

# Support Vector Machine Implementation to Separate Linear and Non-Linear Dataset

Viswanatha V.<sup>1\*</sup>, Ramachandra A. C.<sup>2</sup>, Rayyan Mohammed<sup>3</sup>, Sai Kiran V.<sup>3</sup>, Parth Sheetal Kumthekar<sup>3</sup>

<sup>1</sup>Assistant Professor, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

<sup>2</sup>Professor, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

<sup>3</sup>UG Student, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

DOI: [10.36348/sjet.2023.v08i01.002](https://doi.org/10.36348/sjet.2023.v08i01.002)

| Received: 08.12.2022 | Accepted: 21.01.2023 | Published: 26.01.2023

\*Corresponding author: Viswanatha V.

Assistant Professor, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

## Abstract

This paper presents the implementation of separating the linear and non-linear data using support vector machine (SVM) algorithm. First let us understand what linear and non-linear datasets are. Linear datasets are the data that can be easily separable using a straight line. Such data are usually easy to implement in Artificial Neural Networks as they require a smaller number of hidden layers for its computation. Less layers implies a smaller number of weights assigned to the nodes present between the layers and less amount of time needed is needed to compute and update the weights for the current neural network. Hence linear datasets are easy to train and model. Whereas non-linear datasets are those datasets which cannot be separated by a straight line. For such datasets more hidden layers and weights are required and also more time and computational power is needed for a system to update the weights and train the model to give a better and sophisticated output data. As a result, training and modeling such neural network is tedious due to its complexity. To solve this problem SVM comes into picture. SVM stands for Support Vector Machine which is a machine or an algorithm that helps in classification and diversifying the data given to it. The data provided to the Support Vector Machine (SVM) should be a labeled one. Then these datasets are given to a training model where the training process of the neural network is being undergone. Once the training is completed, the next step is to predict the output. For this process we have to provide a new data that may or may not belong to the dataset, so that the neural network can predict the output of it. If the prediction is wrong, again the training is done until we get the actual output matching with the desired output given by the designer for verification purposes. This is the basic working process under the SVM algorithm. The linear data that is used for this separation is an Iris dataset that contains various information about the different plant-life growing from 2002-2004.

**Keywords:** Support Vector Machine, Kernel function, Feature extraction, Deep learning.

**Copyright © 2023 The Author(s):** This is an open-access article distributed under the terms of the Creative Commons Attribution **4.0 International License (CC BY-NC 4.0)** which permits unrestricted use, distribution, and reproduction in any medium for non-commercial use provided the original author and source are credited.

## 2. INTRODUCTION

Nowadays technological advances have boosted artificial networks to advance at a rapid speed. Integration of this advanced hardware with Artificial Neural Networks (ANN) are paving the road for different applications in areas such as control engineering, autonomous navigation and automated robotics [1]. This is possible only through extensive research in the fields of machine learning and artificial intelligence that we can use the systems for such advanced applications. The combination of the highly

specialized hardware with latest help to build systems in a robust manner. One such type of networks used for data classification and implementation and data regression is known as a SUPPORT VECTOR MACHINE (SVM). An SVM is basically an algorithm that makes data analysis and classification easy and is also widely used in many of the ARTIFICIAL NEURAL NETWORKS (ANN) and Machine Learning for variables of multiple data [2-4]. The Fig. 1(a) is the Iris flower taken as the input for the linear data and the fig 1(b) is a circular construct for the non-linear data.



