Eksplorium *p*-ISSN 0854-1418
Volume 46 No. 1, 2025: 1446–1457 *e*-ISSN 2503-426X

Transforming Clustered Non-Linearly Separable Data using Nonlinear Mapping Functions in SVM for Enhanced Classification

Pavithra C[1], Saradha M[2] and Antline Nisha B[3]
[1]Department of Mathematics, REVA University, Bangalore, India.
pavithracshekar5@gmail.com
[2]Department of Mathematics, REVA University, Bangalore, India.
ishuharisri88@gmail.com
[3]Department of Mathematics, St. Joseph's Institute of Technology, Chennai, India.
antlinenisha@gmail.com

Article Received: 25 Feb 2025, Revised: 21 April 2025, Accepted: 04 May 2025

Abstract: Support Vector Machines is renowned for their robustness in handling classification and regression. They work by finding the best hyperplane that is able to class data into different classes very well. But when dealing with data that is clustered as well as non-linearly separable, SVM can suffer from issues in establishing crisp decision boundaries. To overcome this, using nonlinear mapping functions is useful. These operations assist in mapping information into higher-dimensional feature spaces in which the nonlinear patterns can be specified more distinctively. Our study investigates the capability of non-linear mapping functions to transform cluster, non-linearly separable information into a feature space without increasing the level of dimension complexity. SVM are reported to be capable of distinguishing information by determining optimal hyperplanes that separate distinct classes. Using nonlinear mapping functions, we establish different linear decision boundaries in the feature space, thus improving the accuracy of classifying non-linear data. The research explores the influence of altering the parameter on such a transformation and includes comparative results for, and to prove the sustainability of the method. Further, the research advances the knowledge of SVM and kernel techniques while enabling examination of the significance of different sets of features and encouraging the creation of machine learning techniques.

Keywords: separable, SVM, kernel, distinguishing, sustainability

1. INTRODUCTION

Kernel Trick is a strong idea employed within Support Vector Machines (SVMs) for dealing with data that is not linearly separable. SVMs proceed by identifying the optimal possible hyperplane that can distinguish between classes of data. This is easy where the data exists as clear-cut groups within a straight line or plane. But with most real-world issues, the data points are intertwined in a manner such that no straight line can differentiate them. The kernel trick circumvents this by transforming the data into a high-dimensional space, where a linear separation is feasible. Rather than explicitly computing this transformation (which can be inefficient), the kernel trick employs a kernel function to compute the relationship between points as if they were pre-transformed into that space. This makes the process efficient and scalable. Well-known kernels are the polynomial kernel, radial basis function (RBF), and sigmoid kernel each designed for different kinds of patterns in the data. With an appropriate kernel, SVMs can actually be used to classify data which appears inseparable in lower dimensions. In plain language, the kernel trick allows SVMs to discover patterns when none appear to be present by "lifting" the data into a new space where the underlying structure is exposed. It's a clever, elegant solution for addressing challenging classification problems. These parameters have a significant influence on the classification outcomes, emphasizing the need for a thorough understanding of SVM to provide the best possible performance [3][4].

SVM are binary classifiers that find the best hyperplane to distinguish between data samples, maximizing the margin, hard or soft. It is done by solving a quadratic programming problem. The resulting solution in the dual space is mapped back into the original space's surface of classification, where the structure of the features is the key to its application. The solid theoretical framework and high generalization ability of SVM have attracted much academic attention, and extensive research and developments continue to this day. SVM, based on risk minimization principles, adequately tackle issues of small sample sizes, nonlinearity, and high dimensions with minimal prior knowledge needed. SVM is thus especially well-suited to applications such as network intrusion detection systems. Choosing an adequate kernel function for practical projects is imperative because it should make the training set linearly separable in the feature space. Nonlinear models are needed for precise classification of nonlinear problems since linear SVM can be insufficient. A kernel function, linked to a semi-positive definite kernel matrix, implicitly defines the feature space [5].

Support Vector Machines (SVM) is a strong supervised learning technique employed for classification and regression problems, which are known to work well with high-dimensional data and reliably obtain the best decision boundary between classes. SVMs try to determine the hyperplane that most separates the data into classes by maximizing the margin, i.e., the distance between the hyperplane and the closest data points of any class, referred to as support vectors. Support vectors are essential components of the training set, which determine the position and orientation of the hyperplane, and the elimination of any support vector would alter the position of the hyperplane. For linearly separable data, SVM identifies a linear hyperplane, but for non-linear data, it applies the kernel functions to project the data into a higher-dimensional space where linear separation can be achieved. The kernel trick enables SVM to function in high-dimensional spaces without directly calculating the coordinates of the data in that space, with typical kernels being the linear kernel, polynomial kernel, and radial basis function (RBF) kernel [6][11].

