**EEG Based Early Detection of Autism by Using SVM Classifier**

**ABSTRACT**

Autism spectrum disorder (ASD) is a developmental disability caused by differences in the brain. Some people with ASD have a known difference, such as a genetic condition. Other causes are not yet known. Scientists believe there are multiple causes of ASD that act together to change the most common ways people develop. Electroencephalography (EEG) signals are captured for analyzing brain and its functions. EEG signals have different features such as, alpha, beta, gamma, delta and theta etc. which are extracted from the frequency domain of the signal. Those frequency variations can help identify the autism in children. In this paper, we have implemented the EEG based autism detection based on features of EEG signals, later those features are used to train a Machine Learning Classifier known as Support Vector Machine (SVM). The classifier is used to validate the data whether the EEG signal have autism or not. The Classifier is later tested for its accuracy which resulted in more than 90%.

**Keywords:** Autism spectrum disorder (ASD), Electroencephalography (EEG), Machine Learning Classifier, Support Vector Machine (SVM).

**INTRODUCTION**

**Autism spectrum disorder**

A serious developmental disorder that impairs the ability to communicate and interact. Autism spectrum disorder impacts the nervous system and affects the overall cognitive, emotional, social and physical health of the affected individual. The range and severity of symptoms can vary widely. Common symptoms include difficulty with communication, difficulty with social interactions, obsessive interests and repetitive behaviors. Early recognition, as well as behavioral, educational and family therapies may reduce symptoms and support development and learning.

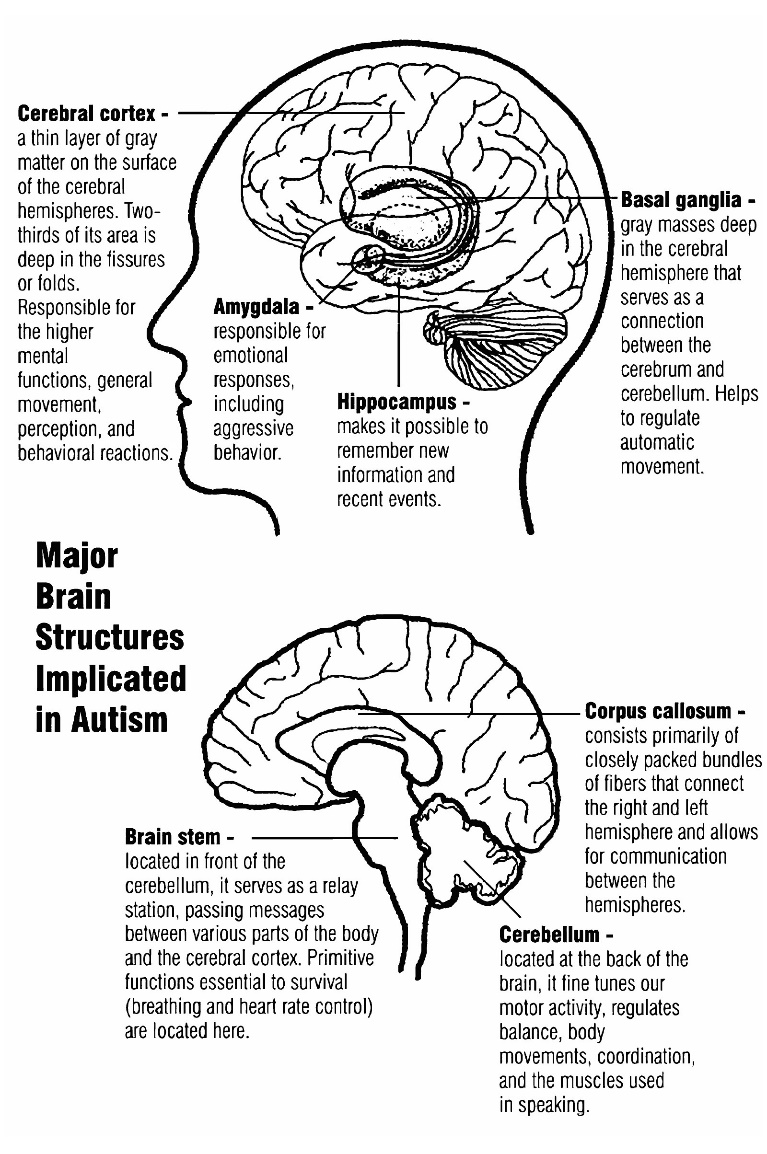
Autism, formally called Autism Spectrum Disorder (ASD) and Autism Spectrum Condition (ASC), and sometimes called the autism spectrum, is a neurodevelopmental disorder characterized by deficits in social communication and social interaction, and repetitive or restricted patterns of behaviors, interests, or activities, which can include hyper- and hyporeactivity to sensory input.

Autism is a spectrum disorder, meaning that it can manifest very differently in each person. Because of this, there is wide variation in the support needs of autistic people. For example, some are nonspeaking, while others have proficient spoken language. Psychiatry has traditionally classified autism as a neurodevelopmental disorder, but the autism rights movement and some researchers see autism as part of neurodiversity, the natural diversity in human thinking and experience, with strengths, differences, and weaknesses. Autistic people still have a disability, but need to be accommodated, rather than cured. This view of the condition has led to significant controversy among those who are autistic alongside advocates, practitioners, and charities.

There are many theories about what causes autism; it is highly heritable and believed to be mainly genetic, but many genes are involved, and environmental factors may also be relevant. The syndrome frequently co-occurs with other conditions, including attention deficit hyperactivity disorder, epilepsy, and intellectual disability. Disagreements continue about questions such as what should be included as part of the diagnosis, whether there are meaningful subtypes of autism, and the significance of autism-associated traits in the wider population. The combination of broader criteria and increased awareness has led to a trend of steadily increasing estimates of autism prevalence, causing a misconception that there is an autism epidemic and perpetuating the myth that it is caused by vaccines.

**Causes of Autism**

Many causes of autism have been proposed, but understanding of the theory of causation of autism, or otherwise known as autism spectrum disorder (ASD) is incomplete. Attempts have been made to incorporate the known genetic and environmental causes into a comprehensive causative framework. ASD is a complex developmental condition marked by persistent challenges to social interaction, speech and nonverbal communication, and restricted/repetitive behaviors and its phenotypes vary significantly.



**Fig: Showing the illustration of Autism in Brain**

Research indicates that genetic factors predominate. The heritability of autism, however, is complex, and it is typically unclear which genes are involved. In rare cases, autism is associated with agents that cause birth defects. Many other causes have been proposed. Numerous epidemiological studies have shown no scientific evidence supporting the controversial but popular theory that autism is caused by vaccines.

Autism involves atypical brain development which often becomes apparent in behavior and social development before a child is three years old. It can be characterized by impairments in social interaction and communication, as well as restricted interests and stereotyped behavior, and the characterization is independent of any underlying neurological defects. Other characteristics include repetitive-like tasks seen in behavior and sensory interests. This article uses the terms autism and ASD to denote classical autism and the wider dispersion of symptoms and manifestations of autism, respectively.

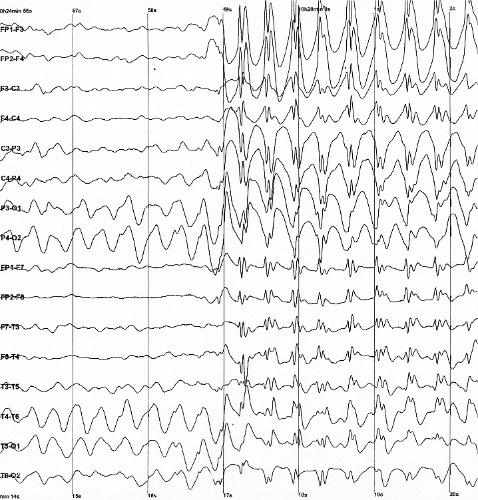
Genetic factors may be the most significant cause of autism. Early studies of twins had estimated heritability to be over 90%, meaning that genetics explains over 90% of whether a child will develop autism. This may be an overestimation, as later twin studies estimate the heritability at between 60 and 90%. Evidence so far still suggests a strong genetic component, with one of the largest and most recent studies estimating the heritability at 83%. Many of the non-autistic co-twins had learning or social disabilities. For adult siblings the risk for having one or more features of the broader autism phenotype might be as high as 30%.

In spite of the strong heritability, most cases of ASD occur sporadically with no recent evidence of family history. It has been hypothesized that spontaneous de novo mutations in the father's sperm or mother's egg contribute to the likelihood of developing autism. There are two lines of evidence that support this hypothesis. First, individuals with autism have significantly reduced fecundity, they are 20 times less likely to have children than average, thus curtailing the persistence of mutations in ASD genes over multiple generations in a family. Second, the likelihood of having a child develop autism increases with advancing paternal age and mutations in sperm gradually accumulate throughout a man's life.

**Electroencephalography (EEG)**

Electroencephalography (EEG) is a method to record an electrogram of the spontaneous electrical activity of the brain. The bio-signals detected by EEG have been shown to represent the postsynaptic potentials of pyramidal neurons in the neocortex and allocortex. It is typically non-invasive, with the EEG electrodes placed along the scalp (commonly called "scalp EEG") using the International 10-20 system, or variations of it. Electrocorticography, involving surgical placement of electrodes, is sometimes called "intracranial EEG". Clinical interpretation of EEG recordings is most often performed by visual inspection of the tracing or quantitative EEG analysis.