Fig. 1(a): Iris flower for linear data

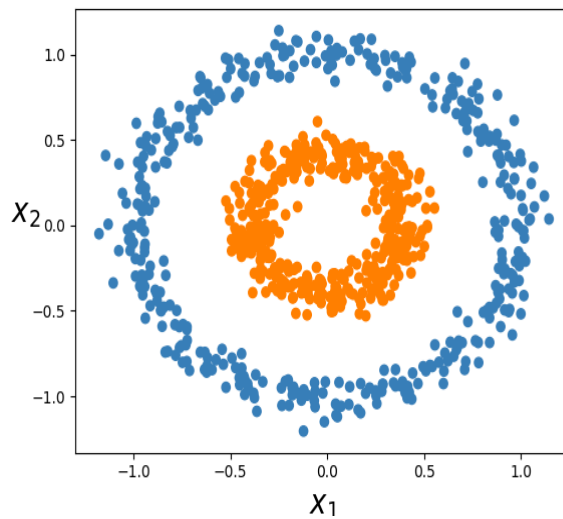


Fig. 1(b): Circular construct non-linear data

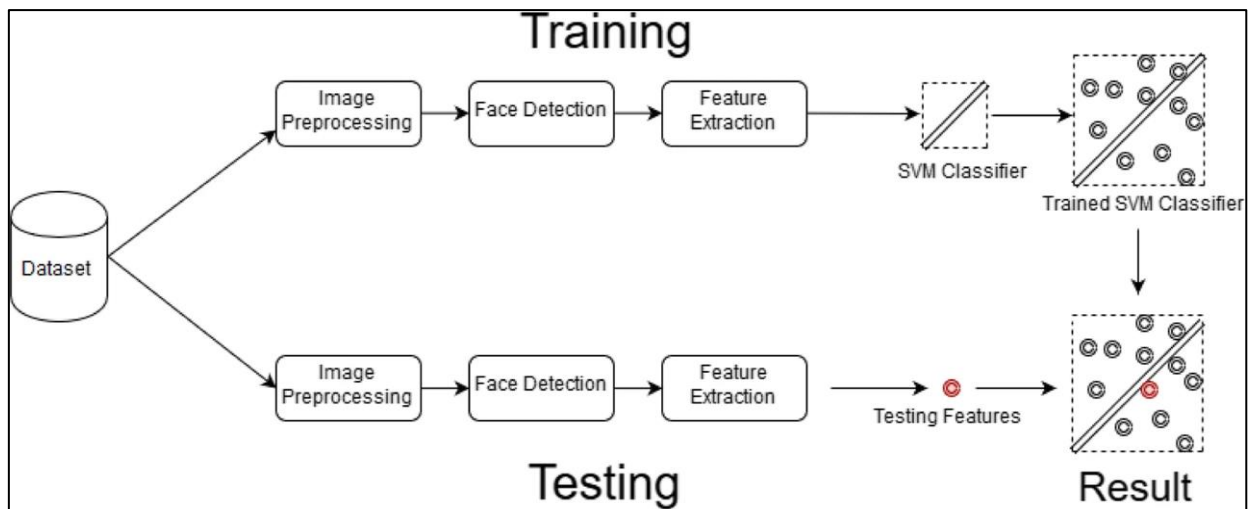


Fig. 2(a): Outline of SVM

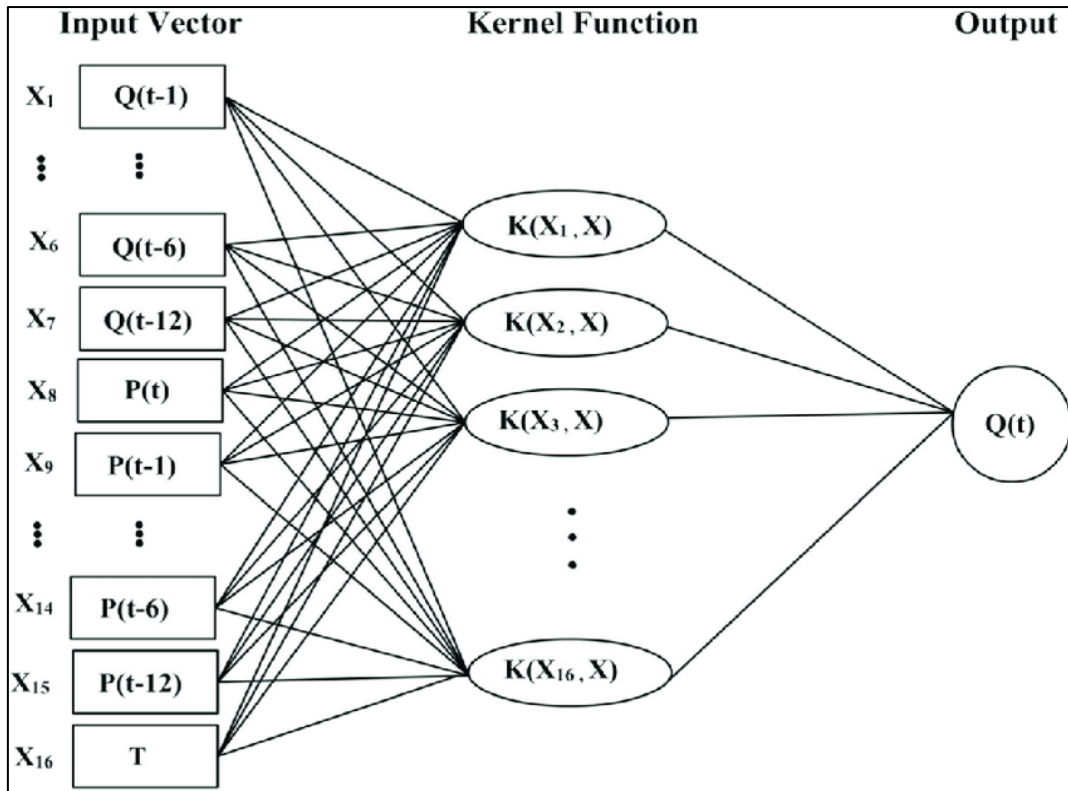


Fig. 2(b): Kernel function

SVM can be classified into two distinct types:

- i. Simple (or) Linear SVM
- ii. Kernel (or) Nonlinear SVM

**i. Simple (or) Linear SVM:**

The linear type SVM is referring to a SVM that is commonly used for the classification of any linearly separated data. This suggests that a set of data

can only be separated into various classes or categories with a singular straight line, in this case it is called a linear SVM, and the related data is called as linearly distinct or separable. Also, the classifier that separates the data is called a linear SVM classifier. A common SVM is generally used to help with categorizing address and analysis problems [5-7].

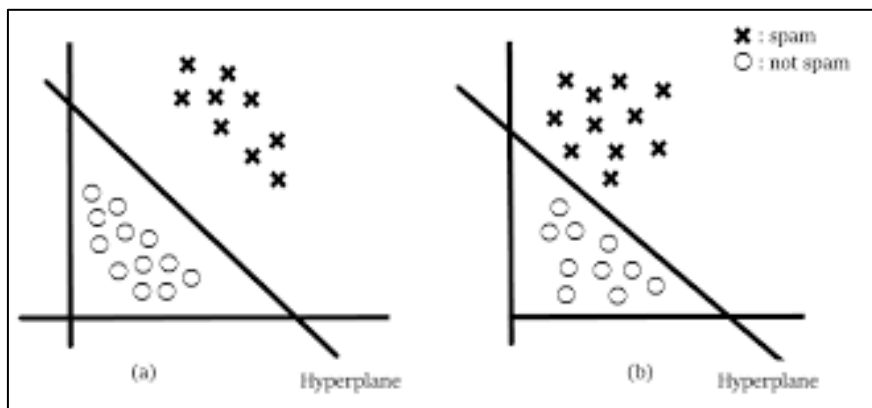
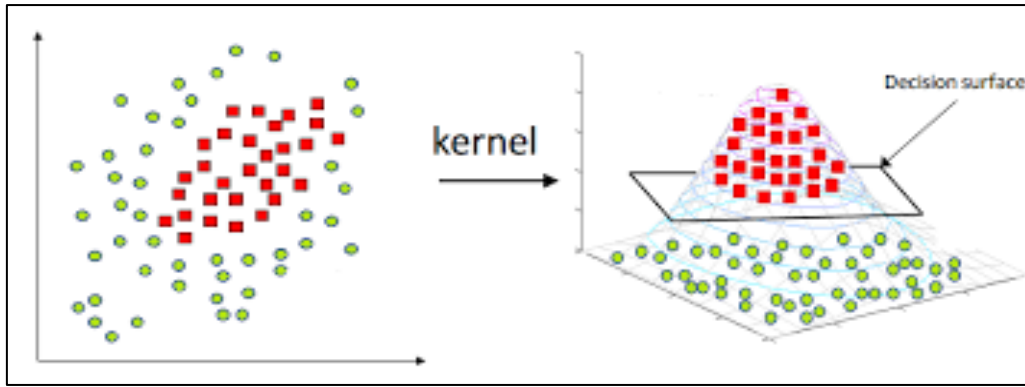


