A STUDY TO INVESTIGATE MACHINE LEARNING BASED MODELS UTILIZING OPTIMAL KERNEL-GENERATED SURFACES TO ADDRESS CLASSIFICATION CHALLENGES

A Thesis

Submitted towards the Requirement for the Award of Degree of

Doctor of Philosophy

In

COMPUTER SCIENCE & ENGINEERING

Under the Faculty of ENGINEERING & TECHNOLOGY

By

V RAJANIKANTH TATIRAJU

(Enrollment No.: 161588517531)

Under the Supervision of

Dr. Rohita Yamaganti
Professor
P.K. University

2025

P.K. UNIVERSITY

NH-27,Vill. Thanra (P.O. - DINARA),

Shivpuri M.P. India–473665

www.pkuniversity.edu.in

CERTIFICATE OF THE SUPERVISOR

This is to certify that the work entitled **"A Study to Investigate Machine Learning Based Models Utilizing Optimal Kernel-Generated Surfaces to Address Classification Challenges "** is a piece of research Work done by **Mr. V Rajanikanth Tatiraju** Under My Guidance and Supervision for the degree of Doctor of Philosophy of **faculty of Computer Science & Engineering, P.K. University (M.P) India.**

I certify that the candidate has put an attendance of more than 240 day with me. To the best of my knowledge and belief the thesis:

I –Embodies the work of the candidate himself/herself.
II – Has duly been completed.
III –Fulfil the requirement of the ordinance relating to the Ph.D. degree of the University.

Signature of the Supervisor

Date: …….

DECLARATION BY THE CANDIDATE

I declare that the thesis entitled "**A Study to Investigate Machine Learning Based Models Utilizing Optimal Kernel-Generated Surfaces to Address Classification Challenges"** is my own work conducted under the supervision of **Dr.Rohita Yamaganti** Approved by Research Degree Committee. I have put more than 240 days of attendance with supervisor at the center.

I further declare that to the best of my knowledge the thesis does not contain my part of any work has been submitted for the award of any degree either in this University or in any other University

Without proper citation.

Signature of the candidate

Date: ……

Place: …….

FORWARDING LETTER OF HEAD OF NSTITUTION

The Ph.D. thesis entitled "**A Study to Investigate Machine Learning Based Models Utilizing Optimal Kernel-Generated Surfaces to Address Classification Challenges** " Submitted by **Mr. V Rajanikanth Tatiraju is** forwarded to the university in six copies. The candidate has paid the necessary fees and there are no dues outstanding against him/her.

Name ……..…….…. Seal ……

Date: ……

Place: ……...

(Signature of Head of Institution where the Candidate was registered for Ph.D.degree)

Signature of the Supervisor

Date: ……..

Address: ……..

Place: …….

……..

ACKNOWLEDGEMENT

The Acknowledgement of a research endeavor is an opportunity to express appreciation for the joint effort of those who have been sources of motivation and support, and who have contributed invaluable knowledge towards achieving success in the pursuit of knowledge. With utmost humility and pride, I reflect on the inspiring individuals who have propelled my work forward, and I am filled with immense pleasure, gratitude and thanks for their unwavering dedication.

First for most I would like to thank almighty god for perseverance, courage and strength bestowed on me to undertake this project.

For most, I extend my deepest appreciation to my esteemed Supervisor, **Prof. (Dr.) Rohita Yamaganti**, Professor of Computer Science & Engineering, P.K University Shivpuri (M.P) without whom this work would not have come to fruition. Her exceptional guidance, unwavering support kind cooperation and provision of a conductive work environmental have been invaluable beyond measure. I remain indebted to her instilling in me a relentless pursuit of excellence, a strong sense of honesty, and respect for ethical principles that govern our profession.

Furthermore, I extend my sincere thanks to the **Prof. (Dr.) Shri Jagdeesh Prasad Sharma** Chancellor, P.K University Shivpuri (M.P) for providing the necessary infrastructure and facilities that were essential during my research work. My profound sense of gratitude, faith, and awe also goes to **Prof. (Dr.) Yogesh Chandra Dubey** Vice Chancellor, P.K University Shivpuri (M.P) for his generous and valuable support throughout my research.

I take this pleasant opportunity to express my deep sense of gratitude and reverence to his holiness **Prof. (Dr.) Jitendra Kumar Mishra** Director; P.K University Shivpuri (M.P) for his constant encouragement and providing me with the required facilities that enabled me to complete my project work.

I would also like to extend my gratitude to and deep appreciation to respected **Prof (Dr). Bhaskar Nalla,** Dean Research of P.K University M.P, I/c Ph.D cell of P.K University (M.P). I also extend my deepest appreciation to **Prof. (Dr.) Jitendra Kumar Malik,** Dean of Faculties, P.K University Shivpuri (M.P) for his valuable support and

encouragement throughout my research journey. **Prof (Dr.) Deepesh Naamdev** Registrar P.K University Shivpuri (M.P) who has inspired me with his encouraging guidance and support right from the beginning of this work in spite of his busy schedule . I am short of words to thank him for his affectionate behavior and patience throughout the duration of research work.

I express my gratitude to **Prof. Dr. Aiman Fatima,** Dean Academics of P.K University Shivpuri M.P

I owe a great deal of application and gratitude to the entire faculty members Department of Computer Science & Engineering P.K University, Head of Department, Computer Science & Engineering, Prof. (Dr.) Balveer Singh, Prof. Renu, Prof. Megha (Asst.Prof), and all non-teaching staff of the department without their support it would have been difficult to shape my research work

I also express my gratitude to **Miss Nisha Yadav,** Librarian of P.K University along with Mr. Anand Kumar, Assistant Librarian and other staff of P.K University. I am also thankful to Mr. Kamlesh Yadav and Mr. Alok Khare, I.T. Cell of P.K. University.

I sincerely thank my loving parents who has always been there with me in every situation. Last but not the least, I am extremely thankful to my beloved brother Mr.T.V.S.Ramu for their moral support, patience and encouragement throughout the research work. Above all, I thanks to almighty for granting me health, strength & the wisdom to commence this research work and enabling me to its completion

Last but not least I would like to honestly appreciate the fact that it is my pleasure to have an opportunity to pursue my higher education at P.K University, Shivpuri (M.P). I owe my permanent debt towards it, this is having the special place in my heart for shaping my care.

Date: **(V.Rajanikanth Tatiraju)**

Place:

LIST OF ABBREVIATIONS

Abbreviations	Indication
RBF	Radial Basis Function
ML	Machine Learning
SVMs	Support Vector Machines
ANN	Artificial Neural Networks
NLP	Natural Language Processing
ACFSVM	Acquired Conditional Probability Support Vector Machine
SGTSVM	Stochastic Gradient Twin Support Vector Machines'
KRR	Kernel Ridge Regression
GPs	Gaussian Processes
TWSVM	Twin Support Vector Machines
SML	Supervised Machine Learning
MDT	Multinomial Decision Trees
MLR	Multinomial Logistic Regression
MNN	Multilayer Neural Networks
PCA	Principal Component Analysis
K-NNs	K-Nearest Neighbors
FPS	Frames Per Second
NLI	Natural Language Inference
IDS	Intrusion Detection System
TSVR	Twin Support Vector Regression
URALTSVR	Unconstrained Robust Asymmetric Lagrangian N-Twin Support Vector Regression
RILTELM	Regularized Based Implicit Lagrangian Twin Extreme Learning Machine

LIST OF TABLES

LIST OF FIGURES

P.K.UNIVERSITY
SHIVPURI (M.P.)

UniversityEstablished Under section 2F of UGC ACT 1956 Vide MP Government Act No 17 of 2015

CENTRAL LIBRARY

Ref. No. PKU/C.LIB /2025/PLAG. CERT./204 Date: 17.06.2025

CERTIFICATE OF PLAGIARISM REPORT

1. Name of the Research Scholar	:	V Rajanikanth Tatiraju
2. Course of Study	:	Doctor of Philosophy (Ph.D.)
3. Title of the Thesis	:	A Study to Investigate Machine Learning Based Models Utilizing Optimal Kernel-Generated Surfaces to Address Classification Challenges
4. Name of the Supervisor	:	Dr. Rohita Yamaganti
5. Department	:	Computer Science & Engineering & IT
6. Subject	:	Computer Science & Engineering
7. Acceptable Maximum Limit	:	10% (As per UGC Norms)
8. Percentage of Similarity of Contents Identified	:	3%
9. Software Used	:	Drillbit
10. Date of Verification	:	20.05.2025

(Librarian, Central Library)
P.K.University, Shivpuri (M.P.)

LIBRARIAN
P.K. University
Shivpuri (M.P.)

ADD: VIL: THANRA, TEHSIL: KARERA, NH-27, DIST: SHIVPURI (M.P.) -473665
MOB: 7241115902, Email: library.pku@gmail.com

DrillBit

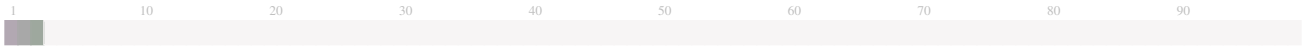
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

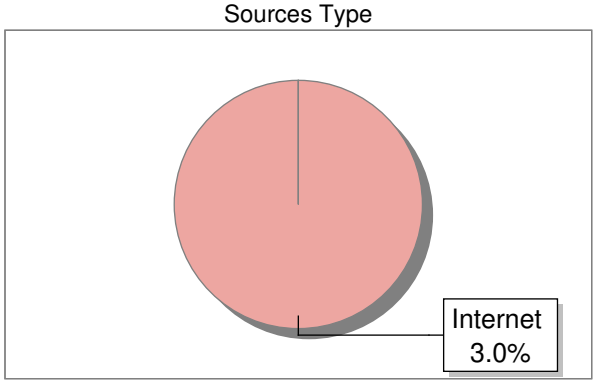
Author Name	V RAJANIKANTH TATIRAJU
Title	A STUDY TO INVESTIGATE MACHINE LEARNING BASED MODELS UTILIZING OPTIMAL KERNEL-GENERATED SURFACES TO ADDRESS CLASSIFICATION CHALLENGES
Paper/Submission ID	3637678
Submitted by	library.pku@gmail.com
Submission Date	2025-05-20 11:54:31
Total Pages, Total Words	254, 55646
Document type	Thesis

Result Information

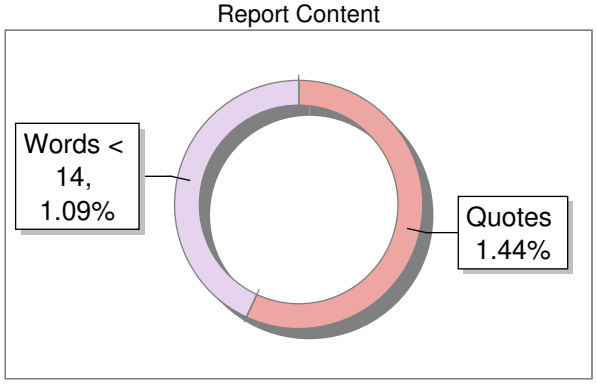
Similarity **3 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	No
Internet or Web	Yes
Institution Repository	No

A Unique QR Code use to View/Download/Share Pdf File

COPYRIGHT TRANSFER CERTIFICATE

Title of the Thesis: **"A Study to Investigate Machine Learning Based Models Utilizing Optimal Kernel-Generated Surfaces to Address Classification Challenges "**

Candidate's Name: **Mr. V Rajanikanth Tatiraju**

COPYRIGHT TRANSFER

The undersigned hereby assigns to the P.K. University, Shivpuri all copyrights that exist in and for the above thesis submitted for the award of the Ph.D. degree.

Date: **Mr. V Rajanikanth Tatiraju**

ABSTRACT

From healthcare diagnostics to financial forecasts and picture identification, machine learning (ML) has become an essential tool for tackling difficult categorization issues in many different domains. Out of all the ways out there, kernel-based approaches have grown popular because they can deal with data that has non-linear correlations by converting it into higher-dimensional spaces where classes are linearly separable. When dealing with multidimensional, noisy, or unbalanced datasets, optimizing these kernel-generated surfaces becomes much more challenging. The purpose of this research is to examine machine learning models that successfully handle these classification problems by making use of optimum kernel-generated surfaces.

In order to create new, better models that can surpass existing methods, this study aims to examine current models for regression and classification-based learning problems, especially twin variants of SVM and ELM models. We propose a number of regression models that improve prediction accuracy while addressing some of the shortcomings of TSVR-based models, including inefficiency in processing, noise and outlier impacts, overfitting, and lack of knowledge about data distribution. In order to obtain lowest learning, cost with higher prediction performance, we solve a system of linear equations or use popular gradient-based algorithms to address unconstrained minimization issues. We further investigate a small number of improved models for classification problems that are based on optimal non-parallel kernel generated surfaces. These models aim to address the aforementioned challenges, such as reducing the substantial computational overhead, improving generalizability, and reducing noise sensitivity. Here, we bypass the need for QPPs in their dual problems by transforming the limited optimization issues into unconstrained minimization problems. Then, we solve these problems using either a generalized derivative technique, smoothing schemes, or functional iterative approach. We also address resilient loss functions for various twin versions of support vector machines (SVMs) used for classification and regression learning. In order to compare all of the suggested methods to different state-of-the-art methods on different performance measures, they are all tested extensively. There is promising evidence from the experiments that the proposed methods work. This study adds to the growing body of knowledge on machine learning techniques and sheds light on how to tackle important classification problems using kernel-based approaches. This approach might

be useful in areas where accurate categorization is crucial, such healthcare, banking, and NLP.

.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Machine learning (ML) has changed several industries by giving new ways to solve old issues, especially those involving categorization. Researchers have been putting a lot of effort into using kernel-based approaches to improve classification accuracy. By raising the dimensionality of the data to a level where linear separation is possible, these techniques improve how well traditional algorithms work. Investigating the most effective kernel-generated surfaces is an innovative approach that might greatly alleviate the classification problems encountered by more traditional models.

The kernel technique gives algorithms the ability to work in high-dimensional feature spaces without having to manually calculate the data's coordinates; this makes kernel-based approaches fundamental. In situations when the distribution is not linearly separable, this method allows for the extraction of intricate patterns and correlations from the data. Several machine learning models have made use of different kernel functions, including linear, polynomial, and radial basis function (RBF) kernels. Nevertheless, in order to get the greatest classification performance, it is essential to pick the appropriate kernel for a particular dataset. In order to improve machine learning models' classification accuracy, this work aims to investigate how to develop and use appropriate kernel-generated surfaces.

An algorithm's performance is heavily dependent on the kernel function that is used. There are advantages and disadvantages to each kernel function that could influence how well the model understands the data. For example, RBF and other non-linear kernels perform better on complicated datasets, but linear kernels excel on linearly separable data. Parameters, in addition to the kind of kernel, are the primary determinants of the kernel's efficacy. Thus, optimizing kernel parameters is crucial for improving classification results by honing the model's decision limits.

Problems like the curse of dimensionality may arise when working with high-dimensional data, which is a major obstacle to using machine learning for categorization. This issue arises when the feature space is too sparse, which hinders the ability of models to apply training data to new, unknown occurrences.

Optimal kernel-generated surface research is an attempt to solve this problem by providing a more understandable and tractable data representation. The goal of the research is to improve the models' capacity to capture the important data features while reducing the negative impacts of high dimensionality by identifying the best-fit surfaces.

Finding the optimum settings for a kernel and choosing the right kernel function are two parts of the optimal kernel-generated surfaces idea. Finding the optimal parameter settings for optimal classification performance is a common challenge in this process and often calls for advanced optimization methods like grid search or Bayesian optimization. Incorporating methods like cross-validation further guarantees that the parameters chosen are strong and can be applied to fresh data. We hope that by thoroughly examining these surfaces, we might shed light on the connections between model complexity, classification accuracy, and kernel parameters.

Evaluating the final classification models is an important part of this study, much as kernel optimization is. The success of the machine learning models using optimum kernel-generated surfaces will be evaluated using standard performance measures including recall, accuracy, precision, and F1 score. In addition, confusion matrices may help us understand the kinds of categorization mistakes we're making, which helps us fine-tune our models. This thorough assessment will help shed light on the effect of kernel settings on classification performance across different datasets.

1.2 MACHINE LEARNING

Machine learning (ML) is a subfield of AI concerned with creating models and algorithms that computers may use to discover new patterns in data and perform better on their own, without human intervention. In the last few decades, ML has grown into a major technical breakthrough across many different industries, influencing fields as diverse as marketing, healthcare, finance, and autonomous systems.

Machine learning, in mathematics, is the practice of developing algorithms that can learn from past data and use that knowledge to make predictions about new, unknown data. Machine learning (ML) is all about how well it can adapt to new situations; as it learns from data, it becomes better at performing similar jobs in the future.

Supervised, unsupervised, semi-supervised, some of the most famous techniques in machine learning include reinforcement learning and other similar approaches. Different approaches employ different types of data (labeled vs. unlabeled) and different kinds of feedback (trial and error, etc.) to discover patterns or make judgments.

1.2.1 MACHINE LEARNING TECHNIQUES

Algorithms for machine learning aim to learn autonomously, without any help from humans. Since learning is fundamental to intelligence, machine learning forms the backbone of AI. A variety of machine learning approaches are available, including:

Supervised Learning

Predictions are made for certain data samples using these algorithms. The data and labels used to generate the entry are classified as spam or non-spam. A training technique gets a model ready for use in making predictions and, if necessary, in making adjustments to those predictions. The model is trained until it reaches the critical accuracy of the training data.

Support Vector Machine

This is designed to address issues with regression and classification. Support vector machines (SVMs) divide training data into classes by finding a hyperplane (line). Your chances of generalizing unseen data improve if you find the hyperplane that optimizes the distance between classes.

In terms of classification performance, or the accuracy of the training set, SVM provides the best option. The data is not overflown. Time series analysis is where support vector machines (SVMs) shine. [1]

SVM refrains from making robust data assumptions. Make better use of resources to ensure accurate data categorization in the future.

There are two types of support vector machines: linear and non-linear. A linear method uses a line—a hyperplane—to depict the training data.

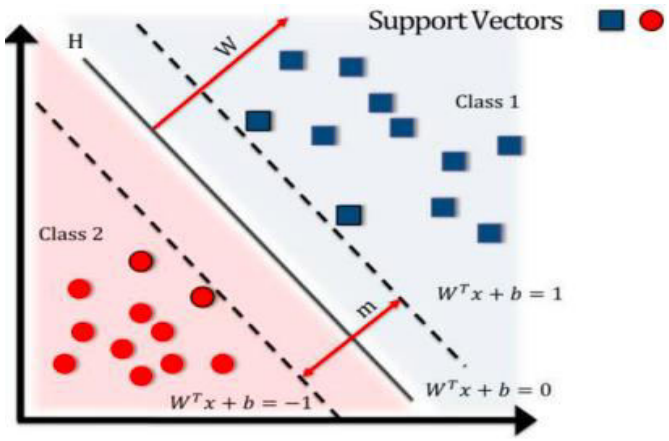


FIGURE 1. 1 SVM Classifier

(Source: Secondary source taken from https://link.springer.com/article/10.1007/s42452-024-06244-y)

k-Nearest Neighbor (kNN)

For issues involving classification and regression, kNN is used (Figure 1.2). In terms of categorization algorithms, this is among the most basic. Finds the value of the parameter k, which represents the count of the closest neighbors. In order to classify new data points, the training data is used to find their closest neighbors. One of the three distance measures—the Minkowski, the Mahalanobis, or the one based on the equations of geometry—is used to determine the distance. A higher value for k indicates a more accurate categorization.

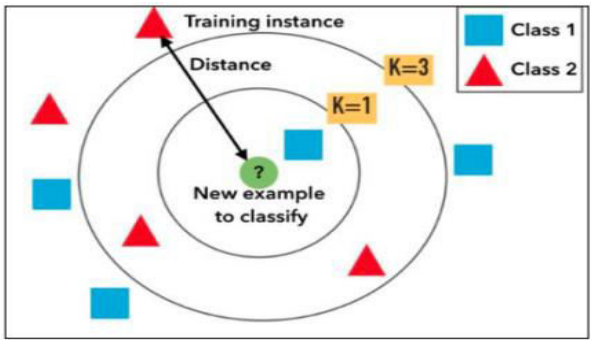


FIGURE 1. 2 KNN Algorithm

(Source: Secondary source taken from https://github.com/mrolarik/basic-machine-learning-using-scikit-learn/blob/master/007-K-Nearest-Neighbor-(KNN).ipynb)

Decision Tree Algorithms

The decision model benefits from the particular values of the data's attributes. Prior to making a prediction judgment, all records are kept in the decision interval. A destination variable is predefined for it. Classification and regression problems are addressed by training decision trees using the available information. In machine learning, decision trees are well-liked because of how quickly and accurately they work. [2] It functioned well with both continuous and categorical input and output data. This method takes the input variables as a starting point and uses them to partition the population or sample into two or more similar subpopulations on top of each other.

Decisions in the strategic branch are based on a tree diagram. As a result, the tree's dependability is severely diminished. This criterion for decision-making is distinct for classification trees and regression trees, as shown in Figure 1.3. The decision to split a node into two or more subnodes is made by decision trees using unique methods. For each available variable, the trees split the nodes, and the tree with the most homogeneous branches is chosen. C4.5, C5.0, and CART are the most popular decision tree algorithms (Classification and Regression Tree).

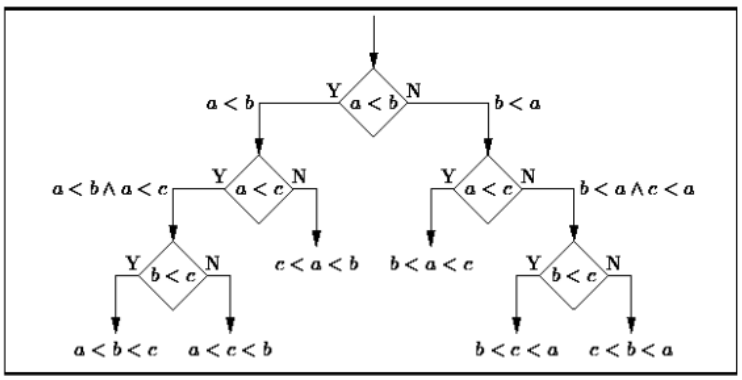


FIGURE 1. 3 Decision Tree Algorithm

(Source: Secondary source taken from https://www.linkedin.com/pulse/10-algorithms-machine-learning-engineers-need-know-james-le)

Neural Network

Classification is seen by artificial neural networks (ANN) (Figure 1.4) as a very active field of study and application. As the amount of entities and sets of phrases grows, the

biggest challenge with RNA is determining how to best organize training, learning, and transfer functions for record classification. When using ANN as a classifier, we look at the various function combinations and how they work, as well as the accuracy of these functions on various types of data.

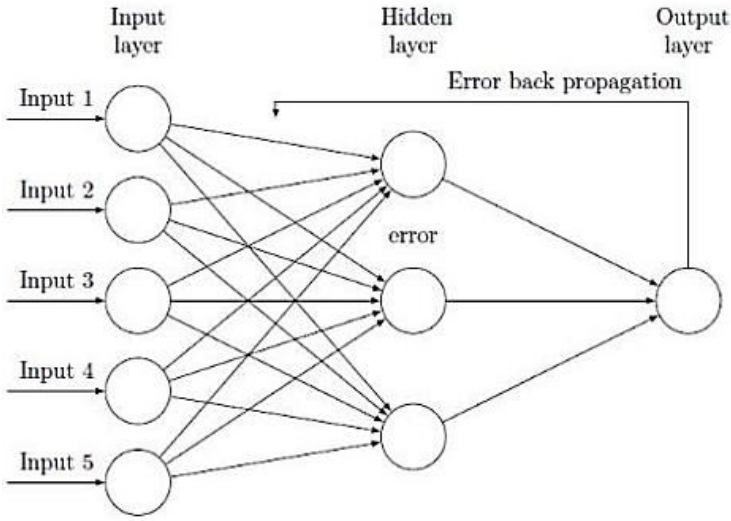


FIGURE 1. 4 Artificial Neural Network

(Source: Secondary source taken from https://www.researchgate.net/figure/Structure-of-a-multilayer-neural-network-with-only-one-hidden-layer_fig5_372300002)

The healthcare industry is a real-world example of a domain where multidimensional datasets have proven useful. Important considerations went into the categorization and naming of these documents. The registration process is separate from the training method and is utilized for educational games and game examinations. Using these data, the results are generated and put to use in the testing. Part two of the recording serves as the training set, while part three serves as the test set. This is accomplished by assessing the precision that these records have yielded via testing. This leads to using the same data to replicate the network. To guide the neural network, the back propagation technique is used.

To lower the mean square error between the network's output and the real error rate, the gradient decay technique (GDM) was used.

Unsupervised Learning

The goal of unsupervised learning algorithms in machine learning is to derive conclusions from datasets that include input data without indicated responses. [3]

Cluster analysis is the most popular non-supervised learning technique. It is used to examine exploration data in order to uncover latent models or group data. When modeling clusters, metrics like Euclidean or probabilistic distance are used to establish the similarity measure.

Popular methods for clustering data consist of:

- **Hierarchical clustering:** builds a multilevel hierarchy of clusters by creating a cluster tree.
- **K-Means clustering:** partitions data into k distinct clusters based on distance to the centroid of a cluster.
- **Gaussian mixture models:** models clusters as a mixture of multivariate normal density components.
- **Self-organizing maps:** uses neural networks that learn the topology and distribution of the data.
- **Hidden Markov models:** uses observed data to recover the sequence of states.

1.2.2 ROLE OF MACHINE LEARNING IN CLASSIFICATION

Classification tasks, which include sorting data points into predetermined groups or labels using input attributes, rely heavily on machine learning (ML). [4]

To succeed in classification, one must first create a model that can take labelled training data and use it to reliably assign a class or label to previously unknown occurrences.

Machine learning is so effective because it can sift through massive datasets for structures, correlations, and patterns; this allows models to generalize effectively and swiftly complete complicated categorization tasks.

Pattern Recognition and Generalization

Machine learning's pattern recognition capabilities are crucial in the categorization

process. It is not possible to handle big or complicated datasets using standard rule-based systems since rules to classify data must be defined by people. But machine learning algorithms can figure out the connections between characteristics in the input and labels in the output on their own. To distinguish spam from real emails, a machine learning model examines characteristics like word frequency, sender information, and email metadata, among others. This process is known as spam email categorization. When the model has been trained, it may use the patterns it has learnt to accurately categorize fresh emails.

Machine learning is very good at generalizing, which means it can use what it has learned from training data to solve problems with previously unknown examples. In real-world applications, this capacity is vital since data is dynamic and changes continually. Machine learning models are able to generalize effectively, which means they can handle new data kinds and patterns with ease. As an example, a model that was trained to categorize medical photographs might discover new illnesses by observing commonalities among comparable images, even if the training dataset did not include images of that particular condition.

Handling Complex and High-Dimensional Data

Classification jobs often include high-dimensional and complicated data, which machine learning excels at managing. Datasets having a large number of characteristics, such pictures, audio, or text, could be difficult for traditional statistical approaches to handle. However, these massive datasets are no match for machine learning algorithms. Face recognition is one area where images might have hundreds of pixel values that could represent intricate patterns. In particular, deep learning models of neural networks are able to learn and integrate hierarchical characteristics (such forms, textures, and edges) from raw data in order to provide correct classifications.

Not only that, but machine learning algorithms are able to deal with feature-output label non-linearities. Relationships between variables in real-world issues are usually not linear, therefore this becomes even more crucial. For instance, decision trees and support vector machines (SVMs) are two examples of machine learning models that may categorize consumer behavior using demographic and transactional data. These models can detect non-linear patterns that more conventional linear models might

overlook. Machine learning models thus demonstrate superior performance compared to conventional approaches in non-linear and high-dimensional classification problems.

Scalability and Efficiency

The efficient scalability of categorization jobs is greatly facilitated by machine learning. The capacity to real-time categorize massive information is becoming more and more crucial as data volumes continue to rise across many industries. Data processing and classification speed is of the essence in several applications, including recommendation systems, content moderation, and fraud detection. Through the process of learning from massive volumes of data and creating real-time predictions, machine learning models are able to handle these large-scale datasets.

Online retailers, for instance, use machine learning algorithms to instantly categorize shoppers' orders, tastes, and actions as they peruse products. Businesses may use these categories to better serve their customers by making more informed suggestions, making inventory management more efficient and better for customers overall. Similarly, autonomous vehicles use machine learning models to classify objects like people, vehicles, and traffic signals based on real-time data processed by sensors. The car can now drive itself safely thanks to this.

In addition, the scalability of machine learning allows it to deal with data quantities that are always growing. Machine learning model training on massive datasets is now a breeze thanks to cloud computing and distributed processing. leading to improved classification speed and accuracy. A major benefit of machine learning is its scalability, which makes it possible to use it in situations that are always changing and producing new data.

Adaptability and Real-Time Learning

The flexibility of machine learning is equally crucial when it comes to categorization. Data changes over time in many real-world applications, necessitating model adaptation to new patterns or distributional shifts in the underlying data. Machine learning models may be trained to increase their classification accuracy by continually updating themselves with fresh data. This allows them to learn from ongoing interactions. This is particularly helpful in contexts where user tastes could change over

time, like recommendation systems, or when fraudulent behaviors might develop over time, like fraud detection.

Two models that enable real-time adaptation are online learning and reinforcement learning. By adding fresh data as it becomes available, online learning allows models to be trained progressively. This way, they can adapt to novel patterns without starting the model's training process again. Agents engage in reinforcement learning when they improve their decision-making abilities via trial and error by learning to categorize using environmental input. These methods allow machine learning models to maintain their effectiveness in situations that are constantly evolving.

Probabilistic Predictions and Uncertainty Management

Making probabilistic predictions is a common use case for machine learning classification models, particularly in contexts with inherent uncertainty. These models, instead of giving a definitive designation, may provide probabilities that represent the chances of each class. Domains like medical diagnostics benefit greatly from this as it allows healthcare providers to make more educated judgments based on the likelihood of an illness.

It is fairly uncommon for machine learning models to provide probability scores when employed for illness classification; for instance, a "positive diagnosis" may be 0.85 and a "negative diagnosis" could be 0.15. By doing so, they may gauge the prediction's reliability and take into account other variables, such clinical signs or more testing, before reaching a conclusion. Models that provide probabilistic results, such as ensemble techniques, logistic regression, and Naive Bayes, may help with uncertainty management and better classifications.

Applications like financial fraud detection or credit scoring need careful uncertainty management due to the substantial ramifications of erroneous positives or negatives. The use of machine learning models facilitates the establishment of suitable decision-making thresholds by providing a quantitative assessment of this uncertainty. For example, when calculating risk, a customer's credit score can indicate that the model thinks they have a 70% likelihood of repaying the loan.

Versatility across Domains

A major factor in machine learning's meteoric rise to classification dominance is its adaptability. Natural language processing (NLP), image identification, healthcare, and finance are just a few of the many fields that can benefit from its use. Medical picture classification, illness outcome prediction, and medication development are all areas where machine learning models are finding use in healthcare. They identify fraudulent transactions, categorize creditworthiness, and predict market movements in the financial sector. Language translation, sentiment analysis, and text categorization are just a few examples of the many NLP jobs that make use of machine learning.

Machine learning is able to solve problems more effectively than traditional approaches since it can generalize across domains. As an example, machine learning algorithms sort through photos, videos, and text on social networking sites to find explicit or hate speech. Just like human drivers use machine learning models to safely navigate complicated traffic conditions, autonomous cars also use these models to categorize items in their environment.

1.3 SVM-BASED CLASSIFICATION METHODS

Support vector machine (SVM) was first created to address binary classification issues; it is a reliable classification system that places an emphasis on avoiding structural hazards rather than empirical risks. In order to maximize model complexity while minimizing misclassification errors, support vector machines (SVMs) are used. [5] A positive label of +1 in support vector machines (SVM) denotes one set of classes and a negative label of -1 another. It is the goal of conventional support vector machines (SVMs) to maximize the distance between two parallel hyperplanes and two border hyperplanes, each of which touches one class. These two boundary hyperplanes intersect with data samples that are known as support vectors. The calculation is carried out in such a manner that the last hyperplane for classification goes through the center of the region enclosed by the two hyperplanes on the borders, which are spaced one relative distance apart. Moreover, if the classes in the input space cannot be separated linearly, support vector machines have the capability to change data samples into a higher dimensional feature space by means of a mapping function φ (.). This procedure makes the classes separate in the feature space. Based on the data sample x, the hyperplanes' normal vector b, and the intersection point b, also known as the bias, the final hyperplane for classification is $\varphi(x)t + b = 0$. The boundary hyperplanes for the

±1 class are now given as φ (x)t ω + b = ±1. It would be wise to consider, φ(X) = [φ(x1),φ(x2),…,φ (xl)]t be the feature space data matrix and e denote a vector of length l of binary ones. As an expression, the SVM optimization problem may be written as:

$$min\frac{1}{2}\|\omega\|^2$$

$$s.t.y_i(\varphi(x_i)^t\omega + b) \geq 1, \quad i = 1,2,\dots,l. \tag{1.1}$$

A data sample is regarded correctly categorized if it falls above the border hyperplane of the +1 class, and correctly classed as -1 class if it falls below the hyperplane of the -1 class. Misclassified samples, also known as misclassified error points, are data samples that do not meet the aforementioned criteria. When it's necessary, a slack variable, or error term, is included to include a certain level of error in categorization into the goal functions of support vector machines (SVMs). The optimization problem for support vector machines with tolerance for errors is reformulated as follows:

$$min\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\xi_i$$

$$s.t.y_i(\varphi(x_i)^t\omega + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{1.2}$$

in which ξi where The misclassification error, or slack variable, in the objective function of is denoted by C, the tradeoff parameter between the two halves. Function (1.2). What about the phrase $\sum_{i=1}^{l}\xi_i$ although maximizing under the previously described conditions, where total error loss is the objective function. To begin, SVM's loss function has to be specified:

$$H_s(z) = \begin{cases} 0, & z > s \\ s - z, & z \leq s \end{cases} \tag{1.3}$$

Where s is the location of the hinge point, which is typically 1 for support vector machines. Many people refer to equation (1.3) defines the loss function as the Hinge loss function. This allows us to rewrite (1.2) in a more generic form as follows:

$$\min \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}H_1(y_i(\varphi(x_i)'\omega + b)) \tag{1.4}$$

Nonetheless, to get the solution, a quadratic polynomial in the dual space is solved. One

way to express the dual of equation (1.2) is by finding its matching Lagrangian and using the KKT requirements, which are both necessary and sufficient:

$$\min \quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i \varphi(x_i)^t \varphi(x_j) y_j \alpha_j - \sum_{i=1}^{l}\alpha_i$$

$$s.t. \begin{matrix} \sum_{i=1}^{l} y_i \alpha_i = 0, \\ 0 \le \alpha_i \le C. \end{matrix} \qquad (1.5)$$

At this point, the feature vectors' dot product φ (xi).φ (xj) or φ(xi)t φ(xj) may be substituted by a well-selected kernel function k (xi, x j) = φ(xi).φ(xj). Our underlying assumption seems to be that determining the mapping function in advance is computationally costly and that it is unknown a priori. Hence, we may restate (1.5) as follows:

$$\min \quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i k(x_i, x_j) y_j \alpha_j - \sum_{i=1}^{l}\alpha_i$$

$$s.t. \begin{matrix} \sum_{i=1}^{l} y_i \alpha_i = 0, \\ 0 \le \alpha_i \le C. \end{matrix} \qquad (1.6)$$

In the linear example, we need just think about k(xi, x j) = xtixj. The dual problem (1.6) has a computational complexity of O(l 3), which is typical for quadratic programming problems (QPP). Equation (1.6) may be solved using the support vectors to get the values of ω and b. Which side of the classifying hyperplane an unseen sample x falls on determines the class it is allocated. The classification equation has this form:

$$f_{(\omega,b)}(x) = \text{sign}(\varphi(x)^t \omega + b) \qquad (1.7)$$

Sparse vs dense models describe SVM, which classifies data using only the training dataset's support vectors.

1.3.1 LEAST SQUARES SUPPORT VECTOR MACHINE

The least square support vector machine (LS-SVM) fails to detect the boundary hyperplanes and instead identifies two hyperplanes near the border. By optimizing the distance between the two class hyperplanes, this hyperplane arrangement aims to get the samples from each class as close to them as feasible without sacrificing distance.

Crossing the center of the separation, the classifying hyperplane is located one unit distant from the class hyperplanes. The SVM slack variable's non-negativity condition is also removed when the LS-SVM goal function uses a quadratic least squares loss function. Here is the loss function defined using quadratic least squares:`

$$Q_0(z) = \frac{1}{2}z^2 \tag{1.8}$$

The optimization challenge for LS-SVM is as follows:

$$\min \quad \frac{1}{2}\|\omega\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2$$

$$s.t.\, y_i(\varphi(x_i)^t\omega + b = 1 - \xi_i \tag{1.9}$$

Solving a system of linear equations may provide the answer to the problem mentioned before. Given that LS-SVM doesn't need solving a massive QPP but rather a system of linear equations, training it is much faster than SVM. However, LS-SVM became less sparse as a result of classifying almost all of the training data samples. There is no difference between the classifier used by LS-SVM and the one specified in (1.7).

1.3.2 RAMP LOSS SUPPORT VECTOR MACHINE

As mentioned before, SVM uses the linear Hinge loss function to locate misclassification hotspots. Hinge loss is particularly susceptible to outliers and noise because to its intrinsic fragility.

The likelihood of distant mistake samples being able to aid in optimization is greater than that of close samples, due to their higher score. [6] The ramp loss function flattens the loss function when it hits a pre-specified score, as indicated in equation (1.10) below. This is done to address this.

$$R_s(z) = \begin{cases} 0, & z > 1 \\ 1 - z, & s \le z \le 1 \\ 1 - s, & z < s \end{cases} \tag{1.10}$$

Support vector machines are taught to be more robust against outliers and class noise using the Ramp loss function as opposed to the Hinge loss function. An abbreviated form of the original loss function, this is it. A non-convex cost function is the result of

RSVM using the non-convex Ramp loss function. When the convex and concave halves of this issue are put together, they create the following:

$$R_s(z) = \underbrace{H_1(z)}_{convex} - \underbrace{H_s(z)}_{concave} \tag{1.11}$$

One possible expression for the RSVM main optimization problem, which is the same as (1.11), is:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l} R_s(y_i(\varphi(x_i)^t\omega + b)) \\ & = \underbrace{\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l} H_1(y_i(\varphi(x_i)^t\omega + b))}_{convex} \underbrace{- C\sum_{i=1}^{l} H_s(y_i(\varphi(x_i)^t\omega + b))}_{concave} \end{aligned} \tag{1.12}$$

Equation (1.12)'s convex part is the classic expense function for support vector machines. An easy way to solve the problem we were talking about before is the concave-convex approach (CCCP).

1.3.3 MACHINE LEARNING FOR SUPPORT VECTORS USING PINBALL LOSS

For SVM-type formulations, the pinball loss function is an extra robust choice.. As a rule, the pinball loss function is:

$$L_\tau(z) = \begin{cases} z, & z \geq 0 \\ -\tau z, & z < 0 \end{cases} \tag{1.13}$$

The quantile distance between the two classes is used by SVM with pinball loss (pin-SVM) to optimize the margin between the two classes.

What follows is the primary QPP of pin-SVM:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(\varphi(x_i)^t\omega + b) \geq 1 - \xi_i, \\ & \tau(y_i(\varphi(x_i)^t\omega + b)) \leq \tau + \xi_i. \end{aligned} \tag{1.14}$$

Where, $\tau > 0$ proves to be a positive constant. With respect to $\tau = 0$, to solve the above stated issue, classical SVM is used. Comparable to pinball loss is the generic Hinge loss.

1.3.4 SUPPORT VECTOR MACHINE WITH FUZZY LOGIC

Fuzzy support vector machines (FSVMs) assign fuzzy membership values to data samples using the membership function. Values for sample membership that are incorrect are considered when choosing a final classifier. It follows that it's possible to improve generalization performance by decreasing the membership values of irrelevant samples. So, here is the key optimization challenge with FSVM:

$$\min \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l} m_i \xi_i$$

$$s.t. \quad y_i(\varphi(x_i)^t \omega + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0. \qquad (1.15)$$

On the other hand, the membership value of the i-th sample is called mi. Since well-classified samples do not have membership values, only error samples may utilize them. ξi = 0, as seen in the prior formulation. By determining the optimal membership function, FSVM is able to significantly reduce outliers and noise.

1.3.5 SUPPORT VECTOR MACHINES WITH FUZZY LOGIC FOR LEARNING ABOUT CLASS IMBALANCE

When presented with an unbalanced dataset, SVM favours the majority class. To make SVM more compatible with datasets that contain changing imbalance ratios, Batuwita and Palade (2010) proposed the FSVM-CIL as a fuzzy support vector machine for class imbalance learning. The membership functions used by FSVM-CIL are defined as follows:

$$m_i = \begin{cases} f_{mv}(x_i) r_+, & if \ y_i = +1 \\ f_{mv}(x_i) r_-, & if \ y_i = -1 \end{cases} \qquad (1.16)$$

Here, membership values between 0 and 1 are produced by the function fmv(.), and the

intervals [0, r±] are specified by r±.

The authors proposed setting r+= 1 and r-=l_1/l_2 with the -1 class representing the majority and the +1 class representing the minority.

1.3.6 SUPPORT VECTOR MACHINE FOR FUZZY ENTROPY

The entropy of a datum is the degree to which it is uncertain, according to information theory. The entropy of a sample could be a useful metric for finding out which class it belongs to. [7]

To address the imbalance problem, while entropy is the basis for EFSVM membership values, samples belonging to the minority class are assigned the value 1.

Here is the equation for entropy: where p_{+i} and p_{-i} are the probability of the ith sample belonging to the +1 and -1 classes, respectively.

$$H_i = – p_{+i} \ln(p_{+i}) – p_{–i} \ln(p_{–i}) \quad (1.17)$$

p+i and p–i are calculated across the input space by use of the k-nearest neighbor method. The authors go even farther by proposing a method to divide samples from the same class into several $\{sub_j\}_{j=1}^{q}$, such that, $H_{sub1} < H_{sub2} < ... < H_{subq}$.

The following is how EFSVM determines the samples' membership values in the jth subset: the -1 class is the majority and the +1 class is the minority:

$$m_i = \begin{cases} 1, & if\ y_i = +1 \\ 1 - \psi(j-1), & otherwise \end{cases} \quad (1.18)$$

where, $\psi \in \left(0, \frac{1}{q-1}\right)$ is used as the fuzzy membership specification.

1.3.7 ENTROPY-DRIVEN FUZZY LEAST SQUARES SVM FOR IMBALANCED DATA LEARNING

An modification of EFSVM tailored for class imbalance learning, EFLSSVM-CIL utilizes LS-SVM with entropy-based fuzzy membership values. Here is the optimization problem that EFLSSVM-CIL aims to fix:

$$\min \quad \frac{1}{2}\|\omega\|^2 + \frac{C}{2}\sum_{i=1}^{l} m_i \xi_i^2$$

$$s.t. \quad y_i(\varphi(x_i)^t \omega + b) = 1 - \xi_i \tag{1.19}$$

EFLSSVM-CIL trains more quickly and does not need to answer any QPPs because it solves a system of linear equations.

1.3.8 CLASS PROBABILITY AND AFFINITY-BASED FUZZY SUPPORT VECTOR MACHINE

Similar to EFSVM, another sort of fuzzy support vector machine called Acquired Conditional Probability Support Vector Machine (ACFSVM) uses methods use a majority-only membership metric and a minority-only metric equal to 1. The ACFSVM method, on the other hand, uses the sample's affinity and likelihood of belonging to a class to calculate membership values.

The input/feature space used to generate the hyperplanes is also used to determine the class probability and affinity. The kernel k-nearest neighbor technique and affinity are used to calculate the class probability pi of the i th sample $m_i^{affinity}$ is calculated using the SVDD technique. Finally, we take the -1 Choose the most common category and use it to get the sample membership values:

$$m_i = \begin{cases} 1, & if \ y_i = +1 \\ p_i m_i^{affinity}, & otherwise \end{cases} \tag{1.20}$$

In addition, the following optimization problem shows that ACFSVM classifies the overall mistake linked with the +1 and the -1 classes:

$$\min \quad \frac{1}{2}\|\omega\|^2 + C_1\sum_{j=1}^{l_1} m_j \xi_j + C_2\sum_{k=1}^{l_2} m_k \xi_k$$

$$s.t. \quad y_i(\varphi(x_i)^t \omega + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{1.21}$$

where, in practice, it is defined as $C_2 = C_1 rim$, where rim is the minority-majority

imbalance ratio.

1.3.9 TWIN SUPPORT VECTOR MACHINES

Twin support vector machines (TWSVM) locate two proximal hyperplanes for each class, which need not be parallel to one another, rather than two parallel class hyperplanes. All of the class hyperplanes are within one unit of distance from each class sample and quite close to the samples from their own classes. [8] In the linear situation, the hyperplanes with the +1 class are $x^t\,\omega_1 + b_1 = 0$, while the hyperplanes with the -1 class are $x^t\omega_2 + b_2 = 0$. ω_i, $b_i\{i = 1,2\}$ comprise the hyperplane mysteries. Because it solves two smaller-sized QPPs, TWSVM purportedly reduces time complexity by around four times when compared to SVM. One loss component of TWSVM's objective functions aims to minimize the proximal term, while the other minimizes the error term, which violates the criterion that samples from the opposite class are at least one unit far from the hyperplanes. The following are the matrix expressions of the first two optimization problems of linear TWSVM:

$$\begin{aligned} \min \quad & \frac{1}{2}\| X_1\omega_1 + b_1e_1 \|^2 + C_1e_2^t\xi_1 \\ s.t. \quad & -(X_2\omega_1 + e_2b_1) \geq e_2 - \xi_1, \\ & \xi_1 \geq 0e_2. \end{aligned} \tag{1.22}$$

And

$$\begin{aligned} \min \quad & \frac{1}{2}\| X_2\omega_2 + b_2e_2 \|^2 + C_2e_1^t\xi_2 \\ s.t. \quad & X_1\omega_2 + e_1b_2 \geq e_1 - \xi_2, \\ & \xi_2 \geq 0e_1. \end{aligned} \tag{1.23}$$

The slack vectors are symbolized by and the penalty or tradeoff parameters are $Ci\{i = 1, 2\}$. ξ_i, and the vectors of 1s of length li are called e_i. Here are the related TWSVM dual formulations:

$$\min \quad \frac{1}{2}\alpha_1^tQ(P^tP)^{-1}Q^t\alpha_1 - e_2^t\alpha_1$$

s.t. $0e_2 \leq \alpha_1 \leq C_1 e_2$ (1.24)

And

$$\min \quad \frac{1}{2}\alpha_2^t P(Q^t Q)^{-1} P^t \alpha_2 - e_1^t \alpha_2$$

s.t. $0e_1 \leq \alpha_2 \leq C_2 e_1$ (1.25)

where, $P = [X_1 e_1]$ and $Q = [X_2 e_2]$ and $\alpha_i\{i = 1,2\}$ these vectors represent Lagrange's multipliers. In nonlinear situations, TWSVM finds the class hyperplanes by mapping the input space to a kernel space $k(x^t, D^t)\omega_1 + b_1 = 0$ and $k(x^t, D^t)\omega_2 + b_2 = 0$. One of the main issues with nonlinear TWSVM is:

$$\min \quad \frac{1}{2} \| k(X_1, D^t)\omega_1 + b_1 e_1 \|^2 + C_1 e_2^t \xi_1$$

$$s.t. \quad -(k(X_2, D^t)\omega_1 + e_2 b_1) \geq e_2 - \xi_1, \quad \xi_1 \geq 0 e_2. \tag{1.26}$$

And

$$\min \quad \frac{1}{2} \| k(X_2, D^t)\omega_2 + b_2 e_2 \|^2 + C_2 e_1^t \xi_2$$

$$s.t. \quad k(X_1, D^t)\omega_2 + e_1 b_2) \geq e_1 - \xi_2, \quad \xi_2 \geq 0 e_1. \tag{1.27}$$

As for the Wolfe duals, these are:

$$\min \quad \frac{1}{2}\alpha_1^t S(R^t R)^{-1} S^t \alpha_1 - e_2^t \alpha_1$$

s.t. $0e_2 \leq \alpha_1 \leq C_1 e_2$ (1.28)

And

$$\min \quad \frac{1}{2}\alpha_2^t R(S^t S)^{-1} R^t \alpha_2 - e_1^t \alpha_2$$

s.t. $0e_1 \leq \alpha_2 \leq C_2 e_1$ (1.29)

where, R = [k (X_1, D^t) e_1] and S = [k (X_2, D^t) e_2]. A data sample is assigned a class according to its proximity to a certain class hyperplane.

1.3.10 TWIN SUPPORT VECTOR MACHINES THAT MINIMIZE SQUARED ERRORS;

Even though TWSVM has a lower training time complexity than SVM, two smaller QPPs still need to be solved using it. Similar to TWSVM, LSTSVMs use the same protocol as PSVMs, or least squares twin support vector machines. Two important revisions have been made to the TWSVM models: Because the target functions take into account the squared L2-norm of slack vectors, the non-negativity limitations of slack vectors are rendered superfluous. Consequently, equality requirements supersede any lingering inequality restrictions. Finding optimal solutions using primal variables is the hallmark of LSTSVM, as opposed to the dual space used by LS-SVM. Since the solutions are obtained via matrix inversions, the training cost of LSTSVM is lower than that of TWSVM. Here is the basic set of optimization challenges for LSTSVM:

$$\min \quad \frac{1}{2}\| X_1\omega_1 + b_1e_1 \|^2 + \frac{C_1}{2}\xi_1^t\xi_1$$

s.t. $-(X_2\omega_1+e_2b_1)=e_2-\xi_1$ (1.30)

And

$$\min \quad \frac{1}{2}\| X_2\omega_2 + b_2e_2 \|^2 + \frac{C_2}{2}\xi_2^t\xi_2$$

s.t. $X_1\omega_2+e_1b_2)=e_1-\xi_2$ (1.31)

By substituting ξ_1 and ξ_2 solving for (1.30) and (1.31), In the primordial space, the objective functions that are confined inside them yield solutions. LSTSVM, on the other hand, receives its answers from two matrix inversions, whereas TWSVM obtains its solutions by solving two QPPs. Later on, we will talk about how to apply the nonlinear kernel with LSTSVM and other TWSVM-based algorithms in the linear case formulations described in (1.30) and (1.31).

1.3.11 VECTOR MACHINES WITH DOUBLE BOUNDS

Its goal functions are enhanced by including regularization components and an estimate

of the squared bias, One improvement over TWSVM is twin bounded support vector machines, or TBSVM. Here are the improvements:

$$\min \quad \frac{C_3}{2}(\| \omega_1 \|^2 + b_1^2) + \frac{1}{2} \| X_1\omega_1 + b_1 e_1 \|^2 + C_1 e_2^t \xi_1$$

$$s.t. \quad -(X_2\omega_1 + e_2 b_1) \geq e_2 - \xi_1, \quad \xi_1 \geq 0 e_2. \tag{1.32}$$

And

$$\min \quad \frac{C_4}{2}(\| \omega_2 \|^2 + b_2^2) + \frac{1}{2} \| X_2\omega_2 + b_2 e_2 \|^2 + C_2 e_1^t \xi_2$$

$$s.t. \quad X_1\omega_2 + e_1 b_2 \geq e_1 - \xi_2, \quad \xi_2 \geq 0 e_1. \tag{1.33}$$

Structure-based risk reduction is the premise that TBSVM adheres to. Finding solutions with TBSVM follows a similar process as TWSVM.

1.3.12 VECTOR MACHINES FOR STOCHASTIC GRADIENT TWIN SUPPORT

The TWSVM solutions are iteratively obtained using the stochastic gradient descent approach, which is very efficient for large-scale datasets in terms of time. By maximizing the regularization terms, Probabilistic gradient twin support vector machines' (SGTSVM) underlying premise is structural risk reduction. Furthermore, solutions are generated in the primary space by reducing the total loss of the loss components in the SGTSVM goal functions based on the class sizes. P. What follows is an expression of the SGTSVM optimization problems:

$$\min \quad \frac{1}{2}(\| \omega_1 \|^2 + b_1^2) + \frac{C_1}{2l_1} \| X_1\omega_1 + b_1 e_1 \|^2 + \frac{C_2}{l_2} e_2^t (e_2 + X_2\omega_1 + e_2 b_1)_+ \tag{1.34}$$

And

$$\min \quad \frac{1}{2}(\| \omega_2 \|^2 + b_2^2) + \frac{C_3}{2l_2} \| X_2\omega_2 + b_2 e_2 \|^2 + \frac{C_4}{l_1} e_1^t (e_1 - X_1\omega_2 - e_1 b_2)_+ \tag{1.35}$$

The plus function is defined here as plus(.) = max(0..) by means of techniques for convergent gradient descent, the starting values of ω_i and b_i, where i = 1, 2, are optimized repeatedly to zero. In conclusion, SGTSVM is more efficient than TWSVM and TBSVM when handling large-scale problems because it reduces the reduced the training time complexity from cubical limits to linear bounds?

1.3.13 NON-PARALLEL HYPERPLANE UNIVERSUM SUPPORT VECTOR MACHINE

Unlike SVM, while TWSVM solves two smaller-sized QPPs to obtain two non-parallel proximal hyperplanes, together with two parallel class boundary hyperplanes. Maximizing TWSVM's unknowns (ω1, b1) and (ω2, b2) are distinct procedures as we have shown in the preceding sections. Undoubtedly, optimization involves both classes. However, the unknowns (ω_1, b_1) and (ω_2, b_2) are optimized independently, which may result in discrepancy between training and classification. An NPSVM is more dependable than a hyperplane support vector machine that works in parallel, since it builds all of the hyperplanes at once. In addition to improving classification accuracy, NHSVM is logically consistent across its training and forecasting procedures, setting it apart from previous nonparallel SVMs. But NHSVM has to resolve one big QPP. In this work, Zhao et al. expand NHSVM to address Universum data issues, specifically how to use NHSVM to leverage Universum data that already have embedded prior knowledge. We may formulate the optimization issue for NHSVM using Universum data (U-NHSVM) as follows:

$$\min \quad \frac{1}{2}(\| \omega_1 \|^2 + b_1^2 + \| \omega_2 \|^2 + b_2^2) + \frac{C_1}{2}(\| X_1\omega_1 + b_1 e_1 \|^2 + \| X_2\omega_2 + b_2 e_2 \|^2) + C_2(e_1^t\xi_1 + e_2^t\xi_2 + e_u^t\psi_u + e_u^t\psi_u^*) \tag{1.36}$$

$$\begin{aligned} s.t. \quad & X_1\omega_1 + e_1b_1 + X_1\omega_2 + e_1b_2 \geq e_1 - \xi_1, \\ & X_2\omega_2 + e_2b_2 - X_2\omega_1 - e_2b_1 \geq e_2 - \xi_2, \\ & X_u\omega_1 + e_ub_1 \geq -(1-\varepsilon)e_u - \psi_u, \\ & -X_u\omega_2 - e_ub_2 \geq -(1-\varepsilon)e_u - \psi_u^*, \\ & \xi_1 \geq 0e_1, \quad \xi_2 \geq 0e_2, \quad \psi_u \geq 0e_u, \quad \psi_u^* \geq 0e_u. \end{aligned}$$

where, X_u is dimensional matrix (lu × n) comprising information from Universum, ψ_u and ψ^*u dimensionless vectors linked to Universum samples, and a unit vector of length

lu. In the dual space, U-NHSVM has to solve a big QPP with dimensions l + 2lu. Applying a secure sample screening technique might lead to better computing performance for U-NHSVM. In order to assign a class label to a fresh sample, the closest hyperplane is used.

1.4 KERNEL METHODS

To successfully handle non-linear classification tasks, many machine learning algorithms rely on kernel approaches. Kernel approaches enable the development of very versatile and resilient classifiers by transforming data into high-dimensional feature spaces and then using kernel functions to calculate correlations within those spaces. To transform input data into higher-dimensional feature spaces, a subset of machine learning techniques known as kernel methods use kernel functions. Kernel approaches depend on the kernel trick, which directly computes inner products in the feature space, as opposed to standard methods that explicitly compute this mapping. Because of this implicit mapping, kernel approaches are computationally efficient and do not suffer from the curse of dimensionality. [9]

When it comes to classification, regression, and clustering, kernel approaches are practically indispensable. Their versatility in handling intricate data structures makes them a top pick in a range of fields, including bioinformatics and financial modeling. The idea of feature space transformations is fundamental to kernel approaches. Think about a dataset where each point x is in the set R_n. The data is transformed into a higher-dimensional space $\phi: R_n \rightarrow H$, where H is a Hilbert space, in order to handle non-linear separability. The mathematical representation of this transformation is $\phi(x)$, where the new space permits linear separation.

The dot product of the mapped vectors in the feature space, $\phi(x) \cdot \phi(x')$, is calculated using a kernel function $K(x,x')$ without explicitly executing the mapping. Below we can see how the kernel technique uses this to reduce computations:

$K(x,x') = \phi(x) \cdot \phi(x')$.

If the Gram matrix, which is generated from paired kernel evaluations, is positive semi-definite, then the kernel is valid, according to Mercer's theorem.

1.4.1 APPLICATIONS OF KERNEL METHODS

Because they make it possible to efficiently manage complicated, nonlinear correlations between data points, kernel approaches have revolutionized several areas of machine learning. They offer strategies for dealing with high-dimensional and non-linearly separable data, and they find extensive use in many applications, including as clustering, regression, and classification. An in-depth analysis of these uses follows.

Classification

The use of Support Vector Machines (SVMs) in classification is among the most common kernel technique applications. Straight lines (or hyperplanes in higher dimensions) divide data points in classic linear categorization. Nevertheless, data is sometimes not easily separated in the actual world. By using kernel functions, non-linear support vector machines (SVMs) infer a linear decision boundary from the data by implicitly mapping it into a higher-dimensional feature space.

Common tools for tasks such as handwriting recognition include Radial Basis Function (RBF) kernel support vector machines. By applying a high-dimensional transformation to the handwritten input, the RBF kernel makes the classes (such letters) easier to distinguish. Because the kernel function does not explicitly calculate the transformation, it is computationally efficient to determine the similarity between input samples. Because data points (pixels) in handwritten letters or numbers often create complicated patterns that are not linearly separable, this method has shown to be quite effective in this area.