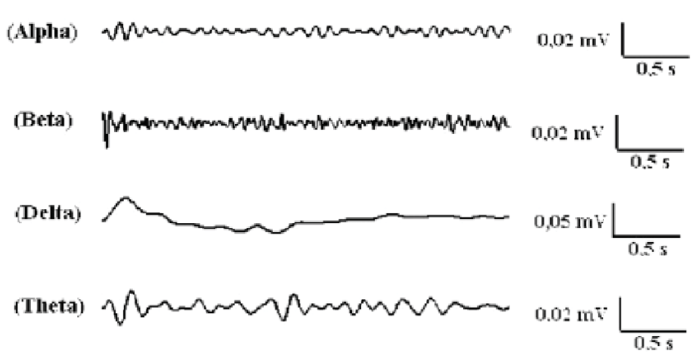
Voltage fluctuations measured by the EEG bio-amplifier and electrodes allow the evaluation of normal brain activity. As the electrical activity monitored by EEG originates in neurons in the underlying brain tissue, the recordings made by the electrodes on the surface of the scalp vary in accordance with their orientation and distance to the source of the activity. Furthermore, the value recorded is distorted by intermediary tissues and bone which act in a manner akin to resistors and capacitors in an electrical circuit. This means not all neurons will contribute equally to an EEG signal with an EEG predominately reflecting the activity of cortical neurons near to the electrodes on the scalp. Deep structures within the brain further away from the electrodes will not contribute directly to an EEG, these include the base of the cortical gyrus, mesial walls of the major lobes, hippocampus, thalamus and brain stem.



**Fig: Showing of an EEG Signal**

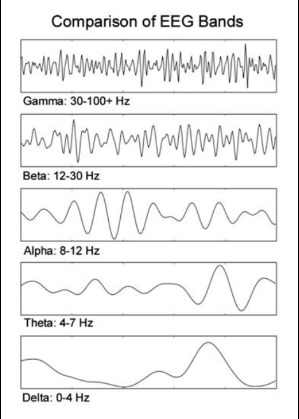
**Features of an EEG Signal**

A healthy human EEG will show certain patterns of activity which correlate with how awake a person is. The range of frequencies one observes are between 1 and 30 Hz and amplitudes will vary between 20 and 100 μV. The observed frequencies are subdivided in various groups, these are alpha (8–13 Hz), beta (13–30 Hz), delta (0.5–4 Hz) and theta (4–7 Hz). Alpha waves are observed when a person is in a state of relaxed wakefulness and are mostly prominent over parietal and occipital sites. During intense mental activity beta waves are more prominent in frontal areas as well as other regions. If a relaxed person is told to open their eyes one observes alpha activity decreasing and an increase in beta activity. Theta and delta waves are not seen in wakefulness and if they are it is a sign of brain dysfunction.



**Fig: Showing Features of an EEG Signal**

EEG can detect abnormal electrical discharges such as sharp waves, spikes or spike-and-wave complexes that are seen in people with epilepsy, thus it is often used to inform the medical diagnosis. EEG can detect the onset and spatio-temporal (location and time) evolution of seizures and the presence of status epilepticus. It is also used to help diagnose sleep disorders, depth of anesthesia, coma, encephalopathies, cerebral hypoxia after cardiac arrest, and brain death. EEG used to be a first-line method of diagnosis for tumors, stroke and other focal brain disorders,[3][4] but this use has decreased with the advent of high-resolution anatomical imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT). Despite limited spatial resolution, EEG continues to be a valuable tool for research and diagnosis. It is one of the few mobile techniques available and offers millisecond-range temporal resolution which is not possible with CT, PET or MRI.



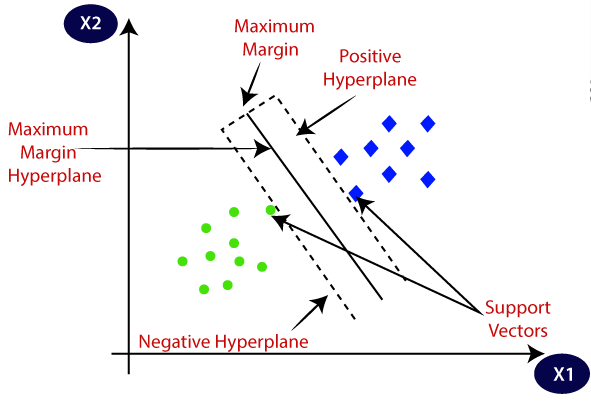
**Fig: Showing different Bands with their Frequency Ranges**

Derivatives of the EEG technique include evoked potentials (EP), which involves averaging the EEG activity time-locked to the presentation of a stimulus of some sort (visual, somatosensory, or auditory). Event-related potentials (ERPs) refer to averaged EEG responses that are time-locked to more complex processing of stimuli; this technique is used in cognitive science, cognitive psychology, and psychophysiological research.

**Support Vector Machine Classifier**

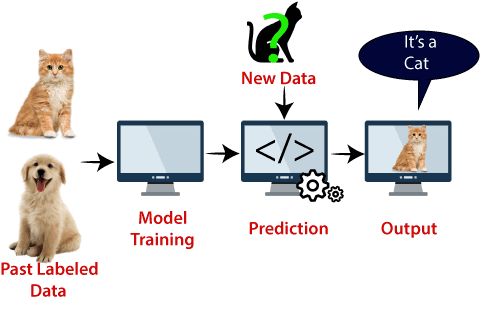
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**Fig: Showing an illustration of SVM Classifier with Hyperplane**

**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



**Fig: Showing Classification using SVM Classifier**

**Types of SVM**

SVM can be of two types:

**Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

**Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

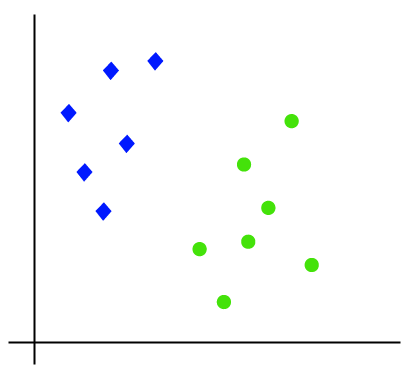
**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane. We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

**Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

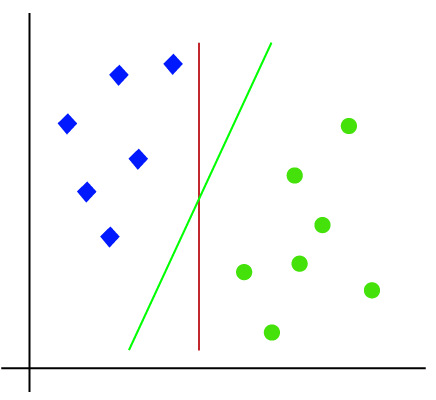
**Linear SVM:**

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



**Fig: Linear Data**

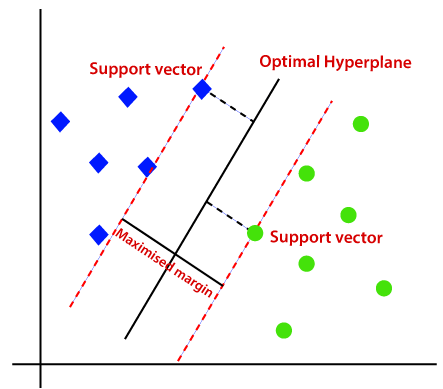
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



**Fig: Showing Hyperplanes to classify two classes**

Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

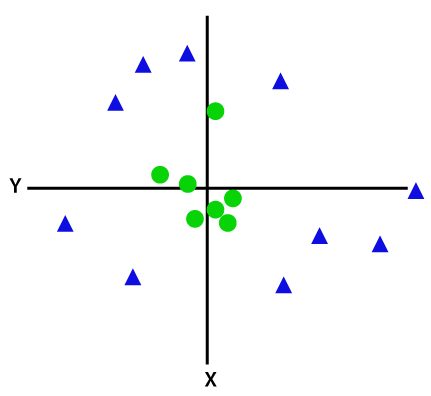
Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.



**Fig: Showing Maximized Margin**

**Non-Linear SVM:**

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

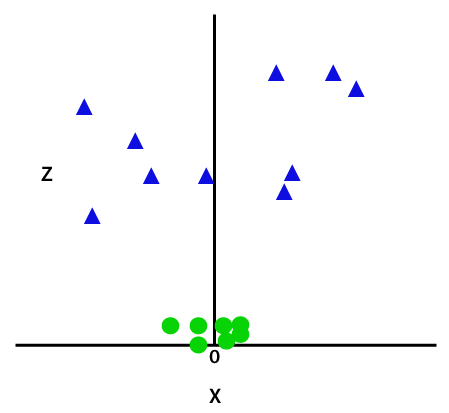


**Fig: Showing a Non-Linear Data**

So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

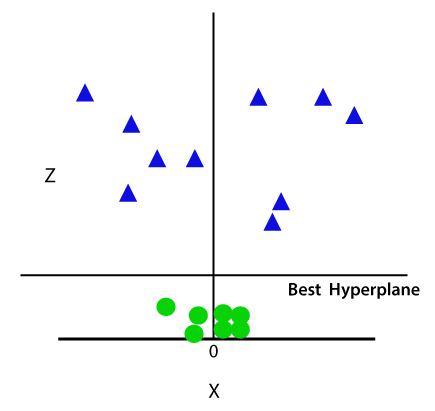
z=x2 +y2

By adding the third dimension, the sample space will become as below image:



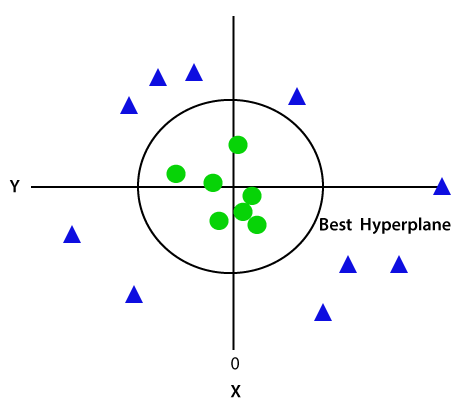
**Fig: Showing the third dimension**

So now, SVM will divide the datasets into classes in the following way. Consider the below image:



**Fig: Showing the Best Hyperplane**

Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



**Fig: Showing the Hyperplane in a Circle**

**LITERATURE REVIEW**

**[1] Thabtah, Fadi, Firuz Kamalov, and Khairan Rajab. (2018):** Autism Spectrum Disorder (ASD) is one of the fastest growing developmental disability diagnosis. General practitioners (GPs) and family physicians are typically the first point of contact for patients or family members concerned with ASD traits observed in themselves or their family member. Unfortunately, some families and adult patients are unaware of ASD traits that may be exhibited and as a result do not seek out necessary diagnostic services or contact their GP. Therefore, providing a quick, accessible, and simple tool utilizing items related to ASD to these families may increase the likelihood they will seek professional assessment and is vital to the early detection and treatment of ASD. This study aims at identifying fewer, albeit influential, features in common ASD screening methods in order to achieve efficient screening as demands on evaluating the items' influences on ASD within existing tools is urgent. To achieve this aim, a computational intelligence method called Variable Analysis (Va) is proposed that considers feature-to-class correlations and reduces feature-to-feature correlations. The results of the Va have been verified using two machine learning algorithms by deriving automated classification systems with respect to specificity, sensitivity, positive predictive values (PPVs), negative predictive values (NPVs), and predictive accuracy. Experimental results using cases and controls related to items in three common screening methods, along with features related to individuals, have been analyzed and compared with results obtained from other common filtering methods. The results exhibited that Va was able to derive fewer numbers of features from adult, adolescent, and child screening methods yet maintained competitive predictive accuracy, sensitivity, and specificity rates.

**Summary:** Analyzed about the new computational intelligence approach to detect autistic features for autism screening.

**[2] Constantino, John N., Patricia D. Lavesser, Y. I. Zhang, Anna M. Abbacchi, Teddi Gray, and Richard D. Todd. (2007):** Teachers routinely observe children in the naturalistic social contexts of their classrooms and provide extremely important input in the evaluation of numerous psychiatric syndromes. Their precision in ascertaining and quantifying autistic symptomatology has not previously been established. In this study, we compared teachers' ratings of autistic symptomatology with those derived from parents, expert clinicians, and trained raters. A total of 577 subjects (ages 4-18 years) with (n = 406) and without (n = 171) pervasive developmental disorders (PDDs) were assessed by one parent and one current teacher using the Social Responsiveness Scale, a quantitative measure of autistic traits. PDD subjects were assessed by expert clinicians, the Autism Diagnostic Interview-Revised, and/or the Autism Diagnostic Observation Schedule. All of the assessments were conducted during the period 1996-2006.

**Summary:** Rapid quantitative assessments by teachers and parents constitute a cost-effective method for measuring and tracking the severity of autistic symptomatology in both educational and clinical settings.

**[3] Daniel Bone, Matthew S. Goodwin, Matthew P. Black, Chi-Chun Lee, Kartik Audhkhasi, and Shrikanth Narayanan. (2015):** Machine learning has immense potential to enhance diagnostic and intervention research in the behavioral sciences, and may be especially useful in investigations involving the highly prevalent and heterogeneous syndrome of autism spectrum disorder. However, use of machine learning in the absence of clinical domain expertise can be tenuous and lead to misinformed conclusions. To illustrate this concern, the current paper critically evaluates and attempts to reproduce results from two studies (Wall et al. in Transl Psychiatry 2(4):e100, 2012a; PloS One 7(8), 2012b) that claim to drastically reduce time to diagnose autism using machine learning. Our failure to generate comparable findings to those reported by Wall and colleagues using larger and more balanced data underscores several conceptual and methodological problems associated with these studies. We conclude with proposed best-practices when using machine learning in autism research, and highlight some especially promising areas for collaborative work at the intersection of computational and behavioral science.

**Summary:** Analyzed about the application of machine learning to facilitate autism diagnostics pitfalls and promises.

**[4] Dennis Paul Wall, J. Kosmicki, T. F. Deluca, E. Harstad, and Vincent Alfred Fusaro. (2012):** The Autism Diagnostic Observation Schedule-Generic (ADOS) is one of the most widely used instruments for behavioral evaluation of autism spectrum disorders. It is composed of four modules, each tailored for a specific group of individuals based on their language and developmental level. On average, a module takes between 30 and 60 min to deliver. We used a series of machine-learning algorithms to study the complete set of scores from Module 1 of the ADOS available at the Autism Genetic Resource Exchange (AGRE) for 612 individuals with a classification of autism and 15 non-spectrum individuals from both AGRE and the Boston Autism Consortium (AC). Our analysis indicated that 8 of the 29 items contained in Module 1 of the ADOS were sufficient to classify autism with 100% accuracy. We further validated the accuracy of this eight-item classifier against complete sets of scores from two independent sources, a collection of 110 individuals with autism from AC and a collection of 336 individuals with autism from the Simons Foundation. In both cases, our classifier performed with nearly 100% sensitivity, correctly classifying all but two of the individuals from these two resources with a diagnosis of autism, and with 94% specificity on a collection of observed and simulated non-spectrum controls. The classifier contained several elements found in the ADOS algorithm, demonstrating high test validity, and also resulted in a quantitative score that measures classification confidence and extremeness of the phenotype. With incidence rates rising, the ability to classify autism effectively and quickly requires careful design of assessment and diagnostic tools. Given the brevity, accuracy and quantitative nature of the classifier, results from this study may prove valuable in the development of mobile tools for preliminary evaluation and clinical prioritization—in particular those focused on assessment of short home videos of children—that speed the pace of initial evaluation and broaden the reach to a significantly larger percentage of the population at risk.

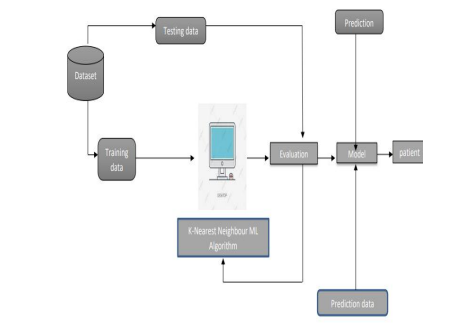
**Summary:** Analyzed about the uses of machine learning to shorten observation based screening and diagnosis of autism.

**[5] Dennis P. Wall, Rebecca Dally, Rhiannon Luyster, Jae-Yoon Jung, and Todd F. DeLuca. (2012):** The Autism Diagnostic Interview-Revised (ADI-R) is one of the most commonly used instruments for assisting in the behavioral diagnosis of autism. The exam consists of 93 questions that must be answered by a care provider within a focused session that often spans 2.5 hours. We used machine learning techniques to study the complete sets of answers to the ADI-R available at the Autism Genetic Research Exchange (AGRE) for 891 individuals diagnosed with autism and 75 individuals who did not meet the criteria for an autism diagnosis. Our analysis showed that 7 of the 93 items contained in the ADI-R were sufficient to classify autism with 99.9% statistical accuracy. We further tested the accuracy of this 7-question classifier against complete sets of answers from two independent sources, a collection of 1654 individuals with autism from the Simons Foundation and a collection of 322 individuals with autism from the Boston Autism Consortium. In both cases, our classifier performed with nearly 100% statistical accuracy, properly categorizing all but one of the individuals from these two resources who previously had been diagnosed with autism through the standard ADI-R. Our ability to measure specificity was limited by the small numbers of non-spectrum cases in the research data used, however, both real and simulated data demonstrated a range in specificity from 99% to 93.8%. With incidence rates rising, the capacity to diagnose autism quickly and effectively requires careful design of behavioral assessment methods. Ours is an initial attempt to retrospectively analyze large data repositories to derive an accurate, but significantly abbreviated approach that may be used for rapid detection and clinical prioritization of individuals likely to have an autism spectrum disorder. Such a tool could assist in streamlining the clinical diagnostic process overall, leading to faster screening and earlier treatment of individuals with autism.

**Summary:** Analyzed about the uses of artificial intelligence to shorten the behavioral diagnosis of autism.

**EXISTING METHOD**

Autism spectrum disorder (ASD) is a psychiatric disorder which leads to neurological and developmental growth of a person which starts in early age and gets carried throughout their life. It is a condition associated with significant healthcare costs and early diagnosis can reduce these. Unfortunately, waiting time is lengthy for an ASD diagnosis and it is cost effective. Due to the increase in economy for autism prediction and the increase in the number of ASD cases across the world is in need of easily implemented and effective screening methods by GUI results. To overcome the time complexity for identifying the disorder advanced technologies can be used such as machine learning algorithms to improve precision, accuracy and quality of the diagnosis process. Machine learning helps us by providing intelligent techniques to discover the affected patient, which can be utilized in prediction and to improve decision making. And hence, we propose the data set features related to autism screening of adult and child to be used for further analysis and to improve the classification of ASD cases.