Semi-consistent hyperplanes generalize the concept of optimal hyperplanes by taking into account situations in which perfect separation of classes is impossible, in most real-world data cases when data points overlap or when data includes noisy points. Semi-consistent hyperplanes seek to maximize separation of data, minimizing misclassified points and maximizing the margin for correctly classified points. To manage non-separable data, SVM introduce the idea of a soft margin, where some misclassifications are permitted and are regulated by a parameter that manages the trade-off between maximizing the margin and minimizing classification errors. Slack variables determine the degree of misclassification for each point, balancing between maximizing the margin and minimizing the sum of slack variables [7]. With this implementation of the concept of semi-consistent hyperplanes into the support vector machines, we have enhanced the model's ability to handle the complexities of real-world data. Soft margin and slack variables allow SVM to keep that high classification accuracy but with some degree of local robustness against overlapping and noisy data, thereby making SVM a robust and versatile machine learning method that achieves an accurate and consistent classification under quite challenging conditions. This is particularly beneficial in areas such as bioinformatics, image recognition, and finance, where data are prone to noise and outliers, and they improve the model's ability to generalize on unseen data, thus lowering overfitting [9].

2. LITERATURE REVIEW

Huang K et al propose a new kernel dictionary learning framework for nonlinear industrial process monitoring, primarily considering the aluminium electrolysis process. The study points out shortcomings of linear process monitoring approaches in retaining non-linear behaviors and encourages movement of input data into a very high-dimensional feature space through kernel mappings. In this space, it is the kernel dictionary learning algorithm which identifies discriminative features for anomaly detection. This method proved superior over standard linear ones in diagnosing processes and has the potential for more widespread industrial application in systems with complex dynamics [1].

Zhou Z and others introduced a randomized version of Kernel Principal Component Analysis (KPCA) to handle large-scale process data more effectively. While the main advantage of KPCA is nonlinear structure modeling, the main drawback of KPCA is its computational demands which hinders it from real-time implementations on large datasets. Zhou's method randomizes the procedure to truncate the feature space without compromising the nonlinear relationship crucial for discriminations. Optimization substantially reduced the computation time as well as memory requirement and also allowed for practical implementation in large-size industrial monitoring applications. The randomized KPCA showed excellent capability in process monitoring, suitable for varying online, real-time industrial processes[2].

Deng J et al, who presented a multi-block dynamic KPCA model for improved monitoring of dynamic and multi-block industrial processes. Classical KPCA tends to have difficulties with time-dependent and multi-source data. Deng's methodology adequately blends multi-block analysis and dynamic modeling to manage both spatial and temporal dependencies. The developed model gave better monitoring sensitivity and fault detection in dynamic nonlinear systems, confirmed by case studies in complicated industrial settings [3].

Zhang et al. proposed an unsupervised Kernel Extreme Learning Machine (KELM) for process monitoring. Being generally fast learners, the new age ELMs are known for their excellent generalization. Better still, the addition of kernel functions enables the ELM to handle nonlinear data. Zhang's framework preserved global data structure enhancing the detection of anomalies. Tested against benchmark data sets, KELM had outperformed older methods in terms of monitoring accuracy and fault sensitivity, thus proving the efficacy of merging ELM framework with kernel mapping for nonlinear process diagnosis [4].

Guo L et al introduced a sparse KPCA approach. Sparsity was achieved using a sequential updating strategy that allowed for compact data representation without sacrificing information fidelity. This not only enhanced monitoring precision but also lowered the computational cost, making it appropriate for industrial real-time use. The performance of the model on real industrial data reaffirmed its ability to effectively identify anomalies with sustaining speed processing and utilizing low system resources [5].

Liu Y et al solved this using Kernel Independent Component Analysis (KICA). Different from regular Independent Component Analysis (ICA), which is not effective in non-Gaussian nonlinear environments, KICA utilizes kernel techniques to reveal sophisticated statistical patterns and independent components related to process performance. The proposed model achieved improved fault detection and a greater understanding of the process dynamics. Implemented in a variety of industrial applications, KICA proved strong in monitoring functions, particularly the detection of hidden faults and intricate inter-variable relationships [6].

Wang L et al emphasized improving KPCA with the incorporation of a double-weighted Local Outlier Factor (LOF) to address local outliers that tend to compromise monitoring precision. The conventional KPCA, as a global approach, remains susceptible to localized noise and data irregularities. The double-weighted LOF scheme facilitated the identification and suppression of the impact of the outliers, enhancing robustness and accuracy in the monitoring model. Empirical testing validated enhanced fault detection rates and immunity to data abnormalities, again supporting the model's utility in actual industrial environments [7].