Fig. 2(c): Data segregation for a linear SVM

**ii. Kernel (or) Nonlinear SVM**

Nonlinear data that will not be classified or segregated into unique classes with the help of just a singular line is referred to as a NON-LINEAR SVM. Here, the classifier is called a nonlinear classifier. The classification process can be executed with the help of a nonlinear data by adding a few features into a certain

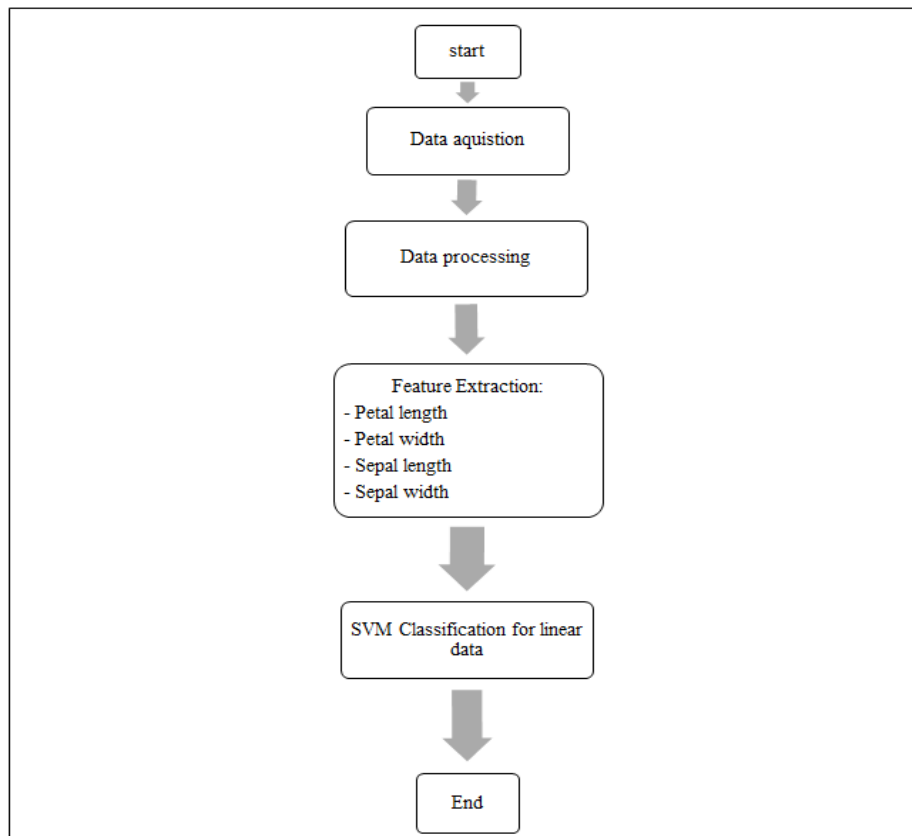
higher dimension instead of relying on 2-D spaces. Here, the latest features fit the hyper-plane which helps to easily separate the classes/categories. KERNEL SVM's are generally used as they can handle optimization problems with multiple complex variables [8-10].



**Fig. 2(d): Data segregation using Kernel function**

SVMs are known to be able solve complicated maths related problems. But smooth SVM is reserved for purposes like data classification because the techniques used for smoothing that reduces the data outliers and constructs a visible pattern are in use. Thus, for problems requiring optimization smooth SVMs use algorithm to handle a large dataset that general SVMs are not capable of doing. Smooth SVM types conventionally are for exploring mathematical properties like as strong convexity for better and precise

classification of data, even for nonlinear data. For linear data, the chosen data sheet is an Iris data set which contains different parameters of an ‘Iris flower’ like petal length, petal width, sepal length and sepal width. These parameters are taken into account to categorise the given data of flower into 2 sub-classes which are divided by a single hyperplane. Here it is considered that the length of the petal and the width of the petal are the primary indicators to segregate the flower data into its respective sub-classes [11-13].



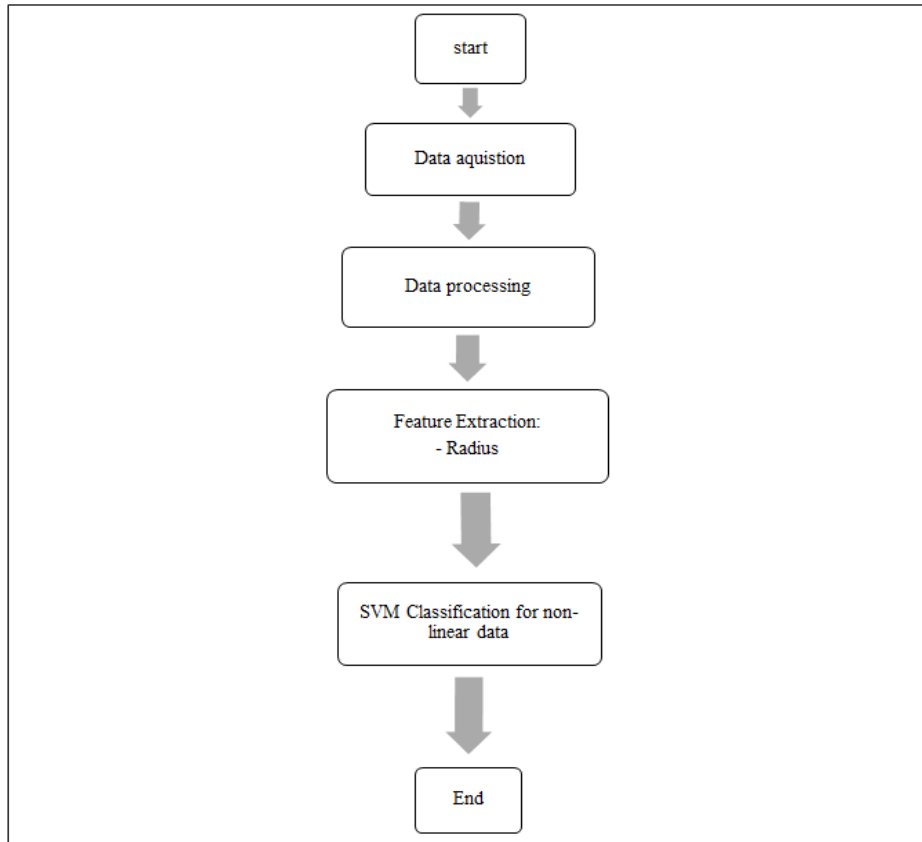
**Fig. 2(e): Flowchart of Iris flower classification**