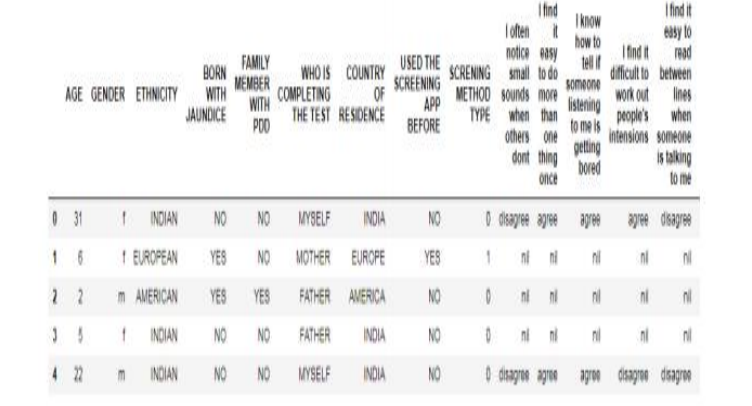


**Fig: Block Diagram of the Existing Method**

**Data Validation and Pre-Processing Techniques**

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters.

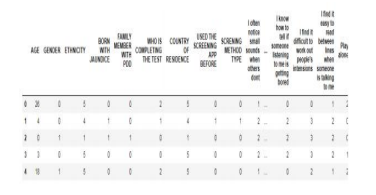
Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. (For example to show the data type format of given dataset) Data Validation/ Cleaning/Preparing Process: Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give 48 an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-vitiate and multivariate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.



**Fig: Data Frame for ASD**

**Data Pre-processing**

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format; for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set.



**Fig: Pre-Processed Data frame**

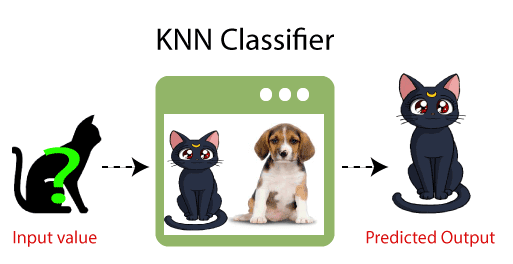
Peak amplitude is a measurement based the instantaneous level of a signal. A different approach would be to analyse the overall amplitude across the signal. Next, let’s look at the common measurement used by audio engineers for this, called the RMS level.

**Comparison of Machine Learning**

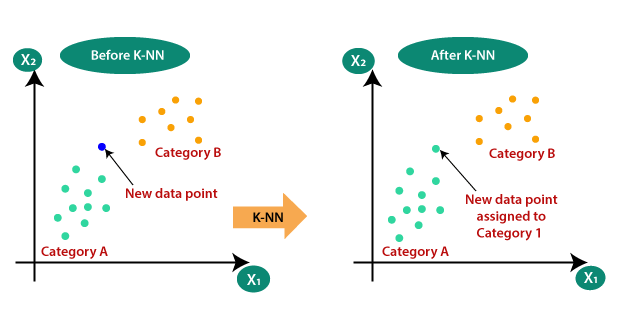
Random Forest Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees‟ habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name Random Forest. The random forest algorithm can be used for both regression and classification tasks.

**K-Nearest Neighbour (KNN)**

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



**Fig: Classification through KNN through Nearest Neighbors**

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

K-Nearest Neighbour is a supervised machine learning algorithm which stores all instance correspond to training data points in n-dimensional space. When an unknown discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors) and returns the most common class as the prediction and for real-valued data it returns the mean of k nearest neighbors. In the distance-weighted nearest neighbor algorithm, it weights the contribution of each of the k neighbors according to their distance using the following query giving greater weight to the closest neighbors. Usually KNN is robust to noisy data since it is averaging the k-nearest neighbors. The k-nearest 1634eighbours algorithm is a classification algorithm, and it is supervised: it takes a bunch of 1634eighbou points and uses them to learn how to label other points. To label a new point, it looks at the 1634eighbou points closest to that new point (those are its nearest 1634eighbours), and has those 1634eighbours vote, so whichever label the most of the 1634eighbours have is the label for the new point (the “k” is the number of 1634eighbours it checks). Makes predictions about the validation set using the entire training set. KNN makes a prediction about a new instance by searching through the entire set to find the k “closest” instances. “Closeness” is determined using a proximity measurement (Euclidean) across all features.

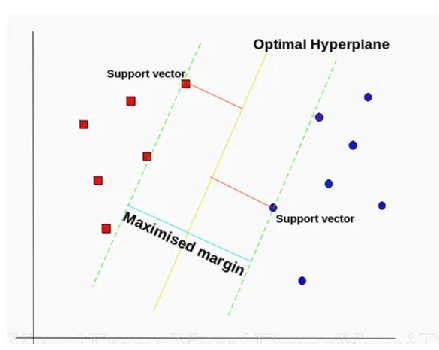
Machine learning are powerful tools for understanding data available in the given data set and to interpret the dataset, but there will be few miss-predictions as these are not completely accurate. Our goal with this study was to apply supervised machine learning algorithm such as KNN (84% accuracy) to a dataset derive from Autism Spectrum Disorder. The Software Development is Flexible and also User-Friendly. Further Functionalities can be added to it in the future.

**DISADVANTAGES:**

* Always needs to determine the value of Kernel which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

**PROPOSED METHOD**

The Support Vector Machine (SVM) Classification is similar to the SVR that I had explained in my previous story. In SVM, the line that is used to separate the classes is referred to as hyperplane. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors which is used to plot the boundary line.

****

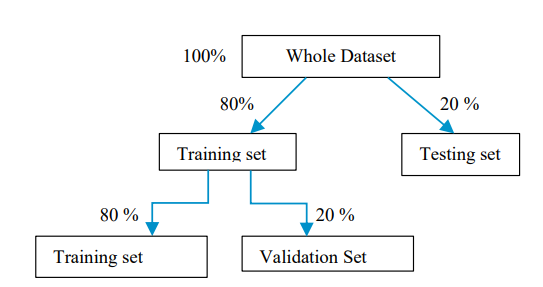
**Fig: Showing an SVM Classifier**

**Data Pre-Processing**

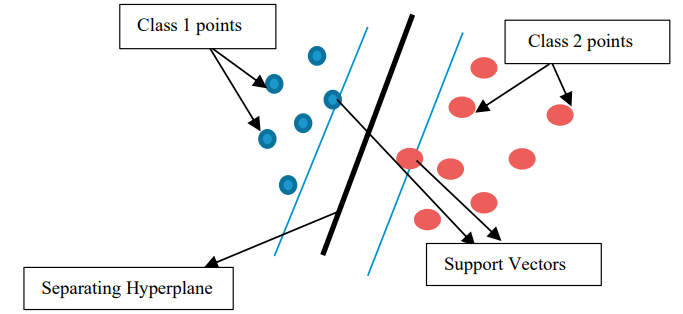
Data pre-processing is a technique in which transform the raw data into a meaningful and understandable format. Real-world data is commonly incomplete and inconsistent because it contains lots of errors and null values. A good pre-processed data always yields to a good result. Various Data pre-processing methods are used to handle incomplete and inconsistent data like as handling missing values, outlier detection, data discretization, data reduction (dimension and numerosity reduction), etc. The problems of missing values in these dataset has been handled by imputation method.

**Training and Testing Model**

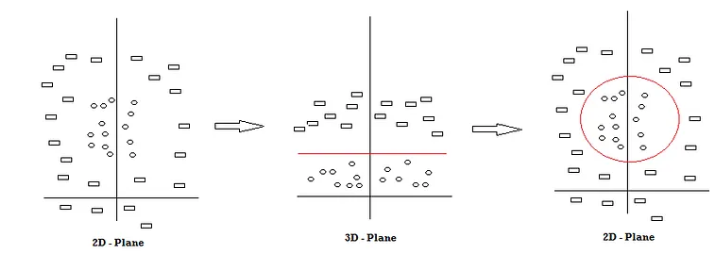
The whole dataset has been split into two parts i.e. one part is training the dataset and the other one is testing dataset with a ratio of 80:20 respectively. For cross-validation purposes again training data has been split into two parts. One part is the training dataset and another part is the validation dataset into an 80:20 ratio respectively. Figure shows the final training, testing and validation sets on which classification has been performed.



**Fig: Showing the Flow of the Proposed Method**



**Fig: Showing Classification using SVM’s Hyperplane**



**Fig: Showing Classification using SVM’s Hyperplane for different planes**

**Hyperplane**

A hyperplane is a decision boundary which separates between given set of data points having different class labels. The SVM classifier separates data points using a hyperplane with the maximum amount of margin. This hyperplane is known as the maximum margin hyperplane and the linear classifier it defines is known as the maximum margin classifier.

**Support Vectors**

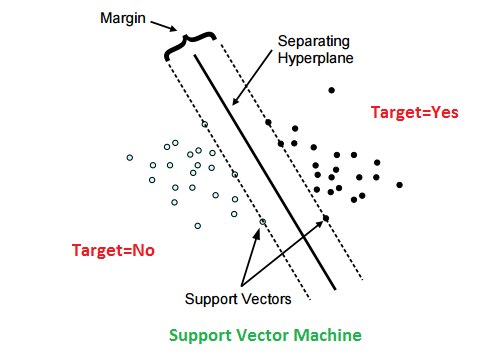
Support vectors are the sample data points, which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.

**Margin**

A margin is a separation gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the line to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get maximum margin.

The following diagram illustrates these concepts visually.