Jiao J et al suggested an optimized Kernel Partial Least Squares (KPLS) approach specifically designed to detect quality-related faults in nonlinear processes. Whereas ordinary PLS models are linear and incapable of capturing nonlinear relations, KPLS utilizes kernel mappings to develop a description of complex associations among process variables and quality factors. Their adapted version improved the method's fault sensitivity to product quality problems. When applied to real industrial data sets, the adapted KPLS was able to catch slight shifts in quality characteristics and was thus a sound tool for ensuring product quality and production consistency in nonlinear manufacturing systems [8].

3. STRUCTURE OF PROPOSED SYSTEM

The system under consideration is designed to leverage nonlinear mapping functions in Support Vector Machines (SVM) to restructure clustered, non-linearly separable data in a manner where linear separation is possible. It involves multiple important modules, each of which performs a different task in the overall data transformation as well as classification process. One of the pivotal steps is the process of choosing and extracting informative features that will effectively assist the nonlinear mapping, such that the transformed feature space provides more distinct class boundaries and enhanced model performance. This process can include feature engineering to construct new features that capture the data more effectively. Train the SVM with the transformed data in the feature space with higher dimensions. Examine the internal workings of the SVM and how the nonlinear mapping functions affect it. Learn how the transformation influences the decision boundary and how the various combinations of features are relevant.

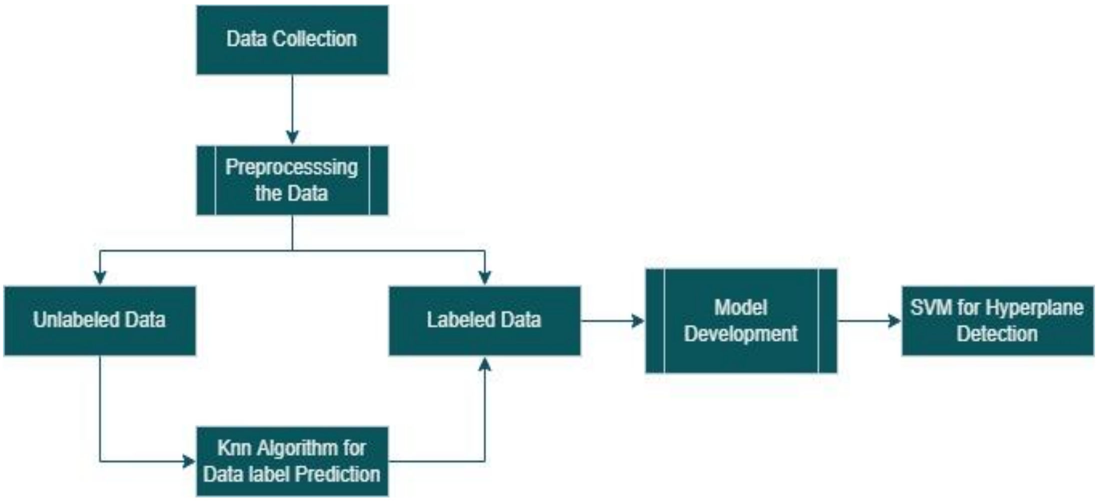

Fig 1: Structure of the proposed system

The application of this systematic method proves that nonlinear mapping functions have the ability to greatly improve SVM's classification performance on non-linearly separable data. The technique adds to greater understanding of SVM operations but also leaves doors open for using comparable techniques in more complicated, real-world datasets. Further work can be done with the investigation of further kernel functions, increasing computational efficiency, and testing the method across many fields.

4. RATIONALE TO EMPLOY KERNEL METHODS

Kernel techniques, especially when combined with Support Vector Machines (SVM), enable data to be projected into higher-dimensional spaces where it is separable in a more straightforward manner. The kernel trick performs this mapping without the actual calculation of the new, higher-dimensional coordinates but rather depends on kernel functions to calculate inner products of data points in this new space from the original input space. This capacity to treat non-linear relationships in the data is a real strength, for it enables more reliable fault detection in complex industrial processes. In process monitoring, the intricacy and nonlinearity found in many industrial processes are great challenges that are commonly beyond the capability of traditional linear methods.

The kernel trick is a powerful technique for allowing linear algorithms to work well with nonlinear data by implicitly mapping data to a higher-dimensional space. Feature representation is enriched without altering the inherent data structure, and performance is enhanced in anomaly detection applications. Through the use of various kernel functions radial basis function, polynomial, and sigmoid models attain the adaptability necessary in order to match various distributions of the data in a way that they can be generalizable in a wide range of complex situations. One of the primary strengths of the kernel trick is computational efficiency: it avoids direct explicit transformation by calculating inner products in feature space, a feat that, for massive datasets, is particularly beneficial. Kernel-based methods are also very robust to noise and outliers, making the output monitoring results robust and more reliable. They also provide a straightforward path to the extension of classical linear models into nonlinear spaces, allowing practitioners to apply known algorithms with little or no modification. In combination, kernel methods provide a beautiful and economical solution to modern industrial monitoring with enhanced accuracy and flexibility for cases with nonlinear dynamics and high-dimensional complexity [4][5].