One further use case is in the field of picture classification. Kernel techniques may be used to map images into a higher-dimensional space, as images are high-dimensional data containing pixel values. Using the kernel method, support vector machines (SVMs) equipped with kernels are perfect for jobs like object identification, picture retrieval, and face recognition because they effectively categorize pictures according to their pixel-level similarity. Kernel approaches are very adaptable, which is great for dealing with picture complexity issues like changing illumination and background noise.

The field of bioinformatics has made substantial use of kernel approaches, particularly support vector machines (SVMs), to analyze gene expression, classify proteins, and forecast diseases. Although linear models struggle to make sense of biological data due

to its high dimensionality, complex patterns may be captured by using suitable kernels, such as polynomial or RBF kernels. To better anticipate outcomes and understand the disease, researchers in the field of cancer utilize kernel-based models to categorize different forms of the disease based on protein structures or gene expression data.

Regression

When trying to predict continuous values from input data, as is the case in regression tasks, kernel approaches are also often utilized. To describe complicated, nonlinear interactions between the input variables and the goal outputs, kernels are used in popular techniques like as Gaussian Processes (GPs) and Kernel Ridge Regression (KRR).

For nonlinear predictions, Kernel Ridge Regression (KRR) integrates kernel functions with ridge regression, a regularized version of linear regression. The data is mapped into a higher-dimensional feature space via KRR's kernel, allowing for the use of linear regression. When working with datasets that display intricate connections between input and output variables, this becomes very helpful. Because of the nonlinear nature of the link between financial data and stock prices, the KRR model has found use in financial forecasting. When compared to standard linear regression, KRR's use of a kernel function allows for more precise prediction.

Regression tools that depend significantly on kernels are Gaussian Processes (GPs). To express the connection between input points, a GP utilizes a kernel function and establishes a distribution across functions. If your data is scarce or noisy, this approach will shine. The use of GPs has spread to many fields, including geostatistics, robotics, and time series prediction. By analyzing the data collected at specific monitoring stations, GPs may make predictions about the amounts of pollutants in as-yet-unmeasured places in environmental models. In GPs, the kernel function aids in estimating uncertainty and making predictions by capturing the continuous and smooth character of the underlying data.

When basic linear models fail to adequately describe the input-output connection, KRR and GPs come in handy. These techniques enhance prediction performance in a wide range of applications by fitting complicated, non-linear models using the power of kernel functions.

Clustering

Clustering, which aims to organize data points into clusters according to similarity, is another major use of kernel algorithms. Complex or non-linearly separable data presents a significant challenge for traditional clustering algorithms like the k-means technique. By using a kernel function, kernel k-means expands the k-means algorithm's applicability to data distributions that are not linear.

Assigning data points to the feature space's closest cluster center is the standard procedure for classical k-means. Without directly altering the data, kernel k-means may calculate the similarity between data points in a higher-dimensional space using the kernel technique. As a result, kernel k-means can handle more datasets than the classic technique since it can cluster data with nonlinear decision limits.

Many fields have discovered uses for kernel k-means, such as document clustering, picture segmentation, and voice recognition. Even when the picture attributes aren't linearly separable, the approach may nonetheless group pixels with comparable textures or color patterns, as shown in picture segmentation. Using kernel approaches, we can get around issues like size, rotation, and illumination that make it hard to capture intricate patterns in the photographs.

One use of kernel k-means in voice recognition is the clustering of audio characteristics collected from speech signals. The use of kernel approaches allows for the identification of clusters that represent various phonemes or words, even though these characteristics are frequently not linearly separable. If word frequency distributions aren't linearly separable, then kernel k-means may group documents according to their semantic content, which improves the accuracy of text data clustering.

1.4.2 TYPES OF KERNEL FUNCTIONS

Machine learning methods that rely on kernels, such Support Vector Machines (SVMs), kernel ridge regression, and Gaussian processes, rely on kernel functions as their mathematical foundation. By using these functions, the model is able to capture non-linear patterns in the data without having to explicitly compute the transformation. The connection between data points in the modified feature space is defined. [10]

Linear Kernel

Dot product of two input space vectors produces the linear kernel, the simplest kind of kernel function. It may be stated mathematically as:

$K(x,y)=x\cdot y$

When the decision boundary is a hyperplane and the data is linearly separable, this kernel works well. For high-dimensional datasets, where the amount of features often surpasses the number of data points—as is common in text categorization and natural language processing—its computing efficiency makes it a popular option. The linear kernel may have trouble capturing complicated connections in nonlinear data, despite its apparent simplicity.

Polynomial Kernel

Problems with polynomial relationships between features and goal variables are well-suited to the polynomial kernel, which is an extension of the linear kernel that introduces nonlinearity. Here is how the function is defined:

$K(x,y)=(x\cdot y+c)^d$

The degree of the polynomial is denoted by d, and the constant c regulates the trade-off between the terms of higher and lower order. Choosing the right degree d is crucial to the polynomial kernel's efficiency, but it may simulate feature interactions. The model's adaptability is enhanced by using higher-degree polynomials; however, over fitting is a potential consequence.

Radial Basis Function (RBF) Kernel / Gaussian Kernel

The RBF kernel's proficiency in dealing with nonlinear data makes it a popular choice among users. A distance-based measure of how close two places are to one another, it is defined as:

$$K(x,y)=\exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

In this case, the parameter σ regulates the kernel's spread. For datasets with intricate,

interdependent class structures, the RBF kernel's ability to generate localized decision boundaries makes it an excellent choice. Nevertheless, achieving a balance between model complexity and generalizability requires careful adjustment of the σ parameter.

Sigmoid Kernel

The hyperbolic tangent kernel, or sigmoid kernel, is a metric that draws from neural networks and is defined as:

$K(x,y)=\tanh(\alpha(x\cdot y)+c)$

In this case, α is a parameter that scales the function, while cc is a constant that moves it. Parameter selection has a substantial impact on the performance of this kernel, which may describe nonlinear interactions. Although it is comparable to neural network activation functions, its numerical instability makes it less popular than the RBF kernel, which is why it is seldom utilized.

Laplacian Kernel

An alternative to the RBF kernel, the Laplacian kernel calculates distances between points using the L1 norm rather than the L2 norm. Here is the definition:

$$K(x,y)=\exp\left(-\frac{\|x-y\|}{\sigma}\right)$$

Since the L_1 norm is less affected by outliers than the L_2 norm, this kernel shines in cases when the data includes them. Although it can't match the RBF kernel's benefits, it can provide superior resilience in datasets with noise.

1.5 STATEMENT OF AIM (TITLE OF THESIS)

The goal of this research is to find out how to solve difficult categorization problems using machine learning models that use optimum kernel-generated surfaces. Conventional approaches may fail miserably when faced with the challenge of classifying datasets that are high-dimensional, nonlinear, and diverse. Improved machine learning model generalizability and high accuracy in varied classification tasks are the goals of this study, which aims to accomplish these goals via the use of kernel approaches. In order to enhance the representational strength of machine learning

algorithms, the research delves into the theoretical underpinnings, design, and optimization of kernel-generated surfaces. Image recognition, bioinformatics, and natural language processing are just a few of the many application domains that their adaptability is tested in. These areas need accurate and reliable categorization. In order to raise the bar for data-driven decision-making, this study intends to make a contribution to machine learning by proposing novel ways to circumvent the shortcomings of current categorization methods.

Therefore we chose our title as, "*INVESTIGATE MACHINE LEARNING BASED MODELS UTILIZING OPTIMAL KERNEL-GENERATED SURFACES TO ADDRESS CLASSIFICATION CHALLENGES*"

1.6 NEED AND SCOPE OF THE STUDY

Need of the study

The need for fast and effective categorization techniques is paramount in this age of rapidly expanding data across several areas. When faced with the complexity of real-world datasets, traditional classification methods often fail. These datasets are typically unbalanced, nonlinear, and high dimensional. It is possible to obtain linear separability by transforming data into higher-dimensional spaces, and machine learning-based models, especially those using kernel-generated surfaces, provide a potential answer for this problem. The optimization of these kernel approaches for varied and ever-changing classification tasks, however, is still severely underdeveloped. To fill these deficiencies, this research must investigate how optimum kernel functions might improve machine learning models' accuracy, scalability, and resilience. Healthcare diagnostics, fraud detection, picture identification, and natural language processing are just a few examples of the many real-world applications that may benefit greatly from further research into these techniques. Meeting the increasing need for efficiency and accuracy in decision-making, this study seeks to systematically develop and evaluate kernel-based techniques in order to contribute to the creation of more reliable and adaptable categorization systems.

Scope of the study

This research delves into the creation and implementation of machine learning models

that tackle categorization problems in many areas by making use of optimum kernel-generated surfaces. Beginning with a thorough examination of kernel approaches, their mathematical underpinnings and their capacity to convert complicated data structures into separable forms in higher-dimensional spaces, the scope spans both theoretical and practical elements.

It goes even beyond, addressing issues like non-linearity, large dimensionality, and unbalanced datasets by developing and improving ML algorithms that use these kernels for strong classification. Relevant areas of research include healthcare (where precise illness categorization might prevent deaths), finance (for the purpose of detecting fraud), and technology (for the purpose of performing tasks such as picture recognition and natural language processing). It also intends to solve problems with scalability and real-time adaptation, making sure the models can handle large-scale, ever-changing situations.

This study aims to help intelligent decision-making systems progress by shedding light on the inner workings of kernel-based categorization models and how they operate in practice. By doing so, it hopes to close the gap between theory and practice.

1.7 OBJECTIVES OF THE STUDY

Following are the main objectives of this study: -

1. To address the limitations of existing classification and regression models in supervised machine learning.

2. To investigate enhanced models based on optimal, non-parallel kernel- generated surfaces for improved classification accuracy.

3. To design robust classification and regression methods capable of fitting training data affected by noise, using various resilient loss functions.

4. To explore three distinct formulations of asymmetric Lagrangian ν-twin support vector regression with pinball loss (URALTSVR), applying gradient-based iterative techniques.

5. To analyze the performance of a regularized, implicit Lagrangian twin extreme learning machine in its primal form (RILTELM).

1.8 DEFINITION OF THE KEYWORDS

Machine Learning (ML)

Machine learning is a subfield of AI that allows computers to automatically process data, find patterns, and draw conclusions or make predictions with little to no human input. Machine learning encompasses supervised, unsupervised, and reinforcement learning approaches.

Kernel Methods

Machine learning techniques that do not directly change data but instead work in a high-dimensional feature space using kernel functions. When dealing with non-linear patterns, they find widespread usage in methods such as support vector machines (SVMs).

Supervised Learning

A machine learning paradigm where a model is trained on labeled data, learning to predict the output based on input features. Classification is a common application of supervised learning.

Support Vector Machine (SVM)

One well-known ML technique for finding the best hyperplane to employ for class separation in a dataset is the kernel approach. It is known for its effectiveness in high-dimensional spaces.

1.9 LIMITATIONS OF THE STUDY

The study on machine learning-based models utilizing optimal kernel-generated surfaces to address classification challenges has some limitations. These include the complexity of selecting appropriate kernel functions, significant computational demands for large datasets, and challenges in fine-tuning hyperparameters. The reliance on specific loss functions and assumptions may restrict generalization to diverse scenarios, while the models' adaptability to dynamic datasets remains unexplored. Additionally, interpretability of the advanced techniques and dependency on high-quality data pose challenges, alongside limited validation in real-world applications. These constraints offer scope for future research to enhance the models' robustness and

applicability.

1.10 PLAN OF WORK

Chapter 1: Introduction

This chapter establishes the foundation for the thesis, Addressing the urgent need to tackle regression and classification problems using state-of-the-art machine learning algorithms. It explains what support vector machines (SVMs) are and how they work, with an emphasis on how approaches based on kernels and resilient loss functions may improve model performance. This chapter also lays out the goals and scope of the study, as well as why it's important to create cutting-edge models like extreme learning machines and twin support vector regression for real-world use.

Chapter 2: Literature Review

The literature review provides a comprehensive analysis of existing research in the domain of support vector machines, twin support vector machines (TWSVM), and other advanced models.

It delves into the evolution of regression techniques, regularization methods, and robust loss functions like pinball and Huber loss. The chapter identifies research gaps and highlights the limitations of current models, paving the way for the proposed methodologies and their application to complex classification and regression problems.

Chapter 3: Regularization-Based and Robust Asymmetric V-Twin Support Vector Regression Using Pinball Loss Function

This chapter introduces a novel regularization-based V-twin support vector regression framework incorporating the pinball loss function. It discusses how the asymmetric nature of the loss function enhances robustness against outliers and addresses imbalanced data. Mathematical formulations, optimization techniques, and experimental evaluations are presented to validate the model's effectiveness in regression tasks.

Chapter 4: Huber Loss Regularized Twin Support Vector Regression with Least Squares Large Margin Distribution Machine

This section delves into a mixed method that combines Huber loss with twin support vector regression and the least squares large margin distribution machine. It examines the benefits of Huber loss in handling noise and outliers and demonstrates how this integration leads to improved generalization. The chapter includes detailed algorithmic development and performance comparisons with existing models.

Chapter 5: Iterative Methods for Twin Bounded Support Vector Machines with Squared Pinball Loss and Intuitionistic Fuzzy Least Squares Twin Bounded SVMs

Here we'll examine iterative functional approaches for enhancing twin bounded support vector machines. It brings round pinball loss and intuitionistic fuzzy-based mechanisms, providing a detailed mathematical framework and optimization strategies. The chapter emphasizes the models' adaptability to complex datasets and discusses their experimental outcomes.

Chapter 6: Regularized Implicit Lagrangian Twin Extreme Learning Machine in Primal for Pattern Classification

This chapter introduces a novel pattern classification machine that uses a regularized implicit Lagrangian twin extreme learning algorithm. Primordial space is where the suggested model functions, offering computational efficiency and superior classification accuracy. Theoretical analysis and extensive experimental evaluations are provided to illustrate the model's advantages over traditional approaches.

Chapter 7: Conclusion, Recommendations, and Future Scope

The concluding chapter summarizes the research findings and highlights the contributions of the proposed models to the field of machine learning. It discusses the practical implications of the study and provides recommendations for deploying the models in real-world scenarios. Additionally, the chapter outlines potential directions for future research, including the exploration of alternative loss functions, scalability improvements and applications in emerging domains.

CHAPTER -2
REVIEW OF LITERATURE

CHAPTER -2
REVIEW OF LITERATURE

Alnuaimi, Amer et al., (2024) [1] (ML) is an important part of AI, which is a larger area that uses statistical approaches to teach computers to learn and make choices on their own, without human intervention or programming. Computers can learn from data, spot patterns, and form conclusions with little to no human input; this is the basic idea. Supervised, unsupervised, semisupervised, and reinforcement learning are the four primary subfields of machine learning. The two main types of supervised learning, classification and regression, both include training models using labeled datasets. If you want your output to be continuous, you should use regression; if it's categorical, you should use classification. Improving models' ability to forecast class labels from given input attributes is the main goal of supervised learning. The purpose of classification is to provide predictions about related data using the values of a class variable or category goal. When used to different kinds of statistical data, it yields useful results. Data mining, predictive modeling and picture categorization are just a few of the many uses for these algorithms. This study's overarching goal is to serve as a convenient reference for the most popular machine learning fundamental categorization algorithms, including their benefits and drawbacks. It goes without saying that no one article could hope to cover every supervised machine learning classification method. Academics and researchers alike will find it useful; it introduces the subject to beginners and helps them better understand categorization procedures.

Almuqati, Mohammed et al., (2024) [2] Automated insights, forecasts, and decision-making are the hallmarks of data science and machine learning, two cutting-edge fields in contemporary technology. Important paradigms in this ever-changing field include supervised and unsupervised learning, which each have their own set of problems. Both supervised and unsupervised learning present complex problems, and this article covers them all. Studies published in the years 2019–2023, inclusive, are reviewed in this article. In this piece, we'll look at the difficulties of both supervised and unsupervised study. Data labeling, overfitting, low generalizability, and balancing error equivalence and decision-making objectives are some of the difficulties in supervised learning. Overfitting, selecting the right method, and understanding outcomes are all examples of challenges in unsupervised learning. Among these tasks is the management of

outliers and noise, as well as the evaluation of clustering quality and the determination of the appropriate cluster size. Whether you're new to machine learning or have years of experience under your belt, this article should help shed light on these obstacles. To get around these complications, researchers and practitioners are always inventing new ways of doing things. For scholars and specialists in the subject, this article is a vital resource that will equip them to successfully manage these issues. To fully harness the potential of these effective technologies, it is crucial to have a complete grasp of these obstacles as technology progresses. Lastly, a number of suggestions were made to help academics in the future use machine learning in data-driven discovery and automation, a path that will be fraught with both possibilities and obstacles.

Mohalder, Rathindra Nath et al., (2024) [3] Supervised Machine Learning is more often known as Supervised Learning (SL) or SML. Being a subset of both AI and ML, it falls under the umbrella of artificial intelligence. In order to train algorithms that accurately anticipate outcomes or categorize data, it is characterized by the use of entitled datasets. Part of the cross-validation procedure involves gradually feeding the input information into a supervised machine learning model so that it can synthesis its weights and get a good match. A supervised learning machine may help a business with a wide range of practical issues. SML is on the lookout for algorithms that used externally provided occurrences to generate common hypotheses, in order to prepare predictions for when similar situations occur again. Effective intelligent systems often finish the supervised Machine Learning (SML) classifications. This article presents an overview of supervised learning algorithms, compares several types of supervised learning, and ultimately determines which algorithm is the most successful for a given collection of examples, variables, and features in machine learning. In this article, we'll look at eight distinct SML algorithms. Those were the ones that were being imagined: ANNs, Bayesian Networks, KNNs, Random Forests, DTs, Linear Regressions, SVMs, and Logistic Regressions. The programming language Python is the basis for these eight algorithms. Justify the performance of each method by using a sample dataset. Using throughput, reaction time, and accuracy as metrics, please defend the algorithms above. Predetermined parameters are the basis of the supervised learning approach. When evaluating the efficacy and capability of a machine learning system, the performance indicator is crucial. Based on the results, Decision Tree provides the most accuracy, reaction time, and throughput among the prediction algorithms discussed in this

research. After the DT method, the next two accurate SML techniques are SVM and Logistic Regression.

Oluchukwu C, Asogwa et al., (2024) [4] Using a tested data set for the opinions of Nigerian citizens during the naira redesign policy period, this research empirically compared the performance of three supervised machine learning models: Multinomial Logistic Regression (MLR), Multilayer back propagated Neural Networks (MNN), and Multinomial Decision Trees (MDT). The models were trained using a classification matrix criterion. About 600 copies of surveys about the views of Nigerian people on their wellbeing during the era of naira redesign. A total of three models were evaluated, and the results showed that ANN achieved the highest accuracy rate (94.4%), followed by MLR (93.5%), and MDT (90.0%).

Rahaman, Md. Jamaner. (2024) [5] What we term "machine learning" (ML) really refers to the process by which computers learn new tasks and tasks alone with the aid of algorithms. These days, it seems like everyone wants everything done quickly and automatically. The efficiency of machine learning has brought about a dramatic shift in that regard. A smart machine can do tasks at a higher rate than a person. By using ML, the occurrence of mistakes is significantly reduced. This paper aimed to provide a description of several ML algorithms, including supervised, unsupervised, semi-supervised, and reinforcement learning, along with their definitions, pros and cons, and areas of work, in order to help people understand which algorithm to use based on improving the necessity of ML algorithms in the present situation. In particular, supervised learning methods such as Support Vector Machines (SVMs), Decision Trees, K-Nearest Neighbors (K-NNs), Linear Regression, and Logistic Regression. Principal component analysis (PCA) and K-Means clustering are tools for unsupervised learning. A crash course on reinforcement learning and semi-supervised learning. By the end of the article, readers will have a good grasp of the most popular machine learning methods.

Zhang, Zheng et al., (2024) [6] One helpful way to spot unusual fish is to look for certain surface characteristics that are out of the ordinary. Problems with present approaches include high levels of subjectivity, low levels of accuracy, and subpar performance in real time. In response to these difficulties, we provide YOLOv5s-based real-time precise surface feature detection for in-water fish. Among the particular

improvements are: 1) In order to enhance the model's capability to identify small targets, we optimize the full intersection over union and non-maximum suppression using the normalized Gaussian Wasserstein distance metric. 2) We use MobileViTv2 to increase detection speed and the DenseOne module to improve the reusability of aberrant surface features into the feature extraction network. 3) To address the difficulty of extracting deep features from complicated backdrops, we combine the omni-dimensional dynamic convolution and convolutional block attention modules in accordance with the ACmix concept. With 160 validation sets of aberrant fish in water, we conducted comparison studies and achieved a recall of 99.5%, a precision of 99.1%, a mAP50 of 73.9%, and a frames per second (FPS) of 88. By1.4,1.2,3.2,8.2%, and 1 FPS, respectively, our model outperforms the baseline. In terms of comprehensive assessment indices, the upgraded model also beats other top-tier models.

Heydari, Zahra et al., (2024) [7] A precise assessment of domestic water end uses (such as showers, toilets, faucets, etc.) is necessary for water sustainability in the built environment. We utilize real (measured) and synthetic (labeled) data sets to assess how well four models—Random Forest, RF; Support Vector Machines, SVM; Logistic Regression, Log-reg; and Neural Networks, NN—classify the end-use of water in residential areas. Conditional Tabular Generative Adversarial Networks were used to create synthetic labeled data. Training each model with its optimal hyperparameters was then accomplished using grid search. In terms of overall model performance, the RF model was the best, but in terms of computational efficiency for specific end uses, the Log-reg model had the shortest execution times under various balanced and imbalanced (based on number of events per class) synthetic data scenarios. Although it took more time to run than the other classification models, the NN model performed quite well. All models in the balanced data set scenario obtained F1-scores that were quite near to each other, with values ranging from 0.83 to 0.90. Nevertheless, the RF and NN models demonstrated superior performance when confronted with unbalanced data that mirrored real-life situations, whereas the SVM and Log-reg models performed worse. In general, we found that when it comes to water end-use data, decision tree-based models are the best option for categorization tasks. Our research contributes to the advancement of home smart water metering systems by generating synthetic labeled end-use data and shedding light on the relative merits of several supervised machine learning classifiers for this purpose.

Laurer, Moritz et al., (2023) [8] The use of supervised machine learning to sift through massive political text corpora is on the rise. The need for thousands of training data points that are manually annotated is the primary drawback of supervised machine learning. Because most novel research issues in the social sciences need fresh training data for a task designed to address the subject at hand, this is an especially pressing concern in that field. Deep transfer learning's ability to build "prior knowledge" in language models is examined in this research as a potential solution to this problem. By training on general tasks such as natural language inference (NLI; "task knowledge"), models such as BERT may acquire statistical language patterns during pre-training ("language knowledge"). This allows them to rely less on task-specific data. Using eight different activities, we show that transfer learning is beneficial. Our BERT-NLI model, which was fine-tuned using 100 to 2,500 texts, outperformed classical models that did not include transfer learning by an average of 10.7 to 18.3 percentage points across all eight tasks. In comparison to traditional models trained on around 5,000 texts, our research shows that BERT-NLI fine-tuned on 500 texts delivers comparable performance. On top of that, we prove that transfer learning excels when faced with unbalanced data. Finally, we outline new avenues for political science research and talk about the constraints of transfer learning.

Wei, Yuzhen et al., (2023) [9] To better understand the role of genes in maize, it is essential to first distinguish between genetically modified (GM) and non-GM kernels. To differentiate between genetically modified (GM) and non-GM maize kernels, a comprehensive and innovative detection system was developed using near-infrared spectra. A total of seven hundred and seventy-one maize kernels of three different types were photographed using hyperspectral imaging equipment, and their average spectra were then retrieved for use in the modeling process. The backpropagation neural network-genetic algorithm model outperformed the other standard feature engineering-based modeling approaches with a prediction accuracy of 0.861. Next, innovative deep learning-based modeling approaches were created. Before building the deep learning models, the original spectra were converted into two-dimensional matrices to extract the interaction information between bands and make them suitable for the application situations. At last, we built a VGG net—a modified convolution neural network—with dilated convolution to categorize the maize kernels, and we achieved a prediction accuracy of 0.961. This study introduces a new and innovative method for identifying

genetically modified (GM) maize kernels. By using deep learning visualization technologies, future study will enhance the detection system for monitoring illicit GM organisms.

Matura, Rishi et al.,(2023) [10] Machine learning has become more popular in recent years due to its extensive industry-specific applications. Various methods of machine learning, including supervised and unsupervised classifiers as well as reinforcement learning, are covered in this paper. We also look at machine learning's downsides, such as how much labeled data is needed and the risk of bias during training, among other things. We have now covered the basics of the discipline and covered some of the possible future advances, such how machine learning may be used in healthcare and finance. Also included are comparisons of two machine learning methods, one of which is the Decision Tree algorithm and the other is the Naive Bayes algorithm. Taken as a whole, this study is a great resource for anybody interested in the present and future of machine learning.

Talaei Khoei, Tala et al., (2023) [11] A number of application sectors, including cybersecurity, have been profoundly affected by the advent of machine learning methods. Data pre-processing, model selection, and parameter optimization are a few of the many steps that must be integrated into the creation of top-notch machine learning applications. While prior studies have provided some insight into these methods, they have mostly targeted narrow fields of application. The absence of an all-encompassing review of the fundamental stages of machine learning architecture in the domain of cybersecurity is a significant void in the existing literature. This study fills that need by offering a comprehensive overview of recent research in machine learning, including methods that may be applied to any field. Reinforcement learning, supervised, semi-supervised, and unsupervised models are the four main types of models. The models for each of these classes are detailed here. The study also covers the latest developments in data pre-processing and hyperparameter tuning methods. Also reviewed are the research gaps and major obstacles that the cybersecurity area is now facing, according to this poll. Our analysis of these gaps leads us to suggest several interesting avenues for further investigation. Ultimately, we hope that this survey will be a helpful resource for scholars looking to learn more about machine learning, and that the insights it provides will help to promote innovation and advancement in a wide

range of application fields.

Ali, Zeravan et al., (2023) [12] Extracting usable information from the massive amounts of data created daily, processing it to learn, and then acting on that knowledge is the main goal of machine learning. Some examples of machine learning's application fields include chemical informatics, medical diagnostics, bioinformatics, search engines, pattern identification, and original language processing. XGBoost is the best machine learning algorithm in terms of categorization variety, interpretability, and prediction accuracy. It was launched not long ago and has shown to be quite good at modeling complicated systems. With its robust architecture, high degree of customization, and portability, XGBoost stands out as an exceptional distributed scaling improvement library. Artificial intelligence algorithms are integrated via augmented scaling. Several data science tasks may be efficiently and effectively handled by this parallel tree enhancement. Because it allows the use of clean low-level libraries and high-level APIs, Python is still the language of choice for scientific computing, data science, and machine learning. This enhances performance and productivity. One of the most well-known Python-based supervised and semi-supervised learning (SSL) methods is presented in this article.

Miric, Milan et al., (2023) [13] Summary of the Research More and more, researchers are building quantitative variables for their analyses from unstructured text data. Traditionally, researchers have used keyword-based ways to accomplish this purpose. These approaches include researchers providing a dictionary of keywords that are mapped to the relevant theoretical ideas. To identify unstructured text documents and generate quantitative variables, one may utilize contemporary machine learning (ML) methods for text classification and natural language processing. In this article, we provide an example of how to use ML techniques for this and talk about one use case for finding AI patents. We show the benefits of the ML approach by comparing and contrasting several ML approaches with the keyword-based approach. To further show how AI technology has evolved in general, we use the categorization results produced by ML models. Executive Synopsis Researchers and business analysts may find a plethora of information in text-based materials. To make these papers usable in future studies, researchers must frequently figure out how to categorize them. In this research, we show how supervised machine learning techniques may be used to automate the task

of grouping textual materials into pre-established categories. We outline the potential applications of such procedures, how they compare to other methods, and the benefits and drawbacks of each. Using the abstract language of all U.S. patents, we use these techniques to detect AI-based innovations. In doing so, we are able to reveal intriguing trends in the evolution of AI innovation nationwide. The data and code used in this article are also made available for future researchers to use.

Taye, Mohammad. (2023). [14] Since its inception, deep learning (DL) has dominated the ML computational landscape, outperforming humans on a number of challenging cognitive tasks while maintaining or improving upon their performance. Thanks to its ability to learn from data, deep learning technology—which evolved from ANN—has become a major player in the computer industry. One advantage of deep learning is its capacity to learn from massive amounts of data. Rapid development and effective application of deep learning have occurred in many more conventional domains in recent years. Popular machine learning methods have been surpassed by deep learning in several fields, such as cybersecurity, bioinformatics, medical information processing, robotics and control, and natural language processing. Also, this essay wants to provide a better overview of the most important parts of deep learning, including the most recent advances in the area, so that people have a better place to start when trying to grasp the topic on a deeper level. The importance of deep learning, as well as several deep learning methods and networks, are also covered in this study. Furthermore, it outlines potential practical domains for using deep learning methods. Finally, we provide some recommendations for further study and indicate certain traits that may be present in further iterations of deep learning models. Academics and professionals in the business world may both benefit from the thorough introduction to deep learning modeling that this essay aims to provide. Finally, we provide more problems and answers to help researchers understand the current gaps in the research. Several methods, deep learning frameworks, tactics, and uses are covered in this paper.

Jain, Sambhav et al., (2022) [15] This research proposes using parametric non-parallel support vector machines to classify binary patterns. The model's sparsity is preserved and its resilience to noise is enhanced by a reevaluation of the support vector machine optimization. Since our model shows characteristics with support vector machines, we may expand other support vector machine-related learning approaches to make it

scalable for large-scale problems. We confirm our assertions with experimental findings on many benchmark UCI datasets.

Liu, Gaoyuan et al., (2022)[16] Our goal is to solve the intrusion detection issue in WSNs by establishing an edge-based intrusion detection system using edge computing, taking into account all of the WSN's combined properties. The WSN is well-defended by an intrusion detection system (IDS), a technology that proactively protects networks from security breaches. We present a WSN intelligent intrusion detection model in this paper. It forms an edge intelligence framework that performs intrusion detection when the WSN encounters a DoS attack by combining the k-Nearest Neighbor algorithm (kNN) from machine learning with the arithmetic optimization algorithm (AOA) from evolutionary calculation. By adjusting the optimization using the Lévy flight strategy and using a parallel method to improve communication between the populations, we may increase the model's accuracy. The benchmark function test shows that the suggested PL-AOA method improves the kNN classifier, and it works. By simulating the WSN-DS dataset in Matlab2018b, we find that our model outperforms the original kNN by around 10% in DoS intrusion detection, and it reaches 99% ACC. The suggested intrusion detection model provides beneficial benefits and is practically significant, according to the testing findings.

Muraina, Ismail et al., (2022)[17] Every day, we all encounter a plethora of decision-making tasks that need careful consideration and, all too frequently, we let our guard down and succumb to a variety of prevalent biases and logical fallacies. Decisions on the machine learning algorithm or model to use for analysis are fraught with peril since they are dependent on a myriad of variables, including the nature of the issue, the criteria for selecting a model, and the anticipated results. The research investigates the potential of an AI-powered expert system to facilitate the prompt selection of appropriate algorithm(s) for achieving set goals. To reach a suitable decision-making method, the research also models a sequence of effective channels using VisiRule software. A variety of algorithms were utilized to guide the selection process, including supervised and unsupervised machine learning, clustering, association rules, dimensionality reduction, and various methods of classification and regression. VisiRule, an AI-based expert system, was utilized for this purpose. With thorough descriptions of each choice, this study's results show the straightforward ways to choose

the most relevant and suitable model or algorithm for the current analysis. With VisiRule, solving decision-making difficulties has never been easier, and you won't even need any code. Artificial intelligence rule-based expert systems might address decision-making problems with no effort, no coding required, and very attainable, accurate results..

Sharma, Shallu & Mandal, Pravat. (2022) [18] Devastating and incurable, Alzheimer's Disease (AD) is a kind of neurodegeneration that affects the brain. Patients with AD are able to maintain a normal lifestyle with the aid of early detection. We have described ML approaches that use several feature extraction strategies to combine complementary and correlated properties of data obtained from various neuroimaging modalities. In order to create an ML-based AD diagnostic system, we detail a number of feature selection, scaling, and fusion approaches, as well as the problems that have been encountered. On top of that, we have included theme analysis to compare the ML process for potential diagnostic solutions. An improved computer-aided early diagnostic method using multi-modal neuroimaging data from AD patients is one area that might benefit from this extensive study.

Fernández Pascual, Ángela et al., (2022) [19] One of the most difficult problems in machine learning is outlier identification, which involves finding data points that are very out of the ordinary. Particular points like these might throw off a model's training and lead to less precise predictions while the model is being constructed. Because of this, the first step in solving a machine learning issue is usually to find and eliminate them before developing a supervised model. There are a plethora of effective outlier detection algorithms available today; however, the key issues with these algorithms are their reliance on unsupervised learning and the hyperparameters that need to be fine-tuned for optimal performance. A novel supervised outlier estimator is presented in this study. To do this, a supervised model is pipelined with an outlier detector in such a manner that the outlier detector's hyperparameters are optimally set by the targets of the supervised model. Using this pipeline-based method, integrating several outlier detectors, classifiers, and regressors is a breeze. Eight regression problems and nine relevant outlier detectors were integrated with three regressors and two classifiers in the trials. Another eight issues were divided between binary and multi-class classification. After analyzing and comparing the nine outlier detectors' efficacy, we

can say that the idea is valuable as an objective and automated technique to properly identify detector hyperparameters..

Jain, Nipun et al., (2022)[20] One key benefit of machine learning is the reliability of the predictions it produces from datasets. As a result, computers may be trained to carry out complicated tasks autonomously. When it comes to analyzing large datasets, machine learning is king. Businesses and entrepreneurs may benefit from machine learning since it speeds up the process of identifying possibilities and hazards. Companies that collect and process massive volumes of data are finding that machine learning is the most effective tool for their data analysis and model building needs. Not only is machine learning fundamental to AI, but it also has a major impact on AI's history and future. The accuracy of classifications achieved by applying algorithms to issues with varying parameter settings varies greatly. Finding the optimal settings for algorithm parameters to address technical issues with performance measures is a difficult task in machine learning. Supervised, unsupervised, and reinforcement learning are only a few of the machine learning techniques covered in this article. A variety of machine learning algorithms including Decision Tree, Naïve Bayes, K-Nearest Neighbor, Random Forest, and SVM Classifier are used mostly in supervised machine learning tasks like classification and regression. Using examples and illustrations, the author provides a clear explanation of all methods that rely on categorization. In addition, the authors provide examples of domains or applications that make use of these categorization techniques.

Sekeroglu, Boran et al., (2022) [21] The use of AI and ML to solve issues or augment human specialists is crucial in almost every aspect of human existence. It remains a difficult issue for academics to determine which machine learning model would generate a better outcome for a specific problem within the broad real-life application domains. Several aspects, including the features of the dataset, the training approach, and the model's responses, might influence the model's performance. Hence, in order to ascertain the efficacy of the proposed tactics and the capability of the model, a thorough evaluation is necessary. Ten standard machine learning models were applied to seventeen different datasets in this research. Training procedures of60:40,70:30, and 80:20 hold-out, in addition to five-fold cross-validation, are used in the experiments. Mean absolute error, mean squared error, and coefficient of determination (R2 score)

were the three metrics utilized to assess the experimental outcomes. The models that were taken into consideration are examined, and the benefits, drawbacks, and data dependencies of each model are highlighted. Decision trees, linear regression, support vector regression with radial and linear basis function kernels, random forests, extreme gradient boosting, deep neural networks, and deep Long-Short Term Memory (LSTM) neural networks all performed poorly in comparison to the deep Long-Short Term Memory (LSTM) model, which emerged as the top performer after an excessive number of experiments. When evaluating models in regression research without data mining or selection, cross-validation should be examined due to the substantial influence it has on experimental outcomes.

Gupta, Monica. (2022) [22] Rather of being expressly programmed to do a given activity, computers may now act and make judgments based on data thanks to machine learning. You can get the answer to your query from the data you have using this tool and technology. When fed fresh data, these systems are meant to become smarter with time. Machine learning is a branch of artificial intelligence that is rapidly expanding its scope. It all starts with the premise that computers should have access to data so they can figure things out for themselves. The goal of machine learning (ML) is to discover rules for optimum behavior and to train computers to adapt to new situations by analyzing datasets for patterns. For decades, many of the underlying algorithms have been known. The article has covered a range of machine learning algorithms. There are many applications for machine learning algorithms, but one might argue that they can learn to handle data management on their own after some initial training..

Pruneski, James et al., (2022)[23] The majority of machine learning approaches used in healthcare research are based on supervised learning. Using a given ground truth, it may categorize situations as positive or negative or make predictions about interesting outcomes. A variety of methods, including supervised learning, are gaining traction in the "big data" movement, from simpler tree boosting to more involved regression modeling. There is a dearth of literature that details the benefits and drawbacks of the various modeling approaches, despite the fact that these tools are booming in use and power. Medical personnel seldom get instruction on how to properly employ machine learning models in the course of their work. It is critical that doctors and other medical professionals have a firm grasp of the mechanisms behind machine learning's growing

influence in the medical field. The goal of this research was to compile a list of popular supervised learning methods with examples from the orthopedic literature that illustrate their application recently. Improving communication inside and across research teams is another objective, as is addressing differences in understanding of these methodologies.

Ono, Sachiko et al., (2022) [24] Machine learning is a set of procedures that computers go through to discover patterns in large datasets. Machine learning has grown and found use in medical research because to the abundance of diverse health data and the recent advancements in computing power. At present, supervised, unsupervised, and reinforcement learning are the three main categories of machine learning. In the field of medicine, supervised learning is often used for prognoses and diagnostics, unsupervised learning for illness phenotyping, and reinforcement learning for optimizing positive outcomes, including overall emergency department patient waiting time optimization. This article gives a quick rundown of four popular prediction algorithms—random forests, gradient-boosted decision trees, support vector machines, and neural networks—and explains the idea and use of supervised learning in medicine, the most popular machine learning approach in the medical field. Deep learning algorithms, which evolved from neural networks, are one kind of algorithm that can handle more complicated problems. Medical imaging, including retinal fundus photos for diabetic retinopathy diagnosis, and basic categorization tasks are two popular applications of deep learning in the medical field. Algorithms may fail in the absence of domain expertise, despite machine learning's potential to improve healthcare by analyzing massive amounts of data that humans just cannot handle. For machine learning to be useful in healthcare, algorithms and human intelligence must work together.

Dahiya, Neelam et al., (2022) [25] With the expansion of human understanding and the proliferation of databases, one of the most pressing issues is figuring out how to extract useful information from massive amounts of raw data. One method that may assist solve this problem more quickly and accurately is machine learning. A key component of machine learning is training the algorithm using training data; from there, the algorithm builds rules; and finally, using test data, assessment is carried out autonomously produce results. We shall examine the many uses and benefits of machine learning in

this post. Following this introduction, the paper delves into a comprehensive catalog of supervised and unsupervised algorithms, detailing their many applications and kinds. With the knowledge gained from this article, the researcher may pinpoint possible uses for machine learning and choose suitable approaches for every situation. Furthermore, the researcher may have a comprehensive grasp of machine learning. This study has the potential to be advanced by comparing and contrasting deep learning with machine learning techniques. There is a lot of hope that this area may lead researchers to solutions for many agricultural problems and medical conditions (including cancer, skin disorders, etc.).

Arista, Artika. (2022). [26] Whether or whether they have COVID-19 is a mystery to many individuals today. A case of COVID-19 manifests itself with a persistent fever, dry cough, and sore throat. See a doctor or visit a clinic without delay if you have any symptoms of coronavirus illness 2019 (COVID-19). Therefore, it is critical to study up on and fully grasp the key distinctions. COVID-19 symptoms may be rather diverse. Specifically, the studies were conducted utilizing the (DT) and (LR) Machine Improving Algorithms for Classes. Python code was written and tested in Jupyter Notebook 6.4.5. The results of the tests performed on the COVID-19 symptoms dataset showed that the DT model had better testing performance and cross-validation than the LR machine learning models. Since the DT model had a cross-validation success rate of 98.0%, it is evident that it is the victor. The DT model has completed performance testing with a 98.0% success rate. Taking into account both the cross-validation performance and the testing results, the LR has achieved the second-best outcome. The LR model obtained a 96.0% accuracy rate in the cross-validation results. With a precision of 97.0%, the LR model has shown itself in performance tests. Therefore, in terms of testing and cross-validation, the DT performs better than the LR on the COVID-19 symptoms dataset.

Bhatt, Prahar et al., (2021) [27] With the capacity to automatically identify surface flaws from photos, industrial applications can't function. There was a subset of issues that could be effectively addressed using traditional image processing methods. Noise, changing illumination, and backgrounds with intricate textures were all challenges that these methods failed to overcome. To automate the process of finding defects, deep learning is being investigated more and more. Three distinct approaches to effort

categorization are offered in this survey report. The context of defect detection, learning approaches, and methods for localizing and classifying defects form the basis of these. This approach categorizes the current literature. Following current tendencies in the deep learning field, the article suggests avenues for further study..

Nair, Nikhitha et al., (2021)[28] Deep learning frameworks have recently emerged and show promise as a unifying paradigm for supervised and unsupervised learning, opening the door to more abstract data representations. Face recognition, text mining, language translation, picture prediction, several fields have profited from deep learning's numerous successful explorations, including action detection and many more. Core vector machines, kernel machines, support vector machines, and extreme learning machines are just a few of the machine learning approaches that can handle both linear and nonlinear data. If we want better data dispersion, these Kernel machines are crucial for mapping the input space data to a Kernel-induced high-dimensional feature space. The data distribution will be better suited to the classification challenge at hand in this Kernel-induced high-dimensional feature space. By picking the right Kernel function, the Kernel technique makes it easy to convert machine learning methods that rely only on inner product calculations between data vectors into a Kernel-based strategy. To compute the inner product of the modified data vectors in an implicitly specified Kernel-induced feature space, Kernel-based methods make use of the Kernel functions. Kernel machines, in contrast to neural networks, ensure that structural risks are minimized and that global optimum solutions are reached. Functionality like as theoretical tractability and outstanding performance in real applications are also shown by the Kernel machines. The researchers were inspired to develop deep Kernel machines by using the rising trends of deep learning with Kernel approaches, thanks to their efforts. To overcome their shortcomings and make the most of their strengths, researchers combine Kernel methods with deep learning networks. Then, they use deep Kernel learning techniques to boost the algorithm's performance in various tasks. Deep Kernel machines can be constructed in various ways by combining Kernel methods with deep learning architectures. These methods include using Kernels as the deep learning network's final classifier, incorporating kernelization into deep neural networks to improve feature enrichment, and constructing deep Kernel machines that use deep or multiple Kernels for different tasks. The purpose of this review is to provide a broad overview of the many methods used to construct deep

Kernel learning architectures, with the goal of improving the characteristics and performance of learning algorithms for use in real-world scenarios.

Wen, Hui et al., (2021) [29] We provide a kernel holistic learning and division (KHLD) based neural network classification improvement algorithm. The suggested approach uses the RBF kernel, or learnt radial basis function, as its research goal. Here, we suggest a kernel that, in the training sample space, may be thought of as a subspace area made up of the same pattern category. By expanding the area of the original examples' sample space, we may access subspace information that is significant across instances, and the classifier's border doesn't have to be close to the original instances; this improves the classifier's generalization performance and resilience. The instance optimization and screening strategy used to describe KHLD is applied in concrete by generating a new pattern vector inside each RBF kernel. Experiments on synthetic datasets as well as many UCI benchmark datasets demonstrate the efficacy of our approach.

Pal, Sujan & Sharma, Prateek. (2021)[30] When it comes to data-driven research in the Earth sciences, machine learning (ML) has made great strides as an AI tool. To provide lower boundary conditions to atmospheric models, Land Surface Models (LSMs) record the water, energy, and momentum exchange between the land surface and the atmosphere. These models are crucial parts of climate models. Focusing on how ML might enhance land modeling and providing a detailed discussion of the most important ML approaches are the goals of this review study. In order to compile a comprehensive list of articles, literature searches were carried out using the appropriate keywords. Additionally, the articles' bibliographies were taken into account. So far, ML-based strategies have improved evapotranspiration and heat flux estimates, optimized parameters, predicted crop yields more accurately, and benchmarked models, all while enhancing the performance of LSMs and reducing uncertainties. Random Forests and Artificial Neural Networks are two popular ML methods that are used for these tasks. We draw the conclusion that land modeling has room for development in areas such as efficient model performance, data assimilation, parameter calibration, reduction of uncertainty, and high-resolution data preparation via the use of machine learning. Long short-term memory, convolutional neural networks, and other deep learning approaches may be used with the standard techniques.

Mazlan, Aina et al., (2021) [31] When it comes to healthcare and medicine, data-driven models that can anticipate outcomes are crucial. Nevertheless, using machine learning (ML) techniques may tackle the most difficult aspect of predictive modeling: building a prediction model. A gene expression dataset is used to train the model utilizing the approaches, which do not need explicit programming. This becomes a tedious and complicated operation when dealing with the massive amounts of gene expression data. In light of the growing interest in cancer classification within bioinformatics and computational biology, this work offers a concise overview of current developments in machine learning (ML) and deep learning (DL). The primary emphasis of this study is on the advancement of ML and DL-based cancer classification algorithms. There have been several approaches to the cancer categorization issue, but newer research indicates that supervised and DL-based algorithms are the most effective. Furthermore, the healthcare dataset's sources are also detailed. The development of many machine learning methods for insight analysis in cancer classification has brought a lot of improvement in healthcare. It would seem that there is an urgent need to handle the growing number of healthcare applications by developing more effective categorization algorithms.

Paturi, Uma Maheshwera Reddy et al., (2021) [32] This study models and optimizes employing machine learning techniques like support vector machines (SVMs), artificial neural networks (ANNs), and wire electrical discharge machining (WEDM) to determine the surface roughness of Inconel 718 and genetic algorithms (GA). As a result, we used surface roughness measurements derived from real-time WEDM trials run with varying degrees of control variables such pulse on/off duration, peak current, servo voltage, and wire feed rate. Using the grid search approach, we were able to modify the SVM parameters and find that the optimal ANN model architecture is 5-10-10-1. The R-value, which measures the degree of agreement between experimental and model predictions, was used to assess the efficacy of the ANN and SVM models in comparison to those of the response surface methodology (RSM). With an R-value of 0.99998 compared to experimental findings and a minimum MAPE of 0.0347%, the SVM predictions were the most accurate of all the models examined. Further, the surface roughness was improved by 61.31% after using the GA technique with the proposed RSM equation as the fitness function. Using the suggested SVM and GA method, we can optimize the WEDM process for Inconel 718 by rapidly predicting and

optimizing the surface roughness.

Saravagi, Deepika et al., (2021)[33] In the last ten years, the healthcare sector has seen a meteoric rise in the popularity and interest in machine learning algorithms across academic groups. New models to study spondylolisthesis (slippage of one vertebra over another) concerns have been developed via interdisciplinary cooperation, and they show great promise and have a lot of potential. Spondylolisthesis detection and prediction machine learning methods are reviewed in this article. From the standpoint of both modeling and applications, it would be an invaluable resource. Searching Scopus, PubMed, IEEE, Google Scholar, ResearchGate, Springer, and Elsevier databases systematically using predefined inclusion-exclusion criteria allowed us to retrieve publications. Title, abstract, and The articles were analyzed using full-text reviews. Finally, we will discuss some of the challenges and opportunities in this area. For every task that was examined, we checked the models and frameworks that were used and the overall performance according to the metrics that were employed. The findings demonstrate that machine learning models may provide remarkably precise results when compared to state-of-the-art image processing technologies.

Hasan, Ruby. (2021). [34] One of the leading killers on a global scale in recent years has been cardiovascular disease. Changes in diet, work practices, and general way of life have all played a role in this worrying problem, which affects countries all over the world, from the most developed to the least. Reducing the expanding patient population and, ultimately, death rate, may be achieved by early diagnosis of the beginning indicators of cardiovascular illnesses and continued medical care. But it's hard to keep tabs on people and provide consultations when there aren't enough medical facilities and specialists. In order to make patient monitoring and treatment easier, technological interventions are necessary. Efficient prediction models for cardiovascular disorders may be developed using healthcare data gathered from numerous medical procedures and ongoing patient monitoring. An exceptional achievement in medicine may be the early detection of cardiovascular diseases, which may help in the decision-making process about lifestyle modifications in high-risk patients, therefore reducing problems. In this research, we take a look at how various machine learning algorithms have been utilized to forecast the occurrence of cardiac problems by analyzing past records and current medical data. In this article, we will go over all of the methods and then compare

and contrast them. Here we take a look at five widely used methods for estimating the likelihood of a heart attack and compare them in the literature. Several methods are used, including KNN, Decision Tree, Logistic Regression, Random Forest, and Gaussian Naive Bayes. The study goes on to detail the pros and cons of each method used to build the prediction models.

Kamiri, Jackson et al., (2021) [35] Because research techniques impact the quality and dependability of the outcomes, they play a crucial role in machine learning. Examining existing approaches to machine learning research as well as new topics and their potential effects on the field were the primary goals of this article. The researchers accomplished this by reviewing 100 publications published in IEEE journals since 2019. Machine learning, according to this study, relies on quantitative research methodologies, with experimental research designs being the prevalent strategy. Researchers today often use many algorithms to tackle an issue, according to the study. Researchers are increasingly relying on optimal feature selection as a means to enhance the efficiency of machine learning algorithms. Even though academics are starting to take processing time into account when evaluating algorithms, confusion matrices and their variants are still the most used approaches. The most popular tools for developing, training, and testing models are the Python programming language and associated libraries. Some of the most popular methods for handling classification and prediction issues include Decision Tree, Artificial Neural Networks, Naïve Bayes, Support Vector Machine, and Random Forest. It is quite probable that the recurrent patterns found in this study will pave the way for new areas of research in machine learning.

Eckart, Li et al., (2021)[36] When dealing with very complicated information, machine learning is a common method for discovering patterns and correlations. Some machine-learning methods are finding practical use thanks to recent developments in storage and processing power. A comparison between traditional statistical methods and machine learning algorithms is the goal of this study. Many scientific disciplines have long made use of these techniques for data grouping and information extraction. The key information about the various approaches, their data set needs, and the limits of each method make it difficult to apply them correctly. It would be much easier to include new machine learning algorithms into the present assessment if it were simpler to choose the correct approaches. Various machine learning algorithms are catalogued in

this work. A detailed comparison is made between four approaches (k-means algorithm, artificial neural network, regression method, and self-organizing map), and various selection criteria are highlighted. Lastly, we provide an estimate of the task and application domains, as well as any constraints, which can aid in making decisions for particular multidisciplinary analyses.

R M, Achshah et al., (2021)[37] Machine learning algorithms are the backbone of artificial intelligence. Machine learning algorithms come in a variety of flavors; developers choose the one that works best with their specific situation by weighing its benefits and drawbacks. This study examines the advantages and disadvantages of many popular ML techniques, including logistic regression, XG-Boost, naive bayes, decision tree, random forest, artificial neural network, convolution neural network, and linear regression. To help newcomers understand and choose the best supervised learning algorithm for their task, it analyzes and contrasts the aforementioned algorithms while outlining the core ideas. Choosing the best ML algorithm for a given application might be challenging for beginners. The purpose of this study is to provide a straightforward method for comparing algorithms' training data in order to choose the most appropriate one. We look at how each method performs with a variety of training datasets. We choose the most effective method by considering the following criteria: speed, dimensionality, normality of distribution, outliers, noise, missing values, and training data preparation requirements. The precision and accuracy of the chosen algorithm are crucial. Training all of the algorithms on the dataset and selecting the one with the highest accuracy score is a massive and laborious undertaking. So, it's peaceful if one can use the suggested method to compare and pick, which saves time.

El Guabassi, Inssaf et al., (2021) [38] As the need for accurate future predictions grows among the world's population, the ability to foretell relevant data in any field is quickly becoming an absolute must. Finding out what may happen is one method to know for sure what the future holds. To this end, machine learning provides a means of efficiently sifting through massive datasets in search of actionable insights. In order to assess students' progress, this study primarily aims to construct a prediction model. The results are therefore trifecta of donations. First, we will train a number of to our instructional dataset using supervised machine learning methods. Decision Tree, Random Forest, Partial Least Squares, Log-linear, Support Vector, ANCOVA, and Logistic Regression

are all part of this class of algorithms. The second objective is to evaluate the prediction model's associated algorithms using various metrics. The final goal is to identify the most critical aspects that impact the pupils' achievement or lack thereof. The findings of the experiments demonstrated that the Log-linear Regression method yields superior predictions, and they also identified the behavioral elements that impact students' performance..

Khalifa, Ramy et al., (2020) [39] In this work, we provide a Logical Analysis of Data (LAD)–based regression model. One method for generating patterns in supervised data mining is LAD, which is a combinatorial Boolean approach. Its primary use is in classification issues, where it has outperformed competing methods in terms of accuracy. In this work, we broaden the use of LAD to handle supervised data with continuous replies. An LAD regression model (LADR) is developed by us. Three discretization techniques are evaluated, each of which converts response values into a set of criteria. At each cutoff, LAD treats the data as a problem of two-class classification and pulls out the corresponding prescriptive patterns. Fitting a numerical continuous dependent response with the patterns created from the original data using cbmLAD software is what LADR regression is all about. As a result, we get a normalized regression model where the independent variables are all binary. When compared to linear regression (LR), support vector regression (SVR), decision tree regression (DTR), random forest (RF), and polynomial regression (PolyR), LADR outperforms all five methods on all six datasets. The Mean Absolute Error (MAE), Coefficient of Determination (R2), and Mean Square Error (MSE) are used to assess the performance, which is based on a 10-fold cross validation.

Apsemidis, Anastasios et al., (2020) [40] Classical process monitoring methods need to evolve to address the growing complexity of contemporary issues in industrial settings. One explanation for the surge in popularity of new Machine and statistics Learning approaches in the statistics world is this precise reason. This article delves into the specifics of process monitoring machine learning kernel methodologies and techniques. We review the process monitoring papers that employ kernel models and how these models are coupled with other Machine Learning techniques after we introduce the principle of kernel methods. In conclusion, we review the whole body of literature and highlight key aspects.

Maulud, Dastan et al., (2020) [41] When it comes to machine learning and statistics, linear regression is among the most popular and all-encompassing algorithms. Discovering a straight line between a few factors is the goal of linear regression. Both simple regression and multiple regression are forms of linear regression (MLR). This study analyzes the performance of linear and polynomial regression based on the best way to improve prediction and accuracy, and it examines many studies by different researchers on the topic. Datasets are the primary emphasis of the reviewed publications; a model's efficacy can only be verified by correlating it with the actual values of the explanatory variables.

Razaque, Abdul et al., (2020) [42] Since it aids in the development of alternate recommendation systems for academically inferior students, predicting students' performance is a critical topic for learning environments. Consequently, several initiatives aimed at enhancing education were put into place. However, most of the present methods don't evaluate students' development. In this study, six machine learning models—Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, Ada Boost, and Stochastic Gradient Descent—were used to evaluate the students' progress. The criteria used to evaluate the performance include sensitivity, accuracy, precision, and f-measure. The findings show that Stochastic Gradient Descent is the most efficient model among the ones we chose for training tiny datasets. On top of that, when compared to other models, it gives results with better precision. The goal of this contribution is to create the most effective model that may be used to draw conclusions about students' academic performance.

Kenge, Rohit. (2020).[43] Computer algorithms and data samples are the building blocks of a mathematical standard model for decision-making that does not need programming, a process known as machine learning. When a computer system learns to do a job automatically, it indicates it has never been trained to do that task before. We dove deep into the notion of machine learning, investigating its applications, methods, models, and constraints, as well as its connections to related disciplines. In the field of machine learning, supervised, unsupervised, and semi-supervised methods are the most common. In addition to this Robot learning, feature learning, sparse dictionary learning, reinforcement learning, and self-learning are some of the concepts in machine learning. Following are a few examples of training models: a Bayesian

network, decision trees, support vector machines, artificial neural networks, regression analysis, and evolutionary algorithms. Machine learning has a few drawbacks that we discovered, including its high installation cost, prejudice, and lack of accuracy and ethics. In order to confirm these restrictions, we used a Google form to poll 400 consumers in the Nashik city and asked them two questions: When using e-commerce mobile applications, do customers experience any bias? When dealing with medical concerns at hospitals, does the consumer feel robbed? Our sample survey data shows that consumers have a negative impression of health care providers due to unethical treatment and a biassed experience while utilizing e-commerce applications. In addition, we suggested a few ways around machine learning's shortcomings, including an online self-declaration form, standardized medical bill proposals, and individualized approaches to hardware installation.

Mahesh, Batta. (2019).[44] The study of statistical models and techniques that computer systems use to carry out a given job autonomously from human programming is known as machine learning (ML). Algorithms for learning in a wide variety of programs that we use often. Learning algorithms that has learnt how to rank online sites is one of the reasons why web search engines like Google operate so well every time someone uses them to search the internet. For example, these algorithms find usage in data mining, image processing, predictive analytics, and many more fields. One major benefit of machine learning is the ability for algorithms to learn and execute tasks autonomously once given data. This article has provided a high-level overview of machine learning algorithms, as well as some predictions about their potential future uses.

Gao, Qian-Qian et al., (2019) [45] For binary classification issues, this research proposes a novel QLSTSVM, which stands for quadratic kernel-free least square twin support vector machine. One benefit of using QLSTSVM for nonlinear classification issues is that the kernel function and associated parameters don't need to be selected. We immediately answer the reformulated consensus QLSTSVM by employing the alternate direction approach of multipliers after applying the consensus procedure. The QLSTSVM may also be solved using the Karush-Kuhn-Tucker (KKT) conditions, which help to decrease CPU time. Two synthetic datasets and several benchmark datasets from the University of California, Irvine (UCI) are used to evaluate the

performance of QLSTSVM. In terms of classification accuracy and operation time, numerical studies suggest that the QLSTSVM may surpass many current approaches for solving twin support vector machines with Gaussian kernels.