**Margin in SVM**



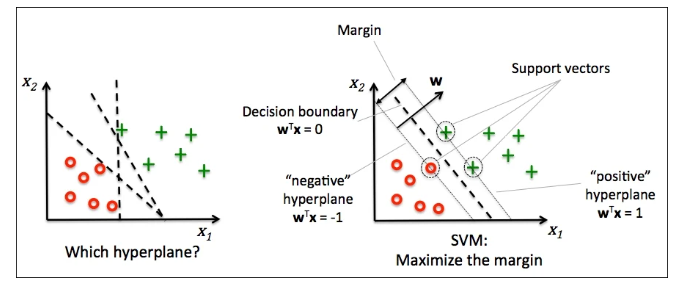
**SVM Under the hood**

In SVMs, our main objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum margin hyperplane in the following 2 step process –

Generate hyperplanes which segregates the classes in the best possible way. There are many hyperplanes that might classify the data. We should look for the best hyperplane that represents the largest separation, or margin, between the two classes.

So, we choose the hyperplane so that distance from it to the support vectors on each side is maximized. If such a hyperplane exists, it is known as the maximum margin hyperplane and the linear classifier it defines is known as a maximum margin classifier.

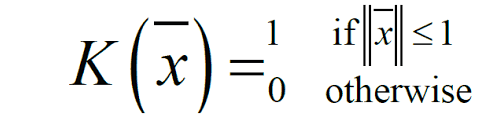
The following diagram illustrates the concept of maximum margin and maximum margin hyperplane in a clear manner

****

**Fig: Showing Maximize Margin**

**Kernel trick**

In practice, SVM algorithm is implemented using a kernel. It uses a technique called the kernel trick. In simple words, a kernel is just a function that maps the data to a higher dimension where data is separable. A kernel transforms a low-dimensional input data space into a higher dimensional space. So, it converts non-linear separable problems to linear separable problems by adding more dimensions to it. Thus, the kernel trick helps us to build a more accurate classifier. Hence, it is useful in non-linear separation problems.



**Fig: Kernel Function of SVM**

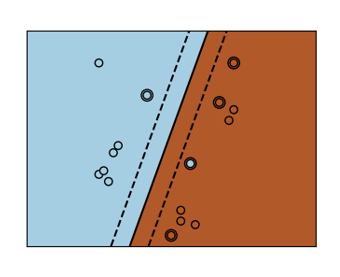
In the context of SVMs, there are 4 popular kernels – Linear kernel, Polynomial kernel, Radial Basis Function (RBF) kernel (also called Gaussian kernel) and Sigmoid kernel.

**Linear kernel**

In linear kernel, the kernel function takes the form of a linear function as follows-

linear kernel : K(xi , xj ) = xiT xj

Linear kernel is used when the data is linearly separable. It means that data can be separated using a single line. It is one of the most common kernels to be used. It is mostly used when there are large number of features in a dataset. Linear kernel is often used for text classification purposes.



**Fig: Linear Kernel**

Training with a linear kernel is usually faster, because we only need to optimize the C regularization parameter. When training with other kernels, we also need to optimize the γ parameter. So, performing a grid search will usually take more time.

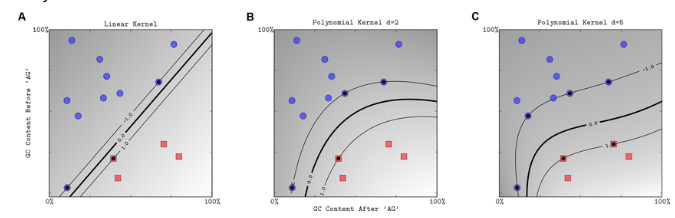
**Polynomial Kernel**

Polynomial kernel represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables. The polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of the input samples.

For degree-d polynomials, the polynomial kernel is defined as follows –

Polynomial kernel : K(xi , xj ) = (γxiT xj + r)d , γ > 0

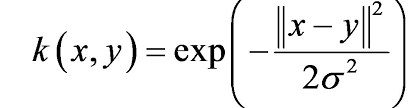
Polynomial kernel is very popular in Natural Language Processing. The most common degree is d = 2 (quadratic), since larger degrees tend to overfit on NLP problems.

****

**Fig: Polynomial Kernel**

**Radial Basis Function Kernel**

Radial basis function kernel is a general purpose kernel. It is used when we have no prior knowledge about the data. The RBF kernel on two samples x and y is defined by the following equation.

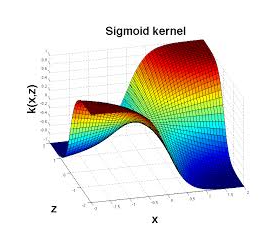


**Fig: Equation of Radial Basis Kernel**

**Sigmoid kernel**

Sigmoid kernel has its origin in neural networks. We can use it as the proxy for neural networks. Sigmoid kernel is given by the following equation.

sigmoid kernel : k (x, y) = tanh(αxTy + c).



**Fig: Sigmoid Kernel**

**Method**

The EEG signals that contains the both autism as well as not autism are taken from the dataset. The dataset is sent for feature extraction process where the different features of an EEG signal are extracted using instantaneous frequencies. The EEG signal which is stored in the dataset will by default be in time domain, later we need to convert it to frequency domain by the application of Fourier Transform (FT). The frequency domain signal will then be sent to the feature extraction process where the different features such as, Alpha, Beta and Theta will be extracted. The features represent the variations in the patient’s brain while capturing the signal which indicates the different emotions and stress patient has undergone while capturing the signal.

The extracted features are then be used to identify a signal with or without autism. The patient’s EEG signal who are suffering from the autism will have an excited levels of all alpha, beta and theta. The values of their frequency range will cross the normal non-autism person. The typical range of beta lies below 20Hz for a non-autism person but, it will cross that value for an autism person. The extracted features are then sent to the classifier for classification.

**Feature Extraction**

Feature extraction decides the rate of classification of the classifier, the more better the features are the more accurate the results or predictions are. The feature extraction method, we’re implementing in this implementation is extraction of instantaneous frequencies. Instantaneous frequencies contains the frequencies of the wavelets of the signal. The frequencies are used for extracting the features such as, alpha, beta and theta etc. which are used for estimating autism of a person. The features are several types depending upon the range of frequencies, but, here, in this process, we too extracted them on the basis of frequencies as well.

The extracted features will be having both the autism based not autism based signals. The features which are extracted on the basis of frequencies are to be stored in a single sheet for a better reference. The one main thing that needs to be implemented for the features before sent for training of the classifier is labelling. Labelling of the features of the data needs to be done because the classifier we’re implementing is a supervised learning classifier. Supervised learning means the classifier will learns on the basis of the labelled data, the learning of classifier needs a supervision under which it will get trained. The labelling for features needs to be done ourselves. So, we need to have the knowledge for differentiating between the two classes of data. We need to label the data in such a way, to make the classifier learns better.

The labelling of the data will be done on the basis of the frequencies we produced in the feature extraction process. The labelling actually refers to the naming of the data on the basis of its state. The state of the features of the data for a not autism based person will be in the normal frequency ranges for all the three features that we extracted, alpha (< 8Hz), beta (8-13Hz) and theta (13-30Hz), so, we label them as ‘Not Autism’, signals. The state of the features of the data for an autism based person will be generally in an excited than the normal frequency ranges for all the three features that we extracted, alpha (8-11Hz), beta (20-25Hz) and theta (>30Hz), so, we label them as ‘Autism’, signals.

Finally, the data will be arranged in such a way, that, the classifier can be easily understand differences between the Autism and Not Autism signals. Both, the features as well as labels are can be fed to the next stage i.e. the training of the classifier for training.

**Classification**

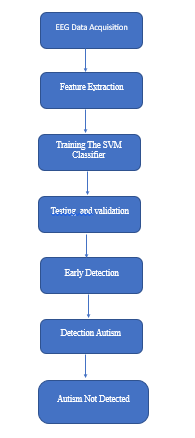
**Training**

The classifier receives the features of the EEG signals of different patients. The dataset is divided into two sets, one is for training and other one is for testing. Initially, the training data is fed to the classifier where the classifier learns about the patterns to identify the autism based person and a non-autism based person. The more the data for training the more the accuracy, but, the more data for training will also increases the time for the completion of training too. So, we need to choose the better suitable features which can distinguish the two kinds of persons very clearly.

The training of the classifier will also need to be fed with labels which are the states of a person with or without autism. We need to feed the labelled data for the classifier, because the SVM classifier is a supervised machine learning classifier, it needs a labelled data for the training. The training will generates the boundaries between the classes for making a comparison. Later, the classifier is tested for its accuracy.

**Testing**

The classifier which got trained in the previous training stage will be tested for it accuracy in this stage. The classifier will be fed with some random testing data samples which we have separated initially will be used here for testing of the classifier’s accuracy. The classifier after testing will produce results in binary data either ‘0’ or ‘1’, because the classifier here, we’re using is a Binary SVM. Later the binary results need to be converted to the labels which we have fed the classifier initially. The classifier has tested for its results, it has produced very closer predictions of the data. The tested results has shown an accuracy of over 90% in this proposed method which is greater than the existing method where they have classified using KNN classifier.



**Fig: Algorithm of proposed method**

The classifier will produce the estimates or predictions based the test data that, we’re feeding to it. The predictions will be closer to the actual status, only if we’ve used an efficient classifier and best feature extraction technique, to extract the best features of the signals that can be used to detect difference between the both classes. After the implementation of the testing of the classifier the accuracy of the classifier showed an accuracy of 90%, which is higher than the previously used KNN classifier which only got an accuracy of just over 80%. The SVM classifier is not only a better classifier than the KNN but, it is quiet easier and has many applications regarding the classifications.