4.1 Kernel Methods in the Machine Learning

Kernel-based techniques have become major tools in modern fault detection system design since they can detect challenging, nonlinear patterns in process data. Unlike traditional linear techniques that could be tested by complexities of real industrial processes, kernel techniques provide more accurate and uniform fault detection through data mapping into higher-dimensional spaces where nonlinear relationships are linearly separable. A important initial step in this process is data preprocessing, specifically normalization. Normalization ensures that all variables regardless of their original units or ranges—make an equal contribution to the model. Without normalization, variables with large magnitudes will dominate the analysis, causing skewed results and impaired model performance. Normalization provides fair input to the learning process through scaling every feature to the same range, which also enhances the sensitivity of the model to extreme process behavior deviations. Normalized data is now poised for advanced modeling. While standard procedure avoids the use of simple linear models due to interpretability, they fall short when faced with the inherent nonlinearities in industrial processes. Kernel techniques address this issue by enabling the transformation of the input data into a new space where the fault patterns and underlying structures are simpler to identify, resulting ultimately in improved monitoring performance and plant robustness.

Supervised Learning: Detection of defects in a process by classifying them from labeled past data.

- Classification: Classification facilitates early detection and accurate identification of distinct faults, allowing for early interventions. Classification is aimed at identifying normal and abnormal operating states.
- Regression: Regression algorithms estimate continuous values from input features. The predictive ability plays an important role in proactive maintenance and optimization of industrial processes.
- Ensemble methods: Ensemble methods are used to combine the predictions of several models to enhance overall accuracy and stability. This method tends to be more effective than a single model since it avoids overfitting and captures the strengths of various models.

Unsupervised Learning: Anomaly detection in a process by identifying patterns and aberrations in unlabeled data.

- Dimensionality reduction: This method decreases the number of variables being considered, making the dataset easier and keeping the critical information intact.
- Clustering: Clustering algorithms place similar data points in the same group according to their features, without known labels. In process monitoring, clustering can be used to find natural groupings of data, which can correspond to various operating conditions or fault states.
- Density Estimation: Density estimation is a method of modeling the probability distribution of a data set. This allows one to determine areas of high or low data density, which may be representative of normal or failure states.

Many real-world datasets are not linearly separable, meaning a straight line (or hyperplane in higher dimensions) cannot perfectly separate the data points into distinct classes. Kernel functions use a technique known as the kernel trick, which allows the algorithm to compute the dot products of the transformed features implicitly, without ever explicitly transforming the data. Imagine you have a dataset where the classes form concentric circles. A linear classifier cannot separate these circles in the original 2D space. [3][5].

Polynomial Kernel Function is given as, $k(x, y) = (x'y + c)^d$ (1)

Where, c is a coefficient term and d is the degree of the polynomial.

The polynomial kernel allows for the creation of a high-dimensional feature space where the original data, which may not be linearly separable, can be separated by a hyperplane. This is particularly useful in scenarios where data exhibits complex relationships that a linear kernel cannot capture.

Gaussian Kernel Function is given as, $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$ (2)

Sigmoidal Function is defined as, $k(x, y) = \tanh[\alpha.(x.y) + c]$ (3)

Where, α is the slope parameter and c is the intercept parameter. Adjust these values to control the characteristics of the kernel function.

Laplacian Function is defined as, $k(x, y) = \exp\left(-\frac{\|x-y\|_1}{\sigma}\right)$ (4)

Where, σ is the scale parameter that controls the width of the kernel.

Within data-driven fault detection, kernel-based methods offer a general framework for the modeling of nonlinear systems. Selecting an appropriate kernel function to well capture the underlying structure of the data is the key stage of this approach. The linear, polynomial, and Gaussian (RBF) kernels are most widely used, with each possessing distinct advantages based on data nature. For instance, the RBF kernel performs very well whenever relations among variables contain complicated, nonlinear patterns, enabling the model to identify subtle patterns that less complex techniques may fail to see. Following selection of an adequate kernel, the model is trained upon a preprocessed dataset in which every feature is normalized to provide balanced representation. In training, the algorithm comes to recognize normal versus faulty states by examining inherent patterns, trends, and interdependencies within data. Trained, the model can analyze new input in real time and mark deviations from expected behavior as possible faults. What makes kernel methods unique is their ability to map data to a space of a higher dimension, facilitating the ability to make unambiguous distinctions between various operational states. This flexibility and accuracy serve them well in those critical industrial environments where timely and accurate fault detection is paramount to system integrity and reduced downtime.