In, this paper, we will try to analyse and implement to see a SVM by looking at the implementation using python, we also are looking at few of the significant implementation. For linear data we use tensor flow library and for the non-linear data

we use the kernel functions to separate the data based on the variable conditions given. Similarly for non-linear data the feature extraction can be viewed as the radius of the given non-linear data, in this case which is a data circle. The algorithm corresponding to the kernel

function considers the radius as its primary indicator and implements it to segregate the input data into distinct sub-brackets. Among all the kernel function

available, in this case an RBF (Radius Bias Function) is selected as it has the highest accuracy and efficiency than compared to its counterparts [14, 15].

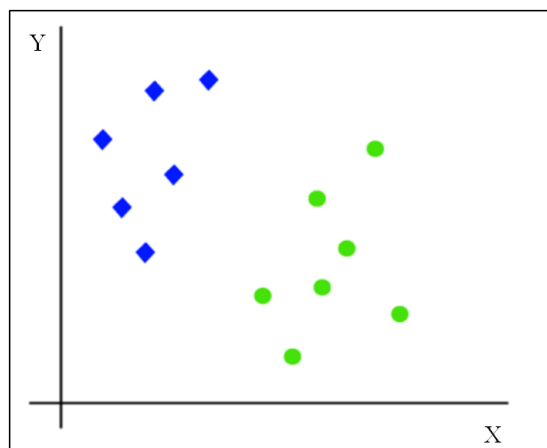


**Fig. 2(f): Flowchart of Circular construct classification**

### 3. METHODOLOGY

The simplest principle for the operation of the Support Vector Machine (SVM) are to generate a hyperplane that isolates a data set and convert to organised classes. We can consider a basic example.

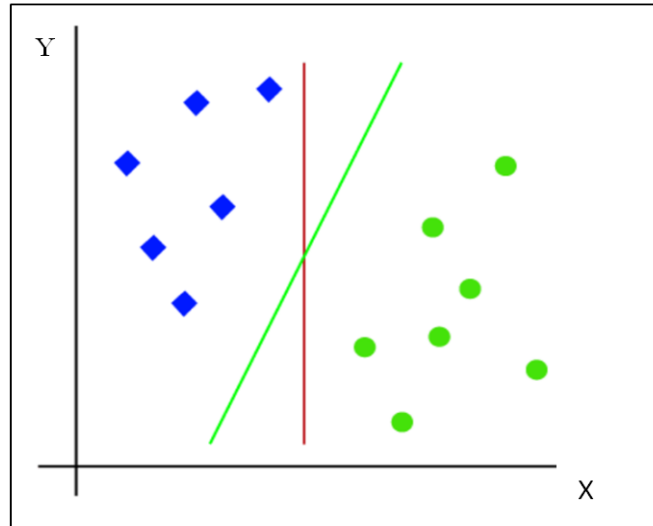
Suppose a given data set, we have to allocate green circles from blue rectangles. Then the agenda is to form an imaginary locus of points that divides data set into different classes, creating a distinct classification connecting the rectangles and circles [16, 17].



**Fig. 3(a): Input unorganized data**

Now we can analyse the plane that segregates the two categories given above in fig 3(a), there can be multiple layers or planes which can perform the task.

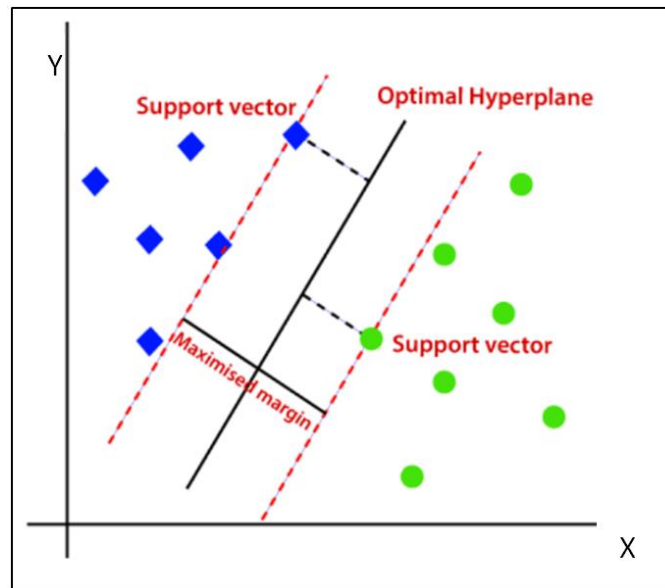
Thus, it complicates the best line which could be used to perform this task as we can see in the figure below.



**Fig. 3(b): Intermediate of plane in data**

In the fig 3(b) mentioned above, we have a red and a green line and either can be an ideal line to partition the data into two classes, however we cannot ignore the fact that infinite number of lines or planes can be generated for the data and any of them can be the most efficient one. Therefore, according to this algorithm, observe the points that are nearest to either

of the categories, thus the points are called support vectors. Thus, proximity between these points of locus and the cleave plane is known as Margin. So, the goal of the SVM algorithm is to enlarge the margin, when the margin arrives at its highest value, the hyper plane enhances the efficient result [18, 19].