M. Pradhan et al., (2019) [46] The potential for enhanced remote sensing technology to use hyperspectral data for a variety of applications has grown thanks to the fast development of multichannel imaging sensors. To get high performance in supervised hyperspectral data classification, it is crucial to collect an appropriate training set. But in many image analysis applications, including hyperspectral images (HSIs), getting a labelled training sample may be a tedious, costly, and time-consuming ordeal. The image analysis framework relies heavily on the active learning (AL) approach to circumvent this issue. According to the research, HSI classification using AL has not yet concentrated on learning rate in terms of calculation time, but on correctness. This study presents an integration of the multiview-based AL approach with the kernel-based extreme learning machine (KELM) classifier. The widely-used kernel-based support vector machine (KSVM) was also compared to our method. Two Hyperspectral Image datasets, one from the Kennedy Space Centre (KSC) and the other from Botswana (BOT), were used to verify our findings. The proposed approach (KELM-AL) achieved the classification accuracy up to 91.15% in KSC dataset while 95.02% in case of BOT dataset with computation time of 149.78 s and 104.98 s, respectively. While KSVM-AL achieved the classification accuracy up to 91.59% in KSC dataset while 95.96% in case of BOT dataset with computation time of 7532.25 s and 6863.60 s, respectively. This shows that classification accuracy obtained by KELM-AL is comparable to KSVM-AL approach but significantly reduces the computational time. As a consequence, the suggested approach reduces computing time significantly while demonstrating promising results with sufficient classification accuracy.

Cao, Jianfang et al., (2019) [47] To improve upon the present state of the art in image classification algorithms, it is suggested to use adaptive feature weight updates. This will help overcome the shortcomings of both basic multifeature fusion methods and algorithms that rely on a single feature for classification. In order to find the best weight combinations, we employ the MapReduce parallel programming paradigm on the Hadoop platform to adaptively fuse hue, local binary pattern (LBP), and scale-invariant feature transform (SIFT) characteristics that are derived from photos. Afterwards, the

best SVM classification model is obtained by using the support vector machine (SVM) classifier for parallel training. This model is subsequently evaluated. The SUN, Pascal VOC 2012, and Caltech 256 databases were used to construct a vast picture archive. In the experiment, we measure the speedup, classification accuracy, and training duration. We find that in a cluster setting, the speedup tends to expand linearly. In comparison to popular classification algorithms like CNN and power mean SVM, this approach outperforms them in terms of hardware costs, performance, accuracy, and time. The classification accuracy rate goes over 95% as the quantity and variety of pictures both grow. The suggested algorithm's training time is only one-fifth of that of conventional methods with a single node when the number of pictures approaches 80,000. The algorithm's efficacy is shown by this outcome, which lays the groundwork for efficient processing and analysis of picture large data.

Mahesh, Batta. (2019). [48] The study of statistical models and techniques that computer systems use to carry out a given job autonomously from human programming is known as machine learning (ML). Algorithms for learning in a wide variety of programs that we use often. Learning algorithms that has learnt how to rank online sites is one of the reasons why web search engines like Google operate so well every time someone uses them to search the internet. For example, these algorithms find usage in data mining, image processing, predictive analytics, and many more fields. One major benefit of machine learning is the ability for algorithms to learn and execute tasks autonomously once given data. This article has provided a high-level overview of machine learning algorithms, as well as some predictions about their potential future uses.

Rong, Shen et al., (2018) [49] The impact of temperature fluctuation on the sale of iced items is the focus of this research. We will begin by gathering information on last year's forecasted temperatures and iced product sales, and then we will compile and sanitize the data. At last, using data mining theory to the cleaned-up data, we will construct a mathematical regression analysis model. The process of investigating the connection between two variables—the independent and the dependent ones—is known as regression analysis. In this work, we provide a linear regression model that fits the real world by first defining a basic model based on an actual issue and then implementing it using Python3.6, the most recent and widely used programming language. Pure

object-oriented programming, platform independence, and a language that is both brief and beautiful are all qualities that Python 3.6 enjoys. To provide the groundwork for the corporation to fine-tune its production on a monthly, weekly, or even daily basis, we will invoke the relevant library function to forecast iced product sales based on temperature variance. This means that overproduction won't happen. Furthermore, the alternative scenario is that the profit will be impacted by the reduced output, since the increase in temperature will also be prevented. As a result, the regression model is useful as a benchmark in other areas of marketing as well.

Y C a, Padmanabha et al., (2018) [50] Despite the low cost of unlabeled data, most application domains do not have enough labelled data. Because skilled domain specialists are needed to provide labels to the unlabeled data patterns, obtaining labelled examples is a challenging task. As a compromise between fully supervised and completely unsupervised learning, semi-supervised learning tackles this issue. A selection of semi-supervised learning (SSL) strategies, including methods for self-training and co-training as well as multi-view learning and TSVMs, are covered in this work. Compared to more conventional supervised and unsupervised learning methods, the accuracy of SSL's conventional classification into semi-supervised clustering and semi-supervised classification is much higher. Scalability and applicability of semi-supervised learning are also covered in the study.

Akinsola, J E T. (2017). [51] Statistical machine learning aims to create algorithms that can learn from human-provided examples, generalize those findings, and then employ those predictions to the future. Supervised classification is a typical task for AI systems. Discover which supervised learning algorithm works best with your dataset, instance count, and characteristics by reading this article's summary of supervised machine learning (ML) classification algorithms, which compares and contrasts several approaches. The seven machine learning algorithms that were considered using the Waikato Environment for Knowledge Analysis (WEKA) application were Decision Table, Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Neural Networks (Perceptron), JRip, and Decision Tree (J48). The 786 classification cases found in the Diabetes data set were used to construct the algorithms. There is a single dependent variable and eight independent factors in the research. When compared to other methods, support vector machine (SVM) proved to be the most

accurate and precise. After Support Vector Machines (SVM), Random Forest and Naïve Bayes were the most accurate algorithms for categorization. There are two primary components, as shown by the results: first, the time needed to build the model and its correctness; second, the kappa statistic and its MAE. As a result, ML algorithms need precision, accuracy, and a low margin of error to achieve supervised predictive machine learning.

Mohamed, Amr. (2017).[52] Decision Tree, K-Nearest-Neighbor, Artificial-Neural-Network, and Support Vector Machine are four popular supervised machine learning algorithms that have been compared. This report primarily focused on the main points of each method, including its benefits and drawbacks. The research concludes with practical application to compare their performance. Their efficacy has been assessed using a number of metrics, including specificity and sensitivity. According to the results of this research, no one metric can reveal all aspects of a classifier's efficiency, and no single classifier can meet all requirements.

Zareapoor, Masoumeh et al., (2017) [53] It may be computationally challenging to do classification with a huge number of features and thousands of classes. Classification performance and computing cost may both be negatively impacted by the inclusion of irrelevant information. Additionally, class-confusability occurs often in classification with thousands or more classes, and training error increases with confusable classes. A feature extractor and a classifier should be wisely combined to create a robust classification model that can handle high-dimensional data with many classes, such as k [U+202F]≥[U+202F]10^4. Although support vector machines with the right kernel show promise for making decisions based on well-behaved features, they may have unintended consequences when trying to model massive datasets with a high number of classes. Architectures with remarkable learning and feature collection capabilities include deep belief networks. In this research, we provide a hybrid system that combines the training of a supervised deep belief network (DBN) to choose generic features with the training of a kernel-based support vector machine (SVM) using those features. Our hybrid model outperforms state-of-the-art methods on real-world datasets with 20,000 to 65,000 classes, thanks to the accuracy-preserving substitution of linear kernels for nonlinear ones caused by the high number of classes.

Fan, Mengbao et al., (2016) [54] When it comes to nondestructive assessment of

product quality and structural integrity, eddy current testing is a common, cost-effective, and non-contact option. One of the most important performance criteria for defect characterization is the excitation frequency. Optimal frequency for detection sensitivity and broad spectrum content have been the subject of several intriguing articles in the literature. Nevertheless, there has been a dearth of study into optimizing frequency in relation to characterisation results. In order to improve the efficiency of surface defect categorization, this research investigates the optimal excitation frequency. Using a support vector machine (SVM) and kernel principal component analysis (KPCA), the effects of excitation frequency on a set of defects were uncovered in terms of detection sensitivity, contrast between defect characteristics, and classification accuracy. When the excitation frequency is adjusted close to the frequency at which the maximum probe signals are recovered for the greatest flaw, it is seen that probe signals are the most sensitive for a group of defects. Optimal hyperplanes are used by the SVM to minimize structural risk after KPCA, which results in optimal margins between defect features. This leads to the highest possible level of categorization accuracy. The major contribution is that the effects of excitation frequency on defect characterization are explained, and methods based on experiments are suggested to find the best excitation frequency for a set of defects, not just one, in terms of characterization performance.

Peng, Chong et al., (2016)[55] We present a novel discriminative regression-based supervised learning model. With the use of class information, this new model can estimate a regression vector that represents the similarity between test and training samples. Because of this, our model is unique compared to traditional regression models and locally linear embedding methods, and it is well-suited to high-dimensional supervised learning challenges. Whether your data is high- or low-dimensional, our model can handle it all, and it's easy to add support for nonlinear relationships. Two optimization techniques are given for the model's convex objective function. Each of these optimization strategies yields a scalable solution with a linear time complexity that can be proven analytically. The experimental findings show that the suggested strategy works well with different types of data. Linear solvers provide encouraging results on large-scale classification, and our technique outperforms several popular classifiers on high-dimensional data while being on par with them on low-dimensional data.

Bai, Yanqin et al., (2015) [56] When it comes to classification problems, support vector machines (SVMs) have shown to be both successful and promising. Classification and prediction of illnesses using real-world data has recently seen the effective use of SVMs. We provide a novel approach to binary classification using a quadratic kernel-free least squares support vector machine (QLSSVM). An benefit of the QLSSVM model over the existing least squares SVM is that it is a kernel-free convex quadratic programming problem. The decision variables of QLSSVM are divided into local and global variables using the consensus approach. The consensus QLSSVM is developed by transforming the QLSSVM into an alternating direction multiplier approach using a Gaussian back substitution, and then the problem is solved. We conclude by demonstrating our QLSSVM via numerical experiments using two distinct training data sets. To validate our QLSSVM's performance, we first apply a numerical test using synthetic data. The second one shows that our model outperforms other existing approaches in illness classification using the diseases data set from the University of California, Irvine, Machine Learning Repository. This allows us to apply our QLSSVM to this domain. More specifically, our numerical example shows how successful our QLSSVM is for a specific illness diagnosis using a customized data set for heart disease given by the Hungarian heart disease database.

Iqbal, Muhammad et al., (2015) **[57]** Giving computers the ability to learn from their own data and experiences is a primary goal of machine learning. Machine learning has already found many useful uses; for example, there are classifiers that can be trained on email messages to differentiate between spam and non-spam, systems that can analyze sales data to forecast client purchasing behavior, fraud detection systems, and many more. In this research, we will concentrate on the strengths and weaknesses of supervised learning classification algorithms, however machine learning may be implemented as association analysis via unsupervised learning and reinforcement learning as well. Using predictor characteristics to construct a succinct model of the distribution of class labels is the objective of supervised learning. When testing cases with known predictor feature values but unknown class label values are encountered, the resultant classifier is used to ascribe class labels to these instances. We hope our work will pave the way for future researchers to evaluate and contrast supervised learning algorithms' efficacy and impuissance, as well as to direct new fields of research.

Tian, Yingjie et al., (2015) [58] A new binary classification method called NSVMOOP—a nonparallel support vector machine based on a single optimization problem—is introduced in this study. Incorporating the structural risk reduction concept, our NSVMOOP is designed to accomplish class separation using the maximum angle between feature space normal vectors and decision hyperplanes. In contrast to previous nonparallel classifiers like the representative twin support vector machine, it employs a modified sequential minimization optimization approach to solve a single quadratic programming problem, resulting in the simultaneous construction of two nonparallel hyperplanes. Both theoretical and experimental analyses are conducted on the NSVMOOP. Results from experiments conducted on synthetic and publically accessible benchmark datasets demonstrate its practicability and efficacy..

Santos, Adam et al., (2015) [59] In this research, we provide four different kernel-based algorithms—one-class support vector machine, support vector data description, kernel principal component analysis, and greedy kernel principal component analysis—for damage identification in different operational and environmental settings. For this performance evaluation, we retrieved acceleration time-series from a lab-based array of accelerometers. This work primarily contributes by demonstrating that the suggested algorithms may be used for damage detection and by comparing their classification performance to that of four other algorithms that have already been established as trustworthy methods in the literature. It turned out that each of the suggested algorithms outperformed its predecessors in terms of categorization accuracy.

Pal, Mahesh et al., (2013) [60] This letter assesses the efficacy of a novel ELM algorithm that uses hyperspectral and multi-spectral remote sensing data for land cover categorization. Support vector machines (SVMs), the most popular methods, are used to compare the outcomes. We compare the outcomes according to computational cost, classification accuracy, and user-defined parameter count for simplicity of use. The interoperability of the two techniques was ensured by using a radial basis kernel function with both the SVM and the kernel-based extreme-learning machine algorithms. As far as classification accuracy goes, the findings show that the new method is on par with or even better than SVM. What's more, it has a far lower computing cost and doesn't need a multiclass approach to work.

Pozun, Zachary et al., (2012) [61] Here, we provide a strategy for improving transition

state theory dividing surfaces via SVM optimization. No previous knowledge or intuition about reaction processes is necessary for the generation of the resultant dividing surfaces. We use a machine-learning cycle that refines the surface via molecular dynamics sampling in order to provide optimum division surfaces. The crucial low-energy saddle points are included in the machine-learned surfaces, as we show. In order to discover unanticipated chemically relevant processes, it is possible to extract reaction mechanisms from machine-learned surfaces. Also, in contrast to a distance-based dividing surface, we demonstrate that machine-learned surfaces considerably enhance the transmission coefficient for an adatom exchange involving several linked degrees of freedom on a (100) surface..

Gönen, Mehmet & Alpaydın, Ethem. (2011) [62] The use of a weighted linear sum of kernels to combine several kernels is an approach that has been suggested in various techniques in the last few years. These many kernels may be using data from a variety of sources, or they might be corresponding to various ways of looking at the same data in terms of similarity. We observe that these approaches include novel regularization parameters that impact the solution quality, in addition to the conventional ones from the canonical support vector machine formulation. In this study, we suggest optimizing them using response surface methodology using cross-validation data. Our suggested regularized variation is compared to multiple kernel learning on several bioinformatics and digit recognition benchmark data sets with respect to accuracy, support vector count, and number of kernels used. We observe that our suggested variation accomplishes comparable or improved accuracy with fewer kernel functions and/or support vectors by implementing appropriate regularization. Additionally, it enables enhanced knowledge extraction by eliminating superfluous kernels and ensuring that the preferred kernels accurately represent the problem's characteristics.

Khemchandani, Reshma et al., (2009) [63] Two similar SVM-type problems, smaller than the one in a standard SVM, are solved to find a pair of non-parallel planes in twin support vector machines (TWSVMs). Nevertheless, the selection of the kernel affects the performance of the TWSVM classifier, much like other classification algorithms. In this work, we define the TWSVM kernel selection issue as an iterative alternating optimization problem over the convex set of finitely numerous basic kernels. Using a few machine learning benchmark datasets developed at UCI, we show that the

suggested categorization technique works.

Jain, Pooja et al., (2009) [64] Applying supervised machine learning techniques to a dataset consisting of 11,360 domain pairs representing protein structures (within a range of 35% sequence identity) and three secondary structural components, we investigate the possibility of automating protein structural categorization. Given a one-dimensional representation of the domain structures, fifteen algorithms from five classes of supervised algorithms are tested for their capacity to learn for two protein domains, the most fundamental shared structural level in the SCOP hierarchy. This model contains evolutionary data in terms of sequence identity and structural data describing the secondary structure components and domain lengths. There are two stages to the assessment process: picking the top performing base learners and then testing boosted and bagged meta learners. With F-measures of 0.97, 0.85, 0.93, and 0.98 for protein categorization to the Class, Fold, Super-Family, and Family levels in the SCOP hierarchy, the most accurate model was determined to be the boosted random forest, a collection of decision trees. Its cross-validated accuracy was 97.0%. By improving the accuracy of instance classification in less populated classes, the meta learning regime—particularly boosting—improved performance.

Agarwal, Sumeet et al., (2008) [65] In this paper, we use kernel-based machine learning techniques to online learning scenarios and examine the associated need to simplify the learned classifier. When dealing with circumstances that entail flowing data, whether in medical or financial applications, online approaches really shine. We demonstrate that a classifier can be constructed using the span of support vectors idea that meets space and time limitations and performs adequately; this classifier may therefore be applicable to such online scenarios.

Kotsiantis, Sotiris. (2007). [66] Finding algorithms that can generalize from examples given to them and use them to predict future occurrences is the goal of supervised machine learning. Basically, supervised learning is all about creating a clear model of how class labels are distributed based on predictor attributes. When testing cases with known predictor feature values but unknown class label values are encountered, the resultant classifier is used to ascribe class labels to these instances. This study provides an overview of several classification strategies in supervised machine learning. Undoubtedly, this article is not meant to be an exhaustive examination of all supervised

machine learning classification algorithms (also called induction classification algorithms). However, we do hope that the references provided will address the key theoretical concerns, leading researchers to intriguing new avenues of inquiry and potentially uncovering unexplored combinations of bias.

Kotsiantis, Sotiris et al., (2006) [67] For example, so-called Intelligent Systems often do supervised categorization. Statistics (Bayesian Networks, Instance-based techniques) and Artificial Intelligence (Logic-based techniques, Perceptron-based techniques) have therefore given rise to a plethora of methods. Using predictor characteristics to construct a succinct model of the distribution of class labels is the objective of supervised learning. When testing cases with known predictor feature values but unknown class label values are encountered, the resultant classifier is used to ascribe class labels to these instances. Different classification techniques are detailed in this work, along with the most current effort to improve classification accuracy, which is called ensembles of classifiers.

Kivinen, Jyrki et al., (2004)[68] With all the training data provided in advance in a batch environment, kernel based techniques like support vector machines have been very successful with a variety of challenges. The so-called kernel technique and the high margin notion are combined in support vector machines. Few online settings that are appropriate for real-time applications have made use of these technologies. Online education in a Reproducing Kernel Hilbert Space is the focus of this research. Our simple and computationally efficient techniques cover a broad variety of tasks, including classification, regression, and novelty detection, by considering classical stochastic gradient descent inside a feature space and using some straightforward strategies. We also consider the need of big margins for classification in the online situation with a drifting objective, and we show that the kernel method may be used there. In addition to demonstrating that the hypothesis converges to the minimiser of the regularized risk functional, we estimate worst-case loss limits. We provide practical data that back up the theory and show how effective the new algorithms are for detecting online innovation.

RESEARCHGAP

In order to tackle classification problems, this work seeks to examine machine learning

models that make use of optimum kernel-generated surfaces. Machine learning models, especially those trained on complicated and high-dimensional datasets, may be improved by investigating the function of kernel approaches. Class imbalance, noisy data, among the common classification issues that this research seeks to address via the optimization of kernel functions are the curse of dimensionality and others. This study aims to provide insight into the potential for optimal kernel-generated surfaces to enhance the accuracy and generalizability of classification algorithms, their use is investigated across different datasets. Improved and more efficient machine learning models with broad applicability (e.g., image recognition, bioinformatics, and data mining) are anticipated outcomes of this strategy.

CHAPTER 3
REGULARIZATION-BASED AND ROBUST ASYMMETRIC V-TWIN SUPPORT VECTOR REGRESSION USING PINBALL LOSS FUNCTION

CHAPTER 3
REGULARIZATION-BASED AND ROBUST ASYMMETRIC V-TWIN SUPPORT VECTOR REGRESSION USING PINBALL LOSS FUNCTION

3.1 REGULARIZATION BASED LAGRANGIAN ASYMMETRIC-V-TWIN SUPPORT VECTOR REGRESSION USING PINBALL LOSS

In regression issues, whether the samples are inside or beyond the range of the estimate functions, and if so, which ones. In this part, a novel twin support vector regression technique is described using the robust pinball loss function, which is an extension of ε-insensitive loss function. By splitting the outliers asymmetrically over both regions, pinball loss limits the fitting error and exploits the properties of ε-insensitive loss. An asymmetric tube may be built by introducing the asymmetric loss function, also known as the pinball loss function.

The computation cost of the suggested model is reduced by using a straightforward linearly convergent approach to get the solution. By including regularization into the SRM theory's goal functions, the problem becomes very stable and convex.

Experiments on common real-world datasets based on many quality measures and on synthetic datasets with symmetric and asymmetric structural noise (e.g.,heteroscedastic and Gaussian noise) demonstrate the efficacy of the proposed approach. Pinball loss also does a better job of surviving outliers than TSVR.

3.1.1 THE LOSS FUNCTIONS

ε -insensitive loss

The ε - is defined as the insensitive loss function.

ε - concept of loss that is not sensitive

$$L_\varepsilon(a) = \begin{Bmatrix} a-\varepsilon, & a \geq \varepsilon \\ 0, & \varepsilon < a < \varepsilon, \\ -a-\varepsilon & a \leq -\varepsilon, \end{Bmatrix}$$

The pinball loss

Here is the description of the pinball loss function:

pinball loss definition

$$L_{\varepsilon}^{\xi}(x)=\begin{cases}\dfrac{1}{2\xi}(x-\varepsilon), & x\geq\varepsilon,\\ 0, & -\varepsilon<x<\varepsilon,\\ \dfrac{1}{2(1-\xi)}(-x-\varepsilon), & x\leq-\varepsilon,\end{cases}$$

where ξ is a variable linked to imbalance and $\xi = 0.5$, it is going to resemble ε - heartless death.

3.1.2 STRENGTHENED STANDARDIZATION THE LASY-N-TSVR IS A LAGRANGIAN ASYMMETRIC V-TWIN SUPPORT VECTOR REGRESSION MODEL THAT INCORPORATES PINBALL LOSS.

This section examines the application of the pinball loss function in dealing with asymmetric noise and outliers in difficult real-world circumstances using the LAsy-□-TSVR, an improved regularization-based technique.

Instead of calculating QPPs, the suggested technique solves the linearly convergent iterative approach, which boosts prediction performance while reducing computational cost.

This linearly convergent iterative technique takes into account the inputs and determines the initial matrix inversion. We substitute its 1-norm for the slack variables vector. ζ_1 and ζ_2, By squaring the vector of slack variables in the 2-norm, we demonstrate that there is a globally unique solution. Our suggested LAsy- □ -TSVR formulation makes the issue enormously convex. Put limitations on regularization.

$$\frac{C_3}{2}(||w_1||^2+b_1^2)\ and\ \frac{C_4}{2}(||w_2||^2+b_2^2)$$

The optimization problem may be modified to adhere to SRM theory by including the TSVR and Asy-□-TSVR scenarios. These scenarios strengthen the dual formulations and ensure that the model is well-posed.

LINEAR LASY-ν -TSVR

The procedures for regression $f_1(x) = w_1^t x + b_1 and\ f_2(x) = w_2^t x + b_2$ when the revised QPPs are solved yield.

$$\min \frac{C_3}{2}(\| w_1 \|^2 + b_1^2) + \frac{1}{2}\| y - (Bw_1 + eb_1) \|^2 + \frac{1}{p} C_1 \zeta_1^t \zeta_1 + C_1 \nu_1 \varepsilon_1^2$$

subject to. $y - (Bw_1 + eb_{1)} \geq -e\varepsilon_1 - 2(1-\xi)\zeta_1$ (3.1)

and

$$\min \frac{C_4}{2}(\| w_2 \|^2 + b_2^2) + \frac{1}{2}\| y - (Bw_2 + eb_2) \|^2 + \frac{1}{p} C_2 \zeta_2^t \zeta_2 + C_2 \nu_2 \varepsilon_2^2$$

Subject to

$$(Bw_2 + eb_2) - y \geq -e\varepsilon_2 - 2\xi\zeta_2 \quad (3.2)$$

where $C_a|\mathrm{a} = 1, \ldots, 4 > 0, \varepsilon_1, \varepsilon_2 \geq 0$ and ν_1, ν_2 are input parameters; $\zeta_1 = (\zeta_{11}, \ldots \zeta_{1p})^1, \zeta_2 = (\zeta_{21}, \ldots \zeta_{2p})^t$ make up the variables that provide a degree of flexibility ξ functions as pinball's loss function. Forget about the slack variables' non-negative requirements at this point in (3.1) and (3.2). Equations (3.1) and (3.2) may be transformed into their Lagrangian functions by using the Lagrangian multipliers α,β > $0 \in \Re^p$as

$$L_1 = \frac{C_3}{2}(\| w_1 \|^2 + b_1^2) + \frac{1}{2}\| y - (Bw_1 + eb_1) \|^2 + \frac{1}{p} C_1 \zeta_1^t \zeta_1 + C_1 \nu_1 \varepsilon_1^2 - \alpha^t (y - (Bw_1 + eb_1) + e\varepsilon_1 + 2(1-\xi)\zeta_1) \quad (3.3)$$

and

$$L_2 = \frac{C_4}{2}(\| w_2 \|^2 + b_2^2) + \frac{1}{2}\| y - (Bw_2 + eb_2) \|^2 + \frac{1}{p} C_2 \zeta_2^t \zeta_2 + C_2 \nu_2 \varepsilon_2^2 - \beta^t ((Bw_2 + eb_2) - y + e\varepsilon_2 + 2\xi\zeta_2) \quad (3.4)$$

In addition, by using the K.K.T. requirements from equation (3.3), we get

$$\frac{\partial L_1}{\partial w_1} = C_3 w_1 - B^1\big(y - (bw_1 + eb_1)\big) + B^1 a = 0, \quad (3.5)$$

$$\frac{\partial L_1}{\partial b_1} = C_3 b_1 - e^1\big(y - (Bw_1 + eb_1)\big) + e^t a = 0, \tag{3.6}$$

$$\frac{\partial L_1}{\partial \zeta_1} = \frac{C_1}{p}\zeta_1 - 2(1-\xi)a = 0, \tag{3.7}$$

$$\frac{\partial L_1}{\partial \varepsilon_1} = 2C_1 v_1 \varepsilon_1 - e^1 a = 0. \tag{3.8}$$

Equations (3.5) and (3.6) are combined to give us

$$\left[\frac{w_1}{b_1}\right] = (D_1^t D_1 + C_3 I)^{-1} D_1^t (y - a) \tag{3.9}$$

where $D_1 = [B, e]$ represents an enhanced matrix.

For primary issue (3.1), the dual QPP may be found by applying equations (3.3), (3.7), (3.8), and (3.9).

$$\min \frac{1}{2}\alpha'\left(D_1(D_1'D_1 + C_3 I)^{-1}D_1' + \frac{4p(1-\xi)^2}{C_1} + \frac{ee'}{2C_1 v_1}\right)\alpha - (D_1(D_1'D_1 + C_3 I)^{-1}D_1' y - y)'\alpha \tag{3.10}$$

In a similar vein, the dual QPP of the primary issue (3.2) is obtained as

$$\min \frac{1}{2}\beta'\left(D_1(D_1'D_1 + C_4 I)^{-1}D_1' + \frac{4p\xi^2}{C_2} + \frac{ee'}{2C_2 v_2}\right)\beta - (-D_1(D_1'D_1 + C_4 I)^{-1}D_1' y + y)'\beta \tag{3.11}$$

By solving the QPPs (3.10) and (3.11), we may determine the values of α and β. By averaging $f_1(x)$ and $f_2(x)$ for every given test sample, we may get the final regression function f, using $x \in \Re^q$:

$$f_1(x) = w_1^t + b_1 = [x^t 1]((D_1^t D_1 + C_3 I)^{-1} D_1^t (y - a)) \tag{3.12}$$

and

$$f_2(x) = w_2^t + b_2 = [x^t 1]((D_1^t D_1 + C_4 I)^{-1} D_1^t (y - \beta)) \tag{3.13}$$

Non-linear LAsy-ν -TSVR

The functions provided by the kernel $f_1(x)K(x^t, B^t)w_1 + b_1$ and $f_2(x)K(x^t, B^t)w_2 + b_2$ choose the appropriate QPPs for the development of non-linear LAsy- ν -TSVR.

$$\min \frac{C_3}{2}(\| w_1 \|^2 + b_1^2) + \frac{1}{2}\| y - (K(B,B^t)w_1 + eb_1) \|^2 + \frac{1}{p}C_1\zeta_1^t\zeta_1 + C_1\nu_1\varepsilon_1^2$$

Subject to. $y - (K(B,B^t)w_1 + eb_1) \geq -e\varepsilon_1 - 2(1-\xi)\zeta_1$ (3.14)

and

Subject to. $(K(B,B^t)w_2 + eb_2) - y \geq -e\varepsilon_2 - 2\xi\zeta_2$ (3.15)

spectively, where $C_a \mid a = 1,...,4 > 0$ a ; $\varepsilon_1, \varepsilon_2 \geq 0$ and ν_1, ν_2 serve as parameters for input ξ serves as the loss function for pinball ζ_1, ζ_2 comprise the factors that allow for some leeway.

With the use of Lagrangian multipliers $\alpha,\beta > 0 \in \Re^p$, , we get the Lagrangian functions of equations (3.14 and 3.15).

$$L_1 = \frac{C_3}{2}(\| w_1 \|^2 + b_1^2) + \frac{1}{2}\| y - (K(B,B^t)w_1 + eb_1) \|^2 + \frac{1}{p}C_1\zeta_1^t\zeta_1 + C_1\nu_1\varepsilon_1^2$$

$$-\alpha^t\left(y - (K(B,B^t)w_1 + eb_1) + e\varepsilon_1 + 2(1-\xi)\zeta_1\right) \quad (3.16)$$

and

$$L_2 = \frac{C_4}{2}(\| w_2 \|^2 + b_2^2) + \frac{1}{2}\| y - (K(B,B^t)w_2 + eb_2) \|^2 + \frac{1}{p}C_2\zeta_2^t\zeta_2 + C_2\nu_2\varepsilon_2^2$$

$$-\beta^t\left((K(B,B^t)w_2 + eb_2) - y + e\varepsilon_2 + 2\xi\zeta_2\right) \quad (3.17)$$

Applying the K.K.T. criterion yields the dual QPPs of main issues (3.16) and (3.17).

$$\min \frac{1}{2}\alpha^t\left(D_2(D_2{}^tD_2 + C_3I)^{-1}D_2{}^t + \frac{4p(1-\xi)^2}{C_1} + \frac{ee^t}{2C_1\nu_1}\right)\alpha - (D_2(D_2{}^tD_2 + C_3I)^{-1}D_2{}^ty - y)^t\alpha \quad (3.18)$$

and

$$\min \frac{1}{2}\beta^t\left(D_2(D_2{}^tD_2 + C_4I)^{-1}D_2{}^t + \frac{4p\xi^2}{C_2} + \frac{ee^t}{2C_2\nu_2}\right)\beta - (-D_2(D_2{}^tD_2 + C_4I)^{-1}D_2{}^ty + y)^t\beta \quad (3.19)$$

where $D_2 = [K(B,B^t e]$ is an augmented matrix.

The final estimate function, f (.), is determined by averaging the following non-linear functions, which yield the non-linear kernel, $f_1(x)$ and $f_2(x)$, after calculating the values

of α and β from (3.18) and (3.19).

$$f_1(x) = [K(x^t, \mathrm{B}^t) \quad 1]\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = [K(x^t, \mathrm{B}^t) \quad 1]((D_2{}^t D_2 + C_3 I)^{-1} D_2{}^t (y - \alpha)) \tag{3.20}$$

$$f_2(x) = [K(x^t, B^t) \quad 1]\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = [K(x^t, B^t) \quad 1]((D_2{}^t D_2 + C_4 I)^{-1} D_2{}^t (y + \beta)) \tag{3.21}$$

An alternative way to rephrase issues (3.18) and (3.19) is as follows:

$$\min_{0 \le \alpha \in \Re^p} L_1(\alpha) = \frac{1}{2}\alpha^t E_1 \alpha - r_1^t \alpha \tag{3.22}$$

And

$$\min_{0 \le \beta \in \Re^p} L_2(\beta) = \frac{1}{2}\beta^t E_2 \beta - r_2^t \beta \tag{3.23}$$

Respectively, where

$$E_1 = \left(D_2(D_2{}^t D_2 + C_3 I)^{-1} D_2{}^t + \frac{4p(1-\xi)^2}{C_1} + \frac{ee^t}{2C_1 v_1} \right), r_1 = D_2(D_2{}^t D_2 + C_3 I)^{-1} D_2{}^t y - y$$

$$E_2 = \left(D_2(D_2{}^t D_2 + C_4 I)^{-1} D_2{}^t + \frac{4p\xi^2}{C_2} + \frac{ee^t}{2C_2 v_2} \right), \text{and } r_2 = -D_2(D_2{}^t D_2 + C_4 I)^{-1} D_2{}^t y + y$$

Classical complimentary problems of the following kind are generated, respectively, by subjecting the QPPs (3.22) and (3.23) to the KKT optimality conditions.

$$0 \le (E_1\, a - r_1) \perp \alpha \ge 0 \tag{3.24}$$

and

$$0 \le (E_2\, \beta - r_2) \perp \beta \ge 0 \tag{3.25}$$

In order to verify the $0 \le x \perp y \ge 0$ if and only if $x = (x - \psi y)_+$ regardless of the parameters, x, and y vectors $\psi > 0$, The following fixed point theorems rephrase the corresponding set of questions from (3.24) and (3.25): to address any $\psi_1, \psi_2 > 0$, the relations

$$(E_1 a - r_1) = \left(E_1 a - \psi_1 a - r_1\right)_+ \tag{3.26}$$

and

$$(E_2 \beta - r_2) = \left(E_2 \beta - \psi_2 \beta - r_2\right)_+ \tag{3.27}$$

One can suggest the following straightforward solving the problems with (3.22) and (3.23), using an iterative approach in the following way.

$$a^{i+1} = E_1^{-1}\left(\left(E_1 a^i - r_1\right) + r_1\right) \tag{3.28}$$

and

$$\beta^{i+1} = E_2^{-1}\left((E_2 \beta^i - r_2)_+ + r_2\right) \tag{3.29}$$

i.e.

$$\alpha^{i+1} = \left(D_2(D_2{}'D_2 + C_3 I)^{-1} D_2{}' + \frac{4p(1-\xi)^2}{C_1} + \frac{ee'}{2C_1\nu_1}\right)^{-1} [\,((D_2(D_2{}'D_2 + C_3 I)^{-1} D_2{}' + \frac{4p(1-\xi)^2}{C_1} + \frac{ee'}{2C_1\nu_1})\alpha^i - \psi_1\alpha^i - (D_2(D_2{}'D_2 + C_3 I)^{-1} D_2{}' y - y))_+ + D_2(D_2{}'D_2 + C_3 I)^{-1} D_2{}' y - y)] \tag{3.30}$$

and

$$\beta^{i+1} = \left(D_2(D_2{}'D_2 + C_4 I)^{-1} D_2{}' + \frac{4p\xi^2}{C_2} + \frac{ee'}{2C_2\nu_2}\right)^{-1} [((D_2(D_2{}'D_2 + C_4 I)^{-1} D_2{}' + \frac{4p\xi^2}{C_2} + \frac{ee'}{2C_2\nu_2})\beta^i - \psi_2\beta^i - (-D_2(D_2{}'D_2 + C_4 I)^{-1} D_2{}' y + y))_+ + (-D_2(D_2{}'D_2 + C_4 I)^{-1} D_2{}' y + y))] \tag{3.31}$$

Remark1: Implications of calculating the inverse of the matrices are readily apparent.

$$\left(D_2(D_2{}'D_2 + C_3 I)^{-1} D_2{}' + \frac{4p(1-\xi)^2}{C_1} + \frac{ee'}{2C_1\nu_1}\right)$$

and

$$\left(D_2(D_2{}'D_2 + C_4 I)^{-1} D_2{}' + \frac{4p\xi^2}{C_2} + \frac{ee'}{2C_2\nu_2}\right)$$

in the aforementioned iterative techniques (3.30) and (3.31) to obtain the LAsy-TSVR solution. These matrices can be calculated from the beginning of the process and are positive definite, in contrast to the Asy- ν-TSVR and TSVR.

Remark 2: A very tiny positive integer must be multiplied by TSVR and Asy-ν-TSVR, δ not to mention the identity matrix For the matrix to be true positive, I need to verify it. But we don't think a tiny term is necessary for our suggested paradigm.

The LAsy- ν-TSVR model consistently offers a distinct worldwide solution because

$$\left(D_2(D_2{}'D_2+C_3I)^{-1}D_2{}'+\frac{4p(1-\xi)^2}{C_1}+\frac{ee'}{2C_1\nu_1}\right)$$

and

$$\left(D_2(D_2{}'D_2+C_4I)^{-1}D_2{}'+\frac{4p\xi^2}{C_2}+\frac{ee'}{2C_2\nu_2}\right)$$ Both of these matrices are positive definite.

Remark 3: Regarding any random vectors $\alpha^0 \in \Re^p$ and $\beta^0 \in \Re^p$, the iterate $\alpha^i \in \Re^p$ and $\beta^i \in \Re^p$ the unique solution is reached via iterative methods (3.28) and (3.29) $\alpha^* \in \Re^p$ and $\beta^* \in \Re^p$ in addition to meeting the prerequisites listed below, as

$$||E_1\ a^{i+1} -\ E_1\ a^*\ || \leq ||\ I - \alpha\ E_1^{-1}||\ ||\ E_1\ a^i -\ E_1\ a^*||$$

and

$$||E_2\ \beta^{i+1} -\ E_2\ \beta^*\ || \leq ||\ I - \beta\ E_2^{-1}||\ ||\ E_2\ \beta^i -\ E_2\ \beta^*||$$

It is possible to extrapolate the aforementioned convergence proof from.

Discussion: Among the many benefits of the proposed LAsy-ν -TSVR:

- The cost function of the provided LAsy-□-TSVR takes into account the 2-norm of the vector of slack variables in order to make the problem highly convex and find the unique global solution.
- For the purpose of using SRM theory, regularization factors are provided to the optimization problem of LAsy-□-TSVR. Herein lies the method's well-posedness.
- Using linearly convergent iterative techniques reduces the calculation cost, leading to the suggested LAsy-□-TSVR result.

3.1.3 COMPUTER-BASED TRIALS

We conducted extensive numerical tests on common baseline real-world datasets for SVR, TSVR, HN-TSVR, Asy-□-TSVR, and RLTSVR to assess the feasibility of the proposed LAsy-□-TSVR. For our numerical studies, we rely on MATLAB software, version 2008b. Using MOSEK's standalone optimization tools, the four formulations (SVR, TSVR, HN-TSVR, and Asy-TSVR) manage QPPs. Exciting datasets covered in these experimental findings include Space Ga Kin900, Pollution, IBM, RedHat, Google, Intel, Microsoft Concrete CS, Boston, Auto-MPG, Parkinson, Gas furnace, and Winequality utilizing Mg17, among many more. This study examines non-linear cases that employ the Gaussian kernel function as well as linear ones.

$$K(x_i, x_j) = \exp(-\mu \left|\left|x_i - x_j\right|\right|^2, For\ i, j = 1, \dots, p$$

where kernel parameter $\mu > 0$. In this case, we've taken all of our parameters from Table 3.1.

TABLE 3. 1 Overview of Parameters in the LAsy-v-TSVR Analysis

Parameters	Parameters	Approaches
ε	{ 0.1 ,0.01}	SVR, RLTSVR
	{ 0.1, 0.3, 0.5, 0.7, 0.9 }	TSVR, HN-TSVR
$\varepsilon = \varepsilon_1^*, \varepsilon = \varepsilon_2^*$	{ 0.1, 0.3, 0.5, 0.7, 0.9 }	HN-TSVR
C	$\{10^{-5}, \dots, 10^{-5}\}$	SVR
$C_1=C_2, C_3=C_4$	$\{10^{-5}, 10^{-3}, 10^{-1}, 10^{1}, 10^{3}, 10^{5}\}$	TSVR, HN-TSVR, Asy-v-TSVR, RLTSVR, LAsy-v-TSVR
$\nu_1 = \nu_2$	{ 0.1, 0.3, 0.5, 0.7, 0.9 }	Asy-v-TSVR
	{ 0.01}	LAsy-v-TSVR
ξ	{ 0.2, 0.4, 0.45, 0.5, 0.55, 0.6, 0.8 }	Asy- v-TSVR, LAsy- v-TSVR
μ	$\{2^{-5}, \dots, 2^{5}\}$	SVR, TSVR, HN-TSVR, Asy-v-TSVR, RLTSVR, LAsy-v-TSVR

Artificial datasets

Eight intentionally created datasets, whose function descriptions are presented in Table 3.2, were subjected to numerical testing in this subsection. To verify the viability of the suggested LAsy-**v**-TSVR in order to account for noise and outliers, we included two forms of noise pollution—symmetric noise and asymmetric noise structure—into synthetic datasets. Functions Synthetic datasets with noise variability caused by symmetric distribution are generated using symmetric noise in equations 1–8.

TABLE 3. 2 Synthetic Dataset Generation Methods for LAsy-v-TSVR

Function Name	Function Definition	Domain of definition	Noise Type
Function 1	$f(x_1, x_2, x_3, x_4, x_5) = 0.79 + 1.27\, x_1 x_2 + 1.56\, x_1 x_4 + 3.42\, x_2 x_5 + 2.06\, x_3 x_4 x_5 + \Omega$	$x_i \in [0,1], i \in \{1, 2, 3, 4, 5\}$	Type A: $\Omega \in U$ $(-0.2, 0.2)$
Function 2			Type B: $\Omega \in N(0, 0.2^2)$
Function 3	$f_1(x) = \frac{\sin(x)}{(x)}$ such that	$x \in U(-4\pi, 4\pi)$ $i = 1,2, \ldots, 200$	Type A: $\Omega \in U$ $(-1,1)$
Function 4	$y_1 = f_1(x_i) + (0.5 - \frac{\lvert x_i \rvert}{8_\pi})\Omega i$		Type B: $\Omega \in N$ $(0, 0.5^2)$
Function 5	$f(x) = \frac{\lvert x-1 \rvert}{4} + \lvert sin(\pi(1 + \frac{x-1}{4})) \rvert + 1 + \Omega$	$x \in U(-10, 10)$	Type A: $\Omega \in U$ $(-0.2, 0.2)$
Function 6			Type B: $\Omega \in N(0, 0.2^2)$
Function 7	$f(x_1, x_2) = 1.9[1.35 + e^{x1} sin(13(x_1 - 0.6)^2) + e^{3(x_2 - 0.5))} \sin(4\pi(x_2 - 0.9)^2)] + \Omega$	$x_1, x_2 \in U(0,1)$	Type A: $\Omega \in U$ $(-0.2, 0.2)$
Function 8			Type B: $\Omega \in N(0, 0.2^2)$

In order to generate the asymmetrical synthetic dataset, functions 3–4 use the heteroscedastic noise pattern, where the noise is significantly dependent on the value of training instances. They also use a uniform probability distribution. $\Omega \in U$ (a,b) Assuming a homogeneous noise interval and a normal distribution (a,b). $\Omega \in N(\mu, \sigma)^2$ given a normal distribution, where and denote the average μ and variance σ^2, respectively. In this case, we combine 500 testing data points devoid of noise with 200 training data points that include additive noise to produce a synthetic dataset. Table 3.4 shows the average ranks of all reported models for simulated datasets using Gaussian

kernels, and Table 3.3 shows that our suggested LAsy-ν-TSVR offers comparable or higher generalization capability compared to earlier approaches, according to RMSE values. The proposed LAsy-ν-TSVR proves to be the most important and reliable among SVR, TSVR, and Asy-ν-TSVR in both linear and nonlinear scenarios, proving its application and dependability. In order to assess the efficacy of the suggested LAsy-ν-TSVR on datasets exhibiting heteroscedastic noise, the prediction graphs for Function 3 using uniform noise are shown in Figure 3.1. We also include Gaussian noise in the Function 4 prediction graphs in Figure 3.2. Using Function 5's Gaussian kernel, Figure 3.3 displays the prediction outcomes for all models and LAsy-ν-TSVR. We use uniform noise to test how well symmetrical noise patterns work. When the noise is Gaussian, Figure 3.4 also shows the prediction graphs for Function 6.

TABLE 3. 3 Average RMSE Rankings of LAsy-v-TSVR and Reported Approaches on Synthetic Data with Linear Kernel.

Dataset	SVR	TSVR	HN-TSVR	Asy-v-TSVR	RLTSVR	LAsy-v-TSVR
Function1	5	4	3	6	1.5	1.5
Function2	6	3	4	5	1.5	1.5
Function3	5	1	6	2	3.5	3.5
Function4	6	1	2	5	3	4
Function5	6	4	3	5	2	1
Function6	6	4.5	4.5	1	2	3
Function7	6	4	5	1	3	2
Function8	3	6	4	5	2	1
Average rank	**5.375**	**3.4375**	**3.9375**	**3.75**	**2.3125**	**2.1875**

TABLE 3. 4 RMSE Rankings: LAsy-v-TSVR vs. Reported Models for Gaussian Kernel on Artificial Data.

Dataset	SVR	TSVR	HN-TSVR	Asy-ν-TSVR	RLTSVR	LAsy-ν-TSVR
Function 1	4	3	5	6	2	1
Function 2	6	5	4	3	1	2
Function 3	6	1	4	5	2.5	2.5
Function 4	6	3	2	4	5	1
Function 5	6	2	3	4	5	1
Function 6	6	2	1	4	3	5
Function 7	6	5	4	3	2	1
Function 8	2	5	4	6	3	1
Average rank	**5.25**	**3.25**	**3.375**	**4.375**	**2.9375**	**1.8125**

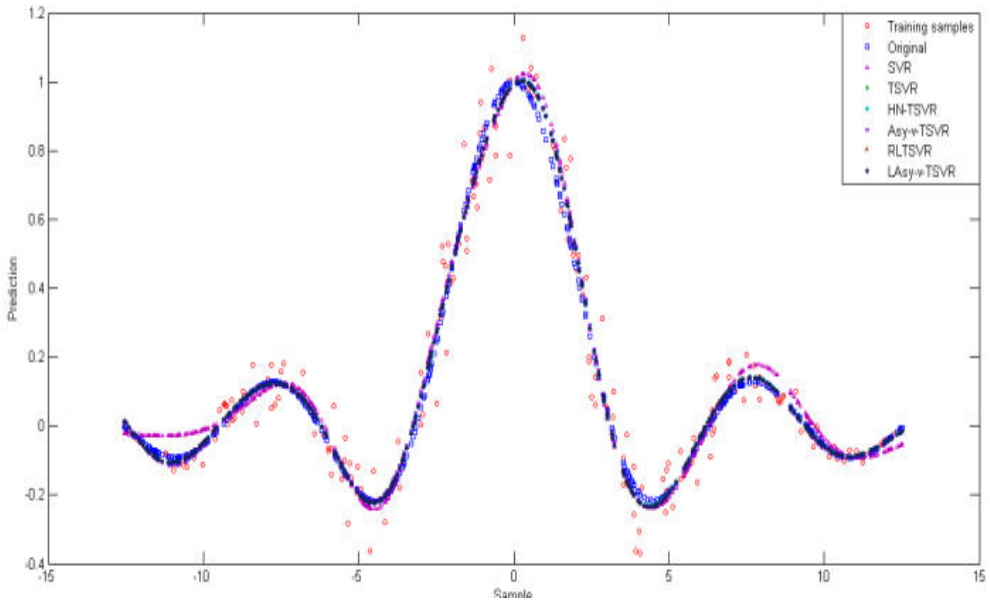


FIGURE 3. 1 Test Set Accuracy Plot for Function 3 with Gaussian Kernel and Uniform Noise.

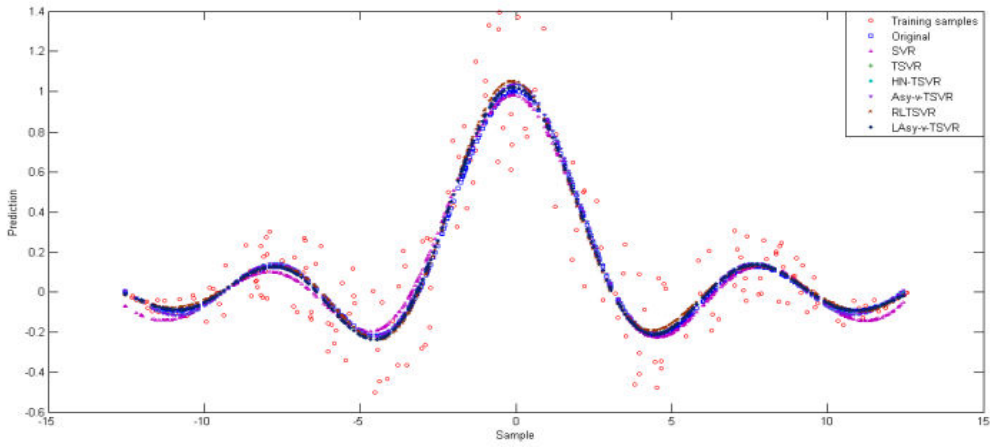


FIGURE 3. 2 Test Set Accuracy Curve Combining Gaussian Noise with a Gaussian Kernel to Evaluate Function 4

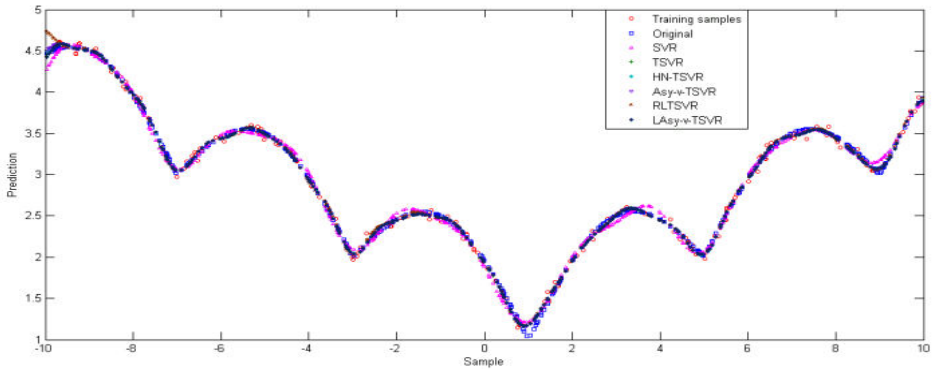


FIGURE 3. 3 Test Set Accuracy Curve Utilizing a Gaussian Kernel for Uniform Noise in Function 5.

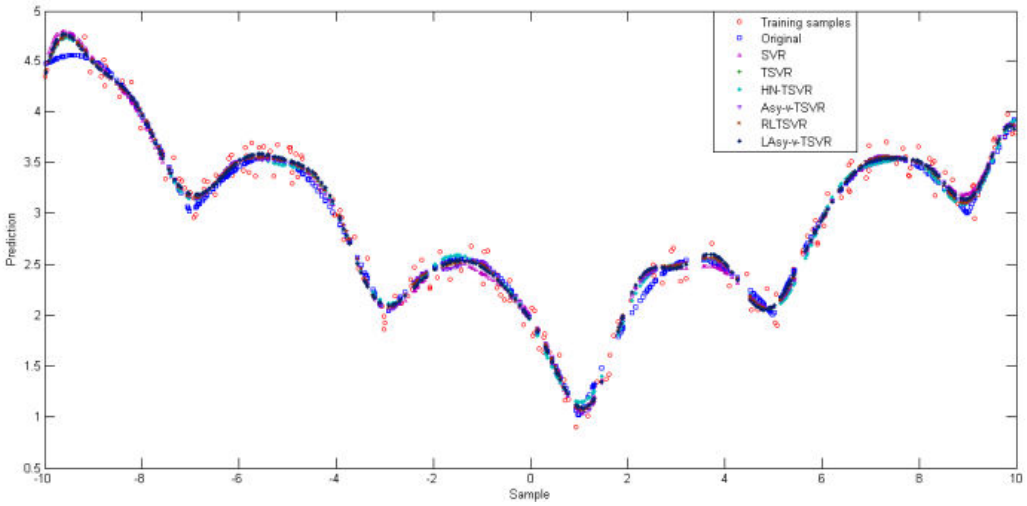


FIGURE 3. 4 Test Set Accuracy Curve Functional 6 with Gaussian Noise and a Gaussian Kernel

When dealing with symmetric noise patterns that include LAsy-□- TSVR clearly beats the stated models when it comes to agreeing with the final projected values, regardless of whether the noise is uniform or Gaussian. Regardless of whether the noise is uniform or Gaussian, the findings demonstrate that LAsy-□ -TSVR performs better when dealing with asymmetric noise topologies.

Real-world Datasets

Using a linear kernel improved the prediction accuracy of LAsy-TSVR in 8 out of 18 real-world datasets, while a Gaussian kernel improved it in 11 of the datasets, demonstrating the model's usefulness and application to noisy datasets.

For the sake of visual representation, Figure 3.5 displays the expected results for Auto-MPG, Gas furnace in Figure 3.7, and Intel in Figure 3.9. Figures 3.6, 3.8, and 3.10 show the similarity and Intel, Gas Furnace, and Auto-MPG's Prediction Error respectively. Based on these findings, it can be inferred that our suggested LAsy-ν-TSVR technique is both practical and useful, since its prediction values are near to goal values when compared to SVR, TSVR, HN-TSVR, Asy-ν -TSVR, and RLTSVR. In order to provide statistical evidence for the effectiveness of our suggested LAsy-ν -TSVR, we have included the average rankings for all the approaches that were compared using linear and nonlinear kernels in Tables 3.5 and 3.6, respectively, based on RMSE values. Both Table 3.5 and Table 3.6 make it quite evident that the suggested LAsy-ν -TSVR ranks worst out of all the options.