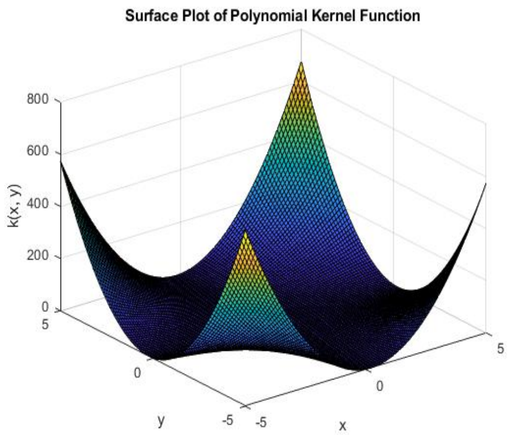

Fig 2: Polynomial kernel function

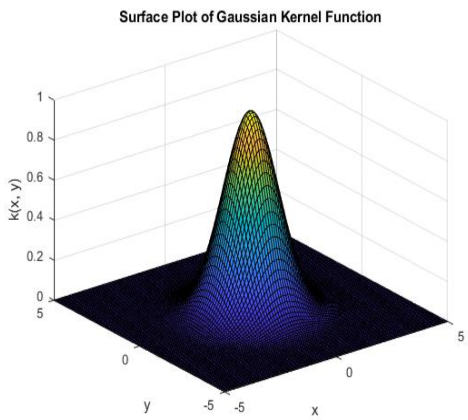

Fig 3: Gaussian kernel function

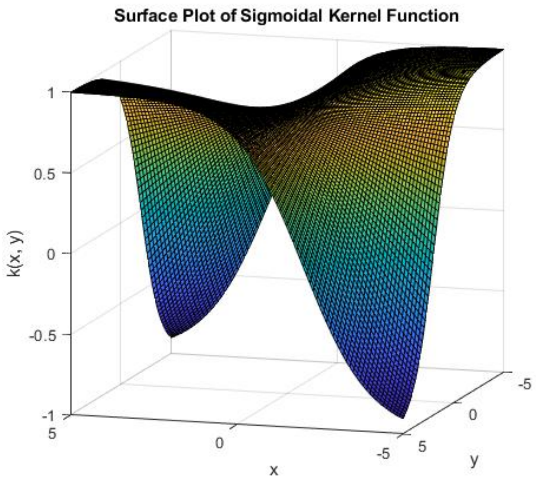

Fig 4: Sigmoidal kernel function

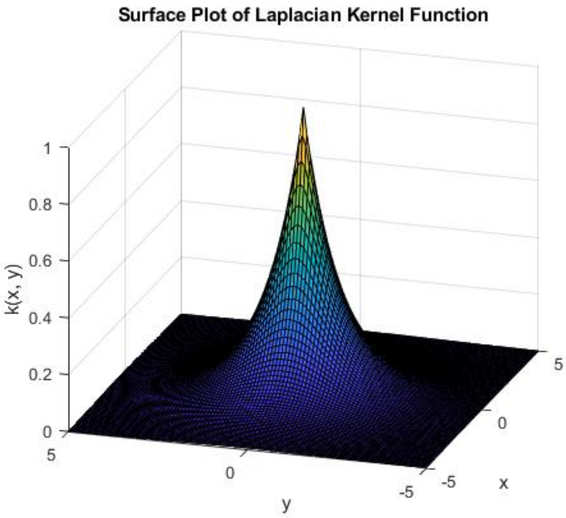

Fig 5: Laplacian kernel function

5. METHODOLOGY AND RESULTS

The kernel function is the application of a series of mathematical operations that transform the data in order to enable SVM to handle non-linear relations. Specifically, the kernel function transforms a non-linear decision surface into a linear equation in higher dimension. This conversion is achieved by computing the inner product between two points in the feature space, referred to as K(x1, x2), representing the inner product of x1 and x2 within this higher space. The properties of the kernel function enable it to execute the "kernel trick," allowing the computation of inner products among observations in the kernel-defined spaces.

We define the equation for the points and it is given as, (x_0, y_0) to a line $Xx + Yy + Z = 0$ is

$$\frac{|Xx_0 + Yy_0 + Z|}{\sqrt{X^2 + Y^2}} = 0$$

Nonlinear mapping functions are responsible for a transformation of data in machine learning, particularly with Support Vector Machines (SVM). For non-linearly separable data, a linear classification method would be insufficient as a result of the intricate structure of the data. Nonlinear mapping functions meet this challenge by mapping the original data into a higher-dimension feature space where the classes can be linearly separated. Some of the most common kernel functions are the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel. All of these functions transform the input data into a higher dimension without actually computing the coordinates in the higher dimension, a method referred to as the "kernel trick." The implicit mapping enables the SVM to work in the transformed feature space but compute on the original input space, thus conserving computational resources and making it easier. The data points are mapped into a new feature space where these data can be linearly separable. In this new space, the SVM finds the optimal hyperplane that maximizes the margin between the different classes. The transformation enabled by the nonlinear mapping function allows the

SVM to construct a linear decision boundary in the new feature space, effectively solving the classification problem.