**ADVANTAGES AND APPLICATIONS**

**Advantages:**

1. The SVM Classifier is better than existing KNN classifier.

2. Instantaneous frequencies based features really helped in maximizing accuracy.

3. The SVM classifier uses less data for training which decreases the training time.

**Applications**

1. Applied in DSP applications.

2. EEG Based diseases Detection.

3. Bio-Medical Signal Processing.

**HARDWARE & SOFTWARE REQUIREMENTS:**

**Software:**

• Matlab R2020a.

**Hardware:**

**Operating Systems:**

• Windows 10

• Windows 7 Service Pack 1

• Windows Server 2019

• Windows Server 2016

**Processors:**

Minimum: Any Intel or AMD x86-64 processor

Recommended: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support

**Disk:**

Minimum: 2.9 GB of HDD space for MATLAB only, 5-8 GB for a typical installation

Recommended: An SSD is recommended a full installation of all Math Works products may take up to 29 GB of disk space

**RAM:** Minimum 4 GB but recommended 8 GB

**SOFTWARE**

Introduction to Matlab

What Is MATLAB?

The name MATLAB stands for Matrix Laboratory. The software is built up around vectors and matrices. This makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. MATLAB has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language, and is one of the easiest programming languages for writing mathematical programs. These factors make MATLAB an excellent tool for teaching and research.

MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems.

MATLAB abilities a family of add-on software program utility software application software program software utility software-unique solutions called toolboxes. Very essential to maximum customers of MATLAB, toolboxes assist you to studies and observe specialized technology. Toolboxes are entire collections of MATLAB abilities (M-files) that increase the MATLAB surroundings to remedy precise schooling of problems. Areas in which toolboxes are to be had embody signal processing, manipulate systems, neural networks, fuzzy correct judgment, wavelets, simulation, and hundreds of others.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

Brief History of MATLAB:

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. The first MATLAB® was not a programming language; it was a simple interactive matrix calculator. There were no programs, no toolboxes, no graphics and no ODEs or FFTs. He designed it to give his student’s access to LINPACK and EISPACK without them having to learn FORTRAN. It soon spread to other universities and found a strong audience within the applied mathematics community. The mathematical basis for the first version of MATLAB was a series of research papers by J. H. Wilkinson and 18 of his colleagues, published between 1965 and 1970 and later collected in Handbook for Automatic Computation, Volume II, Linear Algebra, edited by Wilkinson and C. Reinsch. These papers present algorithms, implemented in Algol 60, for solving matrix linear equation and Eigen value problems.

In the 1970s and early 1980s, I was teaching Linear Algebra and Numerical Analysis at the University of New Mexico and wanted my students to have easy access to LINPACK and EISPACK without writing FORTRAN programs. By “easy access,” I meant not going through the remote batch processing and the repeated edit-compile-link-load-execute process that was ordinarily required on the campus central mainframe computer. Jack little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded Math Works in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK. MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra and numerical analysis, and is popular amongst scientists involved in video processing.

EISPACK and LINPACK:

In 1970, a group of researchers at Argonne National Laboratory proposed to the U.S. National Science Foundation (NSF) to “explore the methodology, costs, and resources required to produce, test, and disseminate high-quality mathematical software and to test, certify, disseminate, and support packages of mathematical software in certain problem areas.” The group developed EISPACK (Matrix Eigen system Package) by translating the Algol procedures for Eigen value problems in the handbook into FORTRAN and working extensively on testing and portability. The first version of EISPACK was released in 1971 and the second in 1976.

In 1975, four of us Jack Dongarra, Pete Stewart, Jim Bunch, and myself proposed to the NSF another research project that would investigate methods for the development of mathematical software. A byproduct would be the software itself, dubbed LINPACK, for Linear Equation Package. This project was also centered at Argonne. LINPACK originated in FORTRAN; it did not involve translation from Algol. The package contained 44 subroutines in each of four numeric precisions. In a sense, the LINPACK and EISPACK projects were failures. We had proposed research projects to the NSF to “explore the methodology, costs, and resources required to produce, test, and disseminate high-quality mathematical software.” We never wrote a report or paper addressing those objectives. We only produced software.

So, I studied Niklaus Wirth’s book Algorithms + Data Structures = Programs and learned how to parse programming languages. I wrote the first MATLAB an acronym for Matrix Laboratory in FORTRAN, with matrix as the only data type. The project was a kind of hobby, a new aspect of programming for me to learn and something for my students to use. There was never any formal outside support, and certainly no business plan. This first MATLAB was just an interactive matrix calculator. This snapshot of the start-up screen shows all the reserved words and functions. There are only 71. To add another function, you had to get the source code from me, write a FORTRAN subroutine, add your function name to the parse table, and recompile MATLAB.

Starting MATLAB:

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

• The Command Window

• The Command History

• The Workspace

• The Current Directory

• The Help Browser

Current Folder: This panel allows you to access the project folders and files.

Command Window: This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).

Workspace: The workspace shows all the variables created and/or imported from files.

Command History: This panel shows or return commands that are entered at the command line.

Help Browser:

The critical way to get assist online is to use the MATLAB help browser, opened as a separate window every through clicking at the question mark photograph (?) on the computing tool toolbar, or through manner of typing assist browser on the spark off in the command window. The assist Browser is an internet browser blanketed into the MATLAB computing tool that shows a Hypertext Markup Language (HTML) files. The Help Browser consists of panes, the help navigator pane, used to find out information, and the show pane, used to view the information. Self-explanatory tabs apart from navigator pane are used to performs are searching out.

MATLAB language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

MATLAB working environment:

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

MATLAB mathematical function library:

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB DESKTOP:

MATLAB Desktop is the precept MATLAB utility window. The computing tool includes five sub home windows, the command window, the workspace browser, the modern-day-day list window, the command records window, and one or greater decide domestic windows, which is probably confirmed high-quality on the identical time due to the truth the client suggests a photo. The command window is in which the character types MATLAB instructions and expressions at the spark off (>>) and in which the output of these commands is displayed. MATLAB defines the workspace because the set of variables that the client creates in a bit consultation. The workspace browser suggests those variables and some facts about them. Double clicking on a variable within the workspace browser launches the Array Editor, which may be used to gain statistics and profits instances edit exceptional homes of the variable.

The modern-day-day-day Directory tab above the workspace tab suggests the contents of the cutting-edge list, whose path is shown inside the modern-day list window. For example, in the home windows on foot machine the path is probably as follows: C: MATLAB Work, indicating that listing “artwork” is a subdirectory of the number one list “MATLAB”; WHICH IS INSTALLED IN DRIVE C. Clicking on the arrow within the modern list window suggests a listing of these days used paths. Clicking at the button to the right of the window permits the individual to trade the present day listing. MATLAB uses a seeking out path to find out M-documents and one-of-a-type MATLAB associated documents, which can be put together in directories within the computer document tool. Any report run in MATLAB need to be dwelling in the modern-day-day listing or in a list that is on is looking for course. By default, the documents supplied with MATLAB and math works toolboxes are included inside the searching out direction. The first-rate manner to look which directories are on the searching out route. The satisfactory manner to appearance which directories are speedy the quest route, or to characteristic or regulate a searching for course, is to pick out outset path from the File menu the computing device, and then use the set course talk discipline. It is proper exercise to feature any generally used directories to the hunt route to avoid again and again having the exchange the cutting-edge-day listing.

The Command History Window contains a file of the instructions a person has entered in the command window, together with every contemporary-day and former MATLAB periods. Previously entered MATLAB instructions can be determined on and re-completed from the command statistics window thru proper clicking on a command or series of commands. This movement launches a menu from which to select numerous options similarly to executing the commands. This is useful to select out abilities options in addition to executing the instructions. This is a beneficial feature at the equal time as experimenting with numerous commands in a piece session.

Using the MATLAB Editor to create M-Files:

The MATLAB editorial manager is a literary substance proofreader particular for growing M-facts and a graphical MATLAB debugger. The supervisor can seem in a window through command facts technique for itself, or it is probably a right-clicking inside the PC. M-information this gadget signified through the use of the expansion .M, as in pixel up.M. The MATLAB editorial supervisor window has a few draws down menus for obligations collectively with sparing, seeing, and troubleshooting facts. Since it plays more than one easy test and furthermore affects utilization of shade to separate among exclusive variables of code, this article editorial supervisor is often supported due to reality the system of a need for composing and altering M-talents. To open the manager, type at enact opens the M-document filename. M in a supervisor window, sorted out for enhancing. As stated earlier than, the file should be inside the cutting-edge posting, or in a posting in the seeking out direction.

Features of MATLAB:

Following are the basic features of MATLAB.

• It is a high-level language for numerical computation, visualization and application development.

• It also provides an interactive environment for iterative exploration, design and problem solving.

• It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

• It provides built-in graphics for visualizing data and tools for creating custom plots.

• MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

• It provides tools for building applications with custom graphical interfaces.

• It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Uses of MATLAB:

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including

• Signal Processing and Communications

• Video and Video Processing

• Control Systems

• Test and Measurement

• Computational Finance

• Computational Biology

Applications of MATLAB:

MATLAB can be used as a tool for simulating various electrical networks but the recent developments in MATLAB make it a very competitive tool for Artificial Intelligence, Robotics, Video processing, Wireless communication, Machine learning, Data analytics and whatnot. Though it’s mostly used by circuit branches and mechanical in the engineering domain to solve a basic set of problems its application is vast. It is a tool that enables computation, programming and graphically visualizing the results. The basic data element of MATLAB as the name suggests is the Matrix or an array. MATLAB toolboxes are professionally built and enable you to turn your imaginations into reality. MATLAB programming is quite similar to C programming and just requires a little brush up of your basic programming skills to start working with.