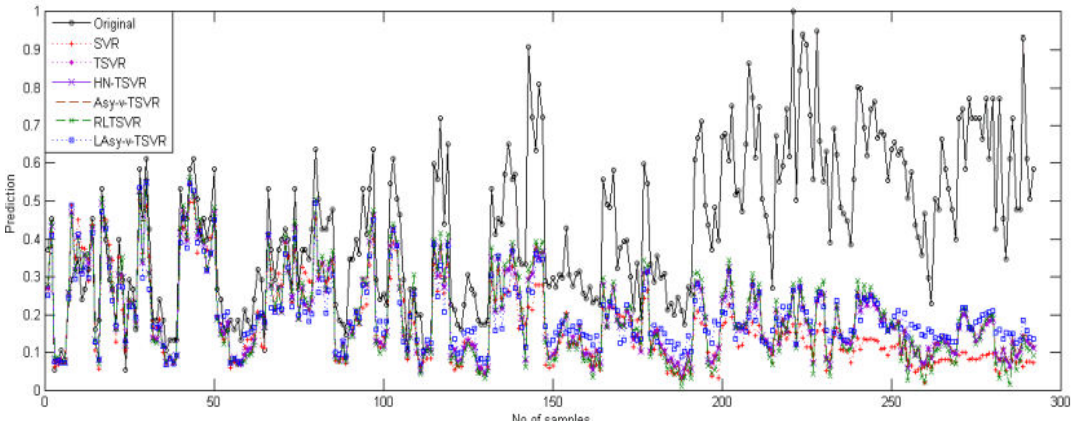


FIGURE 3. 5 Prediction on the Testing Dataset of Auto-MPG Using Gaussian Kernel

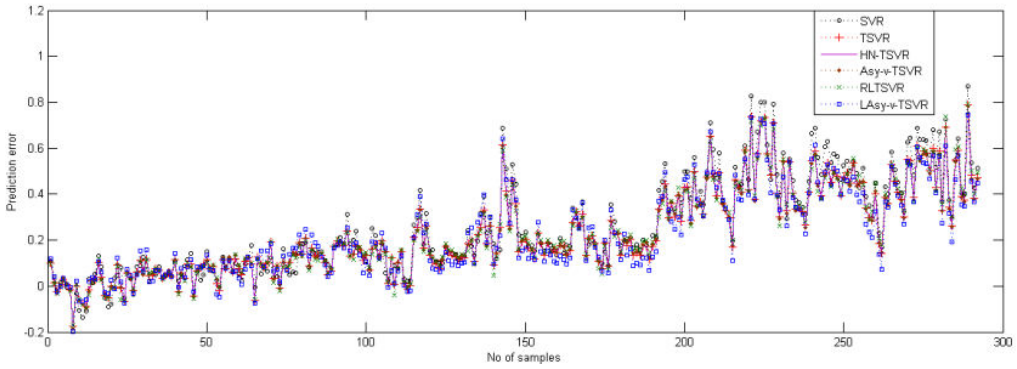


FIGURE 3. 6 Prediction Error on the Testing Dataset of Auto-MPG Using Gaussian Kernel

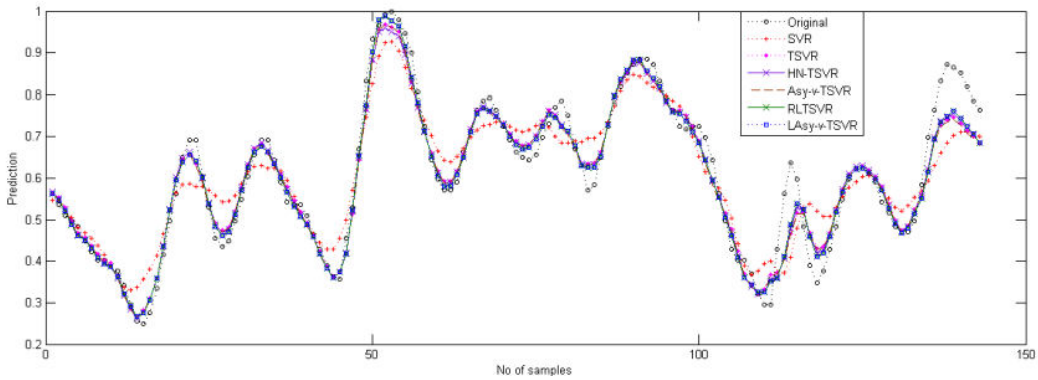


FIGURE 3. 7 Prediction on the Testing Dataset of Gas Furnace Using Gaussian Kernel

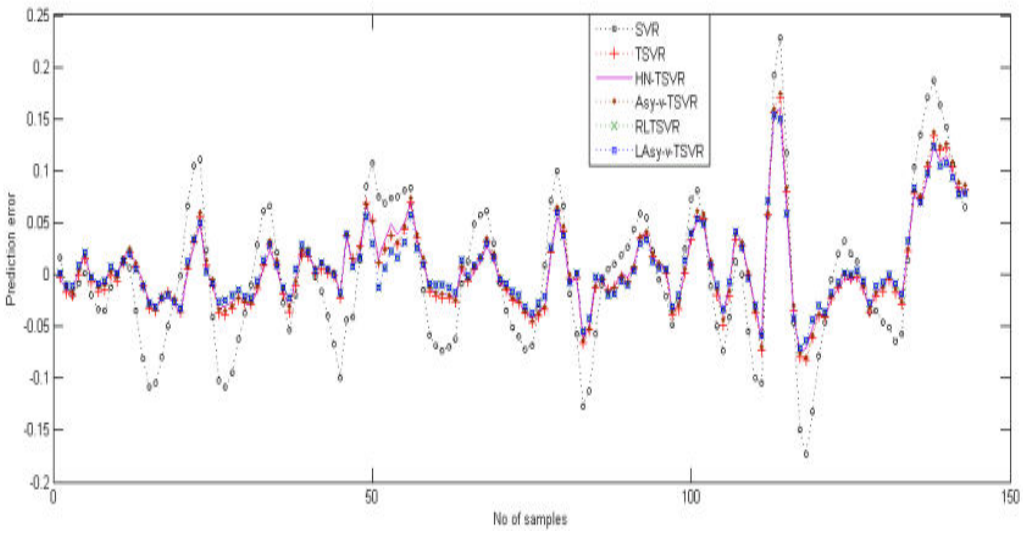


FIGURE 3. 8 Prediction Error on the Gas Furnace Testing Dataset Using Gaussian Kernel

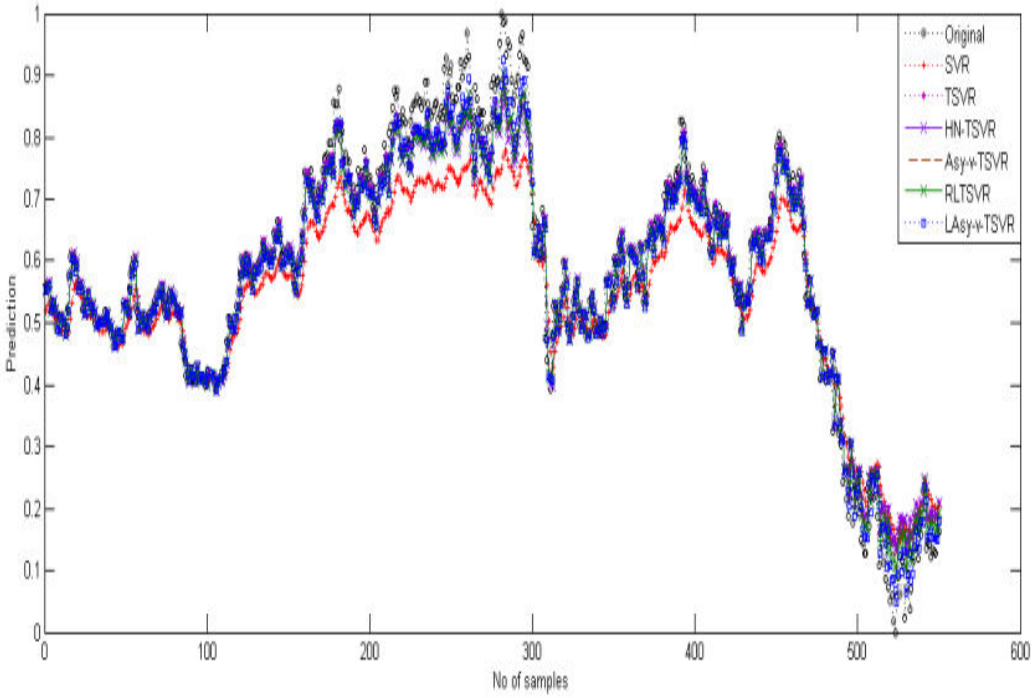


FIGURE 3. 9 Using the Gaussian Kernel, Intel's Testing Dataset for Prediction

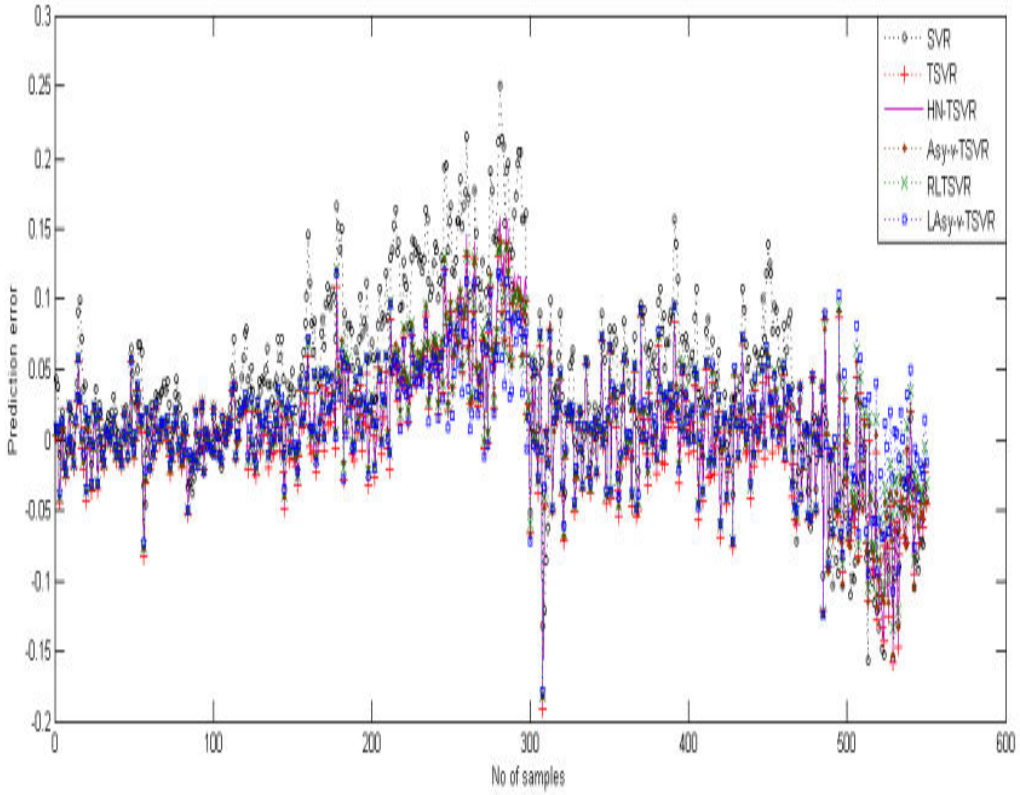


FIGURE 3. 10 Prediction Error using the Gaussian Kernel on the Intel Testing Dataset

TABLE 3. 5 RMSE Rankings of LAsy-v-TSVR and Reported Models on Real-World Datasets with Linear Kernel

Dataset	SVR	TSVR	HN-TSVR	Asy-v-TSVR	RLTSVR	LAsy-v-TSVR
ConcreteCS	6	2	4	3	5	1
Boston	1	4	6	5	3	2
Auto-MPG	1	6	4	5	2	3
Parkinsons	6	3	5	4	2	1
Winequality	3	1	4	6	2	5
Kin900	5	2	4	6	3	1
Demo	6	5	3	4	1	2
Mg17	6	3	3	5	1	3
Google	6	1	5	2	3	4
IBM	6	4	5	3	1.5	1.5
Intel	6	5	2	1	4	3
Microsoft	6	1	3	2	4.5	4.5
RedHat	6	3	4	5	1.5	1.5
Pollution	3	6	5	4	2	1
Gas Furnace	6	5	4	3	2	1
Flexible robot arm	6	5	3	4	2	1
S&P500	6	2	3	1	4	5
Space Ga	6	1	2	3	4	5
Average rank	**5.05556**	**3.27778**	**3.83333**	**3.66667**	**2.63889**	**2.52778**

TABLE 3. 6 RMSE Rankings of LAsy-v-TSVR and Reported Models on Real-World Datasets with Gaussian Kernel

Dataset	SVR	TSVR	HN-TSVR	Asy-v-TSVR	RLTSVR	Lasy-v-TSVR
ConcreteCS	6	4	3	2	5	1
Boston	1	6	4	5	3	2
Auto-MPG	6	5	4	2	3	1
Parkinsons	6	4	5	3	1	2
Winequality	1	3	6	5	4	2
Kin900	6	5	3.5	3.5	2	1
Demo	3	5	6	4	2	1
Mg17	6	5	4	3	2	1
Google	6	4	3	5	2	1
IBM	6	4	3	5	2	1
Intel	6	3	5	4	2	1
Microsoft	6	5	3	4	1.5	1.5
RedHat	4	5	6	3	1	2
Pollution	1	2	5	3	6	4
Gas Furnace	6	4	3	5	1.5	1.5
Flexible robot arm	6	5	4	3	1	2
S&P500	6	3	4	5	2	1

Space Ga	6	3	4	5	1	2
Average rank	4.88889	4.16667	4.19444	3.86111	2.33333	1.55556

Statistical Friedman test

Currently, in order to identify variations in the ranking of RMSE among various algorithms, a non-parametric Friedman test is performed with the matching post hoc test on six algorithms and eighteen datasets. The primary use of this test is in rank-based one-way repeated-measures analyses of variance. Under the null hypothesis, all of these approaches are equally valid.

a) Linear Case

The following is how the Friedman statistic is calculated for the linear example using Table 3.5:

$$\chi_F^2 = \frac{12\times 18}{6\times 7}\left[\left(5.055556^2 + 3.27778^2 + 3.83333^2 + 3.66667^2 + 2.63889^2 + 2.52778^2\right) - \left(\frac{6\times 7^2}{4}\right)\right]$$

$\chi^2_F \approx 22.0873$ and

$$F_F = \frac{17\times 22.0873}{18\times 5 - 22.0873} = 5.5289$$

There is a degree of flexibility in the distribution of FF, as stated by Friedman(6 -1, (6 -1) *(18 -1)) = (5, 85). For F(5,85), the critical value is 2.321. $\alpha = 0.05$. Since $F_F >$ 2.321, All algorithms are not comparable, hence we reject the null hypothesis. After that, we compare one procedure to the other using the Nemenyi post hoc test. Following a rejection of the null hypothesis using the Friedman test, this test is used to compare pairwise performances. Thus, we determine the significant difference (CD) by using

$$q_a = 2.589 \; as \; CD = 2.589\sqrt{\frac{6\times 7}{6\times 18}} = 1.6145$$

For $\theta = 0.10$ given that the worth of q_α the number of algorithms that were reported and the value of θ from the difference of the average ranks of SVR and LAsy-ν -TSVR (5.055556 - 2.527778 = 2.527778), which is higher than CD (1.6145), are used to

determine this. The outcome guarantees that LAsy-ν -TSVR outperforms SVR in terms of prediction performance.

b) Non-linear Case

Secondly, SVR, TSVR, HN-TSVR, Asy-ν-TSVR, RLTSVR, and LAsy-ν-TSVR average ranks in a nonlinear situation using real-world datasets.

$$\chi_F^2 = \frac{12\times 18}{6\times 7}\left[\left(4.88889^2 + 4.16667^2 + 4.19444^2 + 3.86111^2 + 2.33333^2 + 1.55556^2\right) - \left(\frac{6\times 7^2}{4}\right)\right],$$

$$\chi_F^2 \approx 41.8016 \text{ and } F_F = \frac{17\times 41.8016}{18\times 5 - 41.8016} = 14.7438$$

For F(5,85), the critical value is 2.321 α = 0.05 . Since, F_F > 2.321, under these circumstances, the null hypothesis is rejected. Do a pairwise comparison of the approaches using the Nemenyi test. The difference that matters most here is 1.6145.

1. Since the average rank difference between SVR and LAsy-ν-TSVR is more than 1.6145, (4.888889 -1.555556 = 3.333333) therefore LAsy-ν-TSVR is better than SVR.

2. The prediction performance of LAsy-ν-TSVR is much better than that of TSVR, as shown by the higher difference between the average rankings (4.166667 -1.555556 = 2.611111) compared to (1.6145).

3. In comparing HN-TSVR and LAsy-νν-TSVR, the average rank difference is 2.638889, which is higher than 1.6145, suggesting that LAsy-ν- TSVR is superior to HN-TSVR.

4. The fact that the difference in average rank between Asy-ν-TSVR and LAsy-ν-TSVR is more than 1.6145 indicates that LAsy-ν-TSVR is more prevalent and useful than Asy-ν-TSVR.

In this section, we propose a more effective method known as LAsy-ν-TSVR, which stands for enhanced regularization-based Lagrangian asymmetric ν-twin support vector regression. This method uses a pinball loss function and effectively incorporates the

core principle of statistical learning, namely the SRM notion. Unlike SVR, TSVR, HN-TSVR, and Asy-ν-TSVR, which use quadratic programming problems (QPPs), LAsy-ν-TSVR uses the linearly convergent iterative approach to achieve its solution. Thus, in our scenario, an additional optimization toolset is not necessary. Proposed LAsy-ν-TSVR outperforms previously described approaches in terms of efficiency and applicability, and it can handle symmetric and asymmetric patterns with two types of uniform and Gaussian noise for statistical support. Through experiments on several real-world datasets, it has been determined that the proposed LAsy-ν-TSVR outperforms SVR, TSVR, HNTSVR, Asy-ν-TSVR, and RLTSVR with respect to learning speed and prediction accuracy, demonstrating its practicality and adaptability. One of the best ways to approximate regression problems is using unconstrained convex minimization; in the future, we may think about using more efficient convergent iterative methods like the Newton iterative method.

3.2 ROBUST ASYMMETRIC-ν-TWIN SVR UTILIZING PINBALL LOSS FUNCTION

The effectiveness of LAsy-ν-TSVR, an enhanced regularization-based Lagrangian asymmetric ν-twin support vector regression with pinball loss function was shown by the experimental findings in the preceding section, where the topic of regression problem solution was covered. Through transforming the 1-norm of the slack variables' vector ζ_1 and ζ_2, in our proposed regularized LAsy- ν-TSVR formulation, the issue is created strongly convex and ensures the availability of a globally unique solution by using the square of the vector of slack variables in 2-norm. The dual space is where LAsy-ν-TSVR finds its answers using an easy iterative convergent technique. Before applying the newton iterative approach and expanding the study to a more efficient rate of convergence of TSVR, we present an alternate way to obtain the solutions using three implementations, including a generalized derivatives approach and two smoothing approaches-based methods. An example of this is controlling the fitting error within an asymmetric tube using the pinball loss function. Leading to a steady and well-posed dual issue is another advantage. Of the three methods, the smooth approximation function outperforms the others on both real-world and simulated datasets, accommodating symmetric and asymmetric patterns with two types of uniform and Gaussian noise, so supporting statistically.

3.2.1 ROBUST ASYMMETRIC LAGRANGIAN V-TWIN SUPPORT VECTOR REGRESSION WITH PINBALL LOSS AND UNCONSTRAINED MINIMIZATION (URALTSVR)

An effective idea as resilient asymmetric For major and essential real-world applications, we provide Lagrangian-twin support vector regression with pinball loss as an unconstrained minimization problem to handle asymmetric noise and re-sampling instability, thus improving prediction performance. The suggested URALTSVR formulation is described by taking into account the square of the vector of slack variables ζ_1,ζ_2 by adopting the two-norm instead of the one-norm, we can make the model more strong and convex, which in turn proves that there is a globally unique solution and allows us to eliminate the non-negativity criteria for the slack variables as they are met at optimum. Furthermore, regularization words are included $\frac{C_3}{2}(||\ W_1\ ||^2 + b_1^2)$ and $\frac{C_4}{2}(||\ W_2\ ||^2 + b_2^2)$ reduce the overfitting problem and improve the stability in the dual formulations, respectively, by making the problem a positive definite and well-posed model in the objective functions of (3.32 and 3.33). When developing non-linear URALTSVR, the functions that are produced by the kernel are used $f_1(x) = K(x^t, B^t)W_1 + b_1 and\ f_2(x) = K(x^t, B^t)W_2 + b_2$ are decided by the subsequent QPPs as.

$$\min \frac{C_3}{2}(\| w_1 \|^2 + b_1^2) + \frac{1}{2}\| y - (K(B,B^t)w_1 + eb_1)\|^2 + \frac{1}{p}C_1\zeta_1^t\zeta_1 + C_1\nu_1\varepsilon_1$$

subject to. $Y - (K(B, B^t)W_1 + eb_1) \geq -e\varepsilon_1 - 2(1-\xi)\xi_1$ (3.32)

and

$$\min \frac{C_4}{2}(\| w_2 \|^2 + b_2^2) + \frac{1}{2}\| y - (K(B,B^t)w_2 + eb_2)\|^2 + \frac{1}{p}C_2\zeta_2^t\zeta_2 + C_2\nu_2\varepsilon_2$$

subject to. $Y - (K(B, B^t)W_2 + eb_2) \geq -e\varepsilon_2 - 2(1-\xi)\xi_2$ (3.33)

respectively, were C_1, C_2, C_3, C_4 >0: $\varepsilon_1, \varepsilon_2 \geq 0\ and\ \nu_1, \nu_2$ act as parameters for input. ξ functions as a pinball loss. ζ_1, ζ_2 comprise the factors that allow for some leeway.

The dual QPPs of primary issues (3.32) and (3.33) are provided as a result of applying the KKT criteria.

$$\min \frac{1}{2}\alpha^t\left(D_2(D_2{}^tD_2+C_3I)^{-1}D_2{}^t+\frac{4p(1-\xi)^2}{C_1}+\frac{ee^t}{2C_1\nu_1}\right)\alpha-(D_2(D_2{}^tD_2+C_3I)^{-1}D_2{}^ty-y)^t\alpha \tag{3.34}$$

And

$$\min \frac{1}{2}\beta^t\left(D_2(D_2{}^tD_2+C_4I)^{-1}D_2{}^t+\frac{4p\xi^2}{C_2}+\frac{ee^t}{2C_2\nu_2}\right)\beta-(-D_2(D_2{}^tD_2+C_4I)^{-1}D_2{}^ty+y)^t\beta \tag{3.35}$$

where D2 = [K (B, B^t e] is an augmented matrix.

After determining the worth of α and β using equations (3.34), (3.35), and (3.35), the non-linear kernel's final estimate function f (.) is calculated by averaging the following non-linear functions f_1x and f_2x as

$$f_1(x)=[K(x^t,\mathrm{B}^t)\quad 1]\begin{bmatrix}w_1\\b_1\end{bmatrix}=[K(x^t,\mathrm{B}^t)\quad 1]((D_2{}^tD_2+C_3I)^{-1}D_2{}^t(y-\alpha)) \tag{3.36}$$

and

$$f_2(x)=[K(x^t,B^t)\quad 1]\begin{bmatrix}w_2\\b_2\end{bmatrix}=[K(x^t,B^t)\quad 1]((D_2{}^tD_2+C_4I)^{-1}D_2{}^t(y+\beta)) \tag{3.37}$$

Here is another way to rephrase problems (3.34) and (3.35):

$$\min_{0\le\alpha\in\Re^p} L_1(\alpha)=\frac{1}{2}\alpha^tE_1\alpha-r_1^t\alpha \tag{3.38}$$

And

$$\min_{0\le\beta\in\Re^p} L_2(\beta)=\frac{1}{2}\beta^tE_2\beta-r_2^t\beta \tag{3.39}$$

respectively, where

$$E_1=\left(D_2(D_2{}^tD_2+C_3I)^{-1}D_2{}^t+\frac{4p(1-\xi)^2}{C_1}+\frac{ee^t}{2C_1\nu_1}\right),r_1=D_2(D_2{}^tD_2+C_3I)^{-1}D_2{}^ty-y$$

$$E_2 = \left(D_2(D_2{}' D_2 + C_4 I)^{-1} D_2{}' + \frac{4p\xi^2}{C_2} + \frac{ee'}{2C_2\nu_2} \right),$$

And $r_2 = -D_2(D_2^t D_2 + C_4 I)^{-1} D_2^t y + y$

The following two typical complementary issues arise as a result of applying the K.K.T. optimality requirements to QPPs (3.38) and (3.39):

$$0 \leq (E_1 a - r_1) \perp a \geq) \tag{3.40}$$

And

$$0 \leq (E_2 \beta - r_2) \perp \beta \geq 0, \tag{3.41}$$

that is, in turn. Through the use of the $0 \leq x \perp y \geq 0$ if and only if $x = (x - \psi y)_+$ regardless of the parameters, x, and y vectors $\psi > 0$, The following fixed point theorems rephrase the related issues from (3.40) and (3.41): for any $\psi_1, \psi_2 > 0$, the relations

$$(E_1 a - r_1) = (E_1 a - \psi_1 a - r_1)_+ \tag{3.42}$$

And

$$(E_2 \beta - r_2) = (E_2 \beta - \psi_2 \beta - r_2 \tag{3.43}$$

By satisfying the requirements (3.42) and (3.43), the dual issues of restricted minimization (3.38) and (3.39) may be recast as a set of unconstrained minimization problems:

$$\min_{\alpha \in \Re^p} L_1(\alpha) = \frac{1}{2} \alpha^t E_1 \alpha - r_1^t \alpha + \frac{1}{2\psi_1} (\| (E_1 \alpha - \psi_1 \alpha - r_1)_+ \|^2 - \| (E_1 \alpha - r_1) \|^2) \tag{3.44}$$

And

$$\min_{\beta \in \Re^p} L_2(\alpha) = \frac{1}{2} \beta^t E_2 \beta - r_2^t \beta + \frac{1}{2\psi_2} (\| (E_2 \beta - \psi_2 \beta - r_2)_+ \|^2 - \| (E_2 \beta - r_2) \|^2) \tag{3.45}$$

Using the Newton iterative method, one can discover the unknowns in the unconstrained minimization problems with strongly convex, continuous, and piecewise quadratic functions, as seen above α and β . Here, we use the most up-to-date i^{th} iterate

α^i and β^i , the value of α^{i+1} and β^{i+1} are determined at the $(i+1)^{th}$ repeat using the following method:

$$\nabla L_1(a^i) + \nabla^2 L_1(a^i)(a^{i+1} - a) = 0 \; where \; i = 0,1,2 \ldots \tag{3.46}$$

And

$$\nabla L_2(\beta^i) + \nabla^2 L_2(\beta^i)(\beta^{i+1} - \beta) = 0 \; where \; i = 0,1,2 \ldots \tag{3.47}$$

in that order. The minimization problems' gradients, therefore, are (3.44) and (3.45), which are given by $\nabla L_1(\alpha)$ and $\nabla L_2(\beta)$ as

$$\nabla L_1(\alpha) = \left(\frac{\psi_1 I - E_1}{\psi_1}\right)[E_1\alpha - r_1) - (E_1\alpha - \psi_1\alpha - r_1)_+], \tag{3.48}$$

And

$$\nabla L_2(\beta) = \left(\frac{\psi_2 I - E_2}{\psi_2}\right)[E_2\beta - r_2) - (E_2\beta - \psi_2\beta - r_2)_+], \tag{3.39}$$

And in order to get the corresponding Hessian matrix for $L_1(\alpha)$ and $L_2(\beta)$,as a result, the gradient of $L_1(\alpha)$ and $L_2(\beta)$may be differentiable twice in the conventional sense, but their 'plus' function is continuous. We have proposed methods for determining the Hessian matrix, such as the generalized derivative technique or a smooth approximation function in lieu of the 'plus' function.

Remark 1: Reason being that SVM with Hinge loss is quite vulnerable to noise in the features as well as the labels.

The robustness property is examined and studied by taking the pinball loss into account and using the quantile value, which is more resistant to re-sampling and noise than the hinge loss.

Generalized derivative approach for URALTSVR

Using the generalized derivative method, the Hessian matrix is calculated, and the generalized Hessian of the problems (3.44) and (3.45) is provided as

$$\nabla^2 L_1(\alpha) = \left(\frac{\psi_1 I - E_1}{\psi_1}\right)[E_1 + (\psi_1 I - E_1) diag\,((E_1\alpha - \psi_1\alpha - r_1)_*)], \tag{3.50}$$

And

$$\nabla^2 L_2(\beta) = \left(\frac{\psi_2 I - E_2}{\psi_2}\right)[E_2 + (\psi_2 I - E_2) diag\,((E_2\beta - \psi_2\beta - r_2)_*)]. \tag{3.51}$$

To choose the parameter, we use the fact that E_1 and E_2 are positive definite matrices $\psi_k > \|E_k\|\ for$ k= 1,2, then $V^2 L_1(\alpha)$ and $V^2 L_2(\alpha)$ turns becomes a positive definite. So, we can find out how much the unknowns are worth α and in a manner that makes use of the following basic iterative techniques from equations (3.46 and 3.47).

$$[E_1 + (\psi_1 I - E_1) diag\,((E_1\alpha^i - \psi_1\alpha^i - r_1)_*)](\alpha^{i+1} - \alpha^i) = -[(E_1\alpha^i - r_1) - (E_1\alpha^i - \psi_1\alpha^i - r_1)_+], \tag{3.52}$$

And

$$[E_2 + (\psi_2 I - E_2) diag\,((E_2\beta^i - \psi_2\beta^i - r_2)_*)](\beta^{i+1} - \beta^i) = -[(E_2\beta^i - r_2) - (E_2\beta^i - \psi_2\beta^i - r_2)_+]. \tag{3.53}$$

Moreover, by averaging the values of α and β in equations (3.36) and (3.37), we may get the final regressor f (xs) for every test sample x, f_1 (X_s) and f_2 (X_s). The GRALTSVR method is based on the generalized derivative method, and it is called after that. Here is GRALTSVR's time complexity $2(m^3) + 2^*O(i^*m^3)$m in where i is the iteration counter.

Smooth Approach 1 for URALTSVR (SRALTSVR1)

To facilitate the computation of the problem's

Hessian matrix, Lee and Mangasarian offered a common smoothing technique that may be used to transform the non-smooth function into a smooth function. It is clear that issues (3.44) and (3.45) are serving 'plus' missions (x_+) in their goal functions, hence a smooth approximation function should be used to replace these 'plus' functions $\gamma_1(x, \tau)$ it is expressed as:

$$\gamma_1(x,\tau) = x + \frac{1}{\tau}\log(1 + \exp(-\tau x)), \tag{3.54}$$

where $x \in \Re$ and smooth parameter $\tau > 0$.

Indeed, for any t up to p, x is equal to $(x1, x2,...,x) \in \Re^P$, The premise is that $\gamma_1(x,\tau) = (\gamma_1(x_{1,}\tau),\ldots\ldots,\gamma_1(x_p,\tau))^t$. Consequently, the two unconstrained minimization problems (3.44) and (3.45) will be transformed into

$$\min_{\alpha\in\Re^p} L_1(\alpha) = \frac{1}{2}\alpha^t E_1\alpha - r_1^t\alpha + \frac{1}{2\psi_1}(\| \gamma_1((E_1\alpha - \psi_1\alpha - r_1),\tau)\|^2 - \| (E_1\alpha - r_1)\|^2) \tag{3.55}$$

$$\min_{\beta\in\Re^p} L_2(\beta) = \frac{1}{2}\beta^t E_2\beta - r_2^t\beta + \frac{1}{2\psi_2}(\| \gamma_1((E_2\beta - \psi_2\beta - r_2),\tau)\|^2 - \| (E_2\beta - r_2)\|^2) \tag{3.56}$$

that is, in turn. Already provided by (3.48) and (3.49), respectively, are the gradient vectors of (3.44) and (3.45). Their resulting Hessian matrices are then calculated as:

$$\nabla^2 L_1(\alpha) = \left(\frac{\psi_1 I - E_1}{\psi_1}\right)\left[E_1 + (\psi_1 I - E_1)diag\left(\frac{1}{1+\exp(-\tau(E_1\alpha - \psi_1\alpha - r_1))}\right)\right], \tag{3.57}$$

And

$$\nabla^2 L_2(\beta) = \left(\frac{\psi_2 I - E_2}{\psi_2}\right)\left[E_2 + (\psi_2 I - E_2)diag\left(\frac{1}{1+\exp(-\tau(E_2\beta - \psi_2\beta - r_2))}\right)\right], \tag{3.58}$$

and so on. To discover the Lagrangian multipliers, one may use a technique similar to GRALTSVR to solve the following simple iterative schemes α and β as:

$$\left[E_1 + (\psi_1 I - E_1)diag\left(\frac{1}{1+\exp(-\tau(E_1\alpha^i - \psi_1\alpha^i - r_1))}\right)\right](\alpha^{i+1} - \alpha^i) = -[(E_1\alpha^i - r_1) - (E_1\alpha^i - \psi_1\alpha^i - r_1)_+] \tag{3.59}$$

And

$$\left[E_2 + (\psi_2 I - E_2)diag\left(\frac{1}{1+\exp(-\tau(E_2\beta^i - \psi_2\beta^i - r_2))}\right)\right](\beta^{i+1} - \beta^i) = -[(E_2\beta^i - r_2) - (E_2\beta^i - \psi_2\beta^i - r_2)_+] \tag{3.60}$$

In addition, using the means of $f_1(Xs)$ and $f_2(X_2)$ in equations (3.36) and (3.37), we may get the final regressor f(Xs) for every test sample s x. The SRALTSVR1 method uses a smooth approximation function. The vectors α^{i+1} and β^{i+1}, SRALTSVR1 may be

calculated for every integer i from 0 to n using $2(m^3)+2^*O(i^*m^3)$ complexity such that iterations i are fewer than or equal to i plus one.

Smooth Approach 2 for URALTSVR (SRALTSVR2)

To make the function easily twice differentiable, we use another smoothing method to replace the non-smooth plus function with a smooth one. This method is defined as

$$\gamma_2(\tau,\tau_0)=\frac{1}{4}\frac{\tau^2}{|\tau_0|}+\frac{1}{2}\tau+\frac{1}{4}|\tau_0|, \tag{3.61}$$

where $\gamma_2(\tau,\tau_0)$ serves as the function that estimates $\tau+$;τ_0 + has a real value that does not equal zero.

Obviously, when the worth of $|\tau_0|$ is nearer to $|\tau|$, then $\gamma_2(\tau,\tau_0)$ becomes closer to $\tau+$. In fact,$\gamma_2(\tau,\tau_0)=\tau+$ whenever $|\tau_0|=|\tau|\neq 0$.

Then, the two unconstrained minimization problems (3.44) and (3.45) will be transformed into for

$$\min_{\alpha\in\Re^p} L_1(\alpha)=\frac{1}{2}\alpha^t E_1\alpha-r_1^t\alpha+\frac{1}{2\psi_1}(\|\gamma_2((E_1\alpha-\psi_1\alpha-r_1),\alpha_0)\|^2-\|(E_1\alpha-r_1)\|^2) \tag{3.62}$$

$$\min_{\beta\in\Re^p} L_2(\beta)=\frac{1}{2}\beta^t E_2\beta-r_2^t\beta+\frac{1}{2\psi_2}(\|\gamma_2((E_2\beta-\psi_2\beta-r_2),\beta_0)\|^2-\|(E_2\beta-r_2)\|^2) \tag{3.63}$$

in such cases, where the element of $a_{0,}\beta_0, \in\Re^p$ more than zero. Equations (3.48) and (3.49) determine the gradient vectors of equations (3.62) and (3.63), respectively. We can now calculate their Hessian matrices as:

$$\nabla^2 L_1(\alpha)=\left(\frac{\psi_1 I-E_1}{\psi_1}\right)\left[\frac{1}{2}(I-Q_1)E_1+(I+Q_1)\psi_1 I\right], \tag{3.64}$$

and

where

$$Q_1=diag\left(\frac{E_1\alpha-\psi_1\alpha-r_1}{|\alpha_0|}\right) \text{ and } Q_2=diag\left(\frac{E_2\beta-\psi_2\beta-r_2}{|\beta_0|}\right) \tag{3.65}$$

that is, in turn. The following basic iterative approaches for finding the Lagrangian multipliers may be solved by using a similar process to GRALTSVR α and β as:

$$\left[\frac{1}{2}(I-Q_1)E_1+(I+Q_1)\psi_1 I\right](\alpha^{i+1}-\alpha^i)=-[(E_1\alpha^i-r_1)-(E_1\alpha^i-\psi_1\alpha^i-r_1)_+] \tag{3.66}$$

and

$$\left[\frac{1}{2}(I-Q_2)D_2+(I+Q_2)\eta_2 I\right](\beta^{i+1}-\beta^i)=-[(D_2\beta^i-r_2)-(D_2\beta^i-\eta_2\beta^i-r_2)_+] \tag{3.67}$$

where

$$Q_1=diag\left(\frac{E_1\alpha^i-\psi_1\alpha^i-r_1}{|\alpha_0^i|}\right) \text{ and } Q_2=diag\left(\frac{E_2\beta^i-\psi_2\beta^i-r_2}{|\beta_0^i|}\right)$$

Lastly, we determine the final regressor f (x) for every test sample, much as SRALTSVR1 $x_s \in \Re^n$ by taking an average of $f_1(x_s)$and $f_2(x_s)$ In honor of the smooth approximation function, we have dubbed this method SRALTSVR2. Time complexity-wise, SRALTSVR2 is not dissimilar to SRALTSVR1, $2(m^3)+2^*O(i^*m^3)$.

Remark 2: The empirical version of TSVR outperforms SVR, but it has a few drawbacks, like being noise-sensitive, difficult to implement when dealing with big data, and losing control over the complexity of the model, which causes overfitting and less-than-ideal solutions. To explain why URALTSVR learns more quickly than TSVR and SVR, our suggested method solves two systems of linear equations rather than quadratic programming issues.

This improved approach is more resilient while controlling the fitting error within the asymmetric tube with the pinball loss function, and it also produces better or equivalent generalization performance.

Remark 3: Discussion based on SVR vs TSVR vs Asy-ν -TSVR vs URALTSVR

➢ There is a globally unique solution and the non-negativity constraints for slack variables can be discarded because they are inevitably satisfied at optimum, thanks to the model's stronger convexity and the significance of the L2-norm of the square of the vector of slack variables, which is different from other state-of-the-art approaches.

➢ In comparison to TSVR and Asy-κ -TSVR, the dual formulations benefit from regularization's substantial influence, which makes the issue positive definite and produces a well-posed model that mitigates overfitting and enhances stability.

➢ Unlike SVR, TSVR, and Asy-TSVR, the proposed URALTSVR uses gradient-based iterative techniques to solve an unconstrained issue, eliminating the need to tackle huge quadratic programming problems (QPPs).

➢ Their non-differentiable goal function takes into account a non-smooth '+' function in URALTSVR. A generalized derivative technique, a smooth approximation method, or the Newton method is all possibilities for handling this non-smooth function '+'.

➢ In contrast to SVR and TSVR, an external toolbox is not required when using a function iterative technique to handle the unconstrained minimization issue, Asy-ν -TSVR.

3.2.2 NUMERICAL EXPERIMENTS

A desktop computer equipped with a 3.20 GHz 64-bit CPU is taken into account for the numerical experiment Intel® Core™ i5-3470, MATLAB 2008b software compatibility requires a minimum of 4 GB of RAM on a physical device and Windows 10. Also used in the solution of the QPP is an external optimization toolkit called MOSEK.

We have conducted numerical tests with different algorithms, such as SVR and TSVR, to evaluate GRALTSVR, SRALTSVR1, and SRALTSVR2 for their efficiency and Asy-ν -TSVR using sixteen synthetic datasets and seventeen real-world standard benchmark datasets.

We tried the experiment with six methods for the linear and nonlinear cases using ten-fold cross-validation to assess the sufficient performance for each dataset. Essentially, this implies that the training dataset is partitioned into two halves: one half is reserved for training, while the other half is used for testing to determine the best values for the parameters.

We have computed RMSE, SSE/SST, SSR/SST, and SMAPE to compare the prediction performance of our proposed algorithms GRALTSVR, SRALTSVR1, and

SRALTSVR2 with other methods that have been reported. As mentioned in Table 3.7, parameter selection is a critical step in these experiments because it directly impacts the algorithms' performance.

TABLE 3. 7 List of applicable all parameters and their range in URALTSVR.

Parameters	Models	Range
ε	SVR	{0.1,0.01,0.001}
	TSVR	{0.1,0.3,0.5,0.7,0.9}
ν_1 and ν_2	Asy-ν -TSVR, GRALTSVR, SRALTSVR1, and SRALTSVR2	{0.1,0.3,0.5,0.7,0.9}
p	Asy-ν -TSVR, GRALTSVR, SRALTSVR1, and SRALTSVR2	{0.2,0.4,0.45,0.5,0.55,0.6,0.8}
μ	SVR, TSVR, Asy-ν -TSVR, GRALTSVR, SRALTSVR1, and SRALTSVR2	$\{2^{-5},\ldots,2^{5}\}$
C	SVR, TSVR and Asy-ν -TSVR	$\{10^{-5},\ldots,10^{5}\}$
$C_1 = C_2$ and $C_3 = C_4$	GRALTSVR, SRALTSVR1, and SRALTSVR2	$\{10^{-5},10^{-3},10^{-1},10^{1},10^{3},10^{5}\}$

The non-linear version of the Gaussian kernel function looks like this:

$K(x_q, x_r) = \exp(-\mu \| x_q - x_r \|^2)$, for $q, r = 1,2,\ldots, p$

in this case the kernel parameter $\mu > 0$.

Artificial Datasets

We examine sixteen synthetic datasets that are created intentionally and whose definitions are provided in, meaning that the noise is reliant on the value of the input data, in order to assess the robustness of the proposed URALTSVR. In this case, we create the noise by treating the interval (a,b) as a uniform probability distribution and treating it as noise $\Omega \in U(a,b)$ together with the use of standard deviation $N(\mu, \sigma^2)$.

To demonstrate the efficacy of the presented approaches in a noise-free environment, the first twelve synthetic datasets include 500 testing samples randomly created and 200 training samples randomly generated with the inclusion of symmetric noise of both kinds.

To illustrate heteroscedasticity, the last four datasets use two hundred randomly generated samples for training, with the addition of a non-uniform level of noise using both types. To test and validate the reported methods, 500 samples are generated.

Looking at these tables, we can see that out of sixteen synthetic datasets, our suggested methods performed better in ten cases using a linear kernel and nine cases using a Gaussian kernel, leading to better generalization performance.

This allows us to numerically evaluate the efficacy of GRALTSVR, SRALTSVR1, and SRALTSVR2. Additionally, GRALTSVR and SRALTSVR2 need less time to train than SVR, TSVR, and Asy-□ -TSVR. This is because, unlike SVR, TSVR, and Asy-□ -TSVR, these techniques instead discover the solution via solving gradient-based iterative schemes, rather than the QPP.

More specifically, we have used the linear kernel in Table 3.8 and the Gaussian kernel in Table 3.9 to calculate the average rank of the described techniques for all synthetic datasets based on RMSE.

TABLE 3. 8 RMSE Rankings of URALTSVR and Reported Models on Synthetic Datasets with Linear Kernel.

Dataset	SVR	TSVR	Asy-v-TSVR	GRALTSVR	SRALTSVR1	SRALTSVR2
Function 1	6	4	5	1.5	3	1.5
Function 2	1	6	5	3.5	2	3.5
Function 3	1	4	6	2.5	5	2.5
Function 4	6	3	2	4.5	1	4.5
Function 5	6	4	5	2.5	1	2.5
Function 6	1	6	5	2.5	4	2.5
Function 7	6	4	5	1.5	3	1.5
Function 8	6	2	1	4.5	3	4.5
Function 9	6	3	4	1.5	5	1.5
Function 10	2	1	3	5.5	4	5.5
Function 11	4	5	6	1.5	3	1.5
Function 12	1	6	5	2.5	4	2.5
Function 13	6	3	4	1.5	5	1.5
Function 14	1	4	6	2.5	5	2.5
Function 15	6	5	2	3.5	1	3.5
Function 16	5	3	6	1.5	4	1.5
Average rank	4	3.9375	4.375	2.6875	3.3125	2.6875

TABLE 3. 9 RMSE Rankings of URALTSVR and Reported Models on Synthetic Datasets with Gaussian Kernel.

Dataset	SVR	TSVR	Asy-v-TSVR	GRALTSVR	SRALTSVR1	SRALTSVR2
Function 1	6	4	5	1.5	3	1.5
Function 2	1	6	5	3.5	2	3.5
Function 3	1	4	6	2.5	5	2.5
Function 4	6	3	2	4.5	1	4.5
Function 5	6	4	5	2.5	1	2.5
Function 6	1	6	5	2.5	4	2.5
Function 7	6	4	5	1.5	3	1.5
Function 8	6	2	1	4.5	3	4.5
Function 9	6	3	4	1.5	5	1.5
Function 10	2	1	3	5.5	4	5.5
Function 11	4	5	6	1.5	3	1.5
Function 12	1	6	5	2.5	4	2.5
Function 13	6	3	4	1.5	5	1.5
Function 14	1	4	6	2.5	5	2.5
Function 15	6	5	2	3.5	1	3.5
Function 16	5	3	6	1.5	4	1.5
Average rank	4	3.9375	4.375	2.6875	3.3125	2.6875

Figure 3.11 shows the suggested techniques in the non-linear situation, and Figure 3.12 shows the average rankings in terms of RMSE of all the baseline approaches. We have also included additional information by plotting a boxplot for the linear case. Our suggested methods outperform previously published algorithms for binary classification, as shown in Figures 3.11-3.12. Table 3.8 shows that SRALTSVR1 has the lowest average rank among all reported approaches, while Table 3.9 shows that GRALTSVR and SRALTSVR2 have the lowest average rank. This supports the idea that GRALTSVR, SRALTSVR1, and SRALTSVR2 are more applicable and robust than SVR, TSVR, and Asy-v-TSVR. We have shown the prediction accuracy graphs for Function 4 with Gaussian noise in Figure 3.14 and for Function 3 with uniform noise in Figure 3.13.

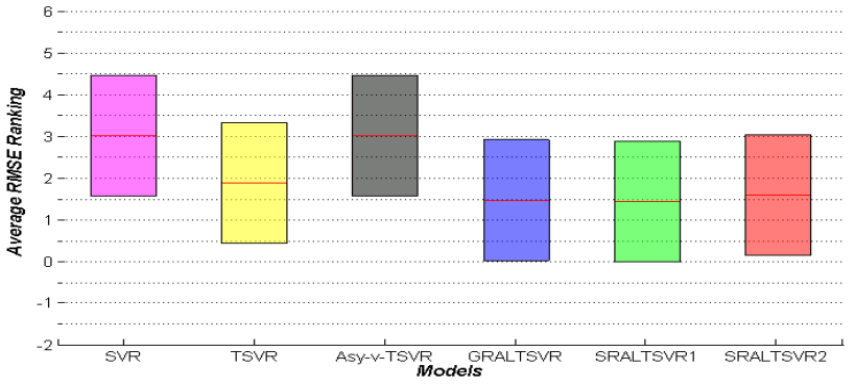


FIGURE 3. 11 Boxplot of Average RMSE Ranks on Synthetic Datasets Using Linear Kernel for URALTSVR and Reported Models

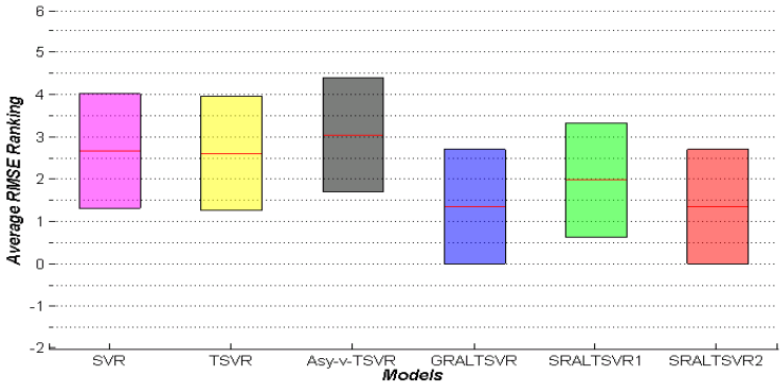


FIGURE 3. 12 Boxplot of Average RMSE Ranks for Synthetic Datasets with Gaussian Kernel

Figures 3.15 and 3.16 show the prediction performance for Function 9 and Function 10, respectively. Our suggested algorithms GRALTSVR, SRALTSVR1, and SRALTSVR2 work well with datasets that have a symmetric noise structure, as shown by the graphs, which demonstrate superior or equivalent regression functions for these approaches. So, for datasets with heteroscedastic error (noise), the input samples have a significant impact on the performance of our suggested approaches GRALTSVR, SRALTSVR1, and SRALTSVR2. Similar to the case of uniform noise, Figures 3.17–3.20 show the accuracy plot for Function 13, 14, 15, and 16, respectively, in synthetic datasets. Our approaches are more beneficial for handling the influence of the heteroscedastic error structure, as can be seen from these figures.

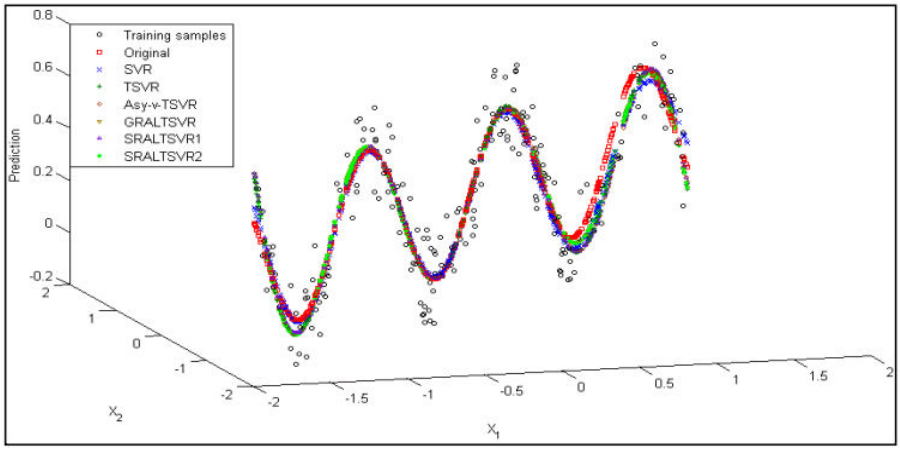


FIGURE 3. 13 Accuracy Visualization for Reported Models on Test Set Implementing Function 3 using Uniform Noise and a Gaussian Kernel.

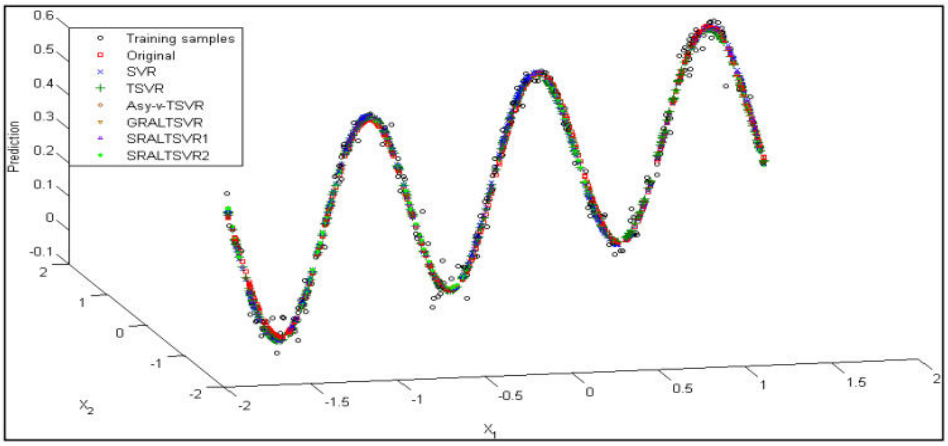


FIGURE 3. 14 Accuracy Visualization for Reported Models on Test Set Combining Gaussian Noise with a Gaussian Kernel to Evaluate Function 4

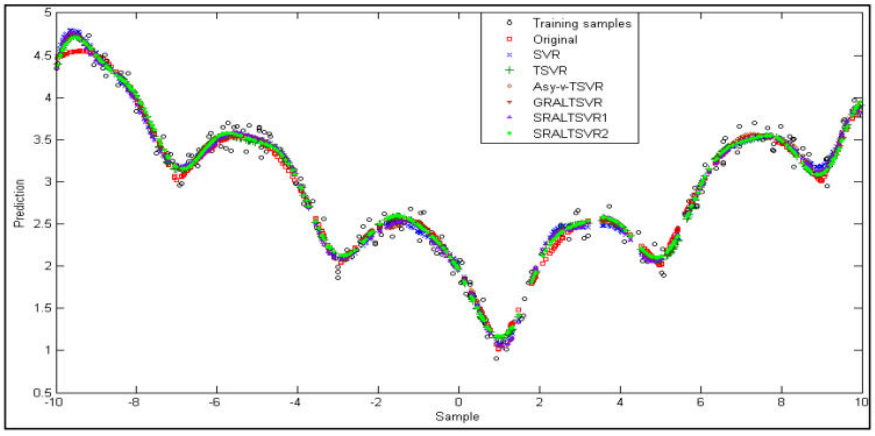


FIGURE 3. 15 Accuracy Plot of Reported Models on Test Set Using Gaussian Kernel for Function 9 (Uniform Noise)

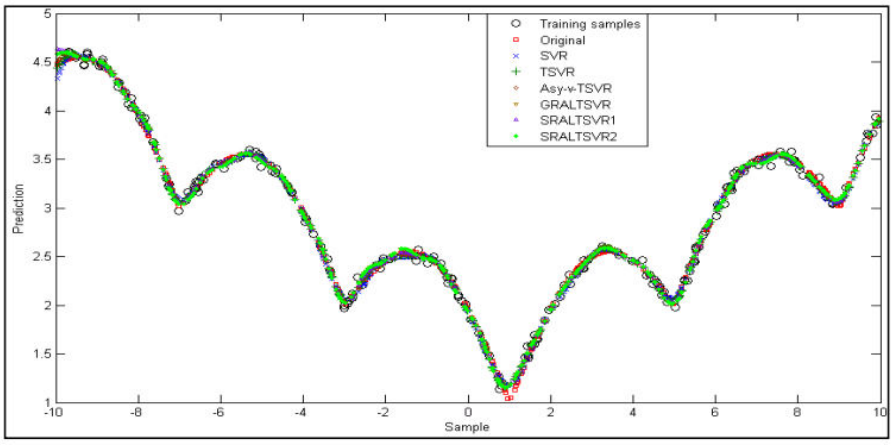


FIGURE 3. 16 Accuracy Plot of Reported Models on Test Set Making Use of a Gaussian Kernel in Function 10 (Hazard))

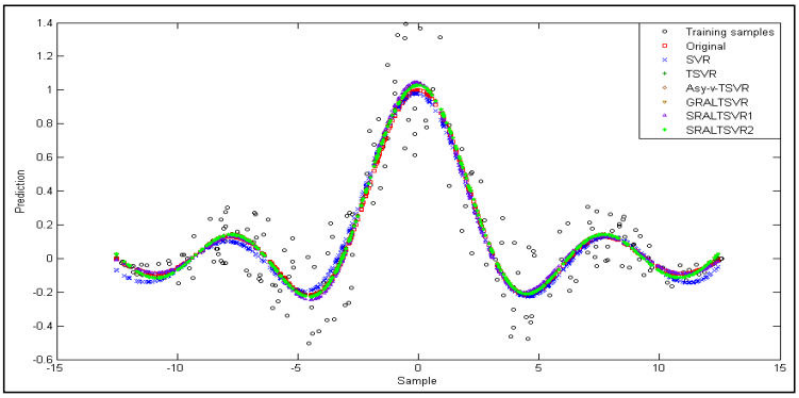


FIGURE 3. 17 Accuracy Plot of Reported Models on Test Set Implementing Function 13 (Uniform Noise) using a Gaussian Kernel)

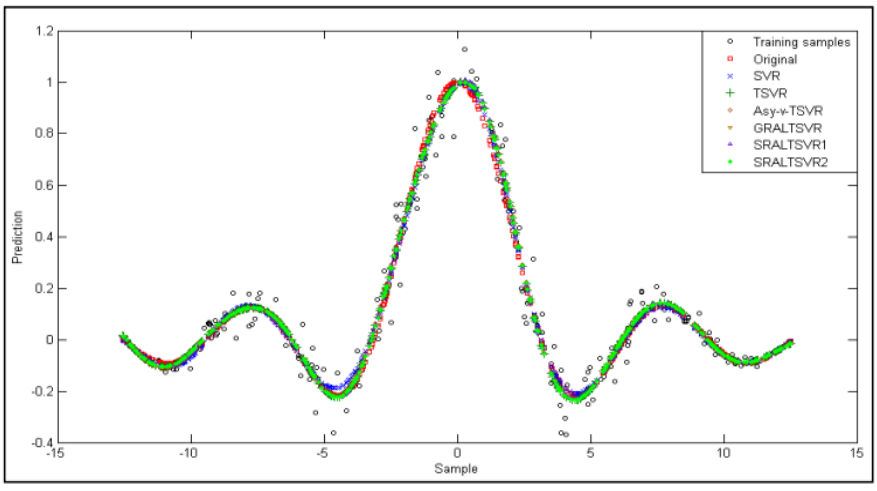


FIGURE 3. 18 Accuracy Plot of Reported Models on Test Set Procedure 14 (Gaussian Noise) using a Gaussian Kernel

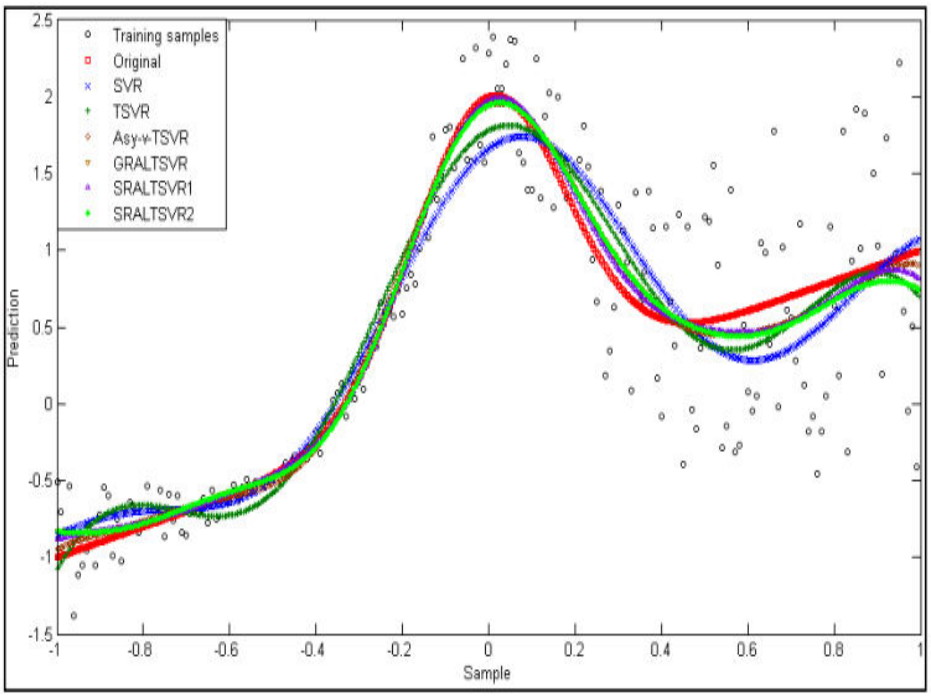


FIGURE 3. 19 Accuracy Plot of Reported Models on Test Set Function 15 (Uniform Noise) Using a Gaussian Kernel)

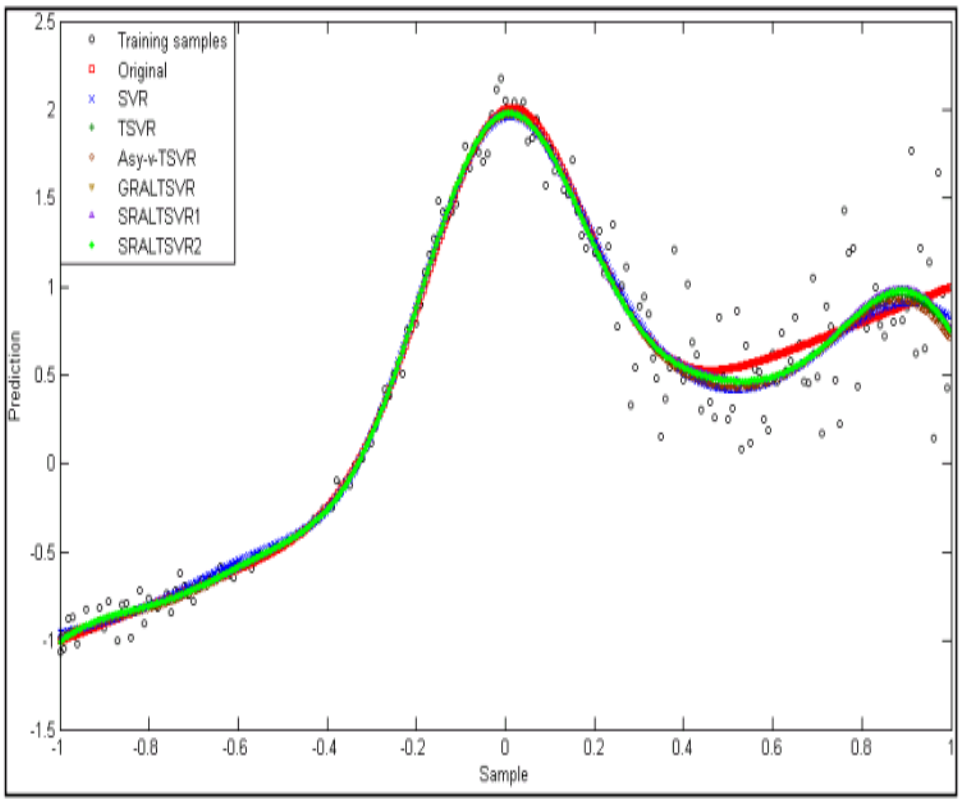


FIGURE 3. 20 Accuracy Plot of Reported Models on Test Set Formula 16 (Gaussian Noise): Applying a Gaussian Kernel

a) Linear case

In this case, we can say that, when comparing the six algorithms, SRALTSVR1 often has the lowest error outcomes.