Our process begins with selecting a suitable kernel function, which implicitly defines the nonlinear mapping, this converts nonlinear data points into a form that can be separated by a hyperplane.

$$\phi(x,y)=\begin{cases}(n_R-x+|x-y|,n_R-y+|x-y|) & \text{if } \sqrt{(x^2+y^2)}\geq n\\(x,y) & \text{Otherwise}\end{cases} \tag{5}$$

Where, n is the highest values of corresponding class.

n_R is the randomly chosen number, where $n_R > n$.

Our research highlights the substantial benefits of using nonlinear mapping functions to transform clustered, non-linearly separable data into a format that can be separated linearly, thus enhancing the effectiveness of SVM. This is fundamental component of supervised machine learning, are adept at performing classification and regression by determining the optimal hyperplane that divides data into separate classes. Nonetheless, when dealing with inherently nonlinear and clustered data, defining a clear decision boundary can be difficult. By implementing nonlinear mapping functions, we convert the data into a higher-dimensional feature space where a linear decision boundary can be more easily identified. This transformation allows for more precise classification of nonlinear data, fully utilizing the capabilities of SVM [11].

To demonstrate the effectiveness of the kernel trick in classifying clustered nonlinear synthetic data, we employed a methodological approach that transforms data into a higher-dimensional feature space using nonlinear mapping functions, denoted as Φ, to make it linearly separable. We generated synthetic data with distinct nonlinear clusters and varied the nonlinear mapping parameter n_R (testing values of 3, 4, and 5) to observe the impact on the distance between the transformed and original data. SVM models were trained on the transformed data for each n_R value, with appropriate kernel functions ensuring linear separability in the feature space. In order to compare the performance of SVM models, we inspected the classification results obtained under various parameter settings. By adjusting the values in a controlled manner, we could see how varying combinations of features shaped the model's behavior and decision-making process. Observing the spatial orientation and location of the separating boundaries through these configurations provided useful insight into how the SVM responds to variations in input and parameter changes. This.ethodical variation not only illuminated the internal workings of the model but also highlighted the significance of using appropriate kernel choice and parameter tuning in the context of nonlinear data structures. The visual plots of the decision surfaces for both settings uncovered changes in the margin and classification boundaries, providing a dramatic illustration of the impact of tuning on model generalization. In addition,

Eksplorium *p*-ISSN 0854-1418
Volume 46 No. 1, 2025: 1446–1457 *e*-ISSN 2503-426X

we provided comparative analysis in graphical form by comparing theoretically computed results against values derived using computational simulation. This double evaluation enabled us to confirm the robustness and consistency of predictions made by the SVM, highlighting the imperative of experimental refinement in creating reliable classification models for complicated datasets