Below are a few applications of MATLAB –

• Statistics and machine learning (ML)

This toolbox in MATLAB can be very handy for the programmers. Statistical methods such as descriptive or inferential can be easily implemented. So is the case with machine learning. Various models can be employed to solve modern-day problems. The algorithms used can also be used for big data applications.

• Curve fitting

The curve fitting toolbox helps to analyze the pattern of occurrence of data. After a particular trend which can be a curve or surface is obtained, its future trends can be predicted. Further plotting, calculating integrals, derivatives, interpolation, etc. can be done.

• Control systems

Systems nature can be obtained. Factors such as closed-loop, open-loop, its controllability and observability, Bode plot, NY Quist plot, etc. can be obtained. Various controlling techniques such as PD, PI and PID can be visualized. Analysis can be done in the time domain or frequency domain.

• Signal Processing

Signals and systems and digital signal processing are taught in various engineering streams. But MATLAB provides the opportunity for proper visualization of this. Various transforms such as Laplace, Z, etc. can be done on any given signal. Theorems can be validated. Analysis can be done in the time domain or frequency domain. There are multiple built-in functions that can be used.

• Mapping

Mapping has multiple applications in various domains. For example, in Big Data, the Map Reduce tool is quite important which has multiple applications in the real world. Theft analysis or financial fraud detection, regression models, contingency analysis, predicting techniques in social media, data monitoring, etc. can be done by data mapping.

• Deep learning

It’s a subclass of machine learning which can be used for speech recognition, financial fraud detection, and medical video analysis. Tools such as time-series, Artificial neural network (ANN), Fuzzy logic or combination of such tools can be employed.

• Financial analysis

An entrepreneur before starting any endeavor needs to do a proper survey and the financial analysis in order to plan the course of action. The tools needed for this are all available in MATLAB. Elements such as profitability, solvency, liquidity, and stability can be identified. Business valuation, capital budgeting, cost of capital, etc. can be evaluated.

• Video processing

The most common application that we observe almost every day are bar code scanners, selfie (face beauty, blurring the background, face detection), video enhancement, etc. The digital video processing also plays quite an important role in transmitting data from far off satellites and receiving and decoding it in the same way. Algorithms to support all such applications are available.

• Text analysis

Based on the text, sentiment analysis can be done. Google gives millions of search results for any text entered within a few milliseconds. All this is possible because of text analysis. Handwriting comparison in forensics can be done. No limit to the application and just one software which can do this all.

• Electric vehicles designing

Used for modeling electric vehicles and analyze their performance with a change in system inputs. Speed torque comparison, designing and simulating of a vehicle, whatnot.

• Aerospace

This toolbox in MATLAB is used for analyzing the navigation and to visualize flight simulator.

• Audio toolbox

Provides tools for audio processing, speech analysis, and acoustic measurement. It also provides algorithms for audio and speech feature extraction and audio signal transformation.

COMMUNICATION:

Communications System Toolbox™ offers algorithms and gear for the layout, simulation, and analysis of communications systems. These capabilities are furnished as MATLAB ® features, MATLAB System gadgets™, and Simulink ® blocks. The machine toolbox includes algorithms for source coding, channel coding, interleaving, modulation, equalization, synchronization, and channel modeling. Tools are supplied for bit blunders charge evaluation, producing eye and constellation diagrams, and visualizing channel characteristics. The machine toolbox additionally provides adaptive algorithms that allow you to version dynamic communications structures that use OFDM, OFDMA, and MIMO techniques. Algorithms support fixed-point facts arithmetic and C or HDL code era.

Key Features

▪ Algorithms for designing the physical layer of communications systems, which includes supply coding, channel coding, interleaving, modulation, channel fashions, MIMO, equalization, and synchronization

▪ GPU-enabled System objects for computationally intensive algorithms together with Turbo, LDPC, and Viterbi decoders

▪ Interactive visualization equipment, consisting of eye diagrams, constellations, and channel scattering capabilities

▪ Graphical tool for evaluating the simulated bit mistakes rate of a machine with analytical outcomes

▪ Channel models, consisting of AWGN, Multipath Rayleigh Fading, Rician Fading, MIMO Multipath Fading, and

LTE MIMO Multipath Fading

▪ Basic RF impairments, along with nonlinearity, section noise, thermal noise, and section and frequency offsets

▪ Algorithms available as MATLAB features, MATLAB System objects, and Simulink blocks

▪ Support for fixed-point modeling and C and HDL code technology

System Design, Characterization, and Visualization:

The layout and simulation of a communications gadget requires analyzing its reaction to the noise and interference inherent in real-world environments, reading its behavior the usage of graphical and quantitative manner, and determining whether the resulting overall performance meets requirements of acceptability. Communications System Toolbox implements a selection of obligations for communications machine layout and simulation. Many of the functions, System objects™, and blocks inside the device toolbox perform computations associated with a specific thing of a communications gadget, consisting of a demodulator or equalizer. Other talents are designed for visualization or evaluation.

System Characterization

The system toolbox offers several standard methods for quantitatively characterizing system performance:

▪ Bit error rate (BER) computations

▪ Adjacent channel power ratio (ACPR) measurements

▪ Error vector magnitude (EVM) measurements

▪ Modulation error ratio (MER) measurements

Because BER computations are fundamental to the characterization of any communications system, the system toolbox provides the following tools and capabilities for configuring BER test scenarios and accelerating BER simulations:

BER tool— A graphical user interface that enables you to analyze BER performance of communications systems. You can analyze performance via a simulation-based, semi analytic, or theoretical approach.

Error Rate Test Console — A MATLAB object that runs simulations for communications systems to measure error rate performance. It supports user-specified test points and generation of parametric performance plots and surfaces. Accelerated performance can be realized when running on a multi core computing platform.

Multi core and GPU acceleration — A capability provided by Parallel Computing Toolbox™ that enables you to accelerate simulation performance using multi core and GPU hardware within your computer.

Distributed computing and cloud computing support — Capabilities provided by Parallel Computing Toolbox and MATLAB Distributed Computing Server™ that enable you to leverage the computing power of your server farms and the Amazon EC2 Web service. Performance Visualization. The system toolbox provides the following capabilities for visualizing system performance:

Channel visualization tool — For visualizing the characteristics of a fading channel

Eye diagrams and signal constellation scatter plots — for a qualitative, visual understanding of system behavior that enables you to make initial design decisions

Signal trajectory plots — for a continuous picture of the signal’s trajectory between decision points

BER plots — for visualizing quantitative BER performance of a design candidate, parameterized by metrics such as SNR and fixed-point word size

Analog and Digital Modulation

Analog and digital modulation strategies encode the facts circulation into a sign this is appropriate for transmission. Communications System Toolbox presents some of modulation and corresponding demodulation abilities. These talents are available as MATLAB features and gadgets, MATLAB System Modulation sorts provided by the toolbox are:

Source and Channel Coding

Communications System Toolbox affords source and channel coding talents that can help you develop and compare communications architectures fast, enabling you to discover what-if eventualities and avoid the need to create coding competencies from scratch.

Source Coding

Source coding, also referred to as quantization or signal formatting, is a manner of processing facts a good way to lessen redundancy or prepare it for later processing. The system toolbox offers a diffusion of styles of algorithms for imposing source coding and interpreting, inclusive of:

▪ Quantizing

▪ Companding (µ-law and A-law)

▪ Differential pulse code modulation (DPCM)

▪ Huffman coding

▪ Arithmetic coding

Channel Coding

▪ orthogonal area-time block code (OSTBC) (encoder and decoder for MIMO channels)

▪ Turbo encoder and decoder examples

The gadget toolbox offers application functions for developing your personal channel coding. You can create generator polynomials and coefficients and syndrome deciphering tables, in addition to product parity-take a look at and generator matrices.

The system toolbox additionally presents block and convolutional interleaving and deinters leaving functions to reduce facts errors as a result of burst mistakes in a conversation machine:

Block, including General block interleaver, algebraic interleaver, helical scan interleaver, matrix interleaver, and random interleaver.

Convolutional, including General multiplexed interleaver, convolutional interleaver, and helical interleaver

Channel Modeling and RF Impairments

Channel Modeling

Communications System Toolbox provides algorithms and tools for modeling noise, fading, interference, and different distortions which might be commonly found in communications channels. The system toolbox supports the subsequent styles of channels:

▪ Additive white Gaussian noise (AWGN)

▪ Multiple-enter multiple-output (MIMO) fading

▪ Single-enter single-output (SISO), Rayleigh, and Rician fading

▪ Binary symmetric

A MATLAB channel object provides a concise, configurable implementation of channel models, enabling you to

specify parameters such as:

▪ Path delays

▪ Average path gains

▪ Maximum Doppler shifts

▪ K-Factor for Rician fading channels

▪ Doppler spectrum parameters

For MIMO systems, the MATLAB MIMO channel object expands these parameters to also include:

▪ Number of transmit antennas (up to 8)

▪ Number of receive antennas (up to 8)

▪ Transmit correlation matrix

▪ Receive correlation matrix

To combat the effects noise and channel corruption, the system toolbox provides block and convolutional coding and decoding techniques to implement error detection and correction. For simple error detection with no inherent correction, a cyclic redundancy check capability is also available. Channel coding capabilities provided by the system toolbox include:

▪ BCH encoder and decoder

▪ Reed-Solomon encoder and decoder

▪ LDPC encoder and decoder

▪ Convolutional encoder and Viterbi decoder

RF Impairments

To model the effects of a non-ideal RF front end, you can introduce the following impairments into your communications system, enabling you to explore and characterize performance with real-world effects:

▪ Memory less nonlinearity

▪ Phase and frequency offset

▪ Phase noise

▪ Thermal noise

You can include more complex RF impairments and RF circuit models in your design using SimRF™.