**Fig. 3(c): Linear data segregation by the hyperplane**

SVM model algorithm enlarges the void existing in between classifications by forming a precise resolve confinement as shown in fig 3(c). However, this data set was in 2-D, thus the hyperplane was linear. But when the data set is in a N-Dimensional cartesian space, we need to generate (N-1) D plane that separates the vector place into unlinked classes. The decision rule for SVM is defined in equation (1).

$$\vec{X} \cdot \vec{w} - C \geq 0 \dots (1)$$

'x' is random point in data space of hyperplane.  $\vec{X}$  is vector of random point. To find position of 'X' whether it lies on left or right side of the plane, we draw perpendicular line from origin to plane is called vector ' $\vec{w}$ '. The distance of ' $\vec{w}$ ' from origin to plane is denoted by 'C'.

$$\begin{aligned} \vec{X} \cdot \vec{w} - C &= 0 \text{ (The data point lies on the decision boundry)} \\ \vec{X} \cdot \vec{w} - C &> 0 \text{ (The data point is positive sample)} \\ \vec{X} \cdot \vec{w} - C &< 0 \text{ (The data point is negative sample)} \end{aligned}$$

### 4. IMPLEMENTATION

The attempt to implement the example dataset into two categories and try to find a hyperplane to separate the data by using the python language. Hyperplane can again be classified into soft-margin and hard-margin hyperplane Firstly, for simple or linear SVM we use the Tensorflow open-source library function that is used to build the given dataset model in the form of tensors. TensorFlow is an opensource software library. TensorFlow is a software library for numerical computations for flowgraphs.

- Junctions on graphs are used to constitute algebraic operations.
- Fringe on graph is used to display the multi-dimensional array elements.

We install the tensorflow open-source library by using the syntax; `import tensorflow.compat.v1 as tf` using, this tensorflow library we are easily able to separate the given dataset using a single hyperplane .The flowchart representation for the python code is as shown below for a linear SVM.

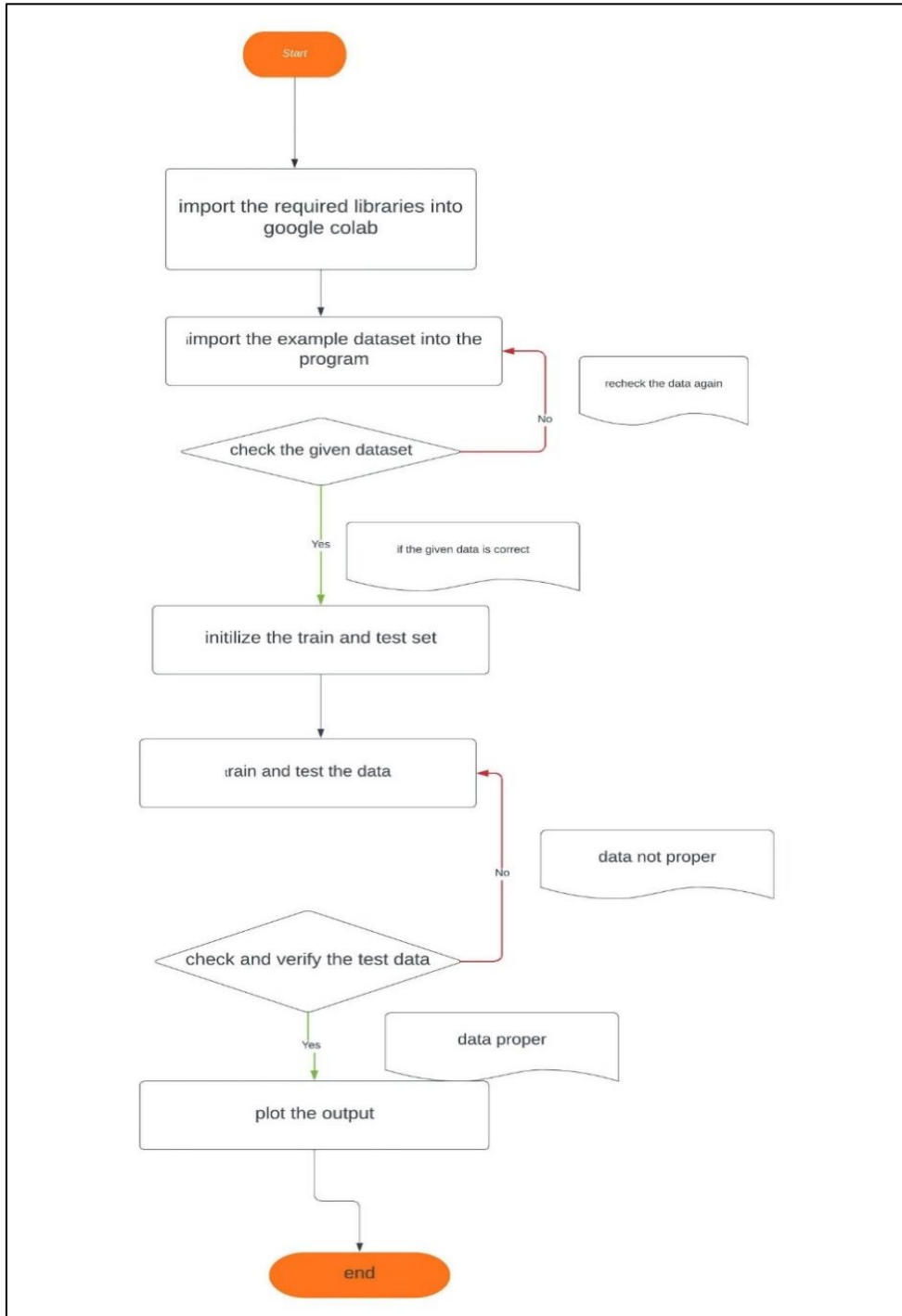


Fig. 4(a): Flowchart of logic for linear and non-linear data classification



The following steps explain the logic used in linear and non-linear data classification and the same is presented in the flowchart shown in fig. 4(a).