Among seventeen real-world datasets, SVR, TSVR, Asy-ν-TSVR, GRALTSVR, SRALTSVR1, and SRALTSVR2 achieved the best results in two, zero, four, seven, and four instances, respectively. The calculation times of GRALTSVR, SRALTSVR1, and SRALTSVR2 are comparable to TSVR and Asy-ν-TSVR, and they are always much faster than SVR. Our suggested techniques GRALTSVR, SRALTSVR1, and SRALTSVR2 all use the linear kernel; Table 3.10 demonstrates that SRALTSVR1 has the lowest average RMSE rank of all the approaches that were reported. Table 3.11 shows the projected average ranks of other performance metrics, including SSE/SST, SSR/SST, and SMAPE, which we used to verify the efficacy of our suggested methods. According to Table 3.11, SRALTSVR1 ranks first for SSE/SST, SSR/SST, and SMAPE, suggesting that the same result is achieved. Figure 3.21 is a bar graph that illustrates the average rank values according to SSE/SST, SSR/SST, and SMAPE. The prediction graphs for the linear kernel for the Hydraulic actuator, Gas furnace, Machine CPU, Pollution, and RedHat datasets are shown in Figures 3.22, 3.24, 3.26, 3.28, and 3.30, respectively. The regression functions of URALTSVR models are greater than those of SVR, TSVR, and Asy-ν-TSVR, as shown in Figures 3.23, 3.25, 3.27, 3.29, and 3.31, respectively.

TABLE 3. 10 RMSE Rankings of URALTSVR and Reported Models on Real-World Datasets with Linear Kernel.

Dataset	SVR	TSVR	Asy-*ν*-TSVR	GRALTSVR	SRALTSVR1	SRALTSVR2
Hydraulic Actuator	6	1	5	3.5	2	3.5
Auto-MPG	5	6	4	2.5	1	2.5
Citigroup	6	1	5	2.5	4	2.5

Concrete CS	6	1	2	4.5	3	4.5
Demo	6	5	4	1.5	3	1.5
Flexible robot arm	2	6	5	3.5	1	3.5
Gas furnace	4	6	5	2.5	1	2.5
IBM	6	5	4	2.5	1	2.5
Kin900	3	2	4	5.5	1	5.5
Machine CPU	5	1	2	3.5	6	3.5
Mg17	6	2.5	5	2.5	2.5	2.5
Pollution	4	6	5	1.5	3	1.5
Parkinson	1	3	4	5.5	2	5.5
RedHat	5	4	6	2.5	1	2.5
SantaFeA	6	4	3	1.5	5	1.5
Sunspots 94	1	2	5	3.5	6	3.5
Wine-quality white	2	1	6	4.5	3	4.5
Average rank	4.35294	3.32353	4.35294	3.14706	2.67647	3.14706

TABLE 3. 11 Comparing URALTSVR with Preexisting Models on Real-World Datasets Using SSE/SST, SSR/SST, and SMAPE as Linear Kernel Metrics

Measures	**SVR**	**TSVR**	**Asy-*v*-TSVR**	**GRALTSVR**	**SRALTSVR1**	**SRALTSVR2**

RatioSSE_SST	3.8824	3.3824	4.2941	3.3235	2.7941	3.3235
RatioSSR_SST	3.7059	3.8824	3.1176	3.6176	3.0588	3.6176
SMAPE	3.9412	3.1177	3.8235	3.5	3.1176	3.5

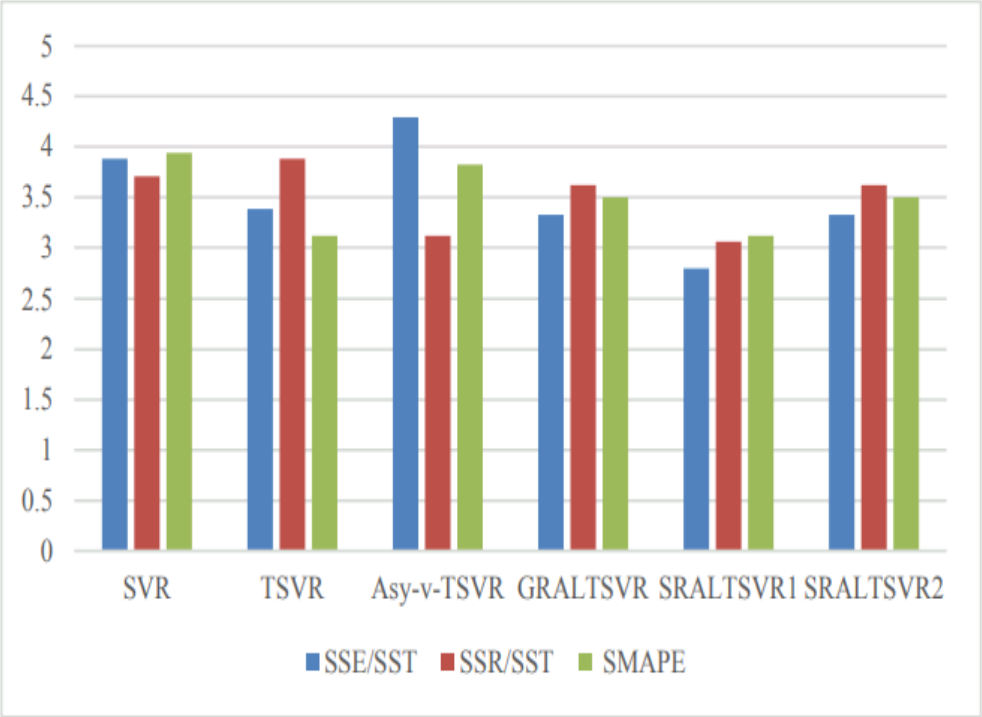


FIGURE 3. 21 Plot of Evaluation Parameter-Based Average Ranks for The Use of Linear Kernels in Various Methods on Real-World Benchmark Datasets.

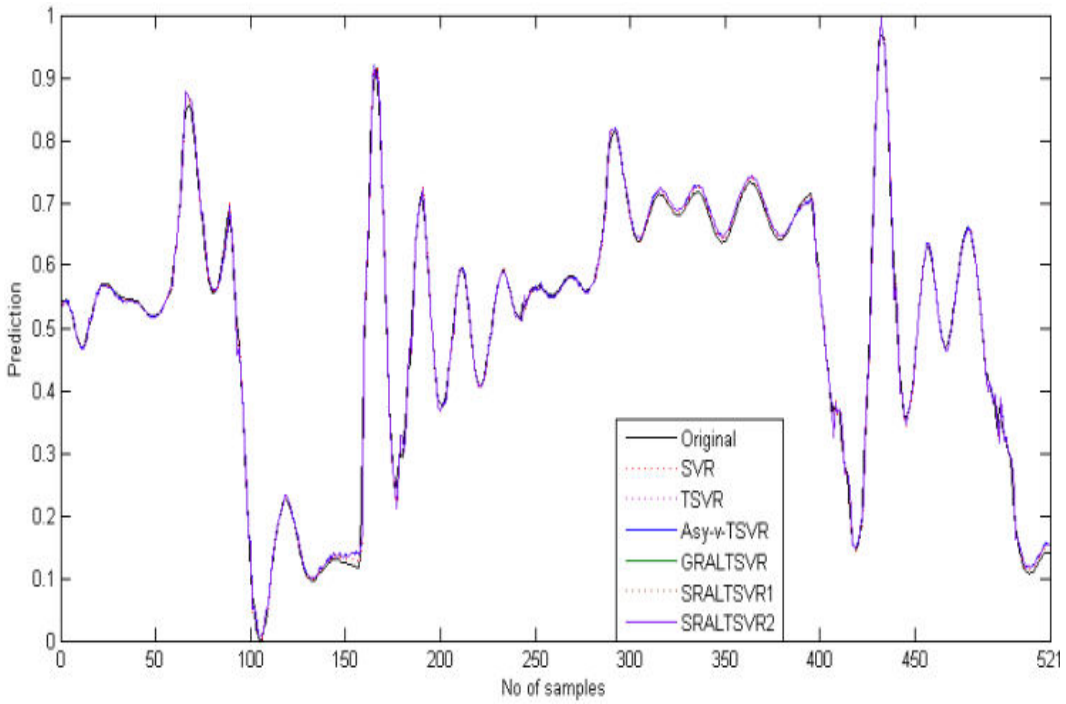


FIGURE 3. 22 rediction Results for Hydraulic Actuator Testing Linear Kernel Dataset.

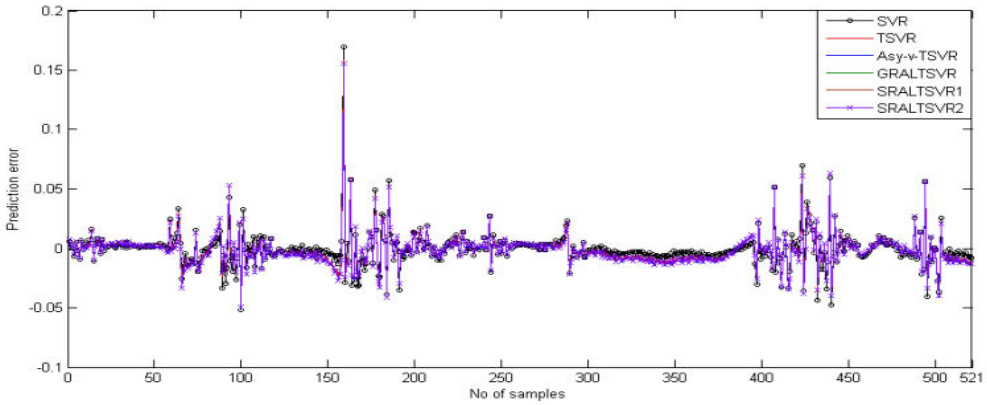


FIGURE 3. 23 Prediction Error on the Linear Kernel-Based Hydraulic Actuator Testing Dataset.

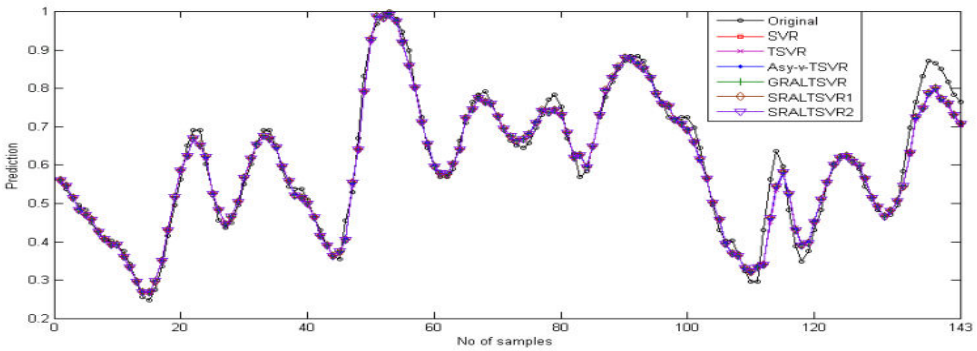


FIGURE 3. 24 Prediction Results for Gas Furnace Testing Linear Kernel Dataset.

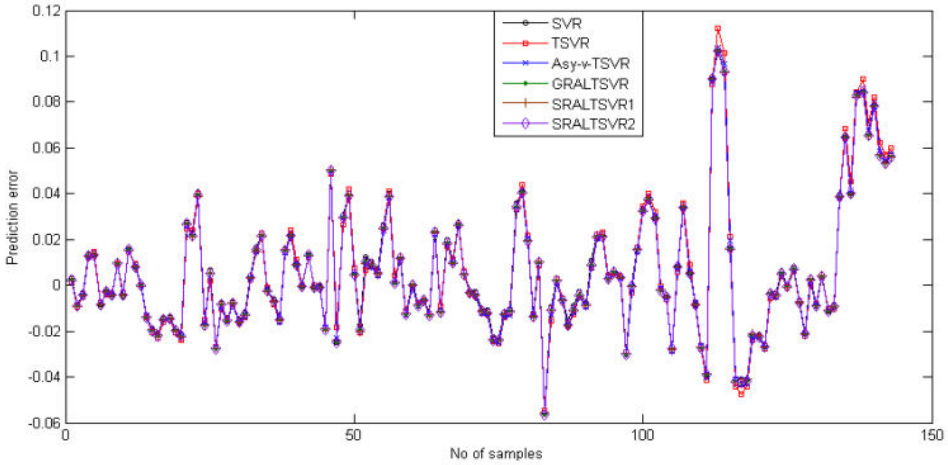


FIGURE 3. 25 Gas Furnace Testing Dataset Prediction Error Using Linear Kernel.

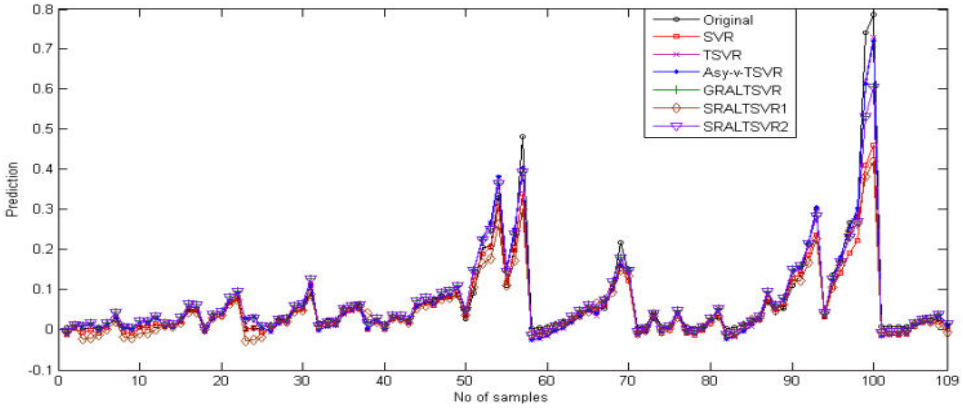


FIGURE 3. 26 Prediction Results for Machine CPU Testing Linear Kernel Dataset.

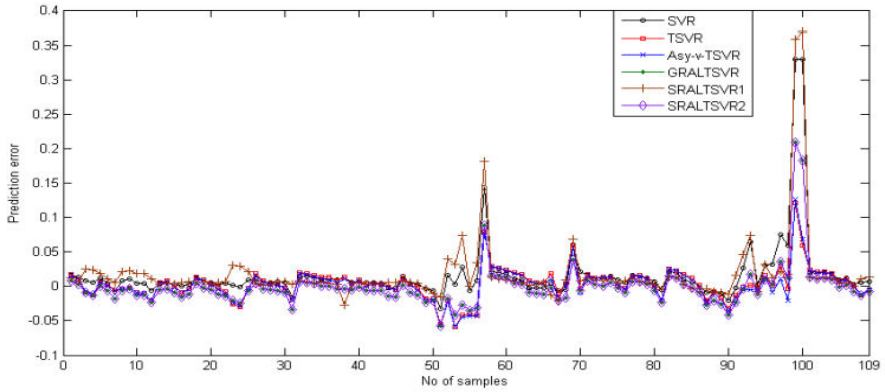


FIGURE 3. 27 Machine CPU Testing Dataset Prediction Error Using a Linear Kernel.

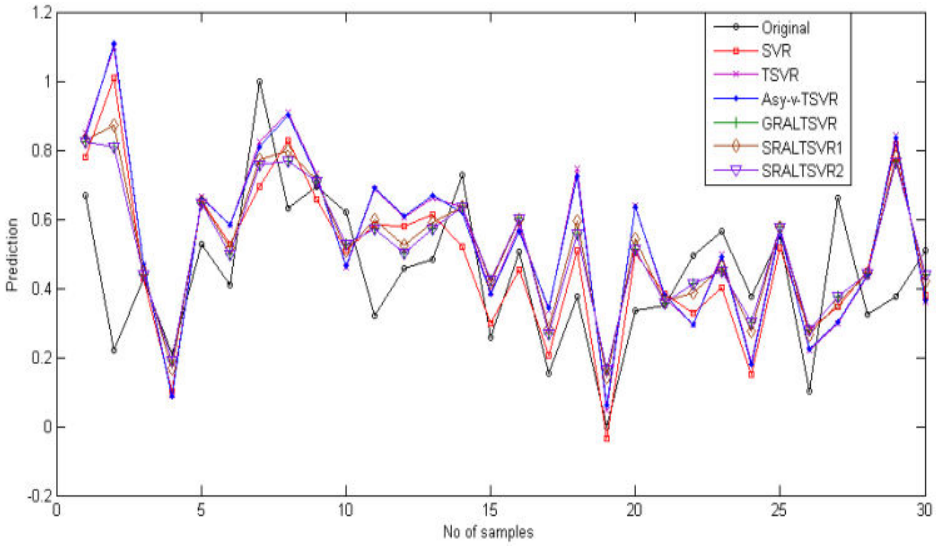


FIGURE 3. 28 Prediction Results for Pollution Testing Linear Kernel Dataset.

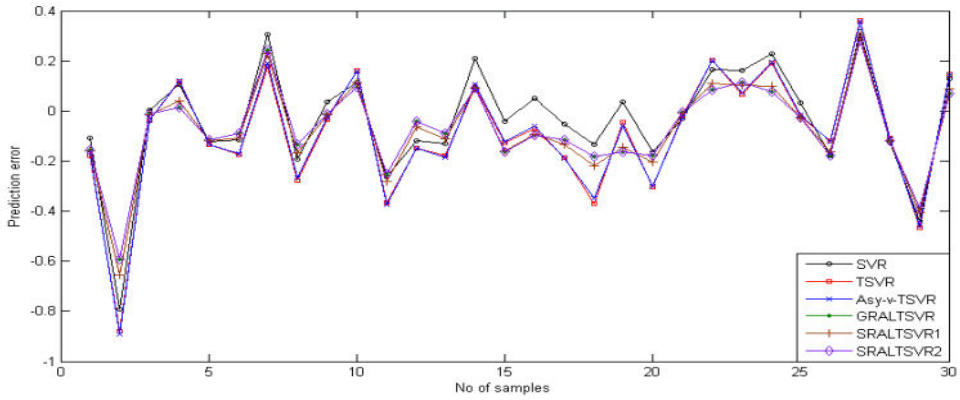


FIGURE 3. 29 Forecast Inaccuracy on the Linear Kernel-Based Pollution Testing Dataset.

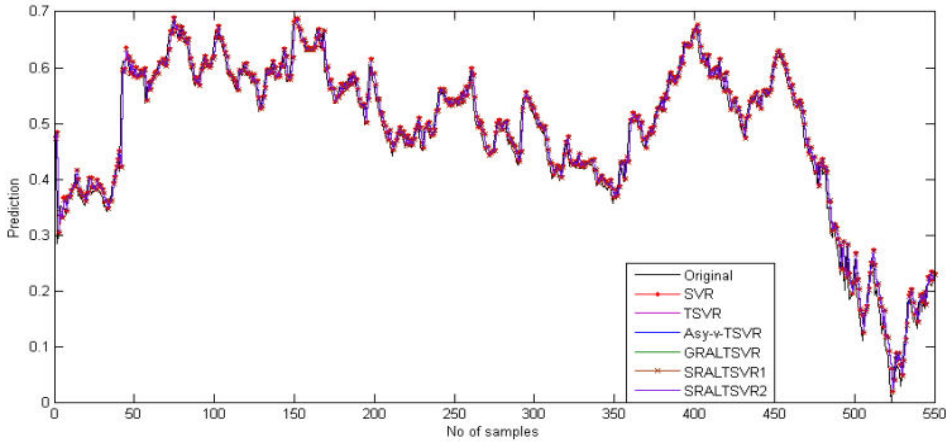


FIGURE 3. 30 Prediction Results for RedHat Testing Linear Kernel Dataset.

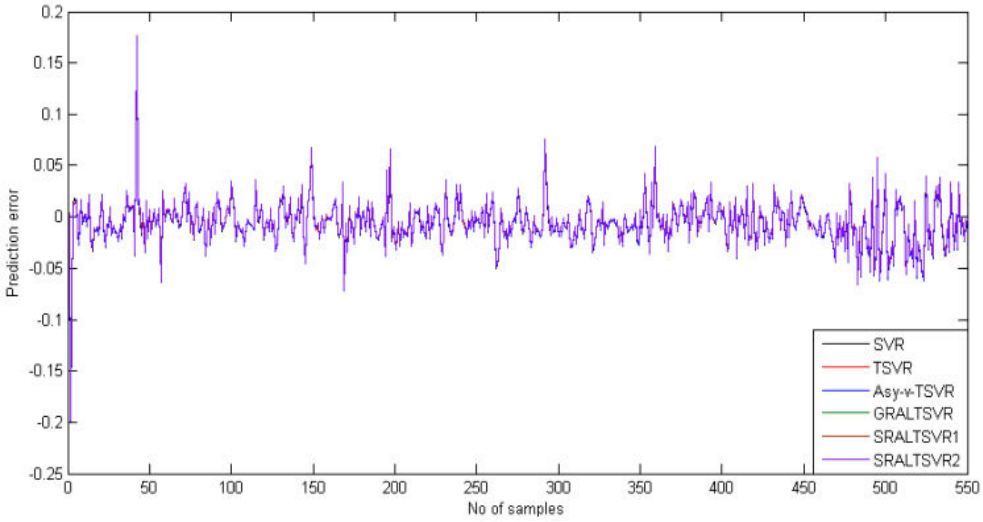


FIGURE 3. 31 Mistakes Made by Linear Kernel Predictions on the RedHat Testing Dataset.

Test for Friedman's hypothesis:

All methods utilized in this numerical experiment were shown to be statistically valid by using the non-parametric Friedman test [142]. Based on the average rankings in Table 3.10 of the following groups: SVR, TSVR, Asy-□-TSVR, GRALTSVR, SRALTSVR1, and SRALTSVR2, the Friedman expression is produced. According to Table 3.10, the formula for the Friedman statistic is as follows, assuming the null hypothesis is constant:

$$\chi_F^2 = \frac{12\times 17}{6\times 7}\left[\left(4.35294^2 + 3.32353^2 + 4.35294^2 + 3.14706^2 + 2.67647^2 + 3.14706^2\right) - \left(\frac{6\times 7^2}{4}\right)\right] = 11.7227$$

and

$$F_F = \frac{16 \times 11.7227}{17 \times 5 - 11.7227} = 2.5596$$

If one believes the Fisher-Snedecor, what Friedman calls There's a little something that FF can do. The F distribution (6 − 1, (6 − 1)×(17 − 1)) = (5, 80) incorporates seventeen common benchmark datasets sourced from real life and six fascinating methods. At α = 0.05, the critical value (CV) of F(5,80) is 2.32872. In this case, the value of Friedman's expression FF is higher than that of the CV, namely, FF > 2.32872. This method-to-method variation is the Achilles' heel of the null hypothesis at the acceptance level. The purpose of using the Nemenyi test was to directly compare two intriguing algorithms. We agree with [142] that this is the most important difference.

$$\text{Critical difference } = 2.589\sqrt{\frac{6\times7}{6\times17}} = 1.6613 \text{ at p } = 0.10\,.$$

In this case, statistical analysis allows us to draw the following conclusions:

i. The numbers that distinguish SVR and Asy-□-TSVR, which are 4.35294 - 2.67647 = 1.67647, may be compared to the average rank of SRALTSVR1. Given that (4.35294− 2.67647 = 1.67647) is more than 1.6613, it is evident that SRALTSVR1 outperforms SVR and Asy-□ -TSVR in terms of generalization performance.

Part two. For the suggested techniques SRALTSVR1 and GRALTSVR or SRALTSVR2, the test does not find a significant difference as the difference between their best and lowest average ranks is less than the critical difference of 1.6613.

iii. Figure 3.32 displays the outcomes of the Friedman statistical tests performed on each of the presented techniques using real-world linear kernel datasets. The findings demonstrate that SRALTSVR1 outperforms the other approaches.

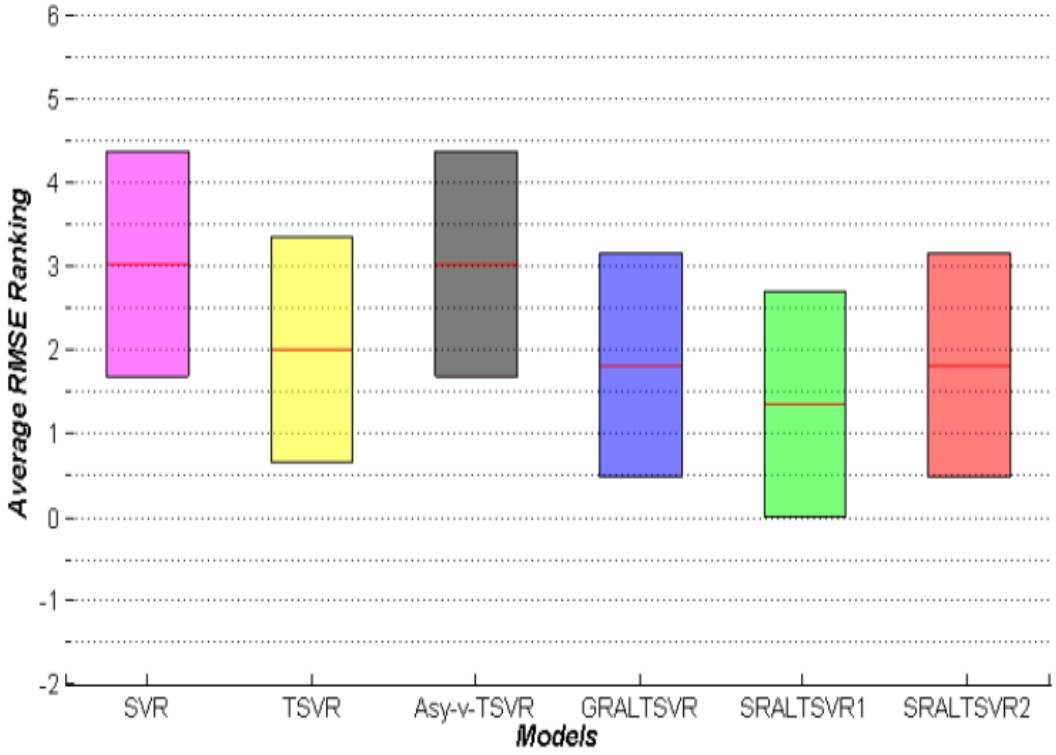


FIGURE 3. 32 A Linear Kernel Boxplot Showing the Average RMSE Ranks of All Presented Models on Real-World Datasets.

b) Non-linear case:

Just as in the linear example, SRALTSVR1 consistently achieves the best results for generalization. Our proposed methods are computationally equal to TSVR and Asy-□-TSVR and perform better than SVR since they do not rely on an external optimization toolbox.

It is evident that the proposed methods also provide the same decision picture, since Table 3.12 displays the average rank of all the evaluated techniques employing a Gaussian kernel, including URALTSVR models. In Figure 3.33, we can see the ranking graphs of SSE/SST, SSR/SST, and SMAPE values derived from real-world datasets.

Refer to Figures 3.34-3.42 for the Hydraulic actuator dataset, Figures 3.36-3.38 for the Gas furnace dataset, Figures 3.40-3.42 for the Machine CPU dataset, and Figures 3.42-3.42 for the Pollution dataset. The RedHat dataset also includes prediction graphs and prediction error graphs.

TABLE 3. 12 RMSE-Based Average Rankings of URALTSVR and Reported Models for Real-World Datasets with Gaussian Kernel.

Dataset	SVR	TSVR	Asy-*v*-TSVR	GRALTSVR	SRALTSVR1	SRALTSVR2
Hydraulic Actuator	2	5	6	3.5	1	3.5
Auto-MPG	6	2	1	4.5	3	4.5
Citigroup	1	3	4	5.5	2	5.5
Concrete CS	6	3	2	4.5	1	4.5
Demo	4	6	5	1.5	3	1.5
Flexible robot arm	1	6	5	3.5	2	3.5
Gas furnace	1	5	6	3.5	2	3.5
IBM	4	2	3	5.5	1	5.5
Kin900	6	5	4	2.5	1	2.5
Machine CPU	4	6	5	2.5	1	2.5
Mg17	2	6	5	3.5	1	3.5
Pollution	6	1	3	4.5	2	4.5
Parkinson	6	5	4	2.5	1	2.5
RedHat	2	6	5	4	1	3
SantaFeA	1	4	6	2.5	5	2.5
Sunspots 94	6	2	1	4.5	3	4.5
Wine-quality white	2	5	6	3.5	1	3.5
Average rank	3.52941	4.23529	4.17647	3.64706	1.82353	3.58824

TABLE 3. 13 Using Gaussian Kernel, Average Rankings of URALTSVR and Reported Models Based on SSE/SST, SSR/SST, and SMAPE Metrics for Real-World Dataset

Measures	SVR	TSVR	Asy-*v*-	GRALTSVR	SRALTSVR1	SRALTSVR2

			TSVR			
SSE/SST	3.5294	4.2353	4.1765	3.6176	1.8235	3.6176
SSR/SST	3.7647	3.3529	3.7647	3.3529	3.4706	3.2941
SMAPE	3.0588	4.0294	4.1471	3.7647	2.2941	3.7059

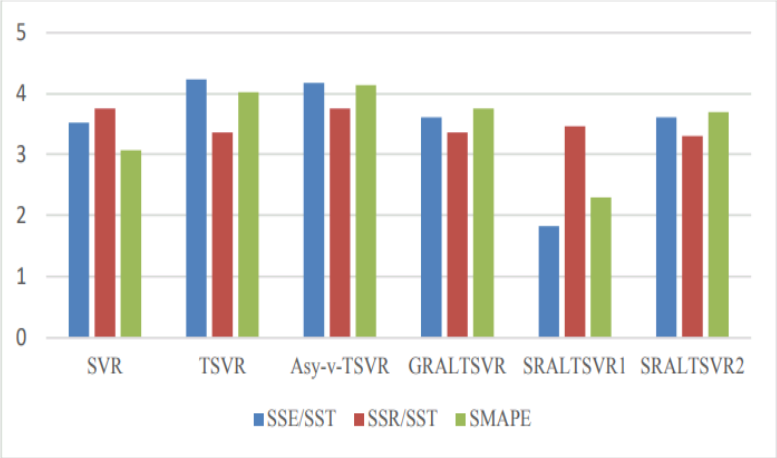


FIGURE 3. 33 Visualization of the Mean Quality Metric Rankings of Different Algorithms on Gaussian Kernel Benchmark Real-World Datasets.

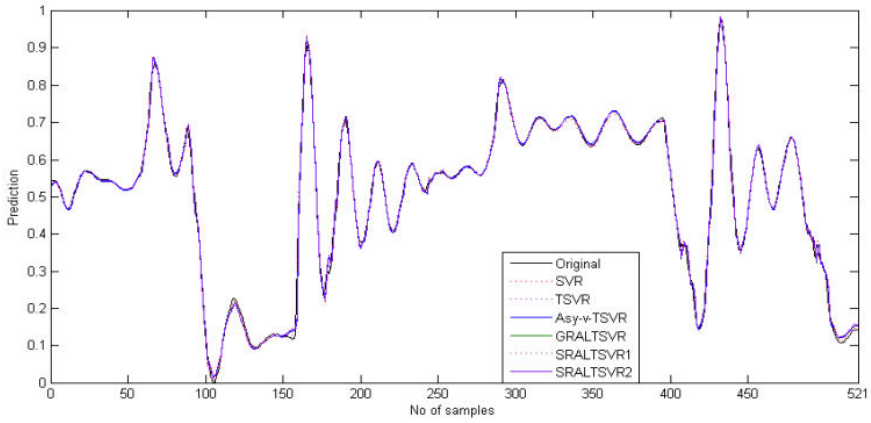


FIGURE 3. 34 Findings from All Presented Models on the Hydraulic Actuator Dataset Employing the Gaussian Kernel for Prediction on the Testing Dataset.

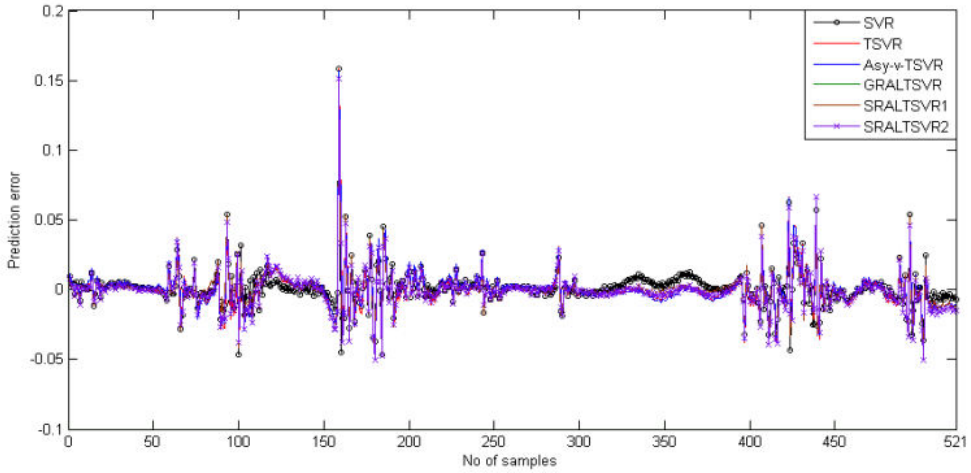


FIGURE 3. 35 The Testing Dataset Prediction Error for All Presented Models on the Hydraulic Actuator Dataset using a Gaussian Kernel.

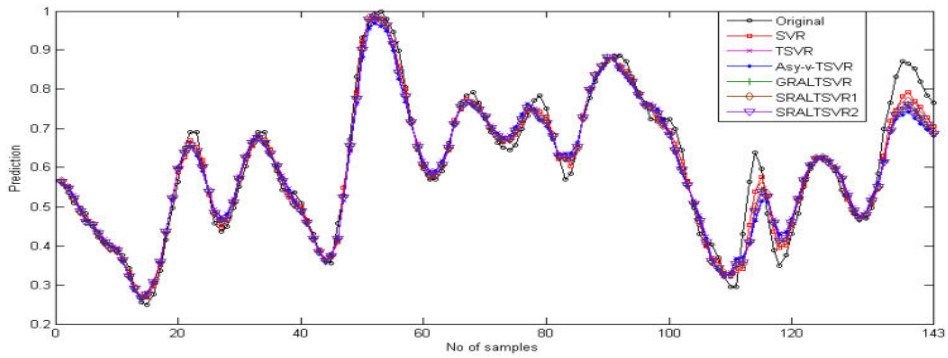


FIGURE 3. 36 Analysis of all published models' predictions using a Gaussian kernel on the Gas Furnace dataset and their validation on the testing dataset.

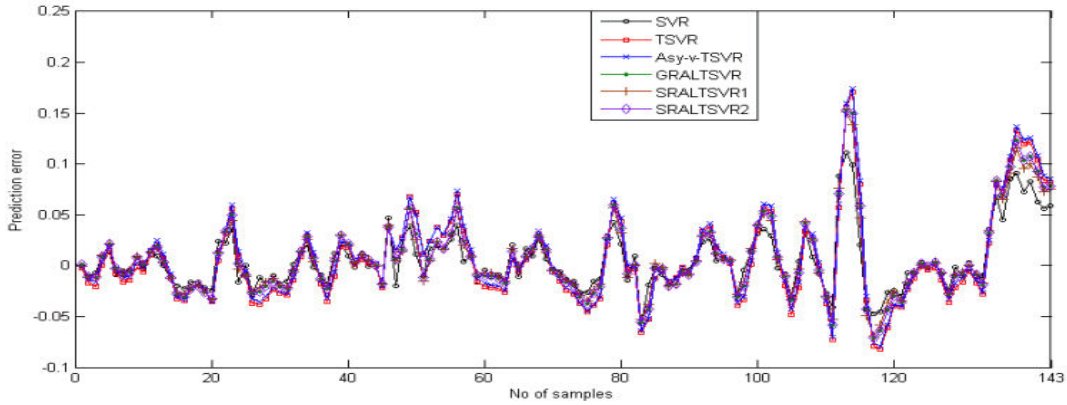


FIGURE 3. 37 The difference between the predicted and actual results on the Gas Furnace dataset using all of the presented models trained using a Gaussian kernel.

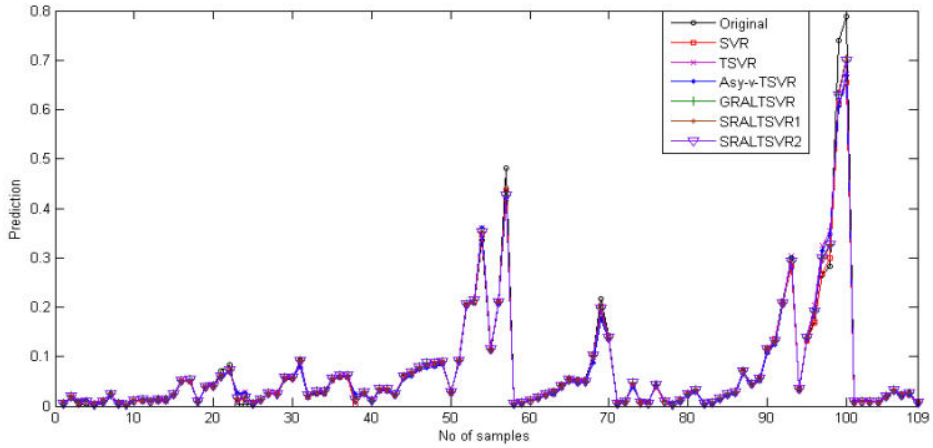


FIGURE 3. 38 Forecasts made by all declared models on the Machine CPU dataset with a Gaussian kernel applied to the testing dataset.

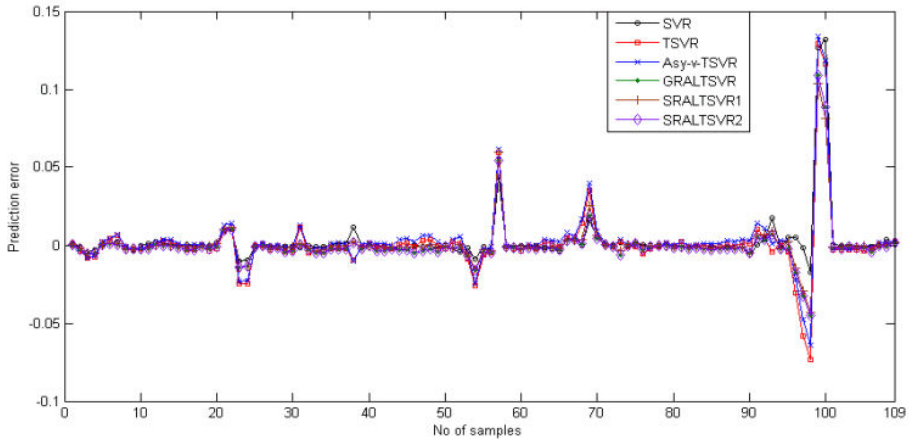


FIGURE 3. 39 Machine CPU dataset prediction error on the testing dataset for all models reported using a Gaussian kernel.

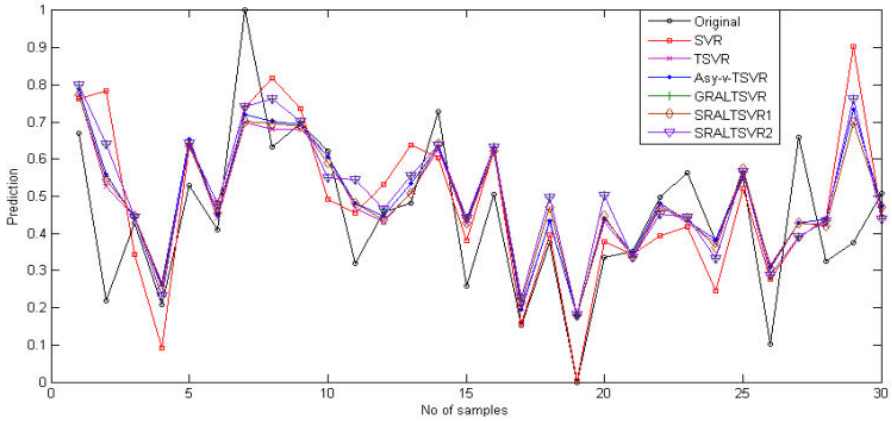


FIGURE 3. 40 Forecasts made by all the models that were reported on the Pollution dataset using a Gaussian kernel on the testing dataset.

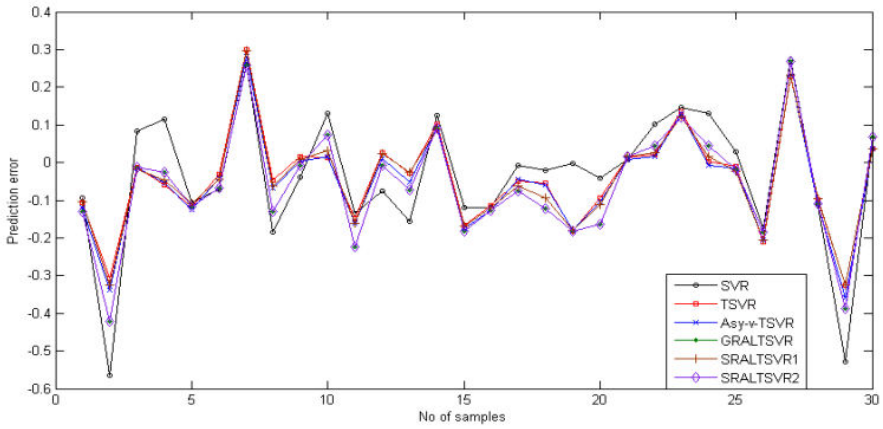


FIGURE 3. 41 Inaccurate predictions made by all the models on the Pollution dataset using the Gaussian kernel on the testing dataset.

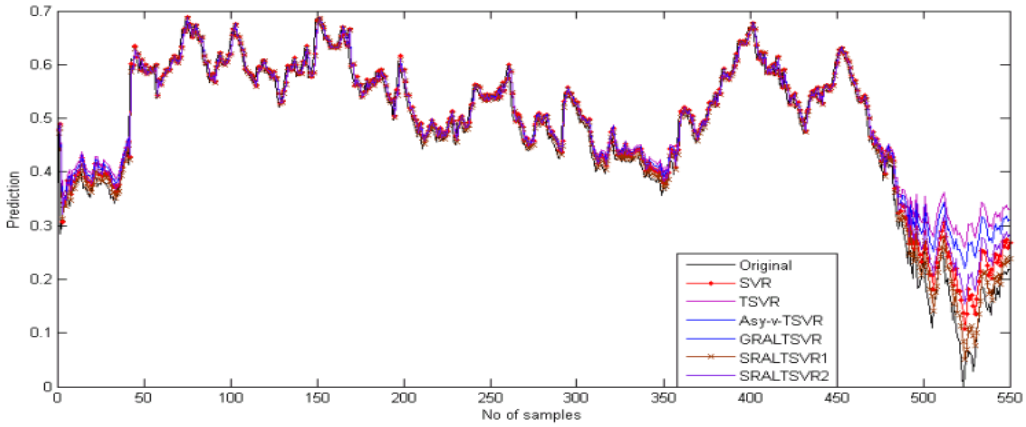


FIGURE 3. 42 All stated models' RedHat dataset predictions using a Gaussian kernel on the testing dataset.

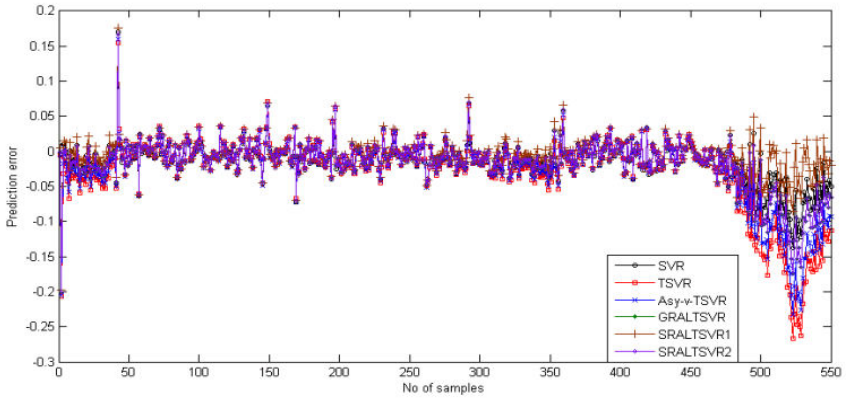
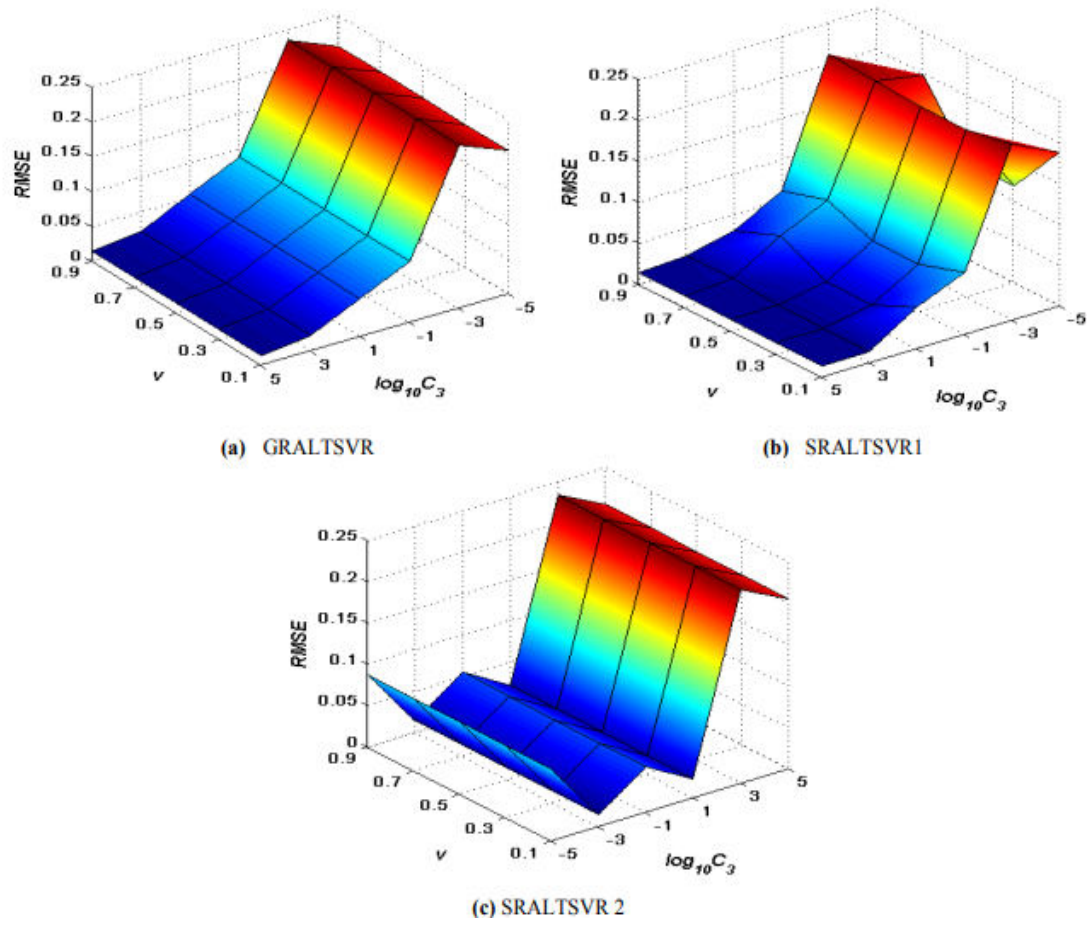


FIGURE 3. 43 Prediction error over the testing dataset by all reported models on the RedHat dataset using Gaussian kernel.

Evaluation of sensitivity

The parameter-insensitive URALTSVR models' generalizability performance was also of interest to us. Data from hydraulic actuators within a preset range for parameters C3 and are used to generate the sensitivity effectiveness plots of the URALTSVR model, as illustrated in Figure 3.44. ν. This allows the obtained numerical test results to be more prominently shown. Figures may be used to investigate less sensitive factors C_3 and ν.



(a) GRALTSVR

(b) SRALTSVR1

(c) SRALTSVR 2

FIGURE 3. 44 Models suggested using a Gaussian kernel and tested on real-world datasets, including Hydraulic Actuator

Friedman statistical test

Additional statistical evaluation of the techniques' prediction accuracy is conducted using the Friedman post hoc test statistic. The Friedman expression may be found using

the following method, which assumes that all processes are identical FF and involves looking at Table 3.12:

$$x_f^2 = \frac{12\times17}{6\times7}\Big[(3.52941^2 + 4.23529^2 + 4.17647^2 + 3.64706^2 + 1.82353^2 + 3.58824^2) - \left(\frac{6\times7^2}{4}\right)\Big] = 1\ 8.647$$

and $F_F = \frac{16\times18.647}{17\times5-18.647} = 4.4964$

Because FF is more valuable than CV, with a value of (4.4964) □ (2.32872), the Nemenyi test is used to compare interesting approaches with the null hypothesis. The constant of variation (CD) is 1.6613 in both the linear and non-linear cases. Next, some statistical inferences will be drawn from their comparative examination.

i. First, we may look examine how SRALTSVR1 ranks on average compared to SVR, TSVR, and Asy-ν -TSVR by using the variation between these three variables. In this case, the results are (3.52941−1.82353= 1.70588), (4.23529−1.82353= 2.41176), and (4.17647−1.82353= 2.35294), respectively. All three of these values are greater than the threshold of 1.6613, proving that SRALTSVR1 comes out on top.

ii. In addition, it is necessary to confirm that the competing techniques vary from each other; for example, when comparing SRALTSVR1 with GRALTSVR and SRALTSVR2, the maximum average rank is 1.82353, and when comparing 3.58824−1.82353=1.76471, the difference is more than 1.6613. Consequently, SRALTSVR1's prediction performance is sufficient, as opposed to STRATVSVR and SRALTSVR2.

iii) As a last step, consider the low-ranking advised approach (GRALTSVR) and choose one of the high-ranking contrasting ways (SVR) so that the gap between their average rankings exceeds the CD. This proves that the GRALTSVR and SVR algorithms are identical in operation.

iv. A boxplot is shown in Figure 3.45, and Table 3.14 presents the statistical significance of SRALTSVR1 with the crucial difference for all baseline techniques in the Gaussian kernel based on RMSE. These distinguish SRALTSVR1 from similar approaches (like the linear example) that accomplish the same objectives.

TABLE 3. 14 Comparison of Models and URALTSVR with Average RMSE Ranks and Statistically Significant Differences at CD for Real-World Datasets.

Average		Rank differs with	Significant different
Models	rank	SRALTSVR1	at CD (1.6613)
SVR	3.52941	1.70588	√
TSVR	4.23529	2.41176	√
Asy-*v*-TSVR	4.17647	2.35294	√
GRALTSVR	3.64706	1.82353	√
SRALTSVR1	1.82353	0	-
SRALTSVR2	3.58824	1.76471	√

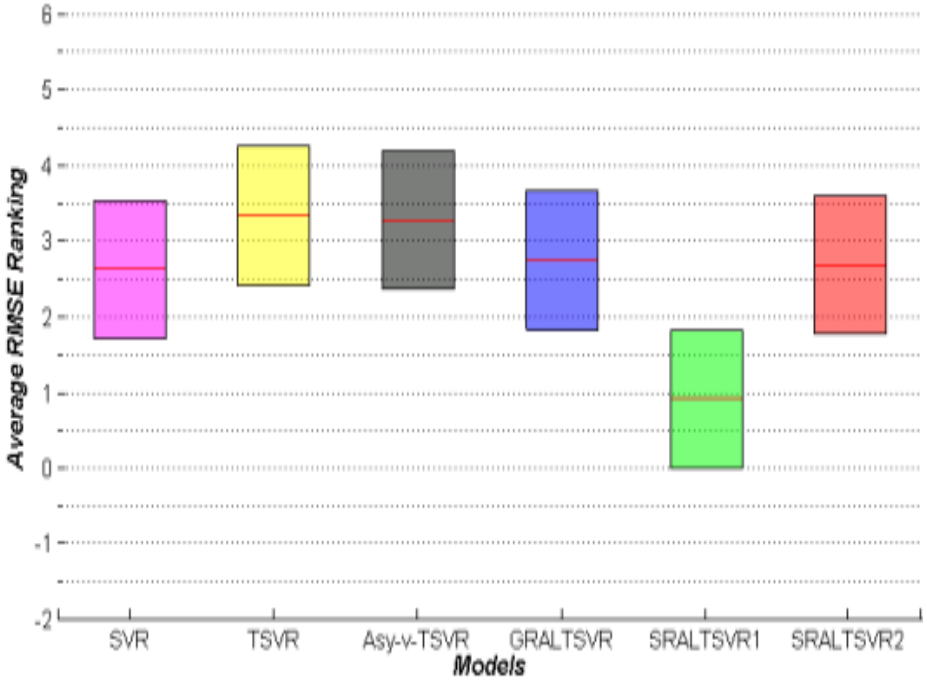


FIGURE 3. 45 Boxplot shows Gaussian kernel-based average rank of models' RMSE on real-world datasets.

Looking at URALTSVR, A robust asymmetric Lagrangian-twin support vector regression issue is solved using gradient-based iterative approaches developed from extended derivative and smoothing procedures and the pinball function. We then solved the problem using Newton iteration. If the settings are right, the asymmetric pinball loss

function in our technique can handle noise-disrupted datasets. Regularization is introduced to the optimization function to guarantee SRM and a stable, well-posed model. Tests on synthetic and real-world datasets show that URALTSVR is acceptable and effective. Comparison of linear and Gaussian kernels in SVR, TSVR, and Asy-□-TSVR reveals that the suggested SRALTSVR1 approach outperforms others. The supplied models have lower or equivalent computing costs as the above techniques.

CHAPTER 4
REGULARIZATION-BASED TWIN SUPPORT VECTOR REGRESSION USING HUBER LOSS AND LEAST SQUARES LARGE MARGIN DISTRIBUTION MACHINE-BASED REGRESSION

CHAPTER 4
REGULARIZATION-BASED TWIN SUPPORT VECTOR REGRESSION USING HUBER LOSS AND LEAST SQUARES LARGE MARGIN DISTRIBUTION MACHINE-BASED REGRESSION

4.1 REGULARIZATION-DRIVEN TWIN SUPPORT VECTOR REGRESSION WITH HUBER LOSS FUNCTION

One of the most important and difficult machine learning research problems is building reliable regression learning models that can fit training data that is contaminated by noise. To make matters worse, the loss function is crucial in mitigating the impact of noise in the training set. For very little mistakes, the Huber loss function acts as a quadratic, whereas for larger ones, it acts as a linear. In this part, we will go over a new huber loss function based regularized twin support vector regression technique.

This function is a hybrid of the Gaussian and Laplace loss functions; it effectively presses the noise of the Gaussian characteristic and suppresses some high noise and outliers. By using the notion of structured risk reduction, our suggested TSVR model is able to overcome the singularity problem and is therefore convex and well-posed.

Several experiments are conducted on various synthetic datasets with uniform, Gaussian, and Laplacian noise, as well as on benchmark datasets from the real world that have varying levels of significant Gaussian noise (0%, 5%, and 10%, respectively), in order to demonstrate the validity and usefulness of the proposed model compared to other models that have been reported.

4.1.1 THE HUBER LOSS FUNCTION

$$c(\lambda_g) = \begin{cases} \frac{\lambda_g^2}{2}, & if \lambda_g \leq \varepsilon \\ \varepsilon|\lambda_g| - \frac{\varepsilon^2}{2}, & otherwise \end{cases} \quad where\ \varepsilon = \varepsilon_1^*$$

where ε^*_1, ε^*_2 are input parameters.

$$c\left(\zeta_{g}\right)=\begin{cases}\dfrac{\zeta_{g}^{2}}{2}, & if\zeta_{g}\leq\ \varepsilon \\ \varepsilon\left|\zeta_{g}\right|-\dfrac{\varepsilon^{2}}{2}, & Otherwise\end{cases} \quad where\ \varepsilon=\varepsilon_{2}^{*}$$

4.1.2 PROPOSED TWIN SUPPORT VECTOR REGRESSION USING REGULARIZATION AND THE HUBER LOSS FUNCTION (RHN-TSVR)

In response, the TSVR provides ε -insensitive omits data with Gaussian noise but preserves loss. In response to this, the author introduced the HN-TSVR method, which deviates from the SRM theory but is compatible with the Huber loss function.