Equalization and Synchronization

Communications System Toolbox lets you discover equalization and synchronization strategies. These techniques are usually adaptive in nature and tough to design and symbolize. The machine toolbox affords algorithms and tools that will let you swiftly select the proper approach on your communications machine. Equalization To compare one-of-a-kind techniques to equalization, the device toolbox offers you with adaptive algorithms which include:

▪ LMS

▪ Normalized LMS

▪ Variable step LMS

▪ Signed LMS

▪ MLSE (Viterbi)

▪ RLS

▪ CMA

These adaptive equalizers are available as nonlinear decision feedback equalizer (DFE) implementations and as

Linear (symbol or fractionally spaced) equalizer implementations.

Synchronization

The device toolbox provides algorithms for each service segment synchronization and timing phase synchronization. For timing section synchronization, the machine toolbox presents a MATLAB Timing Phase Synchronizer object that offers the following implementation techniques:

▪ Early-late gate timing method

▪ Gardner’s method

▪ Fourth-order nonlinearity method

Stream Processing in MATLAB and Simulink

Most verbal exchange structures cope with streaming and frame-primarily based statistics using a aggregate of temporal processing and simultaneous multi frequency and multichannel processing. This form of streaming multidimensional processing can be visible in superior communication architectures consisting of OFDM and MIMO. Communications System Toolbox enables the simulation of advanced communications structures via helping move processing and frame-based simulation in MATLAB and Simulink. In MATLAB, circulate processing is enabled by way of System items™, which use MATLAB objects to symbolize time-based and facts-driven algorithms, sources, and sinks. System objects implicitly manipulate many information of flow processing, including information indexing, buffering, and management of set of rules state. You can mix System gadgets with fashionable MATLAB functions and operators. Most System items have a corresponding Simulink block with the identical abilities. Simulink handles circulation processing implicitly with the aid of coping with the float of information thru the blocks that make up a Simulink model. Simulink is an interactive graphical environment for modeling and simulating dynamic systems that uses hierarchical diagrams to symbolize a machine version. It includes a library of widespread-reason, predefined blocks to represent algorithms, resources, sinks, and device hierarchy.

Implementing a Communications System

Fixed-Point Modeling Many communications systems use hardware that requires a fixed-point representation of your design.

Communications System Toolbox supports fixed-point modeling in all relevant blocks and System objects™ with tools that help you configure fixed-point attributes.

Fixed-point support in the system toolbox includes:

▪ Word sizes from 1 to 128 bits

▪ Arbitrary binary-point placement

▪ Overflow handling methods (wrap or saturation)

▪ Rounding methods: ceiling, convergent, floor, nearest, round, simplest, and zero

Fixed-Point Tool in Simulink Fixed Point™ facilitates the conversion of floating-point data types to fixed point. For configuration of fixed-point properties, the tool tracks overflows and maxima and minima.

Code Generation

Once you've got advanced your set of rules or communications device, you can robotically generate C code from it for verification, rapid prototyping, and implementation. Most System gadgets, functions, and blocks in Communications System Toolbox can generate ANSI/ISO C code the use of MATLAB Coder™, Simulink Coder™, or Embedded Coder™. A subset of System gadgets and Simulink blocks also can generate HDL code. To leverage present highbrow belongings, you can choose optimizations for specific processor architectures and integrate legacy C code with the generated code.

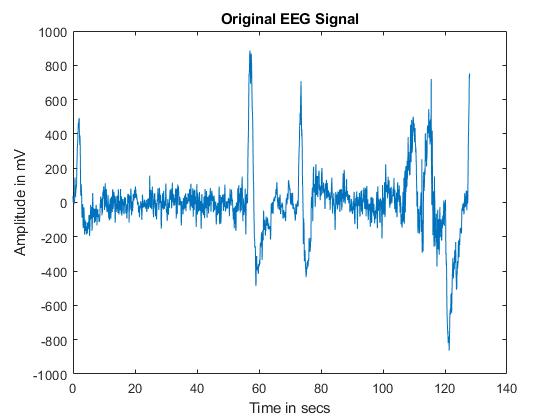
You can also generate C code for both floating-point and fixed-point data types.

DSP Proto typing DSPs are used in communication system implementation for verification, rapid prototyping, or final hardware implementation. Using the processor-in-the-loop (PIL) simulation capability found in Embedded Coder, you can verify generated source code and compiled code by running your algorithm’s implementation code on a target processor. FPGA Prototyping

FPGAs are used in communication systems for implementing high-speed signal processing algorithms. Using the FPGA-in-the-loop (FIL) capability found in HDL Verifier™, you can test RTL code in real hardware for any existing HDL code, either manually written or automatically generated HDL code.

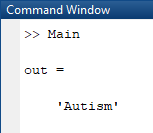
**RESULTS**

**OUTPUTS:**



**Fig: Raw EEG Signal**

**Classification Results:**



**Fig: Prediction of the Classifier**

**CONCLUSION**

After the implementation of SVM Classifier based Autism Detection, we can finally conclude that, the classifiers in the existing and proposed methods all belong to machine learning classifiers. So, we trained the classifiers using the labelled data. The proposed SVM classifier performed better classification results than the existing KNN classifier which makes it more reliable for predictions where the accuracy is more important issue.

**FUTURE SCOPE**

After the implementation of the classification using both supervised learning based machine learning classifiers, we now wants to extend it to the unsupervised learning based classifiers that too from Deep Learning Classifiers which are more accurate than the machine learning classifiers. The Deep learning classifiers are very closer at their predictions which is more suitable for the Bio-Medical Signal applications where the predictions judge the health condition of a person. Any of the deep learning classifiers like, Artificial Neural Networks (ANN) or Convolutional Neural Networks (CNN) are very good classifications.

**REFERENCES**

[1] Thabtah, Fadi. "Machine learning in autistic spectrum disorder behavioral research: A review and ways forward. (2018) " Informatics for Health and Social Care : 1-20.

[2] Thabtah, Fadi, Firuz Kamalov, and Khairan Rajab. (2018) "A new computational intelligence approach to detect autistic features for autism screening." International journal of medical informatics 117: 112-124.

[3] Vaishali, R., and R. Sasikala. "A machine learning based approach to classify Autism with optimum behaviour sets. (2018) " International Journal of Engineering & Technology 7(4): 18.

[4] Constantino, John N., Patricia D. Lavesser, Y. I. Zhang, Anna M. Abbacchi, Teddi Gray, and Richard D. Todd. (2007) "Rapid quantitative assessment of autistic social impairment by classroom teachers." Journal of the American Academy of Child & Adolescent Psychiatry 46(12): 1668-1676.

[5] Daniel Bone, Matthew S. Goodwin, Matthew P. Black, Chi-Chun Lee, Kartik Audhkhasi, and Shrikanth Narayanan. (2015) "Applying machine learning to facilitate autism diagnostics: pitfalls and promises." Journal of autism and developmental disorders 45(5): 1121-1136.

[6] Dennis Paul Wall, J. Kosmicki, T. F. Deluca, E. Harstad, and Vincent Alfred Fusaro. (2012) “Use of machine learning to shorten observationbased screening and diagnosis of autism.” Translational psychiatry, 2(4): e100.

[7] Dennis P. Wall, Rebecca Dally, Rhiannon Luyster, Jae-Yoon Jung, and Todd F. DeLuca. (2012) “Use of artificial intelligence to shorten the behavioral diagnosis of autism.” PloS one, 7(8): e43855.

[8] Fadi Thabtah. (2017). “Autism spectrum disorder screening: machine learning adaptation and DSM-5 fulfillment.” In Proceedings of the 1st International Conference on Medical and Health Informatics, pp. 1-6. ACM.

[9] Daniel Bone, Chi-Chun Lee, Matthew P. Black, Marian E. Williams, Sungbok Lee, Pat Levitt, and Shrikanth Narayanan.(2014). “The psychologist as an interlocutor in autism spectrum disorder assessment: Insights from a study of spontaneous prosody.” Journal of Speech, Language, and Hearing Research, 57(4): pp.1162-1177.

[10] Fadi Thabtah. (2017) “ASD Tests. A mobile app for ASD screening.” www.asdtests.com [accessed December 20th, 2017].

[11] Baihua Li, Arjun Sharma, James Meng, Senthil Purushwalkam, and Emma Gowen. (2017) “Applying machine learning to identify autistic adults using imitation: An exploratory study.” PloS one, 12(8): e0182652.

[12] Fadi Fayez Thabtah (2017), “Autistic Spectrum Disorder Screening Data for Adult”., https://archive.ics.uci.edu/ml/machine-learningdatabases/00426/.

[13] M. S. Mythili, and AR Mohamed Shanavas. (2014) “A study on Autism spectrum disorders using classification techniques.” International Journal of Soft Computing and Engineering (IJSCE), 4: 88-91.

[14] J. A. Kosmicki, V. Sochat, M. Duda, and D. P. Wall. (2015) “Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning.“ Translational psychiatry, 5(2): e514.

[15] Fadi Fayez Thabtah (2017), “Autistic Spectrum Disorder Screening Data for children,” https://archive.ics.uci.edu/ml/machine-learningdatabases/00419/ ,2017.