- We import all the required libraries and functions required to execute the SVM algorithm into the workspace.
- Import the sample dataset that needs to be analysed, modified and separated into the workspace (program).
- Check the dataset that was imported to make sure that every data has been successfully imported.
- If no, then import the dataset again.
- If yes, then proceed with the next step that is to initialize the train and test dataset.
- After that, we test the given dataset and then test the accuracy of this dataset.
- Now, the next step is to check and verify the test data.
- If the data is not proper, then perform the training and testing of the data.
- If the data is proper, the next step is to plot the output of the training set, the accuracy set and also the hyper-plane graph that segregates the dataset into two organised categories.

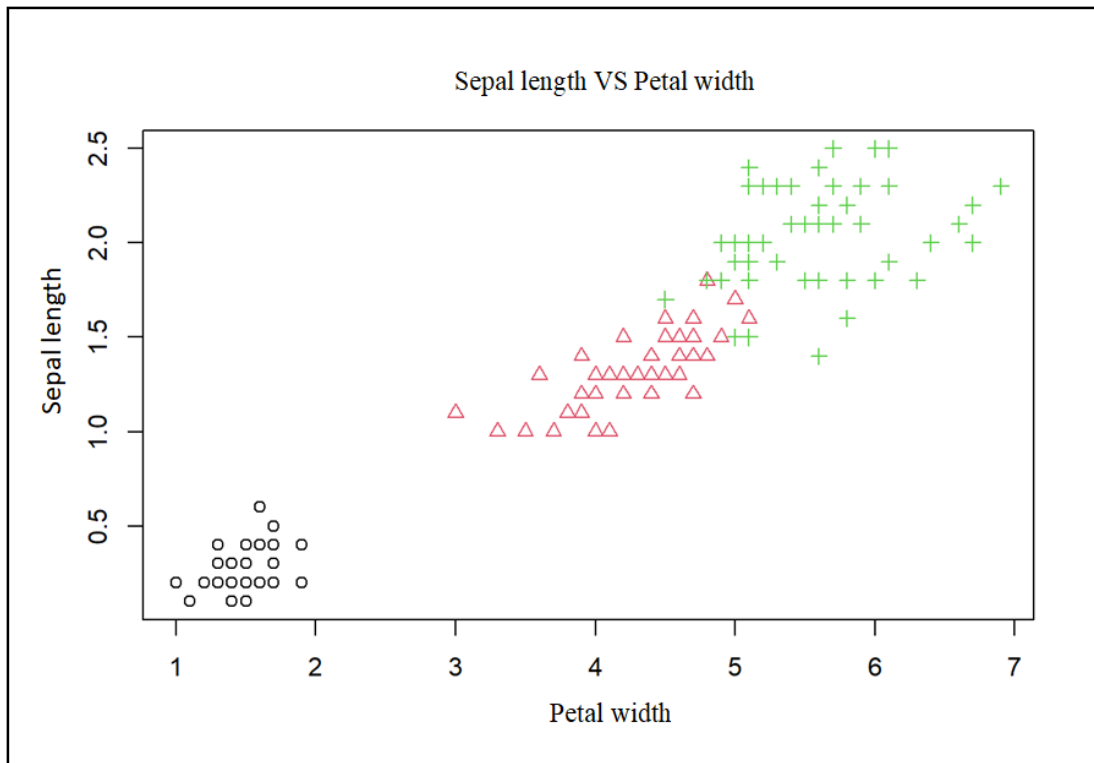
Now, for nonlinear SVM we use kernel function which again has 3 parts namely:

- Linear function.
- Polynomial function.
- Radius bias function (RBF).

For our execution, we choose the kernel function which has more accuracy. That is in this case we use the RBF function.

**5. RESULTS AND VERIFICATION**

The fig 5(a) is the input data of the Iris flower transformed into construct form for the SVM algorithm to process the data and convert it into organised classes. From this we can analyse the python code required for the linear and non-linear data separation and understand the plot that it makes. The graph shown in figure 5(b) shows the output for separation of data using a single hyper-plane. The graph plotted shows the sepal length (vs) petal width values, in which the data is separated by a single hyper-plane that is represented in the graph by the red colour. Here the data has been separated as Iris setosa and Iris non-setosa by comparing the values of sepal length and petal width, the data has been successfully analysed and measured and classified into two classes.



