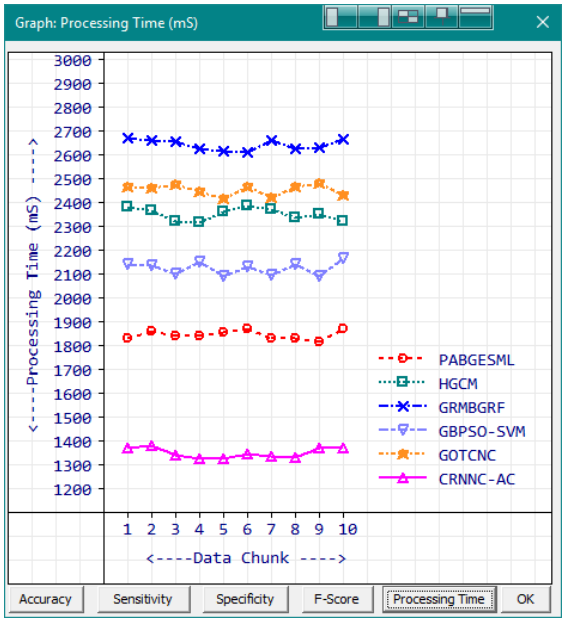

Figure 12. Average Processing Time (mS)

VII. CONCLUSION

Identifying ASD in earlier stages assuring higher probability cures among children. Finding the ASD problems by observing the behavioral patterns in children takes time. A novel method of finding ASD classifications by analyzing the genome sequence of a child is introduced in this work which operates at higher accuracy, specificity and sensitivity. The proposed method uses modern machine learning procedure of RNN to make this technique applicable in real-time model systems with reasonable processing time. By this way the proposed CRNNC-AC can serve millions of children to get rid of their ASD behaviors in the earlier stage itself – which will be a boon to the modern emerging society.

REFERENCES:

1. Hui Liu, Hong-qi Tian, Xi-feng Liang and Yan-fei Li, "Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks" in Applied Energy Volume 157, Elsevier-2015, pp.183-194
2. Santiago Pascual and Antonio Bonafonte, "Multi-output RNN-LSTM for multiple speaker speech synthesis and adaptation" in 2016 24th European Signal Processing Conference, IEEE - 2016, pp. 2325 - 2329
3. Davide Ballabio, Francesca Grisoni and Roberto Todeschini, "Multivariate comparison of classification performance measures" in Chemometrics and Intelligent Laboratory Systems - Volume 174, Elsevier - 2018, pp. 33 - 44
4. A.J.Baxter, T.S.Brugha, H.E.Erskine, R.W.Scheurer, T.Vos and J.G.Scott, "The epidemiology and global burden of autism spectrum disorders" in Psychological Medicine - Volume 45 issue 3, Cambridge - 2015, pp. 601 - 613
5. Deborah L. Christensen et.al., "Prevalence and Characteristics of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2012" in Morbidity and Mortality Weekly Report - Surveillance Summaries, NCBI - 2018, pp. 1 – 39
6. Bruno Falissard, "Early detection of child and adolescent mental disorders: some elements of a necessary debate" in European Child & Adolescent Psychiatry - Volume 15 issue 10, Springer 2016, pp. 1041 – 1043
7. Isabelle Thiffault, Maxime Cadieux-Dion, Emily Farrow, Raymond Caylor, Neil Miller, Sarah Soden and Carol Saunders, "On the verge of diagnosis: Detection, reporting, and investigation of de novo variants in novel genes identified by clinical sequencing" in Human Mutation Variation, Informatics and Disease, Wiley Online Library - 2018, pp. 1505-1516
8. Evie Stergiakouli, George Davey Smith, Joanna Martin, David H. Skuse, Wolfgang Viechtbauer, Susan M. Ring, Angelica Ronald, David E. Evans, Simon E. Fisher, Anita Thapar and Beate St Pourcain, "Shared genetic influences between dimensional ASD and ADHD symptoms during child and adolescent development" in Molecular AutismBrain, Cognition and Behavio, BMC: Springer Nature - 2017, pp. 1 – 13
9. Anne Claire Richard, Anne Rovelet-Lecrux, Elsa Delaby, Camille Charbonnier, Bhooma Thiruvahindrapuram, Eli Hatchwell, Peggy S. Eis, Alexandra Afenjar, Brigitte Gilbert Dussardier, Stephen W. Scherer, Catalina Betancur and Dominique Campion, "The 22q11 PRODH/DGCR6 deletion is frequent in hyperprolinemic subjects but is not a strong risk factor for ASD" in Wiley Online Library - 2016, pp. 1 – 12
10. Rezvan Noroozi, Mohammad Taheri, Abolfazl Movafagh, Soudeh Ghafouri-Fard, Arezou Sayad, Reza Mirfakhraie, Seyed Abdulmajid Ayatollahi, Hidetoshi Inoko, Hanieh Noroozi, Atieh Abedin Do and Amin Abbasi Soureshjani, "Association analysis of the GABRB3 promoter variant and susceptibility to autism spectrum disorder" in Basal Ganglia, Elsevier - 2018, pp.4 - 7
11. Nat Neurosci, "Whole genome sequencing resource identifies 18 new candidate genes for autism spectrum disorder" in Canadian Institutes of Health Research, CIHR - 2017, pp. 1 - 29
12. Diogo Pratas, Armando J. Pinho and Paulo J. S. G. Ferreira, "Efficient Compression of Genomic Sequences" in 2016 Data Compression Conference (DCC), IEEE - 2016, pp. 231 - 240
13. Sorina Maciuca, Carlos del Ojo Elias, Gil McVean and Zamin Iqbal, "A Natural Encoding of Genetic Variation in a Burrows-Wheeler Transform to Enable Mapping and Genome Inference" in International Workshop on Algorithms in Bioinformatics, Springer - 2016, pp. 222 – 233
14. Dong Hoon Oh, Il Bin Kim, Seok Hyeon Kim and Dong Hyun Ahn, "Predicting Autism Spectrum Disorder Using Blood-based Gene Expression Signatures and Machine Learning" in Clinical Psychopharmacology and Neuroscience, PMC-2017, pp. 47 - 52
15. Matt Spencer, Nicole Takahashi, Sounak Chakraborty, Judith Miles and Chi-Ren Shyu, "Heritable genotype contrast mining reveals novel gene associations specific to autism subgroups" in Journal of Biomedical Informatics, Elsevier 2018, pp. 50 - 61
16. Li-Chung Chuang and Po-Hsiu Kuo, "Building a genetic risk model for bipolar disorder from genome-wide association data with random forest algorithm" in Scientific Reports volume 7 - Article number: 39943, Scientific Reports - 2017, pp. 1 - 10
17. Shilan S. Hameed, Rohayanti Hassan and Fahmi F. Muhammad, "Selection and classification of gene expression in autism disorder: Use of a combination of statistical filters and a GBPSO-SVM algorithm" in Plos One - 2017, pp. 1 - 25
18. Thomas P Quinn, Samuel C Lee, Svetha Venkatesh and Thin Nguyen, "Improving the classification of neuropsychiatric conditions using gene ontology terms as features" in BioRxiv - The preprint server for Biology, Cold Spring Harbor Laboratory - 2018, pp. 1 – 25
19. Lichao Mou, Pedram Ghamisi and Xiao Xiang Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification" in IEEE Transactions on Geoscience and Remote Sensing - Volume: 55 - Issue: 7, IEEE - 2017, pp. 3639 – 3655
20. https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/variable.cgi?study_id=phs000267.v5.p2&phv=161300
21. https://www.ncbi.nlm.nih.gov/IEB/ToolBox/SDKDOCS/CORELIB.HTML#_Introduction
22. Sven Amann, Sebastian Proksch, Sarah Nadi and Mira Mezini, A Study of Visual Studio Usage in Practice" in IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), IEEE - 2016, pp. 1 – 11
23. Kostadin Damevski, David C. Shepherd, Johannes Schneider and Lori Polloc, "Mining Sequences of Developer Interactions in Visual Studio for Usage Smells" in IEEE Transactions on Software Engineering - Volume: 43 - Issue: 4, IEEE - 2017, pp. 359-371

AUTHORS PROFILE

S.Padmapriya completed her M.Sc Computer Science in Nehru Memorial College, Puthanampatti in 2001 and her M.Phil degree in Bharathidasan University in 2014. She completed M.Tech(IT) in Bharathidasan university in 2014.She also passed State Level Eligibility Test(SET) for Lectureship in 2012.She is pursuing her Ph.D in Bharathidasan University. She is having 18 years of teaching experience in various institutions and presently working as Assistant Professor in SRM Trichy Arts & Science College, Trichy. Her research areas of interest are Data Mining, Machine Learning and Neural Network and also published many research articles in National and International journals.

S.Murugan received his M.Sc degree in Applied Mathematics from Anna University in 1984 and M.Phil degree in Computer Science from National Institute of Technology formerly known as Regional Engineering College, Trichirappalli in 1994. He is associated with the department of Computer, Nehru Memorial College (Autonomous), affiliated to Bharathidasan University since 1986 where he is currently working as an Associate Professor. He has 32 years of teaching experience in the field of Computer Science. He has completed his Ph.D degree in Computer Science from Bharathiyar University in 2015 with Data Mining specialization. His research interest includes Data and Web Mining and also published many research articles in the National and International journals.