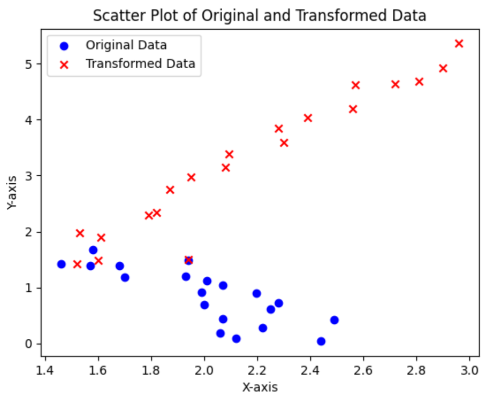

Fig 5: Analytically Solved for $n_R = 3$

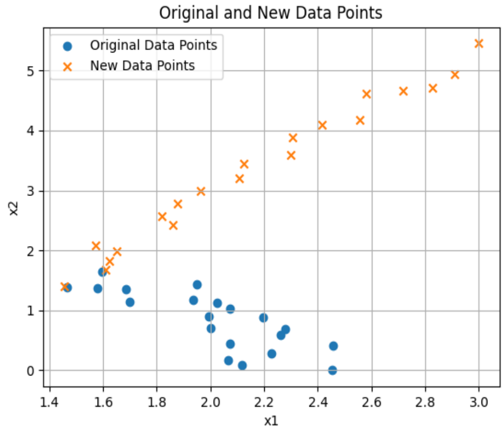

Fig 6: Validation for $n_R = 3$

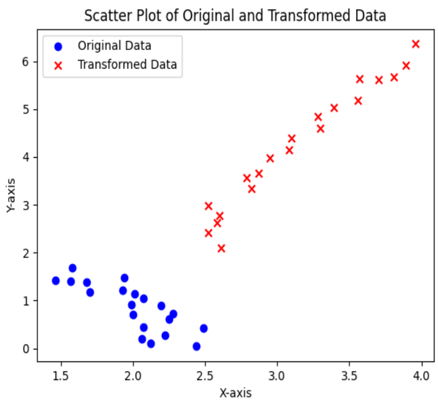

Fig 7: Analytically Solved for $n_R = 4$

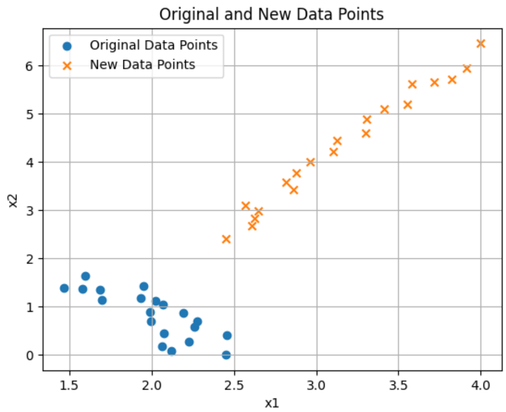

Fig 8: Validation for $n_R = 4$

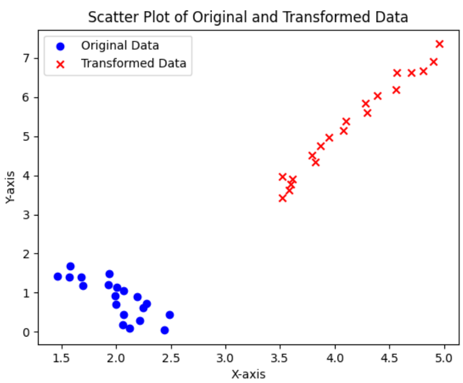

Fig 9: Analytically Solved for $n_R = 5$

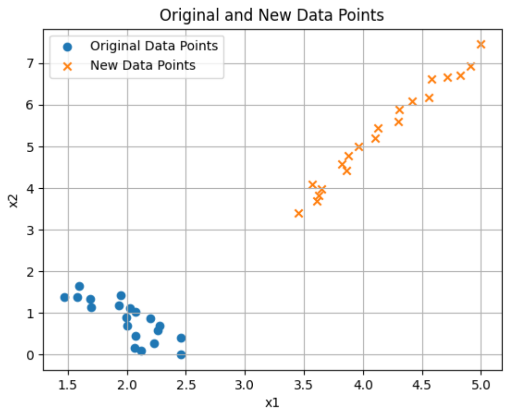

Fig 10: Validation for $n_R = 5$

$$\beta_1\phi_1(s_1).\phi_1(s_1)+\beta_2\phi_1(s_2)\phi_1(s_1)+.......+\beta_m\phi_1(s_1)\phi_1(s_1)=-1$$

$$\beta_1\phi_1(s_1).\phi_1(s_2)+\beta_2\phi_1(s_2)\phi_1(s_2)+.......+\beta_m\phi_1(s_2)\phi_1(s_1)=+1$$

The support vectors are identified from each class, $s_1=\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\end{bmatrix}$ $s_2=\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\end{bmatrix}$ and $s_m=\begin{bmatrix}s_{m1}\\s_{m2}\\\vdots\\s_{mn}\end{bmatrix}$

The add biased value 1 to chosen vectors, $\dot{s}_1=\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}$ $\dot{s}_2=\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}$ and $\dot{s}_m=\begin{bmatrix}s_{m1}\\s_{m2}\\\vdots\\s_{mn}\\1\end{bmatrix}$

$$\beta_1(\overset{*}{s}_1).(\overset{*}{s}_1)+\beta_2(\overset{*}{s}_2)(\overset{*}{s}_1)+.......+\beta_m(\overset{*}{s}_m)(\overset{*}{s}_1)=-1$$

$$\beta_1(\overset{*}{s}_1).(\overset{*}{s}_2)+\beta_2(\overset{*}{s}_2)(\overset{*}{s}_2)+.......+\beta_m(\overset{*}{s}_m)(\overset{*}{s}_2)=+1$$

$$\beta_1\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}.\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}+\beta_2\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}.\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}+........+\beta_m\begin{bmatrix}s_{m1}\\s_{m2}\\\vdots\\s_{mn}\\1\end{bmatrix}.\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}=-1$$

$$\beta_1\begin{bmatrix}s_{11}\\s_{12}\\\vdots\\s_{1n}\\1\end{bmatrix}.\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}+\beta_2\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}.\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}+........+\beta_m\begin{bmatrix}s_{m1}\\s_{m2}\\\vdots\\s_{mn}\\1\end{bmatrix}.\begin{bmatrix}s_{21}\\s_{22}\\\vdots\\s_{2n}\\1\end{bmatrix}=+1$$