**Fig. 5(a): Input data for Iris flower**



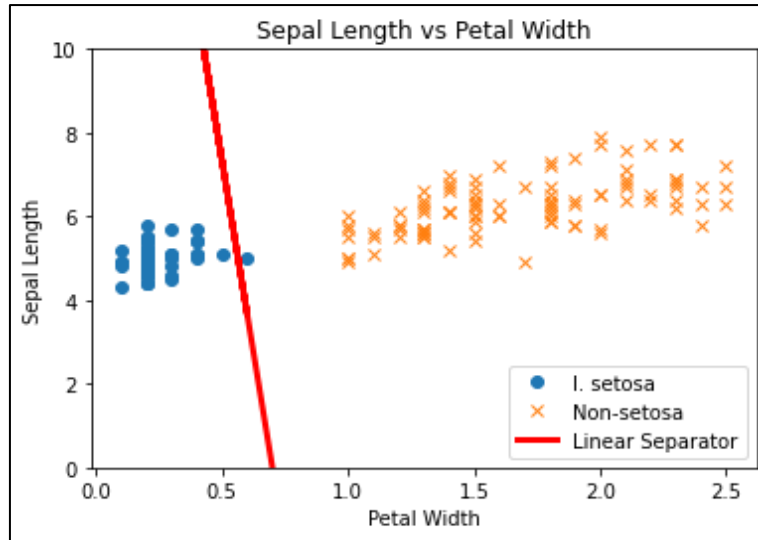


Fig. 5(b): Output of the linear data of the Iris flower

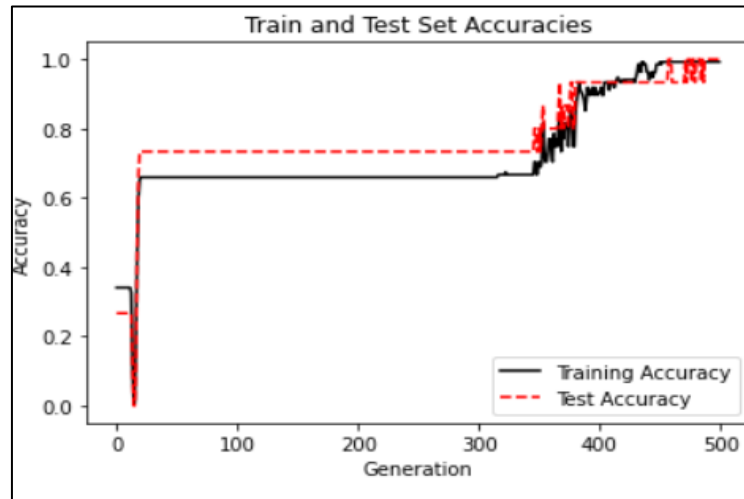


Fig. 5(c): Showing the Accuracy graph of train and test value

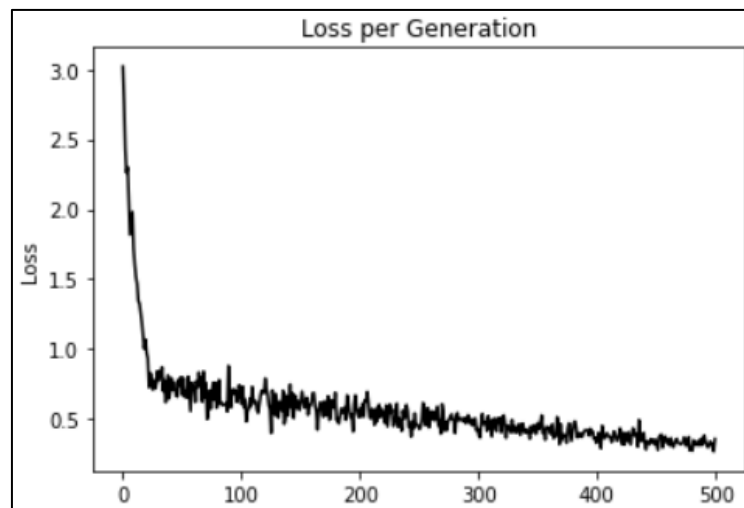


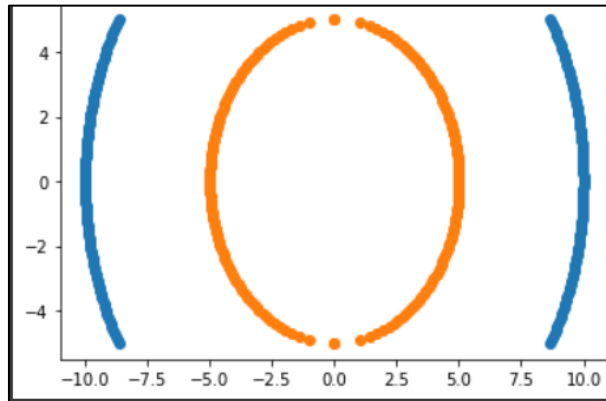
Fig. 5(d): Loss per generations

The graph in the fig.5(c) shown above is representing the loss per generation value. It is a graph plotted between loss (vs) generation graph. From this graph we can understand that the loss value for each

generation is going low and at some point, in the generation value is almost 0.1.

All these above diagrams show that the linear separation of data using SVM has been successfully implemented and executed.

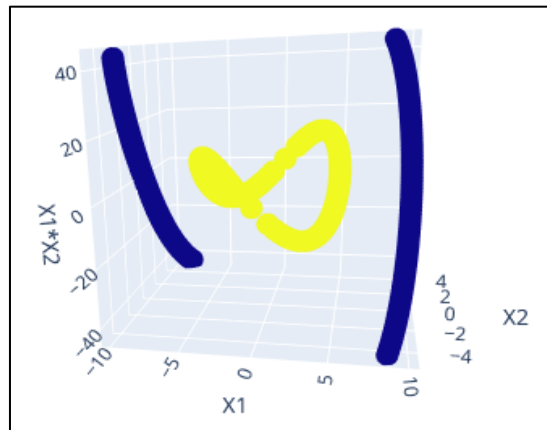
Now the input and output values for non-linear data separation is always as shown in the fig. 5(d).



**Fig. 5(d): Input of non-linear data**

The fig. 5(d) shown above is the input waveform that is obtained by using the circle equation. From the graph it is very clear that the data given is a

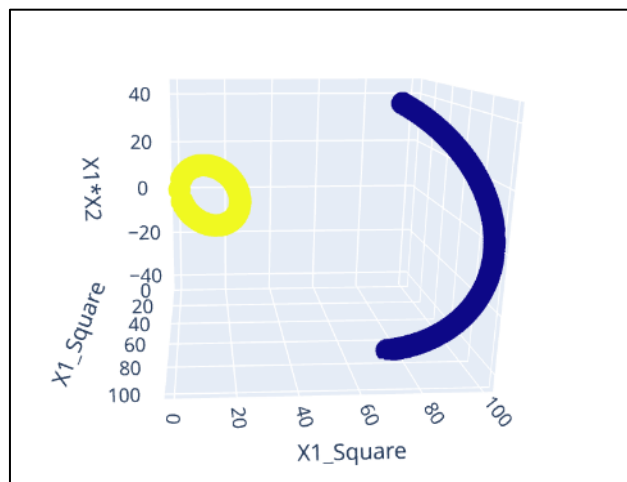
non-linear data that cannot be separated by a single hyper-plane. So, in cases like this, we use a kernel function in this case “RBF” (Radius Bias Function).