We include a single regularization component to the HN-TSVR to reduce its singularity issue $\frac{C_3}{2}(\|w_1\|^2+b_1^2)\ and\ \frac{C_3}{2}(\|w_1\|^2+b_1^2)\ and\ \frac{C_4}{2}(\|w_1\|^2+b_2^2)$ by resolving the two main difficulties (4.1) and (4.2), we achieve a stable and well-posed model called regularization-based Huber loss-twin support vector regression, which fulfills the core principle of statistical learning theory. The RHN-TSVR formula may be expressed mathematically as: Two kernel generating functions are necessary for RHNTSVR to operate as $f_1(x)=K\ (x^1,\ B^1)\ w_1+b_1$ and $f_2(x)=K\ (x^1,\ B^1)\ w_2+b_2$. The following optimization issues are involved with the suggested method as.

$$\min_{w_1,b_1,\lambda}\frac{1}{2}\|\ y-\varepsilon_1 e-(K(B,B^t)w_1+b_1e)\|^2+C_1e^t(\sum_{g\in\rho_1}\frac{1}{2}\lambda_i^{\ 2}+\varepsilon\sum_{g\in\rho_1}(\lambda_i-\frac{1}{2}\varepsilon))+\frac{C_3}{2}(\|\ w_1\|^2+b_1^{\ 2})$$

Subject to, y-(K (B, B^t) w_1+b_1e) $\geq\varepsilon_1 e-\lambda\geq 0$, (4.1)

And

$$\min_{w_2,b_2,\zeta}\frac{1}{2}\|\ y+\varepsilon_2 e-(K(B,B^t)w_2+b_2e)\|^2+C_2e^t(\sum_{g\in\rho_2}\frac{1}{2}\zeta_i^{\ 2}+\varepsilon\sum_{g\in\rho_2}(\zeta_i-\frac{1}{2}\varepsilon))+\frac{C_4}{2}(\|\ w_2\|^2+b_2^{\ 2})$$

Subject to, y-(K (B, B^t) w_2+b_2e)-y $\geq\varepsilon_2 e-\zeta,\zeta\geq 0$, (4.2)

where,

$$\rho_1=\{g\,|\,0\leq\lambda_g<\varepsilon\},\ \rho_1^{'}=\{g\,|\,\lambda_g\geq\varepsilon\},\ \rho_2=\{g\,|\,0\leq\zeta_g\geq\varepsilon\},\ \rho_2^{'}=\{g\,|\,\zeta_g\geq\varepsilon\};$$

loose variables include of λ,ζ ; the parameters that are entered, C_1, $C_2 > 0$;$\varepsilon_1,\varepsilon_2 > 0$. When the Lagrangian multipliers are included α_1,α_2,β ,β_2 while fixing enough parameters in the KKT equations in (4.1) and (4.2)

$$L_1(w_1,b_1,\alpha_1,\beta_1)=\frac{1}{2}\|(y-\varepsilon_1 e-(K(B,B^t)w_1+b_1e))\|^2+C_1e^t(\sum_{g\in\rho_1}\frac{1}{2}\lambda_g{}^2+\varepsilon\sum_{g\in\rho_1}(\lambda_g-\frac{1}{2}\varepsilon))$$

$$+\frac{C_3}{2}(\|w_1\|^2+b_1{}^2)-\alpha_1{}^t(y-\varepsilon_1 e-(K(B,B^t)w_1+b_1e)+\lambda)-\beta_1{}^t\lambda \quad (4.3)$$

$$L_2(w_2,b_2,\alpha_2,\beta_2)=\frac{1}{2}\|(y+\varepsilon_2 e-(K(B,B^t)w_2+b_2e))\|^2+C_2e^t(\sum_{g\in\rho_2}\frac{1}{2}\zeta_g{}^2+\varepsilon\sum_{g\in\rho_2}(\zeta_g-\frac{1}{2}\varepsilon))$$

$$+\frac{C_4}{2}(\|w_2\|^2+b_2{}^2)-\alpha_2{}^t(y+\varepsilon_2 e-(K(B,B^t)w_2+b_2e)+\zeta)-\beta_2{}^t\zeta \quad (4.4)$$

Next, we determine the gradient of equation (4.3) with respect to w1, b1, and λ :

$$\frac{\partial L_1}{\partial w_1}=-K(B,B^t)^t\,(y-K(B,B^t)w_1-b_1e-\varepsilon_1e)+K(B,B^t)^t\alpha_1+C_3w_1^t=0,$$

$$\frac{\partial L_1}{\partial b_1}=-e^t(y,\varepsilon_1e-K(B,B^t)w_1-b_1e)+e^t\alpha_1+c_3b_1=0,$$

$$\frac{\partial L_1}{\partial b_1}=C_1V_g-\alpha_{1g}-\beta_{1g}=0$$

Where

$$v_g=\frac{\partial(c(\lambda_g))}{\partial(\lambda_g)}=\begin{cases}\lambda_g & if\ g\in\rho_1\\ \varepsilon & if\ g\in\rho_1'\end{cases},\ for\ g\in\rho_1.$$

Presented below are, $\lambda_g < \varepsilon$ g thus $v_g \le \varepsilon$. Also, $\beta_{1g} \ge 0$, in order to get $0\le\alpha_{1g} \le C_1V_g$. Hence, it is safe to say that $0\le\alpha_{1g} \le C_1\varepsilon$.

Applying the same logic as in (4.3), determine the value of w2, b2, and ζ the gradient of equation (4.4):

$$\frac{\partial L_2}{\partial W_2} = -K(B, B^t)^t(y - K(B, B^t)W_2 - B_2e + \varepsilon_2 e) + K(B, B^t)^t\alpha_2 + C_4 w_2^t = 0$$

$$\frac{\partial L_2}{\partial W_2} = -e^t(y + \varepsilon_2 e - K(B, B^t)w_2 - b_2 e + e^t\alpha_2 + C_4 w_2^t = 0$$

$$\frac{\partial L_2}{\partial W_2} = C_2 v_g - \alpha_{2g} - \beta_{2f} = 0$$

Where

$$v_g' = \frac{\partial(c(\zeta_g))}{\partial(\zeta_g)} = \begin{cases} \zeta_g & if\ i \in \rho_2 \\ \varepsilon & if\ i \in \rho_2' \end{cases}, for\ i \in \rho_2 .$$

Here, we have, $\zeta_g < \varepsilon$ thus $V_g \leq \varepsilon$. Also, $\beta_{2g} \geq 0$ g, then we can get $0 \leq \alpha_{2g} \leq C_2 V_g$. Therefore, we can conclude that $0 \leq \alpha_{2g} \leq C_2\varepsilon$.

The two-part version of equation (4.1) follows the same procedure, and it reads as

$$\min_{\alpha_1} \frac{1}{2}\alpha_1^t D_1(D_1^t D_1 + C_3 I)^{-1} D_1^t \alpha_1 - (y - \varepsilon_1 e)^t D_1(D_1^t D_1 + C_3 I)^{-1} D_1^t \alpha_1 + (y - \varepsilon_1 e)^t \alpha_1 + \frac{1}{2C_1}\alpha_1^t \alpha_1$$

Subject to. $0 \leq \alpha_1 \leq C_1 \varepsilon_1^* e$ (4.5)

where, D_1=[K(B, B^t)e]is the augmented matrix.

Similar to above, we get the dual formulation of (4.2) as

$$\min_{\alpha_2} \frac{1}{2}\alpha_2^t D_1(D_1^t D_1 + C_4 I)^{-1} D_1^t \alpha_2 - (y + \varepsilon_2 e)^t D_1(D_1^t D_1 + C_4 I)^{-1} D_1^t \alpha_2 + (y + \varepsilon_2 e)^t \alpha_2 + \frac{1}{2C_2}\alpha_2^t \alpha_2$$

Subject to. $0 \leq \alpha_2 \leq C_2 \varepsilon_2^* e$ (4.6)

The values of W_1 W_2 b_1, b_2 can be obtained as

$$\vartheta_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = (D_1^t D_1 + C_3 I)^{-1} D_1^t (u_1 - \alpha_1), \text{ and } \vartheta_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (D_1^t D_1 + C_4 I)^{-1} D_1^t (u_2 + \alpha_2). \quad (4.7)$$

where $u_1 = y = -\varepsilon_1 e$, $u_2 = y + \varepsilon_2 e$.

In order to get the final regressor value, a fresh test sample might be used to average functions $f_1(x)$ and $f_2(x)$

$$f(x) = \frac{f_1(x) + f_2(x)}{2}. \tag{4.8}$$

4.1.3 NUMERICAL EXPERIMENTS

Here we provide the results of a series of tests that compared RHN-TSVR to more conventional methods like SVR and TSVR, ε -AHSVR, ε -SVQR, and HN-TSVR on distinct artificial datasets with different kinds of noise, and on actual datasets with distinct degrees of significant noise (0.0, 0.05, and 0.10, respectively). A Windows 10 PC with 4 GB of RAM and a single CPU will do the trick, a high-speed 64-bit processor (such as an Intel Core i5 3.20 GHz), and the MATLAB program to conduct this experiment. This research takes a nonlinear approach by looking at the Gaussian kernel function as $K(\chi_{z1}, \chi_{z2}) = exp(-\mu\|\chi_{z1}, -\chi_{z2}\|^2), for\ Z_1Z_2 = 1, \ldots, p,$ in this case the kernel parameter $\mu > 0$. All of the parameters and their limitations for the algorithms in the problem are summarized in Table 4.1. All noteworthy algorithms undergo 10-fold cross-validation on both real-world and artificially produced datasets.

TABLE 4. 1 Parameter Ranges and Associated Algorithms in RHN-TSVR.

Parameters	Range	Model
C, C_1=C_2, C_3=C_4	{10^{-5},.....10^{5}}	SVR, TSVR, ε -AHSVR, ε - SVQR, HN-TSVR, RHN-TSVR
μ	{0.1}	SVR
ε	{0.001,0.01, 0.1, 0.3, 0.5, 0.7, 0.9}	TSVR
(ε_α =ε_β)	{0.001,0.01, 0.1}	ε -AHSVR
ε	{0.1, 0.3, 0.5, 0.7, 0.9}	ε -SVQR
(ε_1=ε_2), (ε_1^* =ε_2^*)	{0.1, 0.3, 0.5, 0.7, 0.9}	HN-TSVR, RHN-TSVR
ζ_α,ζ_β,	{0.1, 1.0, 1.345}	ε -AHSVR
θ	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,0.8}	ε -SVQR

Artificial Dataset

a) Gaussian and homogeneous noise

The first fourteen functions employ symmetrical, evenly distributed noise, whereas functions fifteen through eighteen use a heteroscedastic noise structure, where the

initial sample value determines the noise. Table 4.2 displays the average results on synthetic samples for all applicable approaches for the Gaussian kernel. The recommended methodology RHNTSVR has the poorest model, whereas our method performs better than the competitors (Table 4.2). Out of the eighteen artificial functions, RHN-TSVR performed the best while dealing with both uniform and Gaussian noise. It also significantly affects the heteroscedastic and homogeneous noise patterns' variability. The results of the synthetic Functions 13 and 14 are in agreement with the symmetrical, uniformly dispersed noise shown in Figures 4.1 and 4.2. Figure 4.3 displays the predictions made using a normal distribution kernel for Fake Function 15, whereas Figure 4.4 displays the predictions made using a distribution with heteroscedastic noise. Figures 4.1–4.4 demonstrate that RHN-TSVR closely resembles the previous technique, in contrast to the other methodologies that were presented.

TABLE 4. 2 Evaluate RHN-TSVR and Other Gaussian Kernel Models on Artificial Datasets with Uniform and Gaussian Noise Using Root Mean Squared Error (RMSE): An Analysis of Average Rank

Datasets	SVR	TSVR	ε - AHSVR	ε - SVQR	HN-TSVR	RHN-TSVR
Function1	1	2	5	4	3	6
Function2	6	4	2	5	3	1
Function3	4	5	2	3	6	1
Function4	6	4	2	5	3	1
Function5	6	2.5	1	5	2.5	4
Function6	6	3	2	5	4	1
Function7	5	3	4	6	2	1
Function8	6	3.5	2	5	3.5	1
Function9	5	1	4	6	2.5	2.5
Function10	6	4	2	5	3	1
Function11	5	1	4	6	3	2
Function12	6	4	1	5	3	2
Function13	6	5	1	3	4	2
Function14	6	2	5	4	3	1
Function15	6	3	4	5	1.5	1.5
Function16	6	2	5	3	4	1
Function17	3	4.5	2	6	4.5	1
Function18	6	3	2	1	4.5	4.5
Average	5.27778	3.1388889	2.7777778	4.5555556	3.3333333	1.9166667
rank						

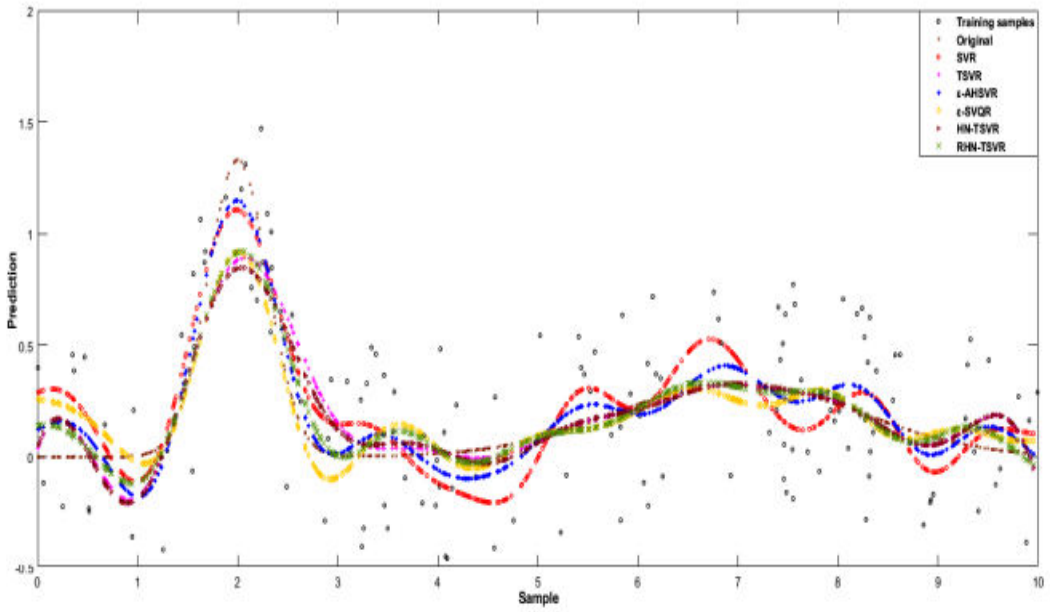


Figure 4. 1 Prediction Results on on the Function 13 Artificial Dataset Utilizing the Gaussian Kernel and Other Previously Announced Models

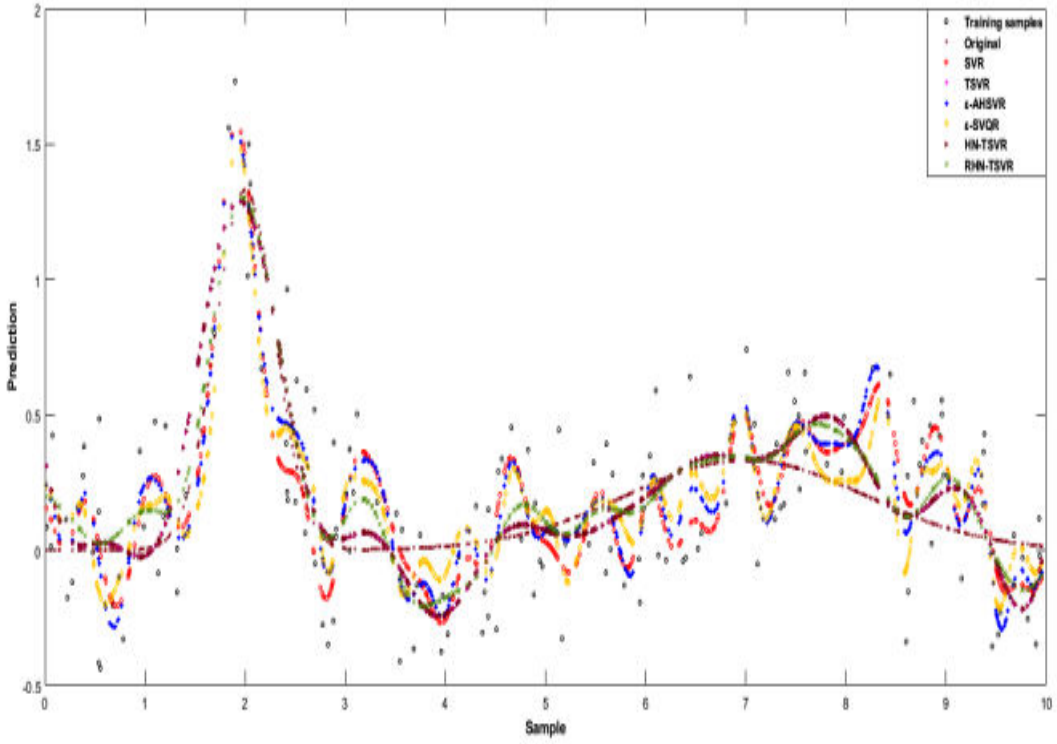


Figure 4. 2 Prediction Results on RHN-TSVR and Other Previously Announced Models for the Function 14 Gaussian Kernel Artificially Generated Dataset have been reviewed.

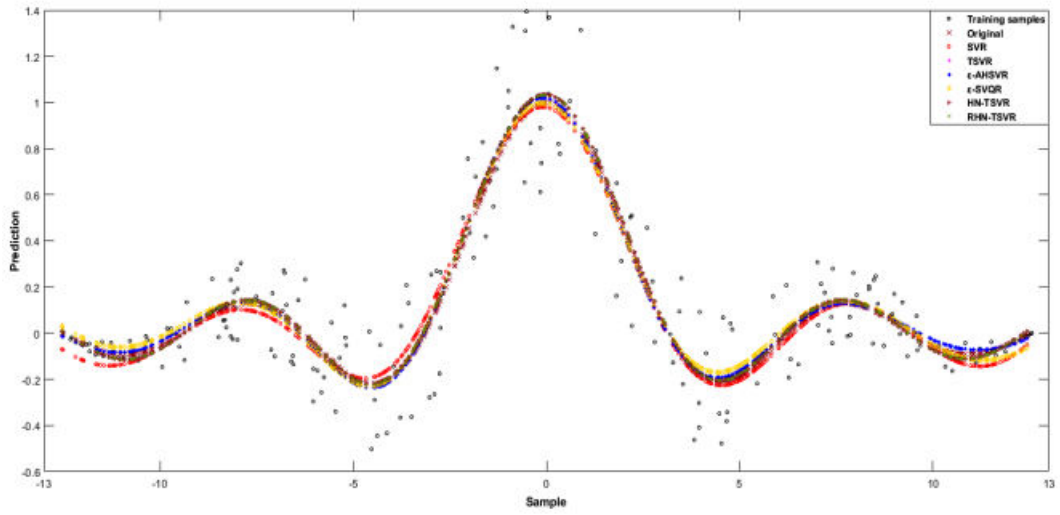


Figure 4. 3 Prediction Results on The Gaussian Kernel-Generated Function 15 Testing Dataset using RHN-TSVR and Other Reported Models.

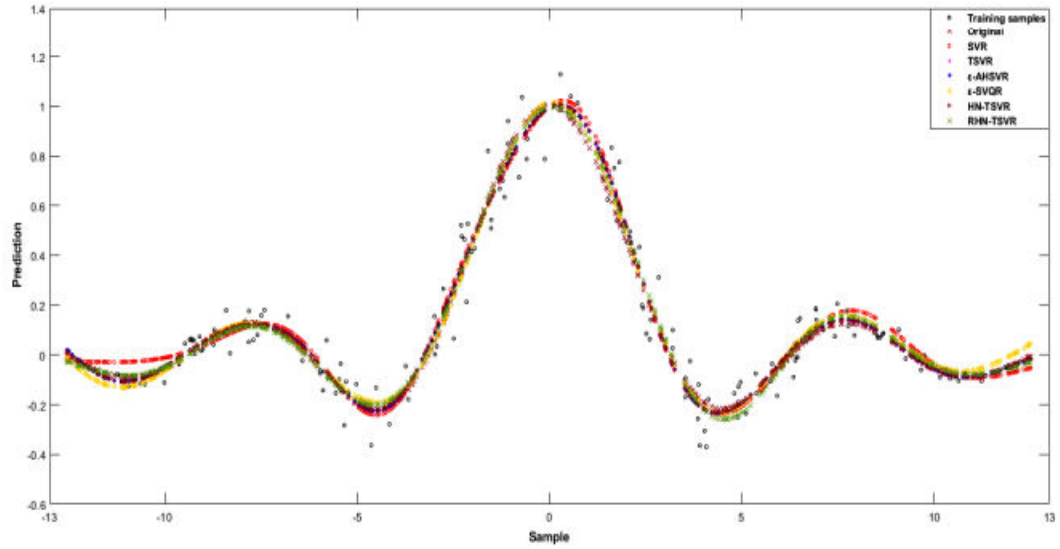


Figure 4. 4 Prediction Results on the Gaussian Kernel-Generated Function 16 Testing Dataset by RHN-TSVR and Other Reported Models.

b) Laplacian noise

In order to evaluate our RHNTSVR method in comparison to SVR and TSVR, Table 4.3 provides a description of the features of HN-TSVR and many similar algorithms. Additionally, we generate synthetic datasets that include a certain kind of noise, namely Laplacian noise, ε -AHSVR, and ε -SVQR. The interval [0,1] $\Psi \in L(\mu,b)$ is used to investigate a Laplacian noise.

TABLE 4. 3 Various man-made functions with Laplacian noise and associated RHN-TSVR definitions

Function name	Function definition	Domain of definition
Function19	$f(x)=\left(\frac{4}{\lvert x_1\rvert+2}\right)+\cos(2x_1)+\sin(3x_1)+\Psi$	$x_1\in[-10,10]$
Function20	$f(x)=((1+\sin(2x_1+3x_2))/(3.5+\sin(x_1-x_2)))+\Psi$	$x_i\in[-2,2]$ $i\in\{1,2\}$
Function21	$f(x_1,x_2)=\exp(x_1\sin(\pi x_2))+\Psi$	$x_1,x_2\in[-1,1]$
Function22	$f(x)=0.02[(12+3x-3.5x^2+7.2x^3)(1+\cos 4\pi x)(1+0.8\sin 3\pi x)]+\Psi$	$x\in[-0.25,0.25]$
Function23	$f(x_1,x_2,x_3,x_4,x_5)=10\sin\pi x_1x_2+20(x_3-0.5)^2+10x_4+5x_5+\Psi$	$x_i\in[0,1]$ $i=1,2,3,4,5$
Function24	$f(x)=0.2\sin(2\pi x)+0.2x^2+0.3$ such that $y_i=f(x_i)+(0.1x_i^2+0.05)\Psi_i$	$x_i=0.01(i-1)-1,$ $i=1,2,\ldots,200$

The proposed RHN-TSVR beats the alternatives in four of the six cases. The average ranks of all the models are also calculated in Table 4.4. The RHN-TSVR has the most features, but we do a lot more. The great level of agreement between the actual and projected values is seen in Figure 4.5, which is related to Function 19. In this case, RHN-TSVR meets or exceeds HN-TSVR in terms of performance.

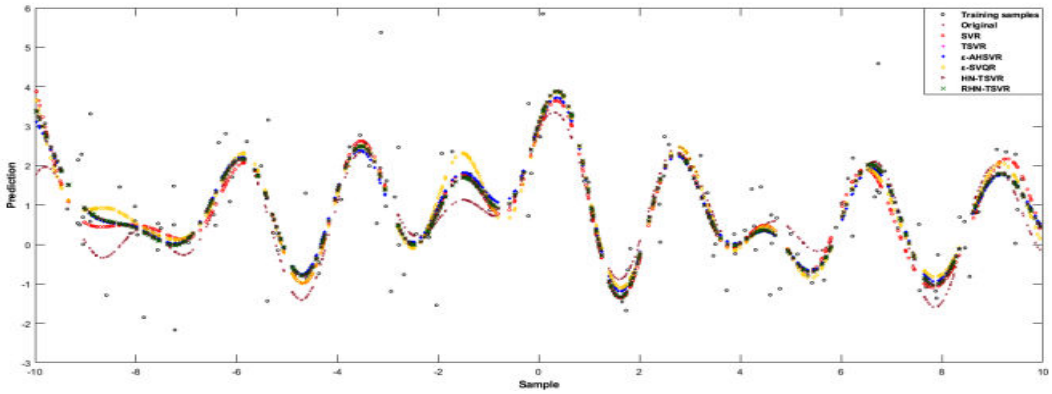


Figure 4. 5 Prediction Results on Function 19 Artificial Dataset Using Gaussian Kernel Testing Dataset using RHN-TSVR and Other Reported Models.

TABLE 4. 4 Average RMSE Ranks of Gaussian Kernel Models for Synthetic Datasets with Laplacian Noise: RHN-TSVR and Others

Datasets	SVR	TSVR	ε - AHSVR	ε - SVQR	HN-TSVR	RHN-TSVR
Function19	5	3	4	6	1.5	1.5
Function20	2	4	6	3	5	1
Function21	2	5	3	6	4	1
Function22	3	4	1	5	6	2
Function23	2	3	5	6	4	1
Function24	5	4	6	1	3	2
Average rank	3.1666667	3.8333333	4.1666667	4.5	3.9166667	1.4166667

Real-world Datasets

We tested RHN-TSVR on 42 real-world datasets with varying levels of statistical significance (0.0, 0.05, and 0.10) to see how well it performed.

At significant noise level 0%

It includes the prediction power, ideal parameter values, time needed to learn each of the techniques offered. When comparing TSVR against SVR, ε -AHSVR, ε -SVQR, RHN-TSVR achieves better results than both HN-TSVR and on 22 real-world datasets. Table 4.5 also shows the results of the statistical test, which is based on averaging the rankings with the RMSE values for the Gaussian kernel. Overall, RHN-TSVR rates the

lowest when compared to the other approaches that have been mentioned. As shown by real-world datasets with a considerable noise level of 0%, RHN-TSVR performs better. Figures 4.6 and 4.7 demonstrate the predicted accuracy for the Machine CPU dataset and the Gas furnace dataset, respectively. In terms of prediction abilities, both plots show that RHN-TSVR is superior than its rivals.

TABLE 4. 5 Evaluation of RHN-TSVR in relation to competing models using root-mean-squared error metrics on a real-world dataset free of noise, trained with a Gaussian kernel.

Datasets	SVR	TSVR	ε-AHSVR	ε-SVQR	HN-TSVR	RHN-TSVR
Forestfires	5.5	2.5	5.5	1	2.5	4
Machine CPU	4	2	6	5	3	1
Auto-original	6	5	1	2	3.5	3.5
Winequality	3	4	1	2	6	5
SantafeA	5	2	6	4	3	1
Gas_furnace	5	3	6	4	2	1
Quake	6	4	1	5	2	3
Flex_robotarm	5	3.5	6	2	3.5	1
S&P500	5	3	6	1	4	2
Space-Ga	5	3.5	6	1	3.5	2
Gauss1	6	2	5	4	1	3
Chwirut2	5	1	6	4	2	3
Roszman1	5	2	3	4	1	6
INFY	6	2	5	4	3	1
ONGC_NS	4	3	6	2	5	1
XOM	5	3	6	2	4	1
ATX	5	3	6	2	4	1
BSESN	6	3	5	1	4	2
DJI	5	3	6	2	4	1
GDAXI	3	2	6	5	4	1
MXX	2	5	3	4	6	1
N225	5	3	6	1	4	2
Wankara	5	2	6	4	3	1
Laser	5	1	6	4	2	3
Dee	3	5	4	1	6	2
Friedman	5	3	6	4	2	1
Mortgage	5	1	6	4	2	3
NNGC1_dataset_E1_V1_001	6	3	1	5	4	2
NNGC1_dataset_F1_V1_008	6	3	5	4	1.5	1.5

NNGC1_dataset_F1_V1_009	6	3	5	4	1.5	1.5
NNGC1_dataset_F1_V1_010	6	1	5	4	2.5	2.5
NNGC1_dataset_F1_V1_006	5	3	6	4	1.5	1.5
NN5_Complete_109	5	4	6	2	3	1
NN5_Complete_104	6	3	4	5	2	1
NN5_Complete_106	6	4	2	5	3	1
NN5_Complete_103	2	5	6	1	4	3
NN5_Complete_101	3	5	4	2	6	1
NN5_Complete_105	2	5	4	3	6	1
NN5_Complete_111	3	4	2	6	5	1
D1dat_1_2000	1	5	2	3	6	4
Vineyard	1	6	3	2	5	4
COVID-19_spain	5.5	1	5.5	4	3	2
Average rank	4.5952381	3.1309524	4.6666667	3.1666667	3.4285714	**2.0119048**

Table 4.5 shows the average algorithm ranks; the next statistical step is to apply the Friedman test to these rankings.

This includes all of the given algorithms, such as SVR and TSVR, ε -AHSVR, ε -SVQR, RHN-TSVR and HN-TSVR, which may be used interchangeably. Here, we must now calculate both 2 F χ and F_F according to Table 4.5, as seen below:

$$\chi_F^2 = \frac{12\times 42}{6\times 7}\left[\left(4.595238^2 + 3.130952^2 + 4.666667^2 + 3.166667^2 + 3.428571^2 + 2.011905^2\right) - \left(\frac{6\times 7^2}{4}\right)\right]$$

$\chi_F^2 = 60.3299$

And $F_F = \frac{(42-1)x60.3299}{(42x6-1))-60.3299} = 16.5265$

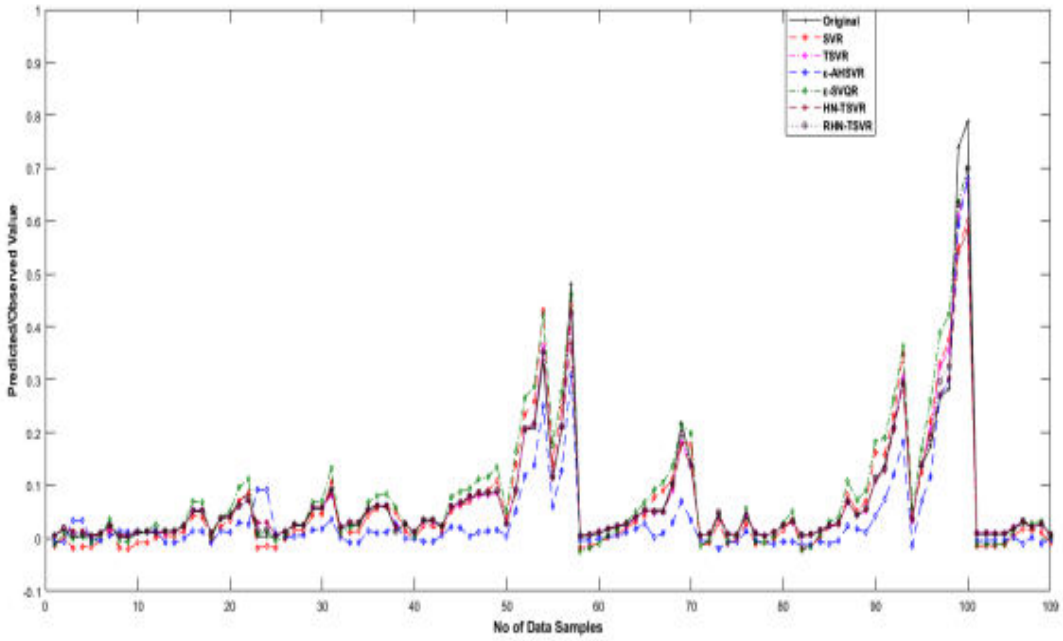


Figure 4. 6 A noise-free machine CPU dataset that has been tested with the RHN-TSVR and other published models.

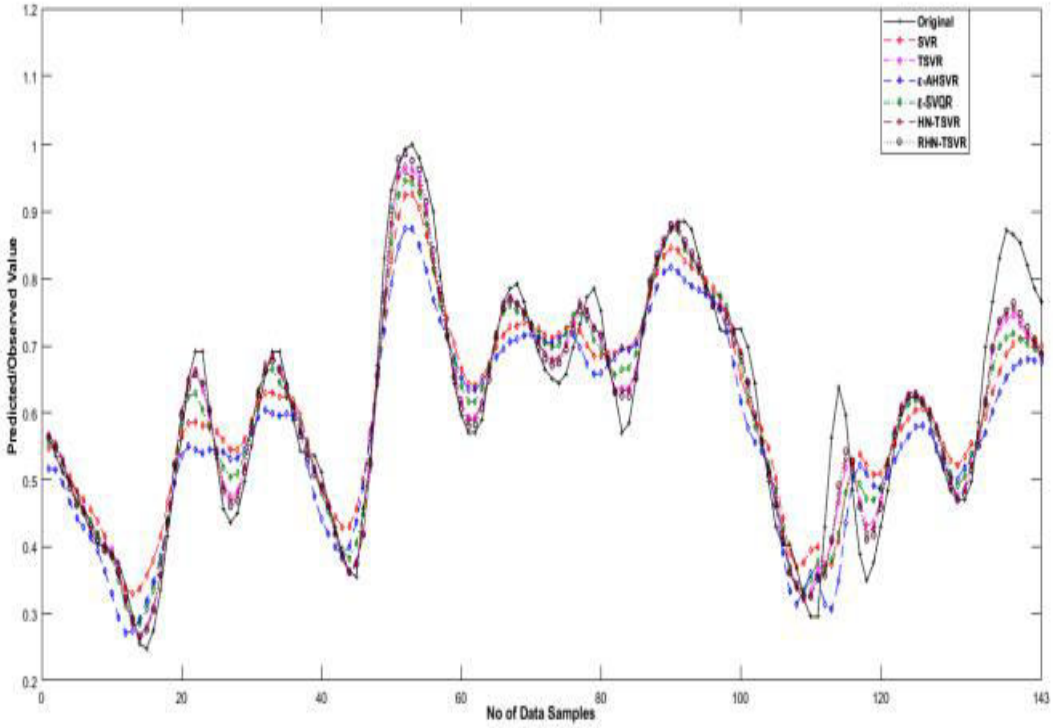


Figure 4. 7 Results from the testing dataset using a Gaussian kernel and the Gas Furnace dataset with zero noise, as predicted by RHN-TSVR and other published models.

The probability level degree of freedom, which is (5, 205), is the most crucial number here. Φ = 0.05 is less in size than F_F (16.5265 > 2.2581).

Thus, the paired Nemenyi test is continued after rejecting the null hypothesis H0. The next step is to determine the statistically significant critical difference (p = 0.10) by

Critical difference = $q_a\sqrt{\frac{kx(k+1)}{6xN}} = 2.589\sqrt{\frac{6x(6+1)}{6x42}} = 1.057,$

where k is the count of published algorithms and N is the count of datasets, and q_α = 2.589. The results of this Nemenyi test are as follows:

Compare the rank of RHN-TSVR with that of SVR, TSVR, ε -AHSVR, and ε -SVQR, while considering the proposed method.

Since the differences above the 1.057 threshold, RHN-TSVR, ε-AHSVR, and ε -SVQR are better than SVR TSVR. For that reason, it is reasonable to expect our suggested RHNTSVR to work.

ii) Compare the average ranks of RHN-TSVR with HN-TSVR. RHN-TSVR is better than HN-TSVR since there is a larger gap than that (1.416667 ε 1.057).

At significant noise level 5%

The RHN-TSVR model was the most prominent in 18 out of all the situations. In Table 4.6, you can see the current rankings of all the models based on the RMSE values that were used in the statistical analysis.

Table 4.6 shows that RHN-TSVR is at the bottom of the list. And yet, for noisy datasets, RHN-TSVR seems to be a good bet.

The prediction value plot for the Gas Furnace dataset is shown in Figure 4.9 at a significance level of 0.05, whereas the plot for the Machine CPU dataset is shown in Figure 4.8. According to this condensed study, RHN-TSVR is more strongly associated with the target result.

TABLE 4. 6 Rankings of competing RHN-TSVR models based on Results of running the RMSE test on a real-world dataset using a 5% noise Gaussian kernel

Datasets	SVR	TSVR	ε-AHSVR	ε-SVQR	HN-TSVR	RHN-TSVR
Forestfires	4	5	3	1	6	2
Machine CPU	3	2	6	5	4	1
Auto-original	1	5	3	4	6	2
Winequality	2	5	3	6	4	1
SantafeA	6	3	1	5	4	2
Gas_furnace	6	3	4	5	2	1
Quake	5	2	4	6	3	1
Flex_robotarm	3	4	6	2	5	1
S&P500	5	1	6	4	3	2
Space-Ga	4	5	3	1	6	2
Gauss1	4	2	6	5	3	1
Chwirut2	3	6	2	1	5	4
Roszman1	4	3	5	6	1	2
INFY	5	3	6	4	1.5	1.5
ONGC_NS	5	4	6	2	3	1
XOM	5	4	6	1	2.5	2.5
ATX	5	4	6	1	2	3
BSESN	5	3	6	4	2	1
DJI	5	3	4	6	1.5	1.5
GDAXI	5	4	6	1	3	2
MXX	5	3	6	4	2	1
N225	6	2	5	4	1	3
Wankara	6	1	5	4	3	2
Laser	6	3	4	5	2	1
Dee	4	5	1	3	6	2
Friedman	5	3	6	2	4	1
Mortgage	6	2	4	5	3	1
NNGC1_dataset_E1_V1_001	1	4	3	6	5	2
NNGC1_dataset_F1_V1_008	6	3	5	4	1.5	1.5
NNGC1_dataset_F1_V1_009	5	2	6	4	3	1
NNGC1_dataset_F1_V1_010	6	1	5	4	2	3
NNGC1_dataset_F1_V1_006	5	1	6	4	2	3
NN5_Complete_109	3	5	1	4	6	2
NN5_Complete_104	6	5	1	3	4	2

DJI	5	3	4	6	1.5	1.5
GDAXI	5	4	6	1	3	2
MXX	5	3	6	4	2	1
N225	6	2	5	4	1	3
Wankara	6	1	5	4	3	2
Laser	6	3	4	5	2	1
Dee	4	5	1	3	6	2
Friedman	5	3	6	2	4	1
Mortgage	6	2	4	5	3	1
NNGC1_dataset_E1_V1_001	1	4	3	6	5	2
NNGC1_dataset_F1_V1_008	6	3	5	4	1.5	1.5
NNGC1_dataset_F1_V1_009	5	2	6	4	3	1
NNGC1_dataset_F1_V1_010	6	1	5	4	2	3
NNGC1_dataset_F1_V1_006	5	1	6	4	2	3
NN5_Complete_109	3	5	1	4	6	2
NN5_Complete_104	6	5	1	3	4	2
NN5_Complete_106	3	6	4	1	5	2
NN5_Complete_103	1	4	6	3	5	2
NN5_Complete_101	3	5	6	1	4	2
NN5_Complete_105	4	2	5	6	3	1
NN5_Complete_111	1	5	6	2	4	3
DIdat_1_2000	5	2	6	4	3	1
Vineyard	4	3	6	1	2	5
COVID-19_spain	6	5	2	1	4	3
Average rank	4.333333	3.404762	4.547619	3.4523809	3.380952	1.880952

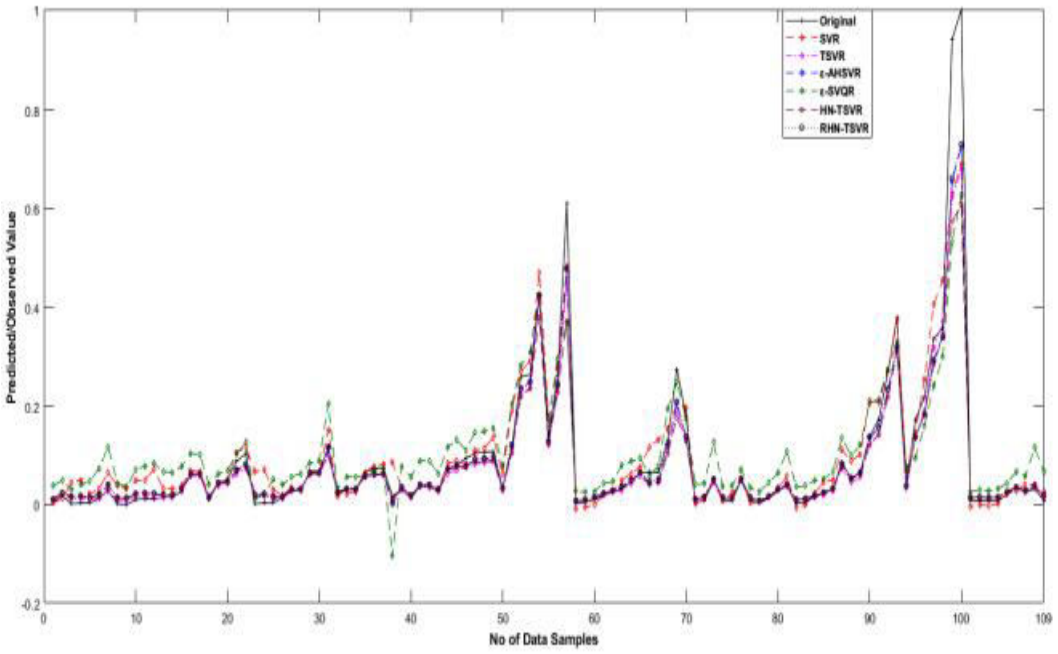


Figure 4. 8 RHN-TSVR and other models employed a Gaussian kernel for testing set prediction on the Machine CPU dataset with 5% noise.

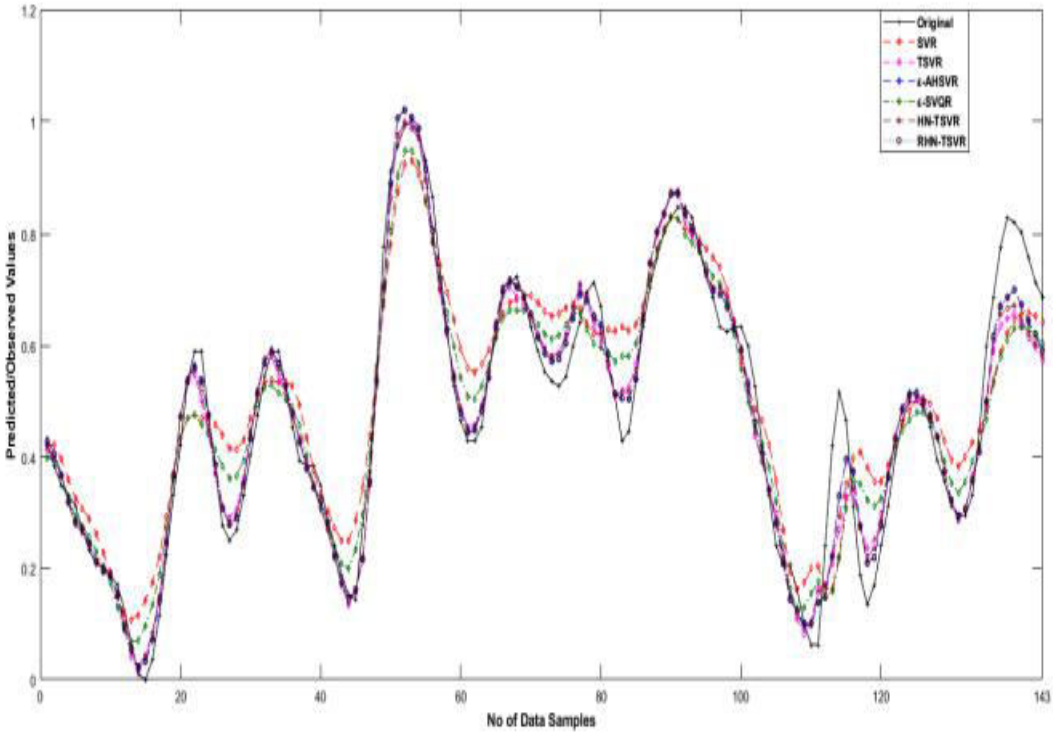


Figure 4. 9 SVR, TSVR, ε -AHSVR, ε -SVQR, HN-TSVR, and RHN-TSVR use a Gaussian kernel for Gas Furnace dataset prediction with 5% noise on the testing dataset.

The suggested method also Table 4.6 shows the results of a Friedman test comparing RHNTSVR's performance on noisy data using SVR and TSVR, with a significance threshold of 5%., ε -AHSVR, and ε -SVQR with HN-TSVR.

$$\chi_F^2 = \frac{12\times 42}{6\times 7}\left[\left(5.047619^2 + 3.2619048^2 + 3.2142857^2 + 3.833333^2 + 3.547619^2 + 2.0952381^2\right) - \left(\frac{6\times 7^2}{4}\right)\right]$$

$= 55.442.$

And

$$F_F = \frac{(42-1)x52.2653}{(42x(6-1))-52.2653} = 13.9336$$

With fewer degrees of freedom (5, 205) and a lower critical value, FF, this situation is ideal (13.9336> 2.2581). Consequently, we test the null hypothesis and do a paired test.

Find the essential difference with a significance level of p = 0.10 in order to run the Nemenyi test. Just as in the previous case, the crucial difference is 1.057. Here are a few things to think about:

i. When comparing RHN-TSVR to SVR, TSVR, ε -AHSVR, and ε -, the value of SVQR is always greater than 1.057. Consequently, RHN-TSVR stands out as the superior option.

ii. There is a larger discrepancy than the essential difference in 3.380952-1.889052=1.5 is the average rank of RHN-TSVR and HN-TSVR., (1.5 > 1.057). It asserts that compared to HN-TSVR, RHN-TSVR is the better method.

Under very loud conditions, 10%

Using real-world datasets, we put the new RHN-TSVR through its paces in a loud environment with a major noise level of 5% and raised the significant noise level to 10%. All of the presented methods are anticipated to rank lower than RHN-TSVR, as shown in Table 4.7.

Prediction graphs for the Machine CPU and Gas furnace datasets with a significant 10% noise level are shown in Figures 4.10 and 4.11, respectively, as in previous cases. Both graphs might lead to the same conclusion.

TABLE 4. 7 Based on RMSE values, the average ranked models and RHN-TSVR utilizing a Gaussian kernel with 10% noise for a real-world dataset.

Datasets	SVR	TSVR	ε - AHSVR	ε - SVQR	HN-TSVR	RHN-TSVR
Forest fires	4	5	2	1	6	3
Machine_8	5	1	4	6	2.5	2.5
Auto-original	6	2	4	5	3	1
Win equality	4	1	5	6	2	3
SantafeA	6	3	4	1	5	2
Gas_furnace	6	4	2	5	3	1
Quake	3	1	5	4	2	6
Flex_robotarm	6	5	1	3	4	2
S&P500	5	3	2	6	4	1
Space-Ga	6	4	3	1	5	2
Gauss1	5	3	1	6	4	2
Chwirut2	3	5	2	4	6	1
Roszman1	6	3	4	5	2	1
INFY	6	3	5	1	2	4
ONGC_NS	6	3	1	5	4	2
XOM	6	4	2	5	3	1
ATX	2	1	5	6	4	3
BSESN	5	6	3	2	4	1
DJI	6	2	5	3	1	4
GDAXI	6	3	2	5	4	1
MXX	3	2	5	1	4	6
N225	6	4	3	5	1.5	1.5
Wankara	6	1	4	5	2	3
Laser	6	3	2	5	4	1
Dee	6	3	1	5	4	2
Friedman	6	2	5	1	3.5	3.5
Mortgage	6	1	4	5	2	3
NNGC1_dataset_E1_V1_001	6	4	2	3	5	1
NNGC1_dataset_F1_V1_008	6	4	3	5	1.5	1.5

NNGC1_dataset_F1_V1_009	6	3	2	5	4	1
NNGC1_dataset_F1_V1_010	6	3	2	5	4	1
NNGC1_dataset_F1_V1_006	6	3	2	5	4	1
NN5_Complete_109	3	5	4	1	6	2
NN5_Complete_104	3	5	2	6	4	1
NN5_Complete_106	3	5	4	1	6	2
NN5_Complete_103	2	4	6	1	5	3
NN5_Complete_101	1	4	3	6	5	2
NN5_Complete_105	5	3	4	6	1	2
NN5_Complete_111	6	5	2	1	4	3
D1dat_1_2000	6	2	5	4	3	1
Vineyard	6	5	3	4	2	1
COVID-19_spain	6	4	5	1	3	2
Average rank	5.047619	3.2619048	3.2142857	3.8333333	3.547619	2.0952381

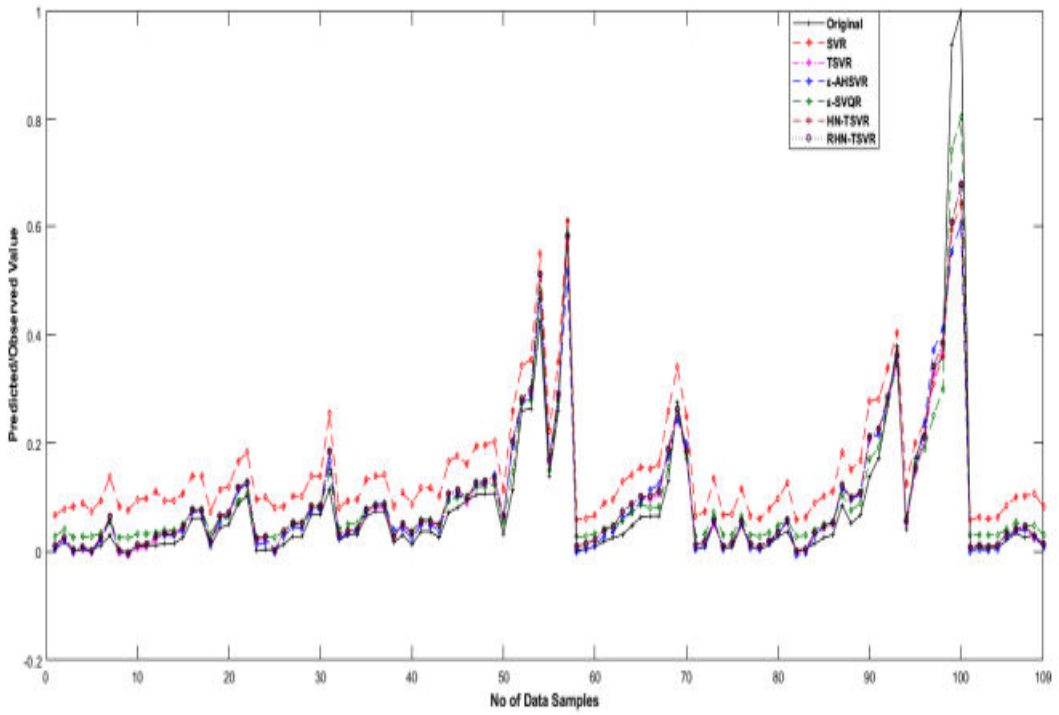


Figure 4. 10 Prediction over the testing dataset by RHN-TSVR and other reported models on the Machine CPU dataset with 10% noise using Gaussian kernel.

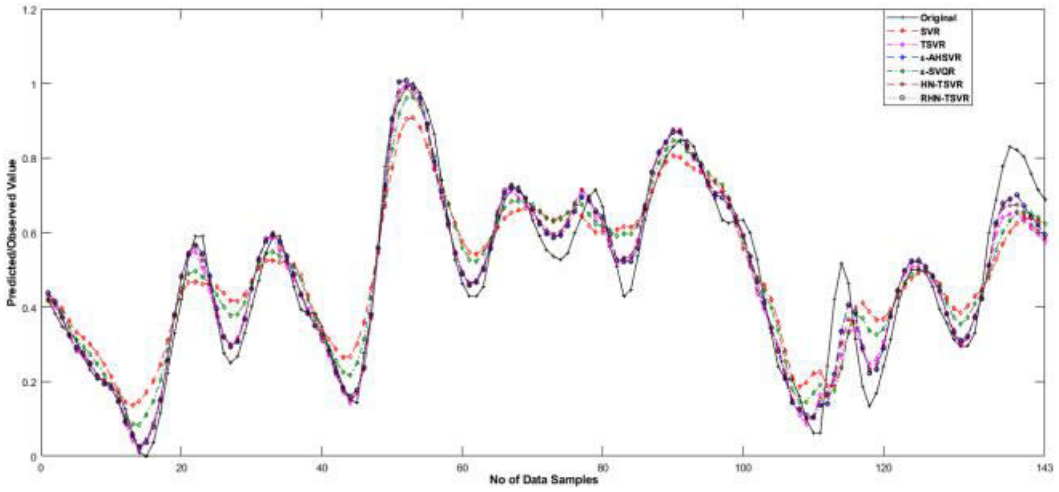


Figure 4. 11 Prediction over the testing dataset by RHN-TSVR and other reported models on the Gas furnace dataset with 10% noise using Gaussian kernel.

Same as to previous cases, compute the values of 2 F χ and FF using Table 4.7 as

$$\chi_F^2 = \frac{12x42}{6x7}\left[(5.047619^2 + 3.2619048^2 + 3.2142857^2 + 3.833333^2 + 3.547619^2 + 2.0952381^2 - \left(\frac{6x7^2}{4}\right)\right]$$

$= 55.442.$

$$F_F = \frac{(42-1)x55.442}{(42x(6-1))-55.442}$$

In this case, FF is likewise larger than the crucial number (14.707 > 2.2581) for the degree of freedom (5, 205).

This scenario likewise rejects the null hypothesis, H0, suggesting that there may be substantial disparities across all of the models. In order to derive various conclusions, let's run the Nemenyi test on these techniques.

Like in the other examples, RHN-TSVR always has an average rank disparity with others that is more than the crucial difference, and it has the lowest average rank when the noise level is 10%. so, 1.057.

The RHN-TSVR has a higher overall effectiveness rating than competing models.

Effect of increasing noise percentage

Here, we showed how a real-world dataset was affected by increasing the noise level by 0.0, 0.50, and 0.10. The next paragraph details our evaluation of the new RHN-TSVR's performance on real-world datasets subjected to varying degrees of substantial noise. At different noise levels, Figure 4.12 shows that the Gas Furnace dataset is quite efficient.

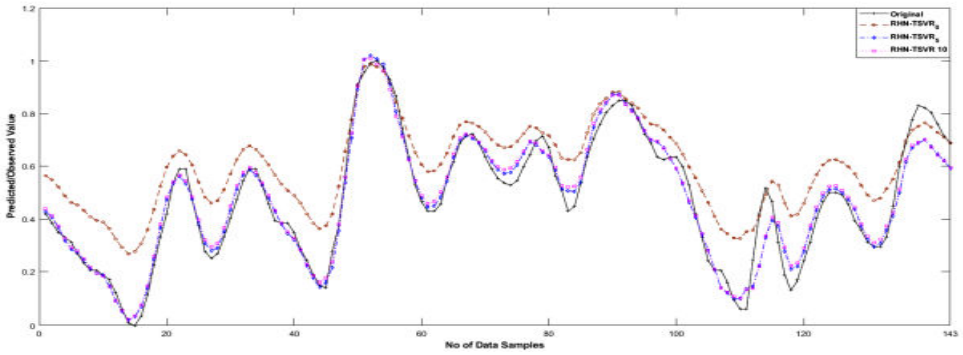


Figure 4. 12 Prediction/Observed Value over the testing dataset by RHN-TSVR on the Gas furnace dataset with 0%, 5% and 10% noise using Gaussian kernel.

A black line represents the instances of testing data, while brown, blue, and pink dotted lines illustrate the RHN-TSVR prediction performance for varying noise levels of 0.0, 0.50, and 0.10, respectively. Figure 4.12 shows the effect of increasing the noise percentage on the proposed RHN-TSVR model. The fact that noisy findings are more closely linked to the intended outcome demonstrates the applicability and dependability of the recommended RHN-TSVR model for noisy contexts.

We suggest a regularized version of TSVR with Huber loss (RHN-TSVR) to address the singularity problem in HNTSVR. This version incorporates the SRM principle and is a regularization-based twin support vector regression with Huber loss. We test the RHN-TSVR's noise insensitivity with different variations of the substantial noise level (0.0, 0.05, and 0.010). The TSVR loss function is □-insensitive, meaning it cannot handle different types of noise or outliers, which is something we can all understand. The basic Huber loss function has a quadratic form for small mistakes and a linear one for bigger ones. The Laplacian loss fusion with the Gaussian loss function provides better prediction performance for data containing Gaussian noise and outliers. We test the proposed method on produced datasets with different kinds of noises for the non-

linear kernel and on real-world datasets with different degrees of importance to see how well it performs. In terms of prediction accuracy and computing time, the proposed RHN-TSVR often surpasses prior methods. A numerical trial-based comparison justifies the RHN-TSVR model's relevance compared to published alternatives, particularly when handling data with noise and outliers. Financial time series forecasting is one area that might benefit from this method. In the future, suggesting an iterative approach might reduce the processing cost.

4.2 LEAST SQUARES LARGE MARGIN DISTRIBUTION MACHINE BASED REGRESSION

Here, we zeroed in on LDM-based regression, a potent subset of regression methods that, unlike SVR, optimizes the distribution of margins rather than minimizing a single point margin. When solving its optimization issue, LDM-based regression takes the margin mean and variation into account. At the same time as minimizing the quadratic loss function, the optimization issue of our suggested model minimizes ε - loss function that is not sensitive.

Therefore, rather than using computationally difficult QPP, the answer is derived from a linear K.K.T. system.

Here we detail the least squares LDM-based regression technique, which reduces data point dispersion while simultaneously increasing robustness against noise and outlier sensitivity. Furthermore, it ε - tube that produces less computational load.

Concurrently avoiding overfitting and making maximum use of the training set are both made possible by our proposed methodology. Through the use of the linear and Gaussian kernels, numerical experiments have been conducted on both synthetically produced datasets and benchmark real-world datasets.

Standard SVR, Twin SVR, and primal least squares are used to analyze all of the experiments of the LS-LDMR that is provided. Dual SVR (PLSTSVR), ε -Huber SVR (ε -HSVR), ε -support vector quantile regression (ε -SVQR), LDMR and minimal deviation regression (MDR), demonstrating the efficacy and practicality of LS-LDMR.

This method has also been confirmed and validated statistically using a number of other criteria.

4.2.1 PROPOSED APPROACH LEAST SQUARES LARGE MARGIN DISTRIBUTION MACHINE-BASED REGRESSION (LS-LDMR)

Keeping with the LDMR concept, we provide a new version of LDMR that uses least squares to reduce computing cost. It is designed for regression-based problems and is called the least squares big margin distribution machine.

Here we develop the LS-LDMR problem formulations using the 2-norm of the slack variable instead of the 1-norm and with equality constraints in the LDMR formulation instead of inequality constraints.

When all that's required to get the answers is to calculate the inverse of the matrix, systems of linear equations are used. Therefore, in contrast to SVR and TSVR, there is no need to resolve the massive size of the QPP, ε - SVQR and LDMR. The issue statement for LS-LDMR is provided as:

Linear LS-LDMR

Linear LS-LDMR model, $f(x)=w^t x+b$ becomes available when the following optimization issue is solved as

$$\min_{w,b,\zeta} \frac{d_2}{2} \| y-(Bw+eb) \|^2 + C\zeta^2 + \frac{d_1}{2} \| w \|^2$$

subject to: $y= (B w+ e b) +e\varepsilon+\zeta$ (4.9)

where ε,d1, d2 > 0 are the input parameters that the user defines; the variables slack and penalty are ζ and C > 0 respectively.

Let us consider, $\vartheta\begin{bmatrix} w \\ b \end{bmatrix}$; $\|w\|^2 = \vartheta^t I_0 \vartheta$ where $I_0 = \begin{bmatrix} I & 0 \\ & : \\ 0 & \ldots 0 \end{bmatrix}$ and p p I × is an identity matrix.