We simplify the equation to obtain the values β_1, β_2 and β_m

Consequently, we find the hyperplane that classifies in the feature space.

$$\overset{*}{w}=X_i\beta_i\overset{*}{s}_i$$

$$\overset{*}{w}=\beta_1.\overset{*}{s}_1+\beta_2.\overset{*}{s}_2+.......+\beta_m.\overset{*}{s}_m$$

Substitute the values in above equation, Therefore, we get $\dot{w}$ and b.

The hyperplane equation is formulated as $y=\overset{*}{w}x+b$ where $\overset{*}{w}$ represents the weighted vector and b denotes the bias term. This mathematical expression precisely defines the decision boundary separating different classes in the feature space.

CONCLUSION

Our research demonstrates the efficacy of using nonlinear mapping functions to transform clustered, non-linearly separable data to a new feature space, thereby enhancing the performance of Support Vector Machines (SVM). By developing and applying these mapping functions, we effectively reveal a clear linear decision boundary within the feature space, facilitating more accurate classification of nonlinear data. We illustrate how varying the parameter n_R influences the transformation, with comparative results for $n_R = 3$, $n_R = 4$ and $n_R = 5$ confirming the approach's robustness. This approach not only enhances our insight into the inner mechanisms of nonlinear models but also facilitates the assessment of the significance of different feature combinations, thereby pushing forward the frontiers of machine learning.

Acknowledgement: The author would like to thank REVA University for their encouragement and support in carrying out this research work.

REFERENCE

[1] Huang, K.; Wen, H.; Ji, H.; Cen, L.; Chen, X.; Yang, C. "Nonlinear process monitoring using kernel dictionary learning with application to aluminum electrolysis process". *Control Eng. Pract.* 2019, 89, 94–102.

[2] Zhou, Z.; Du, N.; Xu, J.; Li, Z.; Wang, P.; Zhang, J. "Randomized Kernel Principal Component Analysis for Modeling and Monitoring of Nonlinear Industrial Processes with Massive Data" *Ind. Eng. Chem. Res*. 2019, 58, 10410–10417

[3] Deng, J.; Deng, X.; Wang, L.; Zhang, X. "Nonlinear Process Monitoring Based on Multi-Block Dynamic Kernel Principal Component Analysis". *In Proceedings of the 2018 13th World Congress on Intelligent Control and Automation (WCICA), Changsha China*, 4–8 July 2018; pp. 1058–1063.

[4] Zhang, H.; Deng, X.; Zhang, Y.; Hou, C.; Li, C.; Xin, Z. "Nonlinear Process Monitoring Based on Global Preserving Unsupervised Kernel Extreme Learning Machine". *IEEE Access 2019*, 7, 106053–106064.

[5] Guo, L.; Wu, P.; Gao, J.; Lou, S. "Sparse Kernel Principal Component Analysis via Sequential Approach for Nonlinear Process Monitoring". *IEEE Access 2019*, 7, 47550–47563.

[6] Liu, Y.; Wang, F.; Chang, Y.; Gao, F.; He, D. "Performance-relevant kernel independent component analysis based operating performance assessment for nonlinear and non-Gaussian industrial processes". *Chem. Eng. Sci*. 2019, 209, 115167.

[7] Wang, L. Deng, X. "Modified kernel principal component analysis using double-weighted local outlier factor and its application to nonlinear process monitoring". *ISA Trans. 2018*, 72, 218–228.

[8] Jiao, J.; Zhao, N.; Wang, G.; Yin, S. "A nonlinear quality-related fault detection approach based on modified kernel partial least squares". *ISA Trans.* 2017, 66, 275–283

[9] Cai, L.; Tian, X.; Chen, S. "Monitoring nonlinear and non-Gaussian processes using Gaussian mixture model-based weighted kernel independent component analysis". *IEEE Trans. Neural Netw. Learn. Syst.* 2017, 28, 122–135.

[10] Yi, J.; Huang, D.; He, H.; Zhou, W.; Han, Q.; Li, T. "A novel framework for fault diagnosis using kernel partial least squares based on an optimal preference matrix". *IEEE Trans. Ind. Electron.* 2017, 64, 4315–4324.

[11] Pavithra C.; Saradha M. "Classification and Analysis of Clustered Non-Linear Separable Data Set using Support Vector Machines". *Migration Letters* Volume: 21, No: S4 (2024), pp. 901-913 ISSN: 1741-8992.