**Fig. 5(e): 3-D graph for representing the output value**

The above fig. 5(e) shows a 3-D plot that shows the separation of 2 non-linear data using the kernel function (RBF) that shows how the data has been

classified in a 3-D plane, where the blue line represents the outer/bigger circle and the yellow line represents the inner/smaller circle.



**Fig 5(f): 3-D graph showing the separation of nonlinear data**

The above graph shows a 3-D plot that shows the separation of 2 non-linear data using the kernel function (RBF) that shows how the data has been classified in a 3-D plane, where the blue line represents the outer/bigger circle, and the yellow line represents the inner/smaller circle.

## 6. CONCLUSION

We have observed that the method used by SVM is an alternative to commonly used regression models especially for nonlinear data; it has an edge over logistic regression model in this aspect. But as a result, the computations time spikes, this is because the time required to study the subject thoroughly hence is not as prominent as logically regression model and the bonus performance is not significant enough to replace logical regression in mainstream.

The higher performance also brings with it some potential risk so most strict institutions due not approve of SVMs use. From, the above project we can classify and implement data for linearly separable and non-linear SVM for data separation using python code. The main result, obtained from this project is that we were successfully able to classify and separate the data in SVM using python code and we were also able to analyse and understand the TensorFlow library used for linear data separation

We were also able to analyse and understand the kernel functions that are to be used for non-linear data classification and separation. We were also able to grasp how to choose the kernel function required implementing the code and successfully classifying and separating data with maximum accuracy. We were also able to understand how SVM can be used in every-day-to-day life and why it is being used in multiple fields including AI and machine learning also in various other day-to-day applications. Despite all the disadvantage of SVM it remains an important concept due to its conceptual and abstract point of view. This was one of the first methods which could be compared with human cognitive abilities to recognise and reconcile variables. Still, it is perceptual concept and cannot be used in real life applications and for intensively complex data applications, until some evolution or progress in the research for SVM.

## REFERENCES

- Piccialli, V., & Sciandrone, M. (2022). Nonlinear optimization and support vector machines. *Ann Oper Res*, 314, 15–47. <https://doi.org/10.1007/s10479-022-04655-x>
- Viswanatha, V., Chandana, R. K., & Ramachandra, A. C. (2022). "Real Time Object Detection System with YOLO and CNN Models: A Review."
- Borah, P., & Gupta, D. (2020). Unconstrained convex minimization based implicit Lagrangian twin extreme learning machine for classification (ULTELMC). *Applied Intelligence*, 50(4), 1327–1344.
- Borah, P., & Gupta, D. (2020). Functional iterative approaches for solving support vector classification problems based on generalized Huber loss. *Neural Computing and Applications*, 32(1), 1135–1139.
- Gangopadhyay, A., Chatterjee, O., & Chakrabarty, S. (2018). Extended polynomial growth transforms for design and training of generalized support vector machines. *IEEE Transactions on Neural Networks & Learning Systems*, 29(5), 1–14.
- Nalepa, J., & Kawulok, M. (2019). Selecting training sets for support vector machines: a review. *Artificial Intelligence Review*, 52(2), 857–900.
- Viswanatha, V., Chandana, R. K., & Ramachandra, A. C. (2022). IoT Based Smart Mirror Using Raspberry Pi 4 and YOLO Algorithm: A Novel Framework for Interactive Display. *Indian Journal of Science and Technology*, 15(39), 2011–2020.
- Lou, Y., Liu, Y., Kaakinen, J. K., & Li, X. (2017). Using support vector machines to identify literacy skills: evidence from eye movements. *Behavior Research Methods*, 49(3), 887–895.
- Yoo, K. H., Koo, Y. D., Ju, H. B., & Na, M. G. (2017). Identification of LOCA and estimation of its break size by multiconnected support vector machines. *IEEE Transactions on Nuclear Science*, 64(10), 1.
- Li, Z. H. A. N. G., Xingning, L. U., & Conglin, L. U. (2017). National matriculation test prediction based on support vector machines. *Journal of University of Science and Technology of China*, 47(1), 1–9.
- Viswanatha, V., Ashwini, K., & Sathisha, B. M. (2022). "Implementation of IoT in Agriculture: A Scientific Approach for Smart Irrigation." 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). *IEEE*.
- Tatsumi, K., Akao, M., Kawachi, R., & Tanino, T. (2011). Performance evaluation of multiobjective multiclass support vector machines maximizing geometric margins. *Numerical Algebra, Control and Optimization*, 1(1), 151–169.
- Gu, W., Chen, W.-P., & Ko, C.-H. (2018). Two smooth support vector machines for  $\epsilon$ -insensitive regression. *Computational Optimization & Applications*, 70(1), 1–29.
- Taherzadeh, G., Zhou, Y., Liew, A. W. C., & Yang, Y. (2016). Sequence-based prediction of protein–carbohydrate binding sites using support vector machines. *Journal of chemical information and modeling*, 56(10), 2115–2122.
- Balasundaram, S., & Gupta, D. (2016). Knowledge-based extreme learning machines. *Neural Computing and Applications*, 27(6), 1629–1641.
- Viswanatha, V., Sathisha, B. M., & Ashwini, K. (2022). "Custom Hardware and Software Integration: Bluetooth Based Wireless Thermal Printer for Restaurant and Hospital Management."

- 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). *IEEE*.
17. Ehrentraut, C., Ekholm, M., Tanushi, H., Tiedemann, J., & Dalianis, H. (2018). Detecting hospital-acquired infections: a document classification approach using support vector machines and gradient tree boosting. *Health informatics journal*, 24(1), 24-42.
  18. Zhang, X., Li, Y., & Peng, X. (2016). Brain Wave Recognition of Word Imagination Based on Support Vector Machines. *Chinese Journal of Aerospace Medicine*, 14(3), 277-281.
  19. Viswanatha, V., Kumari, A., & Kumar, P. (2021). Internet of things (IoT) based multilevel drunken driving detection and prevention system using Raspberry Pi 3. *International Journal of Internet of Things and Web Services*, 6.