Next, get the Lagrangian functions of equation (4.9) by using the Lagrangian multiplier α.

$$L(\vartheta,\zeta,\alpha)=\frac{d_2}{2}\| y-(Bw+eb)\|^2+C\zeta^2+\frac{d_1}{2}\| w\|^2-\alpha^t\big((Bw+eb)+e\varepsilon+\zeta-y\big).$$

It is also possible to rephrase the above equation as

$$L(\vartheta,\zeta,\alpha)=\frac{d_2}{2}\| y-D_1\vartheta\|^2+C\zeta^2+\frac{d_1}{2}\| w\|^2-\alpha^t\big(D_1\vartheta+e\varepsilon+\zeta-y\big), \tag{4.10}$$

where $D_1 = [B\ e]$.

By taking into account ϑ, ζ, and α and setting them equal to zero, we may get the gradient of (4.10).

$$\frac{\partial L(\vartheta,\zeta,\alpha)}{\partial\vartheta}=d_1I_0^t\vartheta-\frac{d_2}{2}D_1{}^t y-\frac{d_2}{2}D_1{}^t y+d_2D_1{}^tD_1\vartheta-D_1{}^t\alpha=0, \tag{4.11}$$

$$\frac{\partial L(\vartheta,\zeta,\alpha)}{\partial\zeta}=2C\zeta-\alpha=0, \tag{4.12}$$

And

$$\frac{\partial L(\vartheta,\zeta,\alpha)}{\partial\alpha}=D_1\vartheta+e\varepsilon+\zeta-y=0. \tag{4.13}$$

To get the value of ϑ, solve the following equations: (4.11), (4.12), and (4.13).

$$\vartheta=\begin{bmatrix}w\\b\end{bmatrix}=\Big(d_1I_0^t+(d_2+2C)D_1{}^tD_1\Big)^{-1}D_1{}^t\big((d_2+2C)y-2Ce\varepsilon\big). \tag{4.14}$$

The best linear LS-LDMR regressor for the new sample q x v is expressed in the following way:

$$f(x)=w^tx+b. \tag{4.15}$$

Non-linear LS-LDMR

We estimate the function K (x^t,B^tw+b) + in the non-linear LS-LDMR model to solve the optimization issue given by

$$\min_{w,b,\zeta} \frac{d_2}{2} \| y-(K(B,B^t)w+eb)\|^2 + C\zeta^2 + \frac{d_1}{2}\| w\|^2$$

subject to

$$Y= (K (B, B^t) w + eb) + e\varepsilon + \zeta, \quad (4.16)$$

The definite kernel matrix is denoted as K (K > 0), the slack variable is denoted as ζ, and the penalty parameter is defined as C > 0.

Let us consider, $\vartheta = \begin{bmatrix} w \\ b \end{bmatrix}$; $\|W\|^2 = \vartheta^t I_0 \vartheta$ and $I_0 = \begin{bmatrix} I & 0 \\ & : \\ 0 & \ldots 0 \end{bmatrix}$

Next, determine the Lagrangian functions of (4.16), using the Lagrangian multiplier Σ.

$$L(\vartheta,\zeta,\beta) = \frac{d_2}{2} \| y-(K(B,B^t)w+eb)\|^2 + C\zeta^2 + \frac{d_1}{2}\| w\|^2 - \beta^t\left((K(B,B^t)w+eb)+e\varepsilon+\zeta-y\right)$$

It is also possible to rephrase the above equation as

$$L(\vartheta,\zeta,\beta) = \frac{d_2}{2} \| y-D_2\vartheta\|^2 + C\zeta^2 + \frac{d_1}{2}\| w\|^2 - \beta^t\left(D_2\vartheta+e\varepsilon+\zeta-y\right), \quad (4.17)$$

where D_2= [K(B, Bt)]is the augmented matrix.

Find the gradient of (4.17), taking into account ϑ, ζ, and β, and set them equal to zero in the following way:

$$\frac{\partial L(\vartheta,\zeta,\beta)}{\partial \vartheta} = d_1 I_0^t \vartheta - \frac{d_2}{2} D_2^t y - \frac{d_2}{2} D_2^t y + d_2 D_2^t D_2 \vartheta - D_2^t \beta = 0 \quad (4.18)$$

$$\frac{\partial L(\vartheta,\zeta,\beta)}{\partial \zeta} = 2C\zeta - \beta = 0, \quad (4.19)$$

And

$$\frac{\partial L(\vartheta,\zeta,\beta)}{\partial \beta} = D_2\vartheta + e\varepsilon + \zeta - y = 0. \tag{4.20}$$

To get the value of , solve the following equations: (4.18), (4.19), and (4.20).

$$\vartheta = \begin{bmatrix} w \\ b \end{bmatrix} = \left(d_1 I_0^t + (d_2 + 2C) D_2{}^t D_2\right)^{-1} D_2{}^t \left((d_2 + 2C)y - 2Ce\varepsilon\right). \tag{4.21}$$

The end regressor of non-linear LS-LDMR may be calculated for any test sample by

F(x)=K(x^t, B^t)$w + b$ (4.22)

Remarks1: With the use of the LDM model's features and the least squares loss function, the LS-LDMR method is suggested. In this case, the answer comes from a linear KKT system, even though the QPP is computationally intensive and requires us to calculate the matrix's inverse as $(d_1 I_0^t + (d_2 + 2C) D_2^t D_2)^{-1}$

Remarks2: The suggested LS-LDMR method uses a system of linear equations, which makes optimization simpler and reduces computing cost. Simultaneously, it maximizes the margin mean and its variation, enables the proposed LS-LDMR to use all training example information, and prevents overfitting.

Discussion: Here, we've gone over why our recommended strategy is better than previous methods that have been described.

1. Based on the features of the LDM model, the suggested least-squares version of LDMR.

2. Working with a system of linear equations that requires us to calculate the inverse of the matrix alone yields the answer.

3. Unlike SVR, TSVR, ε -SVQR, and LDMR, there is no need to resolve the enormous size of the QPP. This means that LS-LDMR requires less processing power.

4. Allow the suggested LS-LDMR to make use of all training example data while also preventing overfitting.

5. By doing numerical tests on both real-world and synthetic datasets, the usefulness and effectiveness of the LS-LDMR model are shown.

4.2.2 NUMERICAL EXPERIMENTS

We compare our proposed method, LS-LDMR, with the traditional SVR, TSVR, PLSTSVR, ε -HSVR, ε -SVQR, MDR, and LDMR on thirty real-world datasets and twenty-eight artificial datasets for both linear and non-linear cases, and we conduct a number of experiments to confirm its efficiency and practicality based on different evaluation parameters and computational cost. To find the best settings, this experiment uses ten-fold cross-validation.

A Windows 10 computer with 4 GB of RAM and the MATLAB 12.0 environment were used to carry out the experiment. For solving the 'quadprog' function in SVR, TSVR, ε -SVQR, and LDMR, an optimization toolbox called MOSEK is also used. All of the data sample input characteristics are scaled to a normal distribution between 0 and 1. The non-linear Gaussian kernel function is defined as

$K(X_r,X_s) = \exp(-\sigma\|Xr - Xs\|^2)$, for r, s=1,….,p.

The input data samples are represented by r x and s x, and the kernel parameter $\sigma > 0$ is selected from the range {2-5,2-4,..., 25} in our studies.

Optional regularization parameters include C, C1, and C2, as well as d1, λ1 and λ2. values for SVR, TSVR, PLSTSVR, ε -HSVR, ε -SVQR, MDR, LDMR, and LS-LDMR range from 10 to 105 on the scale.

The values of ε for SVR, TSVR, PLSTSVR, -HSVR, and LS-LDMR are selected from the following ranges: 0.1, 0.3,..., 0.9 for SVR; 0.001, 0.01, 0.1 for ε -HSVR and LS-LDMR; and { 0.001, 0.05, 0.01, 0.1, 0.5, 1,1.5, 2} for LDMR and ε -SVQR.

For ε -HSVR, the value of ρ is chosen from the interval {0.1,1,1.375}, and for ε -SVQR, ψ is chosen from the interval {0.1, 0.2,...,0.9}. A value of 1 is assigned to d2.

Artificial Datasets

The sounds are created by using a normal distribution N (τ δ2) and The mean and variance are represented by τ and δ2, respectively, in the uniform probability distribution U (θ1 θ2). When it comes to generalization performance, LS-LDMR clearly does better than SVR, TSVR, PLSTSVR, ε -HSVR, ε -SVQR, MDR, and LDMR.. In addition, when compared to all other methods, LS-LDMR is the fastest. Figures 4.13–4.4 and 4.15–4.16 show the prediction performance for Functions 15–16 and 27–28, respectively, which prove that LS-LDMR is capable of making accurate predictions. Our proposed strategy outperforms the alternatives in 12 out of 28 cases.

TABLE 4. 8 Comparison of LS-LDMR to other published models on the synthetic dataset's average RMSE values obtained from a Gaussian kernel

Datasets	SVR	TSVR	LSTSVR	ϵ -HSVR	ϵ -SVQR	MDR	LDMR	LS-LDMR
Function 1	1	8	6	7	4	5	3	2
Function 2	2	8	6	7	4	5	3	1
Function 3	5	2	8	1	7	3	4	6
Function 4	8	3	5	4	7	6	2	1
Function 5	8	1	6	5	7	4	2	3
Function 6	8	7	4	5	1	6	3	2
Function 7	8	7	2	6	1	5	3	4
Function 8	8	7	3	5	1	6	4	2
Function 9	8	4	6	3	5	2	7	1
Function 10	8	7	5	6	1	2	4	3

Function 11	2	1	7	3	8	6	4	5
Function 12	8	6	4	3	7	5	2	1
Function 13	8	2	4	3	6	7	1	5
Function 14	8	7	5	4	3	6	2	1
Function 15	5	6	8	7	2	4	3	1
Function 16	2	8	6	7	4	5	3	1
Function 17	4	8	6	7	1	5	3	2
Function 18	4	8	6	7	1	5	3	2
Function 19	7	8	4	5	3	2	6	1
Function 20	5	7	4	2	8	1	6	3
Function 21	5	4	6	8	7	2	3	1
Function 22	8	4	6	7	1	2	5	3
Function 23	8	1	3	7	5	6	4	2
Function 24	8	3	4	2	5	1	7	6
Function 25	2	8	6	7	4	5	3	1
Function 26	1	8	6	7	4	5	3	2
Function 27	8	7	5	4	6	3	2	1

Function 28	8	7	4	6	5	2	3	1
Average ranks	5.8929	5.6071	5.1786	5.1786	4.2143	4.1429	3.5	2.2857

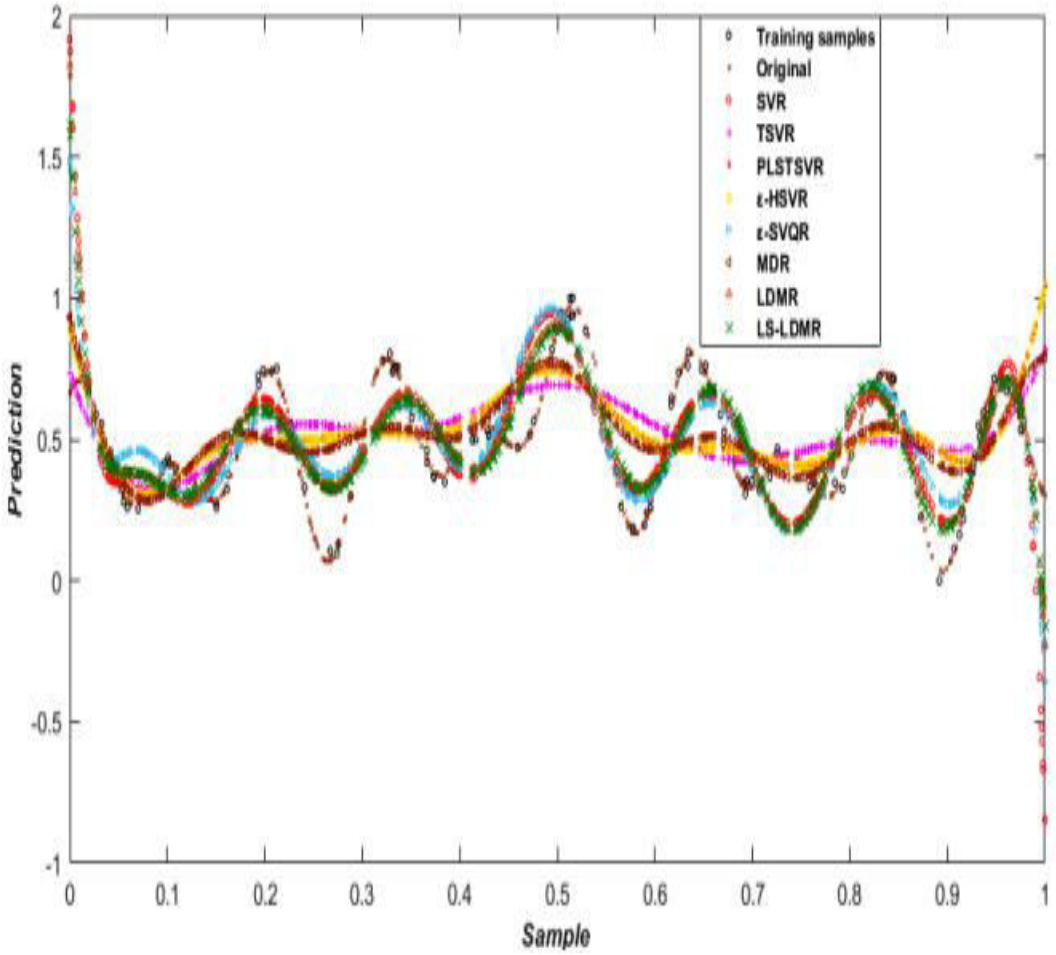


FIGURE 3. 46 Predictions made by several models on the Function 15 synthetic dataset using a Gaussian kernel, including LS-LDMR, over the testing dataset.

For the RMSE evaluating parameter, all models are included in Table 4.8; See Table 4.9 for the ordering of the evaluation parameters: RMSE, MAE, SSE/SST, SMAPE, and MASE. As seen in Tables 4.8 and 4.9, our proposed approach, LS-LDMR, ranks last when contrasted with SVR, TSVR, PLSTSVR, ε -HSVR, ε -SVQR, MDR, and LDMR.

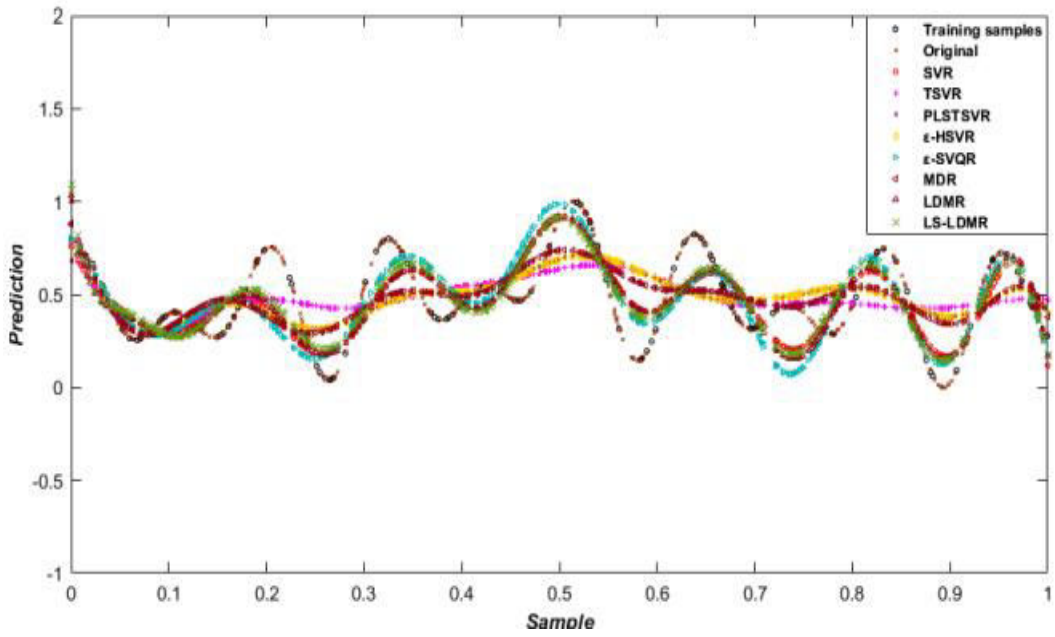


Figure 4. 13 Predictions made by several models on the Function 16 synthetic dataset using a Gaussian kernel, including LS-LDMR, over the testing dataset.

TABLE 4. 9 Comparison of LS-LDMR's MAE, SSE/SST, SMAPE, and MASE to other models that have been reported using a Gaussian kernel on simulated datasets using

Parameters	SVR	TSVR	PLSTSVR	ϵ-HSVR	ϵ-SVQR	MDR	LDMR	LS-LDMR
RMSE	5.8929	5.6071	5.1786	5.1786	4.2143	4.1429	3.5	2.2857
MAE	5.78571	5.25	5.03571	4.92857	4.10714	4.17857	3.67857	3.03571
SSE/SST	5.89286	5.71429	5.21429	5.14286	4	4.10714	3.53571	2.39286
SMAPE	5.85714	5.32143	5.10714	4.60714	4.53571	3.89286	3.78571	2.89286
MASE	5.78571	5.25	5.03571	4.92857	4.10714	4.17857	3.67857	3.03571

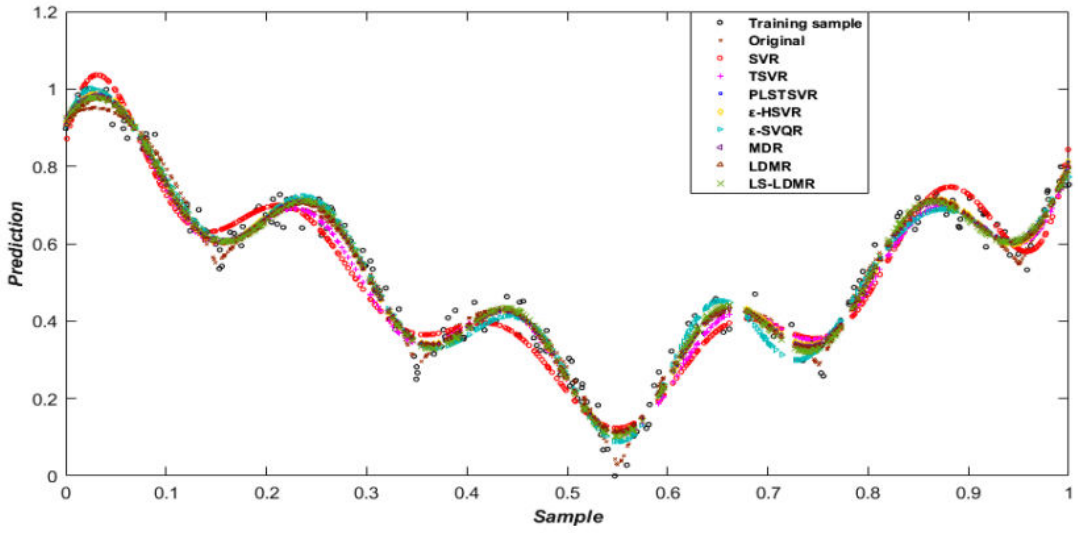


Figure 4. 14 Predictions made by several models on the Function 27 synthetic dataset using a Gaussian kernel, including LS-LDMR, over the testing dataset

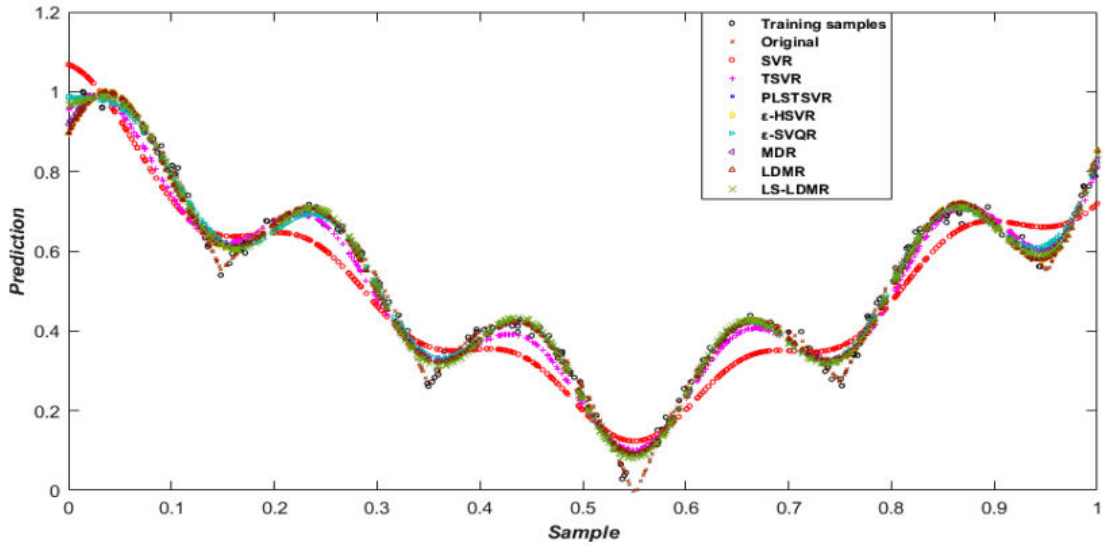


Figure 4. 15 Forecasting on the LS-LDMR test dataset and alternative models on the Function 28 synthetic dataset with a Gaussian kernel.

a) Friedman and Nemenyi test on synthetic datasets

Using the Friedman test and the synthetic datasets listed in Table 4.8, we statistically test our proposed LS-LDMR. The null hypothesis is taken into account. We can tell that these algorithms are comparable by comparing their results on important evaluation metrics like root-mean-square error (RMSE) and average rank (Rj). If you believe Demsar,

$$\chi_F^2 = \frac{12\times 28}{8\times(8+1)}\left[\left(5.8929^2+5.6071^2+5.1786^2+4.2143^2+4.1429^2+3.5^2+2.2857^2\right)-\left(\frac{8(8+1)^2}{4}\right)\right]$$

$\chi_F^2 \approx 47.5952,$

And

$$F_F = \frac{(28-1)\times 47.5952}{(28(8-1))-47.5952} \approx 8.6592.$$

A Friedman statistic FF with 7 degrees of freedom and an F-distribution with 189 degrees of freedom, with a standard deviation of (λ −1) and (λ −1)×(Δ −1). Considering the significance thresholds of 0.05 and 0.10, as well as F(7, 189), the critical value (CV) is 2.05829 and 1.74883, respectively. In this situation, the null hypothesis cannot be accepted since the Friedman statistic FF (8.6592 > 2.05829) is bigger than the CV (8.6592 > 1.74883).

Use the Nemenyi test to find the key difference now by

Critical difference (CD) = $2.78\sqrt{\frac{8x(8+1)}{6x28}} = 1.8199,$

The following are some conclusions reached from the statistical analysis:

i. On average, our proposed LS-LDMR ranks higher than SVR and TSVR. Since the difference between LS-LDMR and SVR and TSVR is more than the CD, the LS-LDMR approach outperforms the other two (3.6071>1.8199; 3.3214 >1.8199)..

ii. Verify the mean rank of LS-LDMR relative to PLSTSVR, □ -HSVR, and □ -SVQR, which are (5.1786 - 2.2857 = 2.8929) and (4.2143 - 2.2857 = 1.9286) respectively. The efficacy of LS-LDMR is justified since the CD is less than the difference (2.8929 >1.8199) and (1.9286 >1.8199).

iii. For example, (4.1429 - 2.2857=1.8571; 3.5- 2.2857=1.2143) is the average rank of LS-LDMR relative to MDR and LDMR. While the discrepancy with MDR (1.8571 >1.8199) is larger, the CD is less when dealing with LDMR, LS-LDMR is better than MDR and equivalent to LDMR because to its higher CD (1.2143 <1.8199).

Real-world Datasets

The following financial time series datasets are used in this analysis, together with thirty real-world benchmark datasets, such as Kin900 and Demo from DELVE: S&P500, INFY, MSFT, IXIC, AT&T, BVSP, and TCS are all included.to assess the effectiveness of the suggested LS-LDMR with SVR, TSVR, PLSTSVR, ε -HSVR, ε -SVQR, MDR, and LDMR; KEEL time-series datasets: KEEL dataset; UCI datasets repositories: Gas_furnace, Flex_robotarm, Motorcycle, Triazines, and Abalone; NLREG repositories: Chwirut2; and OSTI datasets: Mg17.

TABLE 4. 10 Compared to other linear kernel models published on real-world datasets, the average rankings of RMSE, MAE, SSE/SST, SMAPE, and MASE for LS-LDMR

Parameters	SVR	TSVR	PLSTSVR	ε -HSVR	ε -SVQR	MDR	LDMR	LS-LDMR
RMSE	6.55	4.3	3.9167	4.6667	5.3	5.3	3.45	2.5167
MAE	6.7667	3.75	4.1	4.9	5.0667	5.0333	3.6167	2.7667
SSE/SST	6.5667	4.3333	3.9	4.6	5.4333	4.8667	3.8667	2.4333
SMAPE	6.6333	3.75	3.8333	4.5667	5.05	5.7	3.3333	3.1333
MASE	6.8	3.75	4.0667	4.7333	5.0333	5.1667	3.6167	2.8333

TABLE 4. 11 Evaluation of LS-LDMR in comparison to other models utilizing a Gaussian kernel and provided RMSE values for a real-world dataset

Datasets	SVR	TSVR	LSTSVR	ε - HSVR	ε - SVQR	MDR	LDMR	LS-LDMR
Kin900	6	5	8	7	4	3	2	1
Demo	5	6	4	7	8	3	2	1
Mg17	8	6	3	4	7	5	2	1
Gas_furnace	8	6	4	5	7	2	3	1
Flex_robotarm	8	7	5	6	2	3	4	1
Motorcycle	8	3	7	6	2	1	5	4
Triazines	2	3	7	8	6	1	5	4
S&P500	8	7	4	6	1	3	5	2
Abalone	8	5	1	2	6	7	4	3
Chwirut2	8	5	3	4	1	7	6	2
AT&T	8	6	3	4	7	5	2	1
INFY	8	1	5	3	7	2	6	4
MSFT	8	6	5	3	1	7	4	2
TCS.BO	3	8	6	4	7	1	5	2
BVSP	8	7	4	5	6	1	3	2
IXIC	8	7	4	2	5	6	3	1
Wankara	8	6	4	3	7	1	5	2
Wizmir	8	6	3	5	7	1	4	2
Friedman	8	7	5	3	1	4	6	2
Treasury	8	6	4	1	7	3	5	2
NN5_Complete_105	3	7	8	5	4	6	2	1
NN5_Complete_108	5	6	7	8	4	3	2	1
NN5_Complete_111	5	4	8	7	6	3	2	1
NNGC1_dataset_D1_V1_010	8	7	4	6	1	5	2	3
NNGC1_dataset_E1_V1_001	1	4	8	6	7	5	3	2
NNGC1_dataset_E1_V1_008	4	6	8	7	3	5	1.5	1.5
NNGC1_dataset_F1_V1_003	8	1	3	5	7	6	4	2
NNGC1_dataset_F1_V1_006	8	1	5	4	7	3	6	2
NNGC1_dataset_F1_V1_010	8	4	6	5	7	2	3	1
NNGC1_dataset_F1_V1_011	8	2	4	5	6	7	3	1
Average rank	6.7333	5.1667	5	4.8667	5.0333	3.7	3.65	1.85

TABLE 4. 12 Rankings comparing LS-LDMR with other published models utilizing a Gaussian kernel on real-world datasets for RMSE, MAE, SSE/SST, SMAPE, and MASE

Parameters	SVR	TSVR	PLSTSVR	ε -HSVR	ε -SVQR	MDR	LDMR	LS-LDMR
RMSE	6.7333	5.1667	5	4.8667	5.0333	3.7	3.65	1.85
MAE	6.8667	5.6	4.6	4.6333	4.9333	3.8333	3.45	2.0833
SSE/SST	6.6333	5.0333	4.9333	4.9	4.9667	3.8667	3.6167	2.05
SMAPE	7.1667	5.6	4.6	4.8	4.6	3.6333	3.4167	2.1833
MASE	6.8667	5.6	4.5667	4.6667	4.9333	3.8333	3.45	2.0833

If you look at Table 4.11 for the Gaussian kernel, you can see the average rankings tabulated according to RMSE. Table 4.10 displays the results for the linear kernel and Table 4.12 displays the results for the Gaussian kernel, while the average rankings of all stated techniques are computed for MAE, SSE/SST, SMAPE, and MASE, respectively. The success of LS-LDMR is supported by the fact that it has the lowest average rank compared to other techniques using RMSE, MAE, SSE/SST, SMAPE, and MASE for both linear and Gaussian kernels.

Figures 4.17–4.20 show the related prediction performance graphs for the Gaussian kernel and benchmark real-world datasets Flex_robotarm, Gas furnace, Mg17, and AT&T, respectively. From the graphs showing the prediction performance, it is clear that our suggested LS-LDMR outperforms SVR, TSVR, PLSTSVR, and others, ε -HSVR, ε -SVQR, MDR and LDMR.

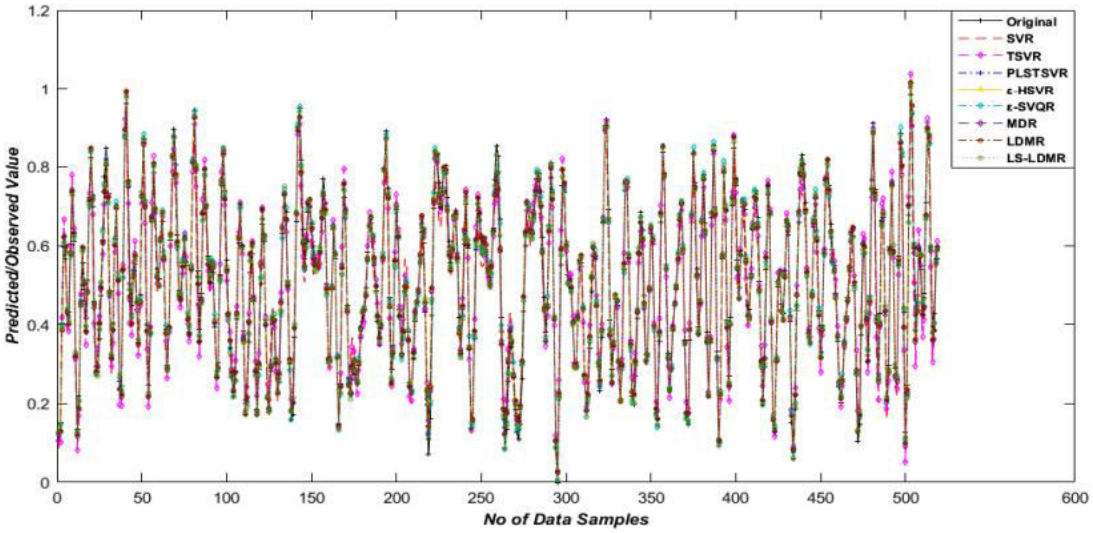


Figure 4. 16 Prediction over the testing dataset by LS-LDMR and other models on the Flex_robotarm dataset using Gaussian kernel.

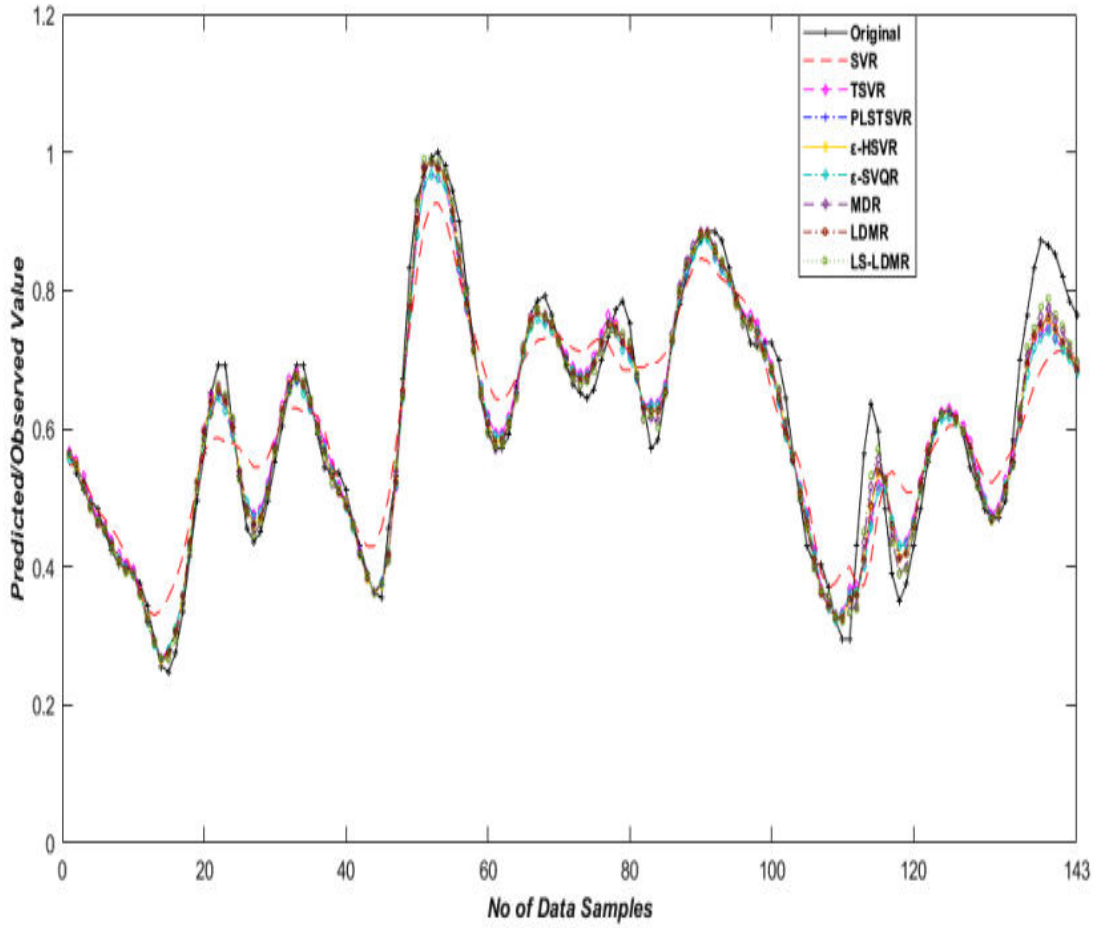


Figure 4. 17 Prediction over the testing dataset by LS-LDMR and other models on the Gas furnace dataset using Gaussian kernel.

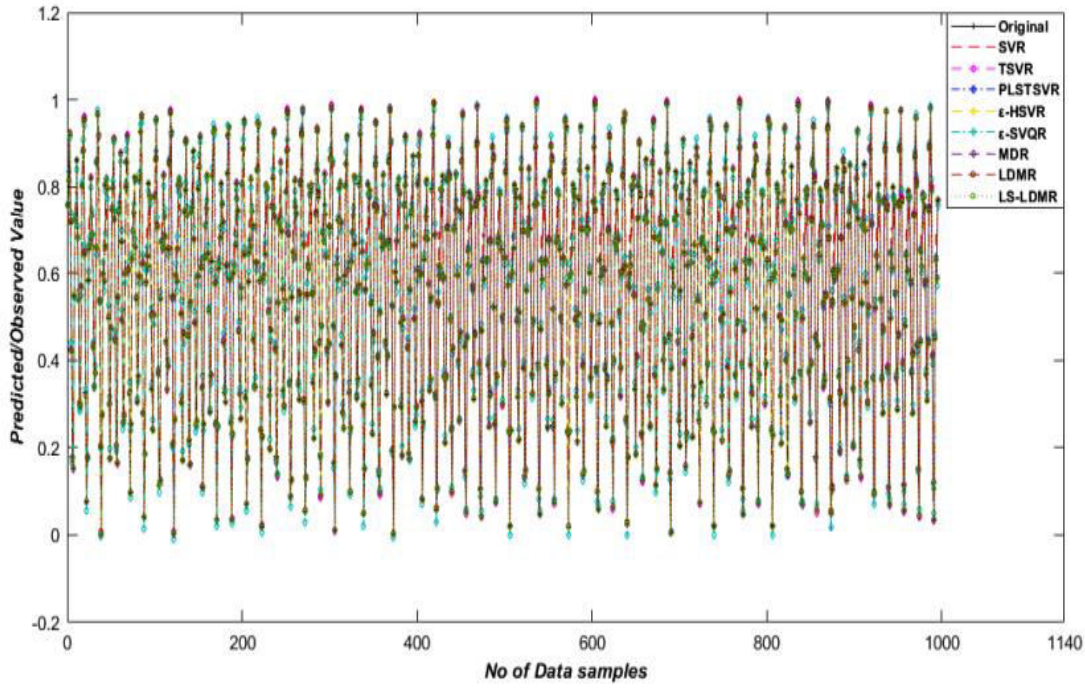


Figure 4. 18 Prediction over the testing dataset by LS-LDMR and other models on Mg17 dataset using Gaussian kernel.

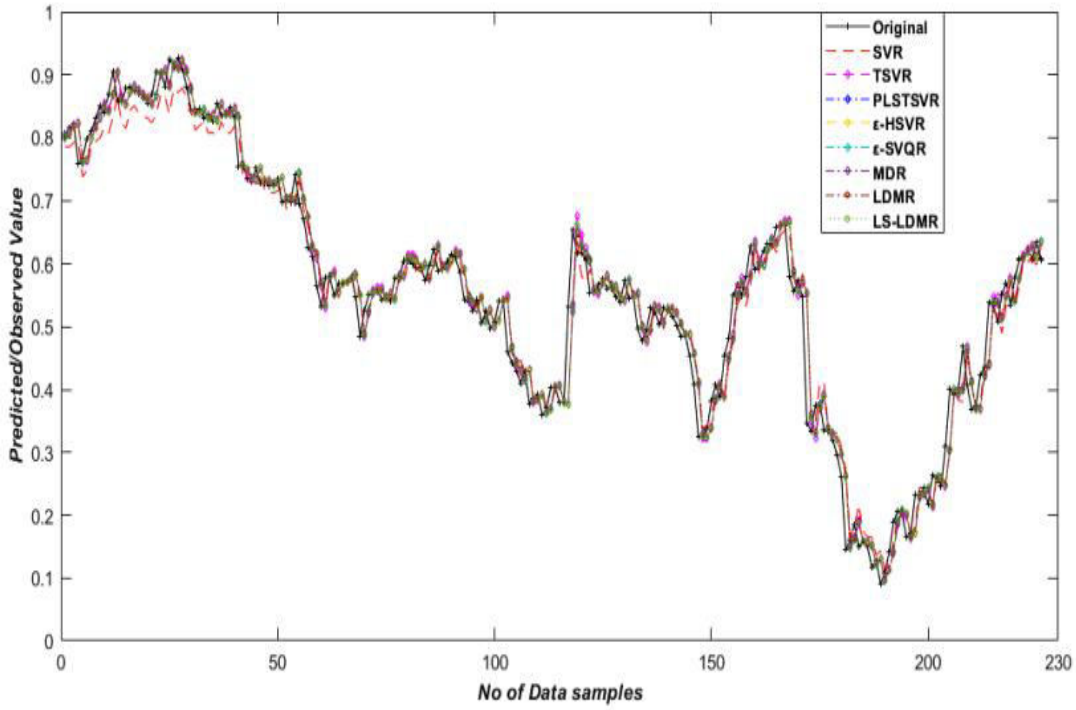


Figure 4. 19 Prediction over the testing dataset by LS-LDMR and other models on the AT&T dataset using Gaussian kernel.

A. Friedman and Nemenyi test on real-world datasets

Our proposed LS-LDMR is further statistically tested for linear and Gaussian kernels using the well-known Friedman and Nemenyi test, which measures RMSE and SSE/SST.

a) For linear kernel based on RMSE from Table 4.10

$$\chi_F^2 = \frac{12\times 30}{8\times(8+1)}\left[\left(6.55^2+4.3^2+3.917^2+4.667^2+5.3^2+5.3^2+3.45^2+2.517^2\right)-\left(\frac{8(8+1)^2}{4}\right)\right]$$

$$\chi_F^2 \approx 54.6703\,,$$

And $F_F \approx \frac{(30-1)X54.6703}{(30(8-1))-54.6703} \approx 10.2069.$

F_F is distributed according to the F-distribution with a probability of 7,203 degrees of freedom level δ= 0.05 and δ= 0.10. the CV is 2.054907 and 1.746585. Here Friedman statistic F_F is greater than the CV (10.2069 > 2.054907 ;10.2069 >1.746585). Consequently, we do not accept the null hypothesis. Based on Demsar's (2006) Nemenyi test, CD is calculated as

$$\text{CD}=2.78\sqrt{\frac{8X(8+1)}{6x30}} = 1.76$$

Here are a few key takeaways from the statistical analysis:

i. When comparing the proposed LS-LDMR with SVR and TSVR, the average rank shows that LS-LDMR is better. The discrepancies between the three methods are larger than the CD, which is (4.033 >1.76) for SVR and (1.783 >1.76) for TSVR.

ii. Verify that 3.917 - 2.517 is 1.4, 4.667 - 2.517 is 2.15, and 5.3 - 2.517 is 2.783, which is the difference between the average ranks of LS-LDMR and PLSTSVR, ε -HSVR, and ε -SVQR.

In this context, the CD represents the LSLDMR method in relation to PLSTSVR, ε -HSVR, and ε -SVQR, where LS-LDMR is similarly effective to PLSTSVR and ε -HSVR.

iii. In comparing the suggested LS-LDMR to MDR and LDMR, the average rank comes out to 2.783 and 0.933, respectively. In this case, the CD is compared to the average rank's difference (1.76 > 2.783; 1.76 > 0.933), indicating that the LS-LDMR method is better than MDR and on par with LDMR.

b) For Linear kernel based on SSE/SST from Table 4.10

$$\chi_F^2 = \frac{12\times 30}{8\times(8+1)}\left[\left(6.567^2+4.333^2+3.9^2+4.6^2+5.433^2+4.867^2+3.867^2+2.433^2\right)-\left(\frac{8(8+1)^2}{4}\right)\right]$$

$$\chi_F^2 \approx 51.7437\,,$$

And

$$F_F \approx \frac{(30-1)\times 51.7437}{(30\times(8-1))-51.7437} \approx 9.4819\,.$$

At = 0.05 and δ = 0.10, the CV will be 2.054907 and 1.746585, respectively, according to the F-test (7, 203).

On this occasion, the Friedman statistic (FF) exceeds the CV (9.4819 > 2.054907 ; 9.4819 >1.746585) . So, the null hypothesis rejects.

Following are a few key takeaways from the Friedman and Nemenyi statistical test:

i. The results obtained by comparing the prediction performance of our suggested method, LS-LDMR, with that of SVR, TSVR, and PLSTSVR are as follows: (4.567 - 2.433 = 4.133), (4.333 - 2.433 = 1.9), and (3.9 - 2.433 = 1.467).

Compared to SVR, TSVR, and PLSTSVR, the LS-LDMR method outperforms them all in terms of SSE/SST, as is seen from the discrepancy with the CD.

ii. In comparison to the CD, the average rank difference of the suggested LS-LDMR with ε -HSVR, ε -SVQR, MDR, and LDMR is higher, with values of (4.6 - 2.433 = 2.167), (5.433 - 2.433 = 3), (4.867 - 2.433 = 2.433), and (3.867 - 2.433 =1.433).

Accordingly, LS-LDMR outperforms ε -HSVR, ε -SVQR, MDR, and is on par with LDMR in terms of SSE/SST.

c) For Gaussian kernel based on RMSE from Table 4.12

$$\chi_F^2 = \frac{12\times 30}{8\times(8+1)}\left[\left(6.733^2+5.167^2+5^2+4.867^2+5.033^2+3.7^2+3.65^2+1.85^2\right)-\left(\frac{8(8+1)^2}{4}\right)\right],$$

$$\chi_F^2 \approx 72.4248,$$

And

$$F_F \approx \frac{(30-1)\times 72.4248}{(30\times(8-1))-72.4248} \approx 15.2667$$

At δ = 0.05 and δ = 0.10, the CV will be 2.054907 and 1.746585, respectively, for the data set F(7, 203). The results show that FF is higher than CV (15.2667 > 2.05497; 15. 2667 >1.746585), thereby rejecting the null hypothesis. In addition, we may draw the following conclusions:

i. The average rank of LS-LDMR is more different to SVR, TSVR, and PLSTSVR than the CD (1.76 = 4.883, 1.76 = 3.317, 1.76 = 3.15). The values of (4.733 -1.85 = 4.883), (5.167 -1.85 = 3.317), and (5 -1.85 = 3.15) are higher, respectively. Accordingly, LS-LDMR outperforms SVR, TSVR, and PLSTSVR by a wide margin.

ii. The LS-LDMR method is deemed better than ε -HSVR, ε -SVQR, MDR, and LDMR based on the root-mean-squared error (RMSE) than the CD (1.76 = 3.017,1.76 = 3.183,1.76 = 1.85,1.76 = 1.80). This is supported by the fact that the average rank difference between LS-LDMR and ε̇ -HSVR, ε̇ -SVQR, MDR, and LDMR is 3.017, (5.033 -1.85 = 3.183), (3.7 -1.85 =1.85), and (3.65 -1.85 =1.80).

d) For Gaussian kernel based on SSE/SST from Table 4.12

$$\chi_F^2 = \frac{12\times 30}{8\times(8+1)}\left[\left(6.633^2+5.033^2+4.933^2+4.9^2+4.967^2+3.867^2+3.617^2+2.05^2\right)-\left(\frac{8(8+1)^2}{4}\right)\right]$$

$$\chi_F^2 \approx 62.9112,$$

And

$$F_F \approx \frac{(30-1)\times 62.9112}{(30\times(8-1))-62.9112} \approx 12.4036.$$

i. The CV (12.4036 > 1.746585) is less than the Friedman statistic FF (12.4036 > 2.05497). The outcome allows us to reject the null hypothesis. The information allows us to draw the following important conclusions: a. The CD values (1.76 = 4.583,1.76 = 2.983,1.76 = 2.883) are not as close to the average rank of the proposed method LS-LDMR as they are to 4.583, (5.033 - 2.05 = 2.983), and (4.933 - 2.05 = 2.883), respectively. This suggests that LS-LDMR is superior than SVR, TSVR, and PLSTSVR.

ii. The rank difference between LS-LDMR and ε -HSVR, ε -SVQR, MDR, and LDMR is bigger than that of the CD, with values of (4.9 - 2.05 = 2.85), (4.967 - 2.05 = 2.917), (3.867 - 2.05 = 1.817), and (3.617 - 2.05 = 1.567), respectively.Because of this, LS-LDMR is thought of as a better approach than ε -HSVR, ε -SVQR, MDR, and LDMR alone.

An effective computer technique for addressing regression issues using a least squares huge Using mathematical formulas from LDMR and PLSTSVR, this study proposes a margin distribution machine. A system of linear equations may be solved using the proposed LS-LDMR. Therefore, instead of calculating the massive size QPP, which is necessary when working with LDMR, ε -SVQR, TSVR, and SVR, we need to calculate the inverse of the matrix. Thus, LS-LDMR does not need the use of an extra optimization toolbox. Both computationally and in terms of prediction ability, our suggested LS-LDMR outperforms the state-of-the-art techniques, as shown by experiments on both real-world and synthetic datasets. Statistics also show that LS-LDMR is more effective and valuable than competing models. Possible future work on class imbalance learning strategies includes exploring the sparse model for LS-LDMR or proposing the Universum approach in conjunction with LS-LDMR.

CHAPTER 5
FUNCTIONAL ITERATIVE APPROACHES FOR TWIN BOUNDED SUPPORT VECTOR MACHINES WITH SQUARED PINBALL LOSS AND INTUITIONISTIC FUZZY-BASED LEAST SQUARES TWIN BOUNDED SUPPORT VECTOR MACHINES

CHAPTER 5
FUNCTIONAL ITERATIVE APPROACHES FOR TWIN BOUNDED SUPPORT VECTOR MACHINES WITH SQUARED PINBALL LOSS AND INTUITIONISTIC FUZZY-BASED LEAST SQUARES TWIN BOUNDED SUPPORT VECTOR MACHINES

5.1 AN OPTIMAL FUNCTIONAL APPROACH TO A TWIN-BOUND SUPPORT VECTOR MACHINE AFFECTED BY SQUARED PINBALL LOSS

Classification accuracy is improved when the loss function is applied to TWSVM-based models, much as it is to regression models. A learning model's ability to generalize depends on its loss function, therefore it's important to choose one that captures the characteristics of the noise in the training data. As an added downside, although being smooth, quadratic loss is more prone to errors and has lower robustness.

This is something that we are aware of. While 1-norm might lessen the impact of noise, it is not a panacea for smoothness. The common reduction techniques for numerical data won't apply since it's not smooth. min-max approaches require longer than smooth loss function minimization because to the non-smooth nature of pinball loss.

The squared pinball loss function, which is sometimes called the asymmetric squared loss function, is supplied by Newey and Powell. This function may be used to quickly estimate the quintile value. In comparison to the pinball ball loss function, the squared pinball loss function is more efficient with respect to time. Our novel model, based on a regularized TWSVM and using elementary's squared pinball loss function, is presented in this chapter.

Following the regularized TWSVM model that is based on the squared pinball loss function, we want to achieve a tiny misclassification error and a minor interior scatter. Instead of employing TWSVM to solve two quadratic programming problems, a basic functional iterative strategy is used to find the answer. Therefore, it does what it set out to do without relying much on any external optimization toolkit. Reason being, in the main space, fundamental knowledge is always preferable to a rough answer.

Among the several advantages of the proposed approach are better resampling stability, less noise sensitivity, greater convexity, and increased resilience. In order to verify the efficacy and excellence of the suggested model, numerical experiments have been conducted utilizing datasets that are publically available as well as datasets that were developed for the University of California, Irvine (UCI). The proposed model's ability to manage corrupted and noisy datasets is shown by comparing its results to those of current and baseline methods such as generalized Huber twin support vector machines (GHTSVM), sparse pinball twin support vector machines (SPTWSVM), and support vector machines (SVM).

5.1.1 THE SQUARED PINBALL LOSS FUNCTION

In Table 5.1, we can see the squared pinball loss function defined as:

Table 5. 1 The squared pinball loss function used in Spin-FITBSVM

Squared pinball loss function
$L^p(u) = \begin{cases} (1-p)u^2 & , \quad u < 0 \\ & , 0 \le p \le 1 \\ pu^2 & , \quad u \ge 0 \end{cases}$ (5.1)
P is the parameter for pinball loss.
The squared pinball function may be rewritten as follows:
$L^p(u) = (1-p)(-u)_+^2 + pu_+^2.$ (5.2)
Where, $u_+^2 = max\{0, u^2\}$ for any $u \in \Re$.

The asymmetric function is shown graphically in Figure 5.1, which also gives a visual depiction of the squared loss function. This tool may be used to view a variety of pinball loss p values. P = 1/j is the source of both the squared hinge loss and the symmetric quadratic loss and it is also feasible to find that p = 1 causes these losses.

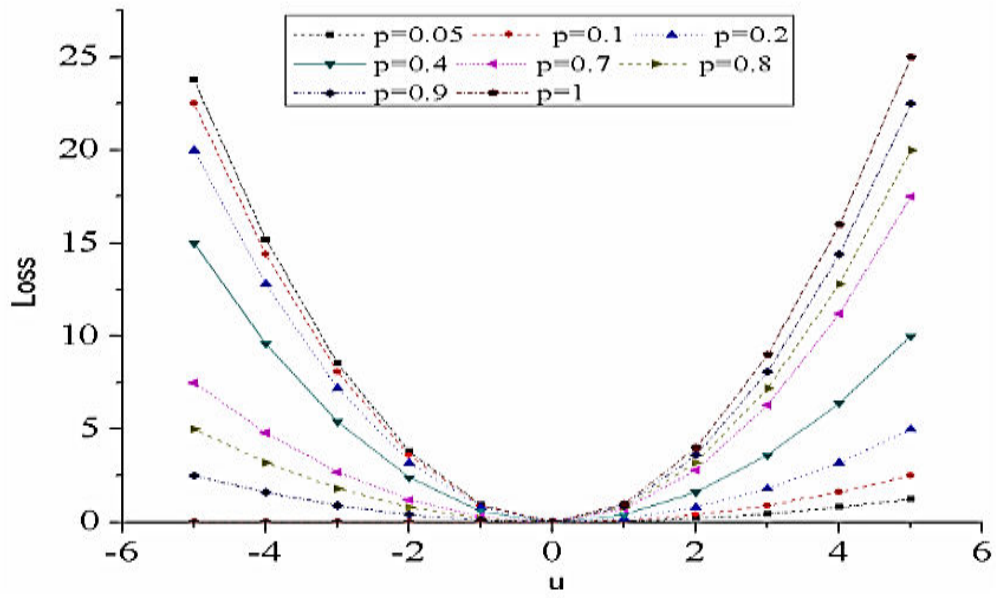


Figure 5. 1 Piston loss function squared for a range of p-values, graphically shown.

5.1.2 A NEW FUNCTIONAL THE SPIN-FITBSVM ITERATIVE METHOD FOR TWIN-BOUNDED SUPPORT VECTOR MACHINES WITH SQUARED PINBALL LOSSES

Iterative techniques for twin bounded support vector machines with squared pinball losses are introduced, and we term it Spin-FITBSVM. Our goal in going this way is to provide a reliable technique for resampling and make the system less susceptible to noise. Using an iterative approach to functions, we have solved two convex minimization problems within the topic of this paper that include squared pinball loss. Up to this point, we have discussed Spin-FITBSVM, a functional iterative approach, is designed for twin bounded support vector machines with squared pinball loss. This method incorporates an extra regularization component.$C_2(\|w_1\|^2 + b_1^2), and\ C_4(\|w_2\|^2 + b_2^2)$in TSVM's fundamental issue. By including the regularization element into the equation, we may resolve the overfitting issue. Two non-parallel hyperplanes using nonlinear data points are necessary for our Spin-FITBSVM model to solve the classification problem. The following is an example of a model that is comparable to the TSVM model:

$$\min_{w_1,b_1} C_2\left(\| w_1 \|^2 + b_1^2\right) + \frac{1}{2}\| [K(B_1, B^t) \quad e_1][w_1 \quad b_1]^t \|^2 + C_1 \sum_{k=1}^{p} L^{\xi}\left([K(x_k, B^t) \quad e_2][w_1 \quad b_1]^t\right), \tag{5.3}$$

And,

$$\min_{w_2,b_2} C_4\left(\| w_2 \|^2 + b_2^2\right) + \frac{1}{2}\| [K(B_2, B^t) \quad e_2][w_2 \quad b_2]^t \|^2 + C_3 \sum_{k=1}^{p} L^{\xi}\left(-\left([K(x_k, B^t) \quad e_1][w_2 \quad b_2]^t\right)\right). \tag{5.4}$$

The regularization parameters are denoted as C (k = 1, 2, 3, 4, k), the pinball function parameter is represented by ξ, the ones vector is 1 2 e, and the squared pinball loss function is given by L (x) ξ.

The regularization term in the first-term demonstrates how to construct the more convex objective functions (5.3) and (5.4), which offers the only solution, within the primary constraints of Spin-FITBSVM, as shown in calculations. Last but not least, a squared pinball loss function is used to provide a dependable resampling solution and reduce noise sensitivity. The first term maintains the hyper plane in the first class; the second term adds up the squared distances from the desired hyper plane to data points in two different classes.

This is the way to rewrite (5.3) and (5.4):

$$\min_{w_1,b_1} C_2\left(\| w_1 \|^2 + b_1^2\right) + \frac{1}{2}\| [K(B_1, B^t) \quad e_1][w_1 \quad b_1]^t \|^2 + C_1(\xi \| ([K(B_2, B^t) \quad e_2][w_1 \quad b_1]^t)_+ \|^2 + (1-\xi) \| (-([K(B_2, B^t) \quad e_2][w_1 \quad b_1]^t))_+ \|^2), \tag{5.5}$$

And

$$\min_{w_2,b_2} C_4\left(\| w_2 \|^2 + b_2^2\right) + \frac{1}{2}\| [K(B_2, B^t) \quad e_2][w_2 \quad b_2]^t \|^2 + C_3(\xi \| (-([K(B_1, B^t) \quad e_1][w_2 \quad b_2]^t))_+ \|^2 + (1-\xi) \| ([K(B_1, B^t) \quad e_1][w_2 \quad b_2]^t)_+ \|^2). \tag{5.6}$$

Using the functional iterative technique, we are able to solve (5.5) and (5.6).

$$L(\vartheta_1) = C_2 \vartheta_1^t \vartheta_1 + \frac{1}{2}\| D_1 \vartheta_1 \|^2 + C_1(\xi \| (D_2 \vartheta_1)_+ \|^2 + (1-\xi) \| (-D_2 \vartheta_1)_+ \|^2), \tag{5.7}$$

And,

$$L(\vartheta_2) = C_4 \vartheta_2^t \vartheta_2 + \frac{1}{2}\| D_2 \vartheta_2 \|^2 + C_3(\xi \| (-D_1 \vartheta_2)_+ \|^2 + (1-\xi) \| (D_1 \vartheta_2)_+ \|^2). \tag{5.8}$$

Where, $\vartheta_2 = \begin{bmatrix} w_k \\ b_k \end{bmatrix}, k = 1,2; D_1 = [K(B_1, B^t)e_1]$and $D_2 = [K(B_2, B^t)e_2]$ are an augmented matrix; the plus function is a+ = max{ 0, a} .

Calculate the gradient of (5.7) and (5.8) and equate to zero.

$$\nabla L_1(\vartheta_1) = 2C_2\vartheta_1 + D_1{}'D_1\vartheta_1 + C_1(2\xi D_2{}'(D_2\vartheta_1)_+ - 2(1-\xi)D_2{}'(-D_2\vartheta_1)_+),$$

(5.9)

And,

$$\nabla L_2(\vartheta_2) = 2C_4\vartheta_2 + D_2{}'D_2\vartheta_2 + C_3(-2\xi D_1{}'(-D_1\vartheta_2)_+ + 2(1-\xi)D_1{}'(D_1\vartheta_2)_+). \quad (5.10)$$

From (5.9) and (5.10), one can find

$$\vartheta_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -2C_1D_2{}'(2C_2I + D_1{}'D_1)^{-1}(\xi(D_2\vartheta_1)_+ - (1-\xi)(-D_2\vartheta_1)_+), \quad (5.11)$$

And,

$$\vartheta_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = 2C_3D_1{}'(2C_4I + D_2{}'D_2)^{-1}(\xi(-D_1\vartheta_2)_+ - (1-\xi)(D_1\vartheta_2)_+). \quad (5.12)$$

For any value of k from 0 to n, use the function iterative technique on equations (5.11 and 5.12, respectively).

$$\vartheta_1^{k+1} = -2C_1D_2{}'(2C_2I + D_1{}'D_1)^{-1}(\xi(D_2\vartheta_1^k)_+ - (1-\xi)(-D_2\vartheta_1^k)_+), \quad (5.13)$$

And,

$$\vartheta_2^{k+1} = 2C_3D_1{}'(2C_4I + D_2{}'D_2)^{-1}(\xi(-D_1\vartheta_2^k)_+ - (1-\xi)(D_1\vartheta_2^k)_+). \quad (5.14)$$

By assuming $R_0 = (2C_2I + D_1^tD_1)^{-1}$ and $S_0 = (2C_4I + D_2^tD_2)^{-1}$ one can write the (5.13) and (5.14) in such a way:

$$\vartheta_1^{k+1} = -2C_1D_2{}'R_o(\xi(D_2\vartheta_1^k)_+ - (1-\xi)(-D_2\vartheta_1^k)_+), \quad (5.15)$$

And,

$$\vartheta_2^{k+1} = 2C_3 D_1{}^t S_o(\xi(-D_1\vartheta_2^k)_+ - (1-\xi)(D_1\vartheta_2^k)_+). \tag{5.16}$$

For any new data point q $x \in \Re$, the class label can be obtained as:

$$class(k) = sign\left(\frac{K(x^t, B^t)w_1 + b_1}{\| w_1 \|} + \frac{K(x^t, B^t)w_2 + b_2}{\| w_2 \|}\right), \tag{5.17}$$

where k ={+1, −1}.

Discussion:

1. We provide a robust, noise-resistant, and highly convex model to address classification issues.

2. Within the context of the idea of structural risk reduction, our proposed Spin-FITBSVM model offers a novel approach.

3. Using a function iterative strategy will help you solve the suggested Spin-FITBSVM model. Because of this, you won't need any third-party optimization toolkit.

4. To demonstrate the method's practicality and feasibility, run extensive numerical tests utilizing real-world benchmark datasets from UCI and synthetic datasets that have been processed using Spin-FITBSVM.

5.1.3. NUMERICAL EXPERIMENTS

Here, we demonstrate that Spin-FITBSVM, the approach we proposed, is applicable to both real-world and synthetic datasets, ranging from publicly available UCI benchmark datasets that are noisy to those that are not. We do this by comparing it against both older and newer techniques, such as baseline support vector machines (SVMs), the well-known twin support vector machines (TSVMs), pin-TSVMs with pinball loss (pin-GTSVMs), GHTSVMs, SPTWSVMs, and general twin support vector machines (GTSVMs). We ran the algorithms on a desktop PC with 32 GB of RAM, an Intel Core i7-4790 CPU, Windows 10, and the MATLAB R2018a environment. Each of the above stated approaches employs a Gaussian kernel with the parameter μ, which is defined as: for $x_r^0, x_s^0 \in \Re^p$.

$$K(x_r^0 - x_s^0) = exp(-\|x_r^0 - x_s^0\|^2/2\mu^2)$$

where $\mu > 0$ is the kernel parameter which is selected from 2^{-5} to 2^5.

Here, we have taken $C_1 = C_2$, $C_3 = C_4$ and $\lambda_1, \lambda_2, \lambda_3$ for the sake of expediency. Through the use of ten-fold cross-validation, the ideal parameter value C *and* C_1, C_2, C_3, C_4 is obtained from 10^{-5} to 10^5 as well as varying pinball loss parameters ξ_k (k =1,2) from the set of {0.5,0.8} and the Huber loss τ. For pin-TSVM, we have additionally computed λ using the set {(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)*C1}. For SPTWSVM, the value of ϵ is taken into account from the interval {0, 0.05, 0.1, 0.2, 0.3, 0.5}. Additionally, the data is often standardized to the interval [0,1]. To determine which of the above methods provide the best results, we use the relevant assessment criteria.

5.1.3.1. ARTIFICIAL DATASET

We begin by running our proposed approach, On a hypothetical dataset we refer to as the "synthetic" dataset, Spin-FITBSVM works with SVM, TSVM, pin-TSVM, pin GTSVM, GHTSVM, and SPTWSVM. With 1,000 data samples altogether, this dataset includes a binary class that corresponds to the following definition: for "+" class $x_1 \in \left[-\frac{\pi}{2,2\pi}\right], -\left(\frac{1}{4}\right) + \sin x_1 \leq x_2 \leq \left(\frac{1}{4}\right) + sinx_1, for\text{-}class\ x_1 \in \left[-\frac{\pi}{2,2\pi}\right], -1.35 + \left(\frac{3}{5}\right) \times sin\left(\frac{x_1}{1.05} + 0.4\right) \leq x_2 \leq -0.85 + (3/5) \times sin\left(\frac{x_1}{1.05} + 0.4\right)$ respectively with added noise $\omega = N(0,0.1^2)$.

Table 5. 2 Features of the synthetic dataset in Spin-FITBSVM

Synthetic data	Mean value	Covariance matrix	Total no. of samples
Positive data	$\begin{bmatrix} 2.3674 \\ -0.1262 \end{bmatrix}$	$\begin{bmatrix} 5.1586 & -0.3724 \\ -0.3724 & 0.5159 \end{bmatrix}$	500
Negative data	$\begin{bmatrix} 2.3544 \\ -1.1513 \end{bmatrix}$	$\begin{bmatrix} 5.2081 & -0.3763 \\ -0.3763 & 0.1977 \end{bmatrix}$	500

Table 5.2 provides a wealth of statistical information. These characteristics include the total number of samples for each class, the mean value, and the covariance matrix, in

that order. The following methods' generalization performance on synthetic datasets employing the Gaussian kernel is described by the numerical experiment results: With SVM, TSVM, GHTSVM, SPTWSVM, pin-TSVM, pin-GTSVM, and spin-FITBSVM. Because it employs an iterative way to derive the answer, Spin-FITBSVM has a quicker learning speed compared to other approaches.

5.1.3.2 REAL-WORLD DATASETS

Using twenty-four real-world benchmark datasets, we ran many tests to verify the efficacy of the suggested approach. Cleveland, WPBC, Asian Credit, and Australian Credit are some of the databases that are available. Among the many areas that have been investigated are bands, cryotherapy, dermatology, the Indian Liver Patient Dataset (ILPD), and breast tissue. Yeast-2 vs. 4, Ecoli2, Glass4, Yeast2 vs. 8, and Autism Adolescent Data are all variables that may be compared with Glass-0-1-4-6_vs_2. The files Glass-0-1-6_vs_2, Ecoli-0-2-6-7_vs_3-5, and ecoli0137vs26 were retrieved from the UCI repository. As part of the non-linear case, you may find 04clover5z-600-5-50-BI, 04clover5z-600-5-70-BI, 04paw02a-800-7-30-BI, 04clover5z-800-7-50-BI, 03subcl5-600-5-50-BI, and 04clover5z-600-5-0-BI. Additionally, we tallied a few statistics using datasets derived from real-world sources. This definition is applied to all real-world datasets in order to normalize them:

$$\bar{x}_{ij} = (x_{ij} - (\min_{i=1}^{p}(x_{ij})))/((\max_{i=1}^{p}(x_{ij})) - (\min_{i=1}^{p}(x_{ij}))),$$

For each input sample element xij, where i is an integer between 1 and j, $B, min_{i=1}^{p}(x_{ij})$ is the smallest value in the jth column of the sample input $_{i=1}^{p}(x_{ij})$ is the maximum value of the j[th] column of the input sample B.

We provide the results for the non-linear case with 10% noise and with no noise at all. The generalization of the Spin-FITBSVM is on par with or better than that of SVM, TSVM, pin-TSVM, GHTSVM, and SPTWSVM on ten datasets, including Cleveland, WPBC, Glass-0-1-5_vs_2, Glass2, Yeast2vs8, Breast Tissue, Indian Liver Patient Dataset (ILPD), Bands, ecoli0137vs26, and 04paw02a-800-7-50-BI. When looking at computation time across all datasets, Spin-FITBSVM stood up as the quickest approach. Figure 5.2 is a bar graph rendering the average training speed rankings that

we created. All of the datasets for additional pertinent methods are included in this graph.

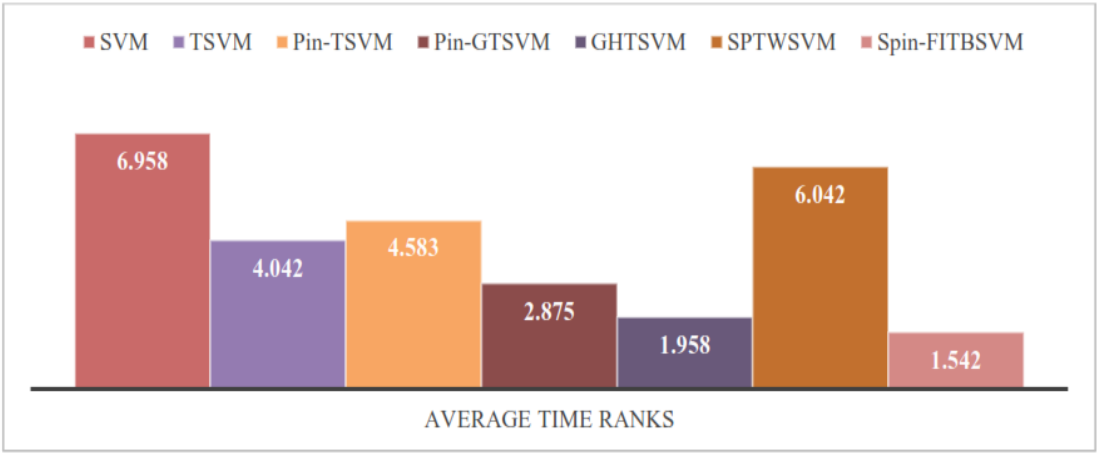


Figure 5. 2 Time series plot of Spin-FITBSVM and other models using noise-free Gaussian kernels applied to UCI benchmark real-world datasets

As shown in Figure 5.2, SpinFITBSVM outperforms competing virtual machines in terms of computation speed. The suggested approach Spin-FITBSVM has a wider gap between it and SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM when comparing the average ranks of several techniques. With data that contains 10% corrupted noise, we also tested our suggested method, SpinFITBSVM, with SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM. The presentation covers training time, optimal parameters, and classification accuracy for data with 10% noise concentration. The ranks were determined by averaging all datasets. On sixteen datasets, including Australian Credit and WPBC, the SpinFITBSVM outperforms or is on par with SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM, according to experimental results, including 5.8. The following groups were considered: Bands, Dermatology, Indian Liver Patient Dataset (ILPD), Spectrum Disorders in Children and Adolescents, and Glass-0-1-4-6_vs_2 are all part of it. The 04clover5z-600-5-50-BI, 03subcl5-600-5-50-BI, and 04paw02a-800-7-30-BI data sets include 10% corrupted noise. The proposed Spin-FITBSVM exhibits substantially improved performance on all datasets exhibiting 10% corrupted noise in comparison to earlier approaches. To display the average ranks of computing speed, we have generated a single bar graph, which can be seen in Figure 5.3. In this network, you may

see every dataset for every interesting approach. The training speed of Spin-FITBSVM is much faster than that of earlier reported methods (Figure 5.3).

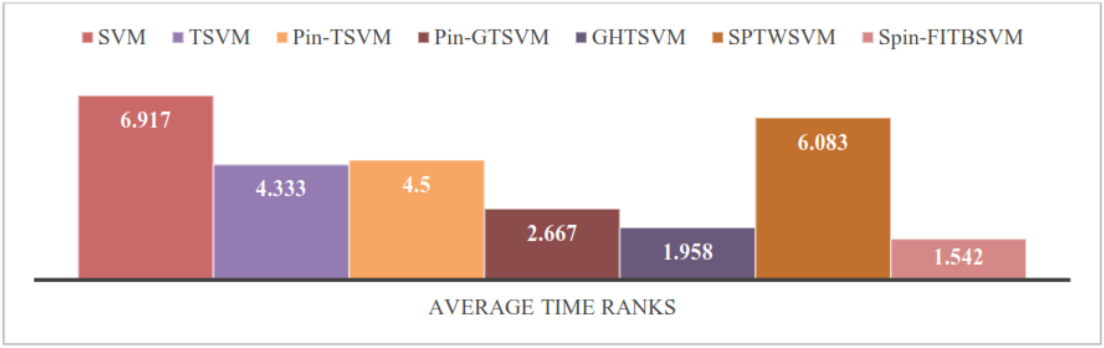


Figure 5. 3 Datasets used as benchmarks by UCI, containing a Gaussian kernel and 10% noise, and time graphs for several models, including Spin-FITBSVM

Furthermore, Figure 5.4 showcases a boxplot that rates the accuracy of all the approaches tested on datasets that were 10% noisy.

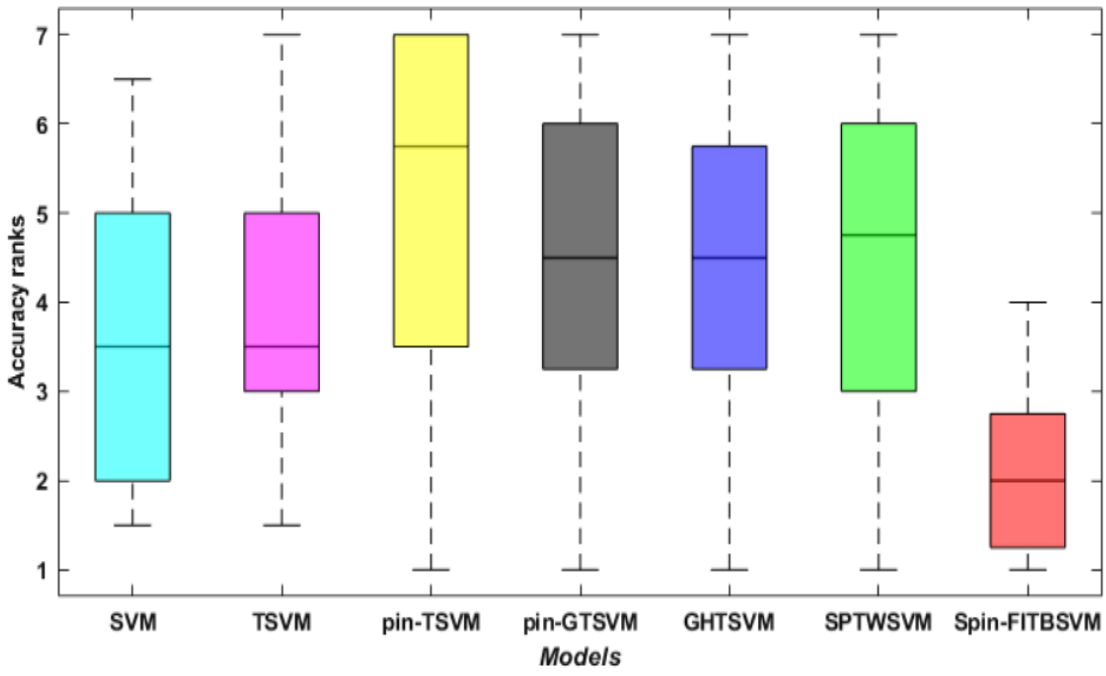


Figure 5. 4 Comparison between Spin-FITBSVM and competing models' accuracy ranks on UCI benchmark real-world datasets trained with a Gaussian kernel and 10% noise

See Figure 5.5 for the suggested approach Spin-FITBSVM's convergence graph.

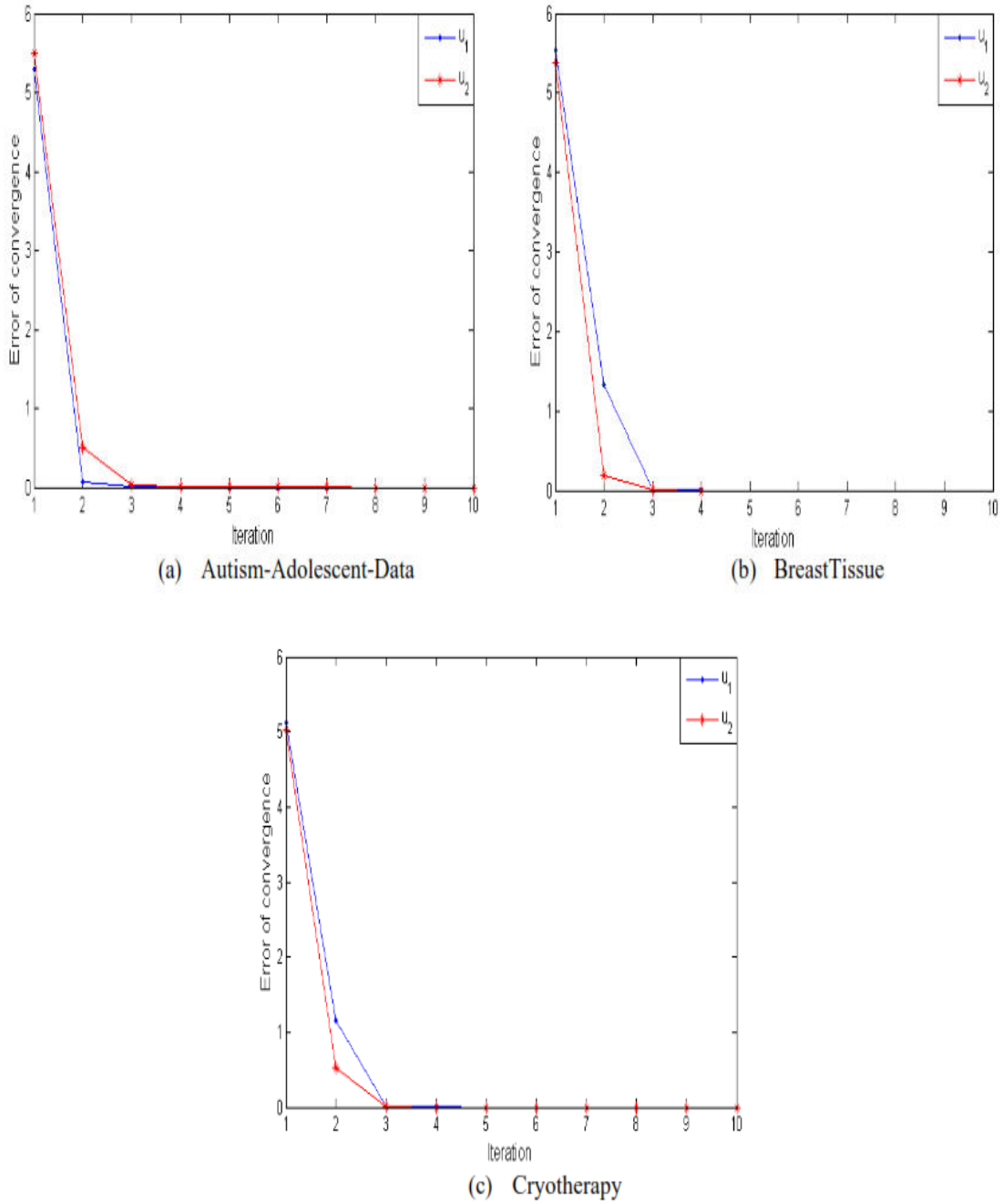


Figure 5. 5 Testing the proposed Spin-FITBSVM method on three real-world datasets: (a) Autism-Adolescent-Data, (b) BreastTissue, and (c) Cryotherapy to determine its noise-free convergence.

This graph is created using data that is free of noise, namely from the Autism-Adolescent-Data, Breast Tissue, and Cryotherapy datasets. Figure 5.6 concludes with a convergence graph displaying a number of real-world datasets, such as Breast Tissue, Cryotherapy, and others, and Yeast-2_vs_4.

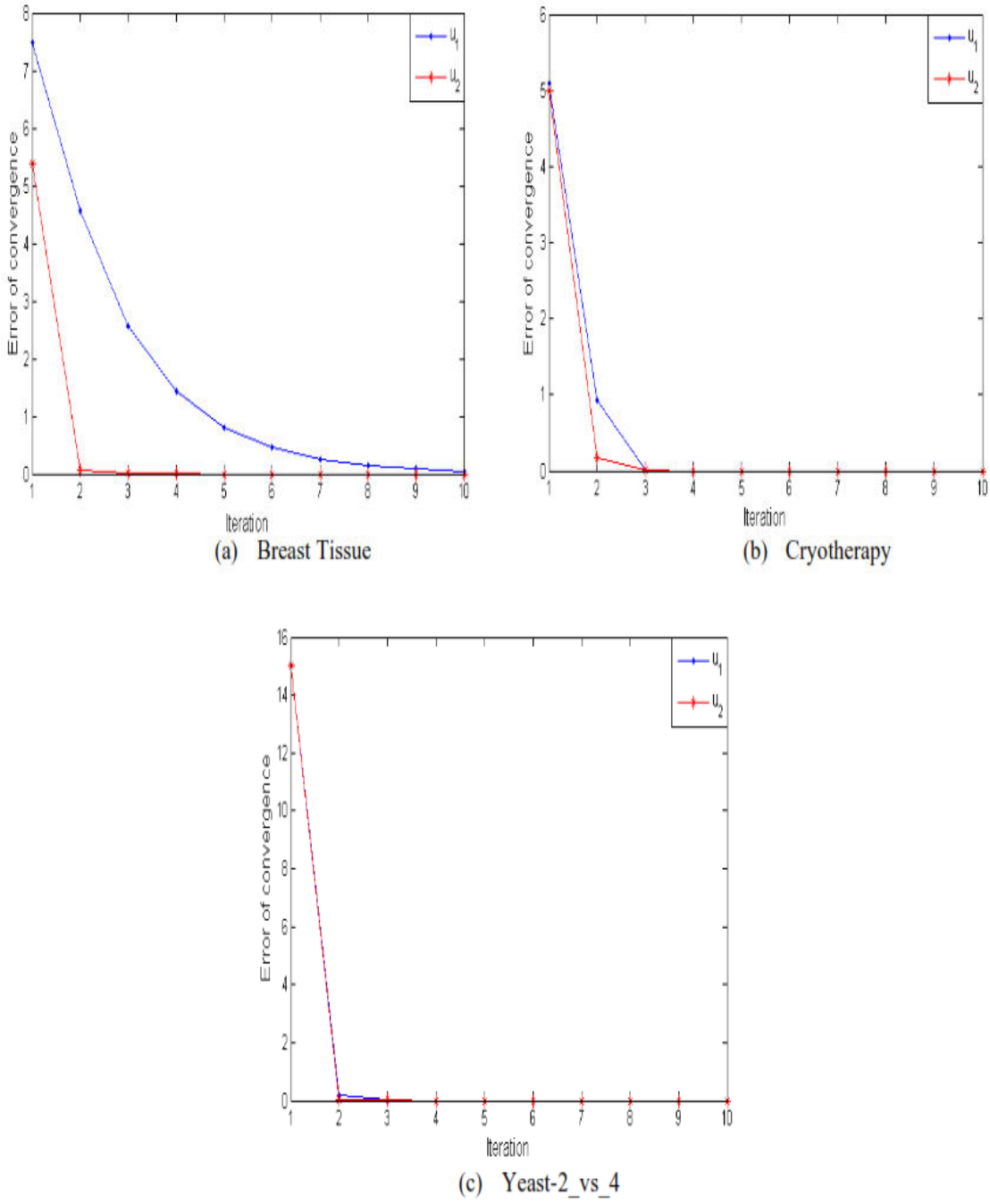
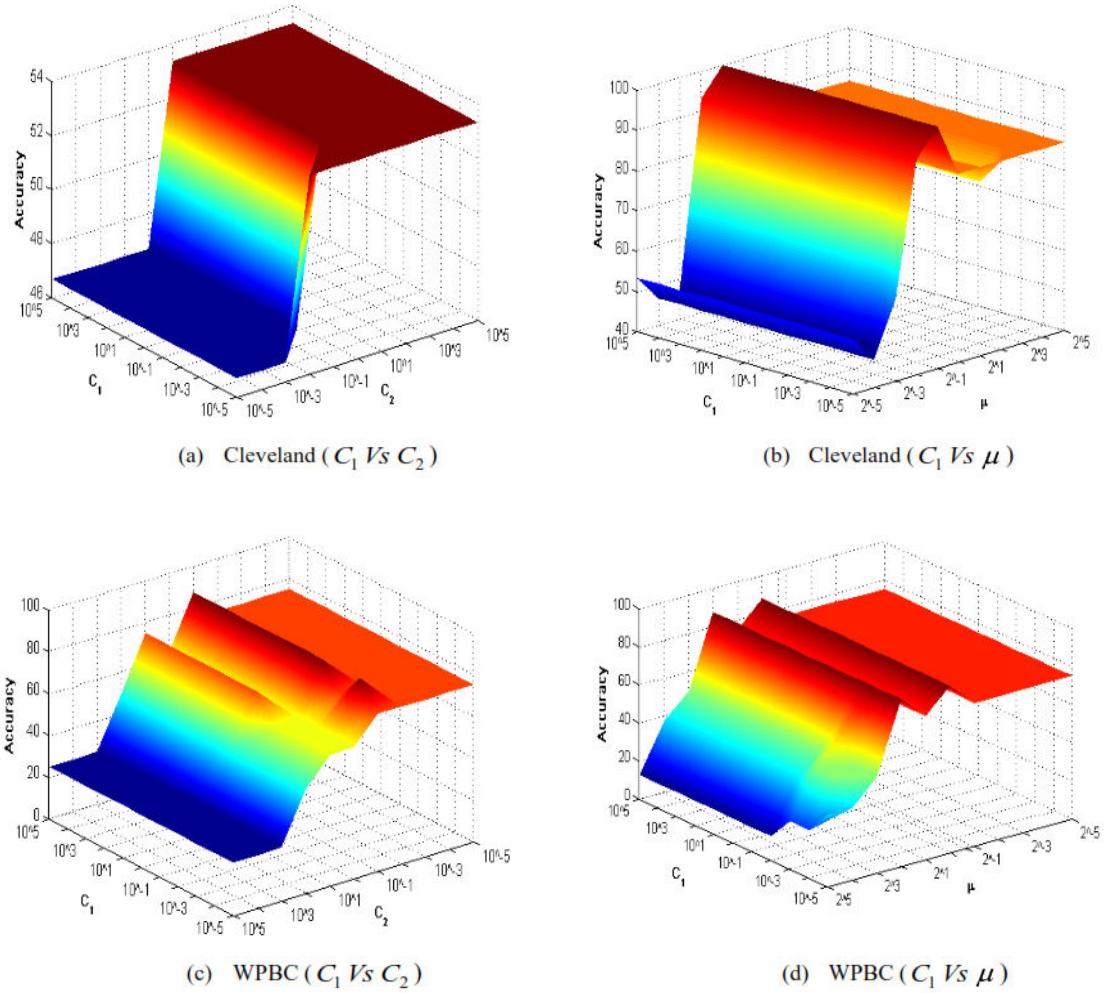


Figure 5. 6 On real-world datasets (a) breast tissue, (b) cryotherapy, and (c) yeast-2 vs. 4, the suggested Spin-FITBSVM method converges with 10% noise using the Gaussian kernel function.

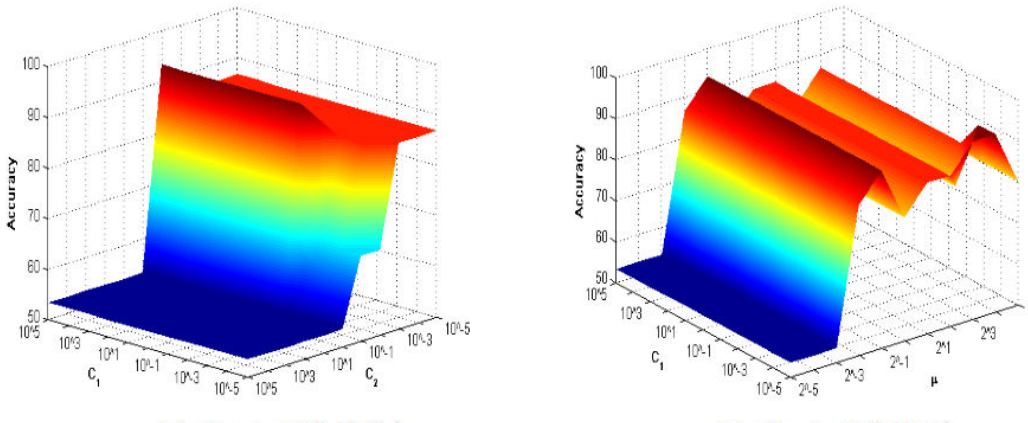
The data in this graph is assumed to have corrupted noise to the tune of 10%. Figures 5.5 and 5.6 show the solution acquired quickly, allowing one to analyze and watch how the convergence is carried out in the least feasible number of iterations. Shown in Figure 5.7-5.9 are the insensitivity graphs for the proposed Spin-FITBSVM and datasets with 0% and 10% noise, respectively.



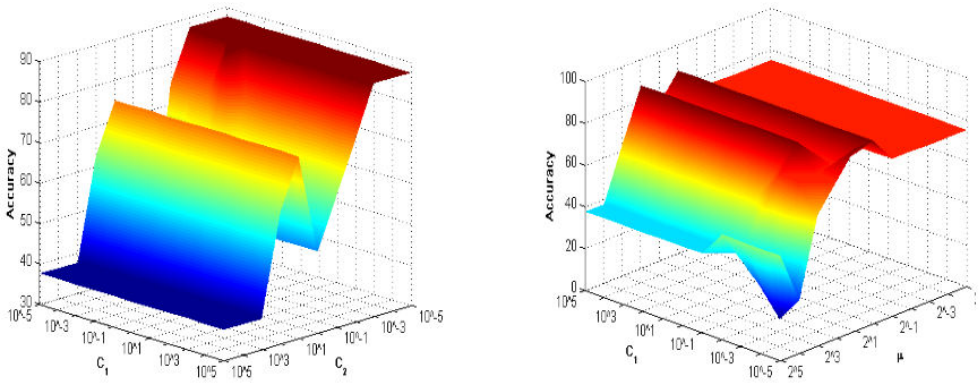
(a) Cleveland (C_1 *Vs* C_2) (b) Cleveland (C_1 *Vs* μ)

(c) WPBC (C_1 *Vs* C_2) (d) WPBC (C_1 *Vs* μ)

Figure 5. 7 Sensitivity plot of the suggested Spin-FITBSVM model for real-world datasets (a-b) with and without noise using the Gaussian kernel function Cincinnati (c-d) The WPBC

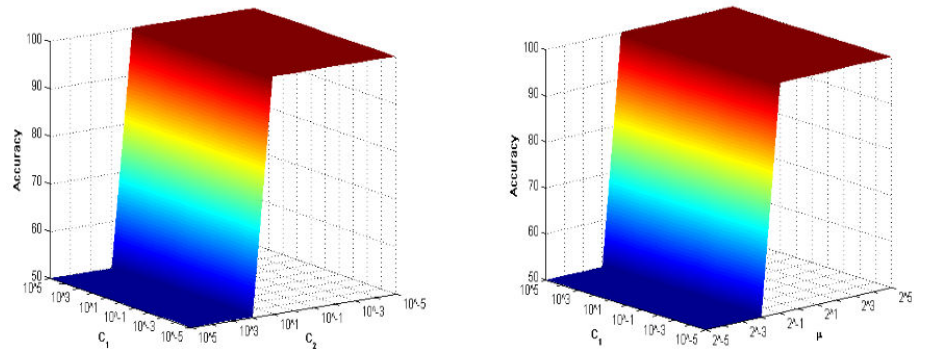


(a) Cleveland (C_1 *Vs* C_2) (b) Cleveland (C_1 *Vs* μ)

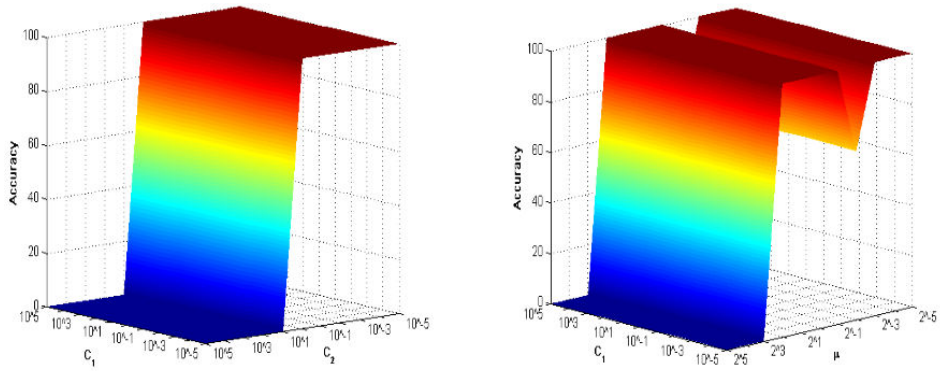


(c) WPBC (C_1 *Vs* C_2) (d) WPBC (C_1 *Vs* μ)

Figure 5. 8 Sensitivity analysis of the suggested Spin-FITBSVM with 10% noise and the Gaussian kernel function applied to real-world datasets (a-b) Cleveland is from c to d The WPBC



(a) Cryotherapy (C_1 *Vs* C_2) (b) Cryotherapy (C_1 *Vs* μ)



(c) Breast Tissue (C_1 *Vs* C_2) (d) Breast Tissue (C_1 *Vs* μ)

Figure 5. 9 Sensitivity analysis of the suggested Spin-FITBSVM with 10% noise and the Gaussian kernel function applied to real-world datasets (a-b) Radiation treatment (c-d) Breast Cancer Patients

Figure 5.7 displays the sensitivity graph with zero noise for the Cleveland dataset (C_1 V_s C_2) and the WPBC dataset (C_1 V_s v). The sensitivity graph for datasets such as Cleveland, WPBC, Cryotherapy, and Breast Tissue is shown in Figure 5.8 for a 10% noise dataset, and for C_1 V_s C_2 and C_1 V_s v, respectively, in Figure 5.9. Both the baseline and its noise-free versions are surpassed by our proposed method, as seen by the sensitivity graph. When compared to other objects, it is less affected by noise. While QPPs in SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM obtain faster and more generalized algorithms, we accomplish a simple and efficient approach by solving the optimization problem using a function iterative technique that employs squared pinball loss. The primary reason for this occurrence is this.

Statistical analysis

The non-parametric Friedman test will be applied to all seven algorithms and run on twenty-four real-world benchmark datasets. By comparing the results of seven separate algorithms, this test verifies the statistical significance. It is a trustworthy, practical, and easy-to-understand exam. Here we talk about the average rankings of all techniques that are significantly different from Spin-FITBSVM's average ranks.

Think about how the suggested method's mean rank differs greatly from the average rank of the interested methods and $R_{\Phi} = 1.916667$ eanticipated the non-linear case's null hypothesis to be: Its distribution is based on $x_F^2\ with\ (\ell - 1)$ with $(\ell - 1)$degree of freedom. N_0 is equal to the sum of all intriguing datasets.

$$\chi_F^2 = \frac{12\times 24}{7\times(7+1)}\left[3.75^2 + 3.89583^2 + 5.14583^2 + 4.375^2 + 4.4583^2 + 4.3125^2 + 2.0625^2 - \frac{7\times 8\times 8}{4}\right]$$

$$\chi_F^2 \approx 28.7411.$$

And $$F_F = \frac{(24-1)\times 28.7411}{(24\times(7-1)) - 28.7411} \approx 5.7353.$$

Taking into account the twenty-four real-world datasets and seven algorithms, the Friedman expression F_F is distributed with $\left((\ell - 1), \left((N_0 - 1) \times (\ell - 1)\right)\right) = \left((7-1), ((24-1)\times(7-1))\right) = (6{,}138)$ amount of leeway as determined by the F-distribution. The pivotal point at which $\mathrm{F}(6{,}138) = 2.1648\ and\ F(6{,}138) = 1.8172\ for\ \alpha_0 = 0.05$ and $\alpha_0 = 0.10$ respectively. No, we will not accept the null

hypothesis since the crucial values are 2.1648<F_F and 1.8172<F_F. To further compare all algorithms, we run them via the Nemenyi pairwise comparison test. The key difference is defined for this test: $= q_{\beta 0}\sqrt{\frac{\ell\times(\ell+1)}{6\times N_0}} = 2.693\sqrt{\frac{7\times(7+1)}{6\times 24}} \approx 1.6794$ for $\beta_0 = 0.10\, where\, q_{\beta 0}$ is considered based on $\ell\, and\, \beta_0$ from Demsar. We arrive at several intriguing decision points by running the Friedman and Nemenyi pairwise test, which are as follows:

- This is because the method found a critical difference, and as compared to the average rank of SVM, the suggested Spin-FITBSVM is 1.6875 places higher. The suggested Spin-FITBSVM may be able to outperform the SVM in terms of generalizability.

- We found that the projected critical difference was 1.83333 above 1.6794, which is bigger than the dissimilarity of 1.83333 with relation to the mean TSVM rank and the proposed Spin-FITBSVM. The results show that Spin-FITBSVM outperforms TSVM when dealing with noisy datasets.

- At 3.08333 and 2.3125, respectively, Comparing pin-TSVM and pin-GTSVM with SpinFITBSVM, the average rank difference exceeds the crucial difference of 1.6794. A significant departure from the crucial difference is shown by this. When it comes to noise-impacted datasets, the results show that Spin-FITBSVM is much more efficient than pin-TSVM and pin GTSVM.

    - The Spin-FITBSVM is used by both GHTSVM and SPTWSVM, with an average rank dissimilarity of 2.3958 and 2.25. At 1.6794, this discrepancy is far less than the essential one. Dealing with noisy datasets leads to the same conclusion: It's clear that Spin-FITBSVM outshines GHTSVM and SPTWSVM and the important difference is smaller.

We present a new technique for sample classification in noisy environments, Spin-FITBSVM, an iterative functional approach to twin bounded support vector machines, reimagines the classic twin model of SVM by using the squared pinball loss function. In the first case, we account for the regularization parameter in order to apply the SRM principle using our suggested approach, Spin-FITBSVM. The cherry on top is that it ensures Spin-FITBSVM achieves its theoretical robustness objectives. On several

datasets, including real-world benchmark and fake ones, as well as SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM, our proposed approach, Spin-FITBSVM, has been computationally compared to other methods that have been published. When compared to the other methods, Spin-FITBSVM uses less computational resources while yet producing better outcomes. The topic of parameter selection requires further research, but it will be addressed eventually. Adding support for scenarios involving many classes would be a nice bonus.

5.2 MACHINE FOR THE INTUITIVE FUZZY SET OF LAST SQUARES WITH TWO BOUNDARIES FOR SUPPORT VECTORS

Outliers and noise are more likely to affect the LS-SVM than the standard SVM. The reasoning for this is because LS-SVM uses hyper planes that are geographically close to the classes and integrates the least squares loss functions. LS-SVM, on the other hand, uses linear equation solutions rather than QPPs to lower the overall number of solutions. A training time that is as complicated as is practically possible. A further limitation of fuzzy-based support vector machines (SVM) in data classification is how well they can make up for the negative impacts of outliers and noise. Since the degree of membership function treats the support vectors of outliers as random noise, it specifies the distance from the input data to the class center in the sample space. The position of the input data in the feature space is irrelevant; this is accomplished. With the right membership function, fuzzy membership may make support vector machine (SVM) based techniques less noise sensitive by assigning membership values depending on the relevance or belongingness of samples to a given class. The best course of action is to choose a membership function. In this case, we circumvent the limitations by creating two acceptable alternatives: intuitionistic fuzzy least squares twin bounded support vector machines and intuitionistic fuzzy least squares support vector machines. Each of these variations derives its fuzzy value from intuitionistic fuzzy numbers using membership and non-membership functions. Finding simple solutions to systems of linear equations may be made easier, which can simplify training. This contradicts the method by which TWSVM acquires the answer by analyzing two instances of quadratic programming. The best techniques have been computationally tested under the non-linear condition on a wide range of publicly

accessible real-world benchmark datasets and artificially generated datasets. The research took into account a broad range of noise levels, from completely silent (0% noise) to severely disturbed (5% noise).

We present a twin model that, with careful noise mitigation, achieves better generalization performance than competing models while requiring much less training time.

The results of the models that are given are further validated by using quality indicators like AUC, F1-score, G-mean, and Precision Predictive Value (PPV).

5.2.1 CONCEPT OF INTUITIVE FUZZY NUMBERS (IFNS)

Functions related to membership and non-membership, along with an explanation of IFN, are laid down below:

1) Membership role: A definition of it is:

$$\psi(x_i)=\begin{cases}1-Z_1, & if\ y_i\ is\ negative\\ 1-Z_2, & if\ y_i\ is\ positive\end{cases},\quad i=1,\ldots,p \tag{5.18}$$

Where

$$Z_1=\frac{\|\varphi(x_i)-Z_C^-\|}{Z_r^-+\theta}\ \text{and}\ Z_2=\frac{\|\varphi(x_i)-Z_C^+\|}{Z_r^++\theta};\theta$$

represent positive variables; Z_r^+, Z_r^- as well as the negative and positive class radii, respectively, where

$$Z_r^+=\max\|\varphi(x_i)-Z_C^+\|,\ y_i=+1$$

And

$$Z_r^-=\max\|\varphi(x_i)-Z_C^-\|,\ y_i=-1;Z_C^+,Z_C^-$$

where the positive and negative classes' cores are located, respectively

$$Z_C^+=\frac{1}{p_1}\sum\varphi(x_i),\ \ y_i=+1\ \text{and}\ Z_C^-=\frac{1}{p_2}\sum\varphi(x_i),\ \ y_i=-1;p_1,p_2$$

represent the sum of all training data points, including positive and negative; ||. || shows how far away the input data sample is from the center of the relevant class, i.e.

$$Z_D = \| \varphi(x_i) - \varphi(x_j) \| = \sqrt{k(x_i, x_j) + k(x_j, x_j) - 2k(x_i, x_j)} .$$

2) Outside of membership: You may think of it as:

$$\tau(x_i) = (1 - \psi(x_i)) \frac{\left| \left\{ x_j \mid Z_D \le \theta, y_i \ne y_j \right\} \right|}{\left| \left\{ x_j \mid Z_D \le \theta \right\} \right|}, \tag{5.19}$$

where |. |represents the cardinality; $0 \le \Psi(x_i) + \tau(x_i) \le 1$;

As a concept, an intuitive fuzzy number (IFN) is:

$$IFN\ (\lambda_i) = (\psi(x_i), \tau(x_i)) = \begin{cases} \psi_i & , \quad \tau_i = 0 \\ 0 & , \quad \psi_i \le \tau_i \\ Z_o & , \quad \psi_i > \tau_i, \tau_i \ne 0 \end{cases}, \tag{5.20}$$

where

$$Z_o = \frac{1 - \tau_i}{2 - \psi_i - \tau_i} .$$

According to the integrated function, it is capable of handling noise and outliers and eliminating them from the support vectors.

Three distinct options are unlocked by this IFN.

It is easy to give the membership value to the data samples in Case 1 when the value of non-membership of one class is zero, since there is no neighborhood of another class.

In the second scenario, we see that noise exists whenever the value of not belonging to a class is greater than or equal to the value of being a member of that class.

Consequently, the IFN will be initialized to zero.

Case3: Support vectors, which are data samples located close to the non-membership value, are few in number if the value of non-membership is more than membership but less than zero.

5.2.2 IMPLICIT FUZZY MODELS THAT ARE SUGGESTED

5.2.2.1 THE IFLSSVM IS AN INTUITIVE FUZZY LEAST SQUARES SUPPORT VECTOR MACHINE.

Rezvani and colleagues have postulated a novel method for training the TWSVM model that makes use of the IFN of the data samples used for training. Exciting features of this information sharing network (IFN) boost the data samples' membership and value to those who do not join. Because of this, it is more resilient to the impacts of outliers and does better with noisy datasets. By using understandable fuzzy values, training data samples may be converted into weighted parameters. To determine how far away the class center is from the supplied data samples, one uses the degree of membership function. The function that specifies the degree of non-membership evaluates the connection between the total number of in harmonic samples and the available neighboring samples. It does a fantastic job at lowering noise and separating it from support vectors. Choosing a membership function that accounts for noise, however, is crucial. Training data located on the border between binary classes may include about the same amount of people from both the positive and negative categories, which is why misclassification happens. In order to address this, we compute and tag each data sample with an IFN; this signifies the degree of non-membership linked to the negative class and the degree of membership function linked to the positive class.

Using fuzzy weighted values and least squares, this work presents a support vector machine (SVM) model. Our model, which we refer to as IFLSSVM, is based on intuitionistic fuzzy numbers, which are useful for evaluating fuzzy membership. It is built from LS-SVM and IFTWSVM. Statistical learning theory's cornerstones must be upheld by the ideal IFLSSVM hyper plane. In this part, intuitionistic fuzzy numbers and slack variables are both considered. Finally, we convert the SVM's inequality criteria into equality constraints during this step. In the context of the non-linear situation, the proposed IFLSSVM is stated as:

Subject to.

$$\min_{w,b} \frac{1}{2}\| w\|^2 + \frac{C}{2}\left(\sum_{a=1}^{p} \lambda_a \zeta_a^2\right)$$
$$- y_a(\varphi(x_a)^t w + b) = \zeta_a - 1, \ \forall a = 1,..., p. \tag{5.21}$$

where C>0 is the penalty parameter; are $\zeta_a = (\zeta_{11}, \ldots, \zeta_{1p})^t$ non-linear mapping is represented by φx_a, and λ_a stands for intuitionistic fuzzy values; slack variables are characterized by this.

Following the implementation of Lagrange's multiplier $\alpha_a > 0$ in (5.21), written as (5.22)

$$L_g(w,b,\zeta_a,\alpha_a) = \frac{w^t w}{2} + \frac{C}{2}\left(\sum_{a=1}^{p} \lambda_a \zeta_a^2\right) - \sum_{a=1}^{p} \alpha_a \left(y_a(\varphi(x_a)^t w + b) - 1 + \zeta_a\right), \tag{5.22}$$

The next step is to find the unknowns' gradient of (5.22), which should equal zero.

$$\frac{\partial L_g(w,b,\zeta_a,\alpha_a)}{\partial w} = w - \sum_{a=1}^{p} \alpha_a y_a \varphi(x_a) = 0, \tag{5.23}$$

$$\frac{\partial L_g(w,b,\zeta_a,\alpha_a)}{\partial b} = -\sum_{a=1}^{p} \alpha_a y_a = 0, \tag{5.24}$$

$$\frac{\partial L_g(w,b,\zeta_a,\alpha_a)}{\partial \zeta_a} = C\lambda_a \zeta_a - \alpha_a = 0, \tag{5.25}$$

And,

$$\frac{\partial L_g(w,b,\zeta_a,\alpha_a)}{\partial \alpha_a} = y_a(\varphi(x_a)^t w + b) + \zeta_a - 1 = 0 \ \text{ for } \ a = 1,..., p. \tag{5.26}$$

The dual of the equation (5.21) may be expressed in this way by using the formula (5.23) -(5.26).

$$\begin{bmatrix} 0 & -Y_g^t \\ Y_g & Z_4 \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ P_g \end{bmatrix}, \tag{5.27}$$

Where,

$$Y_g = [y_1; \ldots; y_p];\ Z_4 = \left(Z_3 Z_3^t + \lambda^{-1} C^{-1} I\right); Z_3 = [\varphi(x_1)^t y_1; \ldots; \varphi(x_p)^t y_p];$$

I is the identity matrix with the right dimensions, and λ is the column vector with the majority class's fuzzy membership values. $P_g = [1; \ldots; 1]$ and $\alpha = (\alpha_1, \ldots, \alpha_p)^t$ Following is the sole way to invert the p × p matrix, as can be shown.

$$Z_4 = \left(Z_3 Z_3^t + \lambda^{-1} C^{-1} I\right) \tag{5.28}$$

and from (5.27) one can get the

$$\delta = Z_4^{-1}(P_g - bY_g) \text{ where } b = \frac{Y_g^t Z_4^{-1} P_g}{Y_g^t Z_4^{-1} Y_g}. \tag{5.29}$$

Using the answer from (5.28), the hyperplane representing the ultimate conclusion for every particular set of test data may be obtained.

The suggested IFLSSVM algorithm uses Sherman-Morrison-Woodbury (SMW)

By examining the SMW formula to reduce the training cost (Golub and loan) in (5.29), it is possible to efficiently evaluate the matrix inverse operation. Formula (5.28), which may be solved using the SMW formula, is described here. As shown in (8.14 5.31), this formula may be used to resolve the problem within the reduced dimensions. We have established the formula for SMW,

$$\left(\frac{I}{C} + AA^t\right)^{-1} = C\left(I - A\left(\frac{I}{C} + A^t A\right)^{-1} A^t\right). \tag{5.30}$$

Where *C* is any positive number; *A* is an arbitrary matrix.

The following expression, for example, may simplify (8.11)

$$\left(\frac{I}{\lambda C} + Z_3 Z_3^t\right)^{-1} = \lambda C\left(I - Z_3\left(\frac{I}{\lambda C} + Z_3^t Z_3\right)^{-1} Z_3^t\right), \tag{5.31}$$

5.2.2.2 THE TWIN-BOUNDED SUPPORT VECTOR MACHINE WITH INTUITIONISTIC FUZZY LEAST-SQUARES (IFLSTBSVM)

We provide a powerful approach that is built on TWSVM and follows a pattern seen in IFLSSVM. To control the impact of sounds while keeping processing costs low, this technique employs intuitionistic fuzzy membership values in conjunction with the principle of least squares.

Our suggested machine learning model is the IFLSTBSVM, which stands for intuitionistic fuzzy last squares twin bounded support vector machine., to get at the heart of the matter. Here is how the IFLSTBSVM's objective function is expressed:

$$\text{subject to: } \min_{w_1,b_1,\zeta_1} \frac{1}{2}\Big(C_1\lambda_2 \| \zeta_1 \|^2 + (K(B_1,B^t)w_1 + e_1b_1)^t (K(B_1,B^t)w_1 + e_1b_1) + C_3(w_1^t w + b_1^2) \Big)$$
$$-(K(B_2,B^t)w_1 + e_2b_1) + \zeta_1 = e_2, \tag{5.32}$$

And,

$$\text{subject to: } \min_{w_2,b_2,\zeta_2} \frac{1}{2}\Big(C_2\lambda_1 \| \zeta_2 \|^2 + (K(B_2,B^t)w_2 + e_2b_2)^t (K(B_2,B^t)w_2 + e_2b_2) + C_4(w_2^t w_2 + b_2^2) \Big)$$
$$(K(B_1,B^t)w_2 + e_1b_2) + \zeta_2 = e_1, \tag{5.33}$$

The column vectors Λ_i (i=1,2) obtained using the formula (5.18) -(5.20), denote the fuzzy membership values for positive and negative data points, respectively. In this case, e_1 represents the ones-vectors, and $\zeta_1, \zeta_2 > 0; C_1, C_2, C_3, C_4 > 0$. the user entered the settings.

The following form, representing the (5.32) and (5.33) as Lagrangian functions, allows us to reformulate them:

$$L_{g_1} = \frac{1}{2}(K(B_1,B^t)w_1 + e_1b_1)^t (K(B_1,B^t)w_1 + e_1b_1) + \frac{1}{2}C_1 \| \lambda_2(K(B_2,B^t)w_1 + e_2b_1) + e_2 \|^2 + \frac{1}{2}C_3(w_1^t w_1 + b_1^2) \tag{5.34}$$

And

$$L_{g_2} = \frac{1}{2}(K(B_2,B^t)w_2 + e_2b_2)^t (K(B_2,B^t)w_2 + e_2b_2) + \frac{1}{2}C_2 \| -\lambda_1(K(B_1,B^t)w_2 + e_1b_2) + e_1 \|^2 + \frac{1}{2}C_4(w_2^t w_2 + b_2^2) \tag{5.35}$$

In order to get to zero, we must first calculate the gradient of 5.34 and 5.35 with regard to the unknowns, which is (5.36), (5.37), and (5.38), (5.39):

$$\frac{\partial L_{g1}}{\partial w_1} = K(B_1, B^t)^t (K(B_1, B^t) w_1 + e_1 b_1) + C_1 K(B_2, B^t)^t \lambda_2^t \lambda_2 ((K(B_2, B^t) w_1 + e_2 b_1) + e_2) + C_3 w_1 = 0,$$

(5.36)

$$\frac{\partial L_{g1}}{\partial b_1} = e_1^t (K(B_1, B^t) w_1 + e_1 b_1) + C_1 e_2^t \lambda_2^t \lambda_2 ((K(B_2, B^t) w_1 + e_2 b_1) + e_2) + C_3 b_1 = 0,$$

(5.37)

And,

$$\frac{\partial L_{g2}}{\partial w_2} = K(B_2, B^t)^t (K(B_2, B^t) w_2 + e_2 b_2) - C_2 K(B_1, B^t)^t \lambda_1^t \lambda_1 (-(K(B_1, B^t) w_2 + e_1 b_2) + e_1) + C_4 w_2 = 0,$$

(5.38)

$$\frac{\partial L_{g2}}{\partial b_2} = e_2^t (K(B_2, B^t) w_2 + e_2 b_2) - C_2 e_1^t \lambda_1^t \lambda_1 (-(K(B_1, B^t) w_2 + e_1 b_2) + e_1) + C_4 b_2 = 0,$$

(5.39)

To get the best value for, now combine (5.36 and 5.37) into (5.40), and (5.38 and 5.39) into (5.41) w_1, b_1, w_2 and b_2,

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -C_1 \left(\begin{bmatrix} K(B_1, B^t)^t \\ e_1^t \end{bmatrix} [K(B_1, B^t) \quad e_1] + C_1 \begin{bmatrix} K(B_2, B^t)^t \\ e_2^t \end{bmatrix} \lambda_2^t \lambda_2 [K(B_2, B^t) \quad e_2] + C_3 I \right)^{-1} \begin{bmatrix} K(B_2, B^t)^t \\ e_2^t \end{bmatrix} \lambda_2^t \lambda_2 e_2,$$

(5.40)

And

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = C_2 \left(\begin{bmatrix} K(B_2, B^t)^t \\ e_2^t \end{bmatrix} [K(B_2, B^t) \quad e_2] + C_2 \begin{bmatrix} K(B_1, B^t)^t \\ e_1^t \end{bmatrix} \lambda_1^t \lambda_1 [K(B_1, B^t) \quad e_1] + C_4 I \right)^{-1} \begin{bmatrix} K(B_1, B^t)^t \\ e_1^t \end{bmatrix} \lambda_1^t \lambda_1 e_1,$$

(5.41)

Assume $D_1 = [K(B_1, B^t)\ e_1]$ and $D_2 = [K(B_2, B^t)\ e_2]$ are the augmented matrix and $\vartheta_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ and $\vartheta_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$ We can rewrite the equations (5.40) and (5.41) as

$$\vartheta_1 = -C_1 (D_1^t D_1 + C_1 D_2^t \lambda_2^t \lambda_2 D_2 + C_3 I)^{-1} D_2^t \lambda_2^t \lambda_2 e_2, \tag{5.42}$$

And,

$$\vartheta_2 = C_2(D_2^t D_2 + C_2 D_1^t \lambda_1^t \lambda_1 D_1 + C_4 I)^{-1} D_1^t \lambda_1^t \lambda_1 e_1 , \tag{5.43}$$

Applying this method to any set of test data will provide better positive and negative class hyper planes formulas to equations (5.42) and (5.43).

$$class\ \ k = \arg\min_{a=1,2} \left\{ \frac{|w_1^t K(x, B^t) + b_1|}{\sqrt{w_1^t K(B_1, B^t) w_1}}, \quad \frac{|w_2^t K(x, B^t) + b_2|}{\sqrt{w_2^t K(B_2, B^t) w_2}} \right\}. \tag{5.44}$$

Sherman-Morrison-Woodbury (SMW) for IFLSTBSVM algorithms

I. In the same way that the TWSVM model is unable to ineluctably perform the matrix inverse operation, neither is our suggested IFLSTBSVM (5.43). As a result, one order of standard two-matrix inverse operations entirely solves the linear IFLSTBSVM. $(q+1)$ twice, where q is shown as a feature space dimension. The dimensionality of the training samples is a good proxy for the computational cost of linear IFLSTBSVM.

II. Two matrix inversion operations of order may be seen for non-linear IFLSTBSVM $(p+1)$ twice is constructed, where p is stands for the sum of all training samples.

III. By taking the SMW formula into account, one may decrease the training cost and easily assess the matrix inverse operation in (5.42) and (5.43). We go over the SMW formula in (5), which may be used to solve equation (5.42) in lower dimensions like $(p_1 \times p_1)\ and\ (p_2 \times p_2)$ where $p_1 \ll p\ and\ p_2 \ll p$,respectively as shown in (5.46).

SMW formula is defined as,

$$\left(\eta + \omega\rho^t\right)^{-1} = \eta^{-1} - \eta^{-1}\omega(I + \rho^t\eta^{-1}\omega)^{-1}\rho^t\eta^{-1}.$$

(5.45)

For instance, (5.42) can be simplified via following formulation

$$\left(D_1^t D_1 + C_1 D_2^t \lambda_2^t \lambda_2 D_2 + C_3 I\right)^{-1} = \left(Z_5 - Z_5 D_2^t \lambda_2^t \left(\frac{I}{C_1} + \lambda_2 D_2 Z_5 D_2^t \lambda_2^t \right)^{-1} \lambda_2 D_2 Z_5 \right),$$

(5.46)

$$\text{where } Z_5 = \frac{1}{C_3 I}\left(I - D_1^t \left(C_3 I + D_1 D_1^t \right)^{-1} D_2 \right).$$

(5.47)

Similarly, one can find the simplified formula of (5.43) using SMW formula.

$$\left(D_2^t D_2 + C_2 D_1^t \lambda_1^t \lambda_1 D_1 + C_4 I \right)^{-1} = \left(Z_6 - Z_6 D_1^t \lambda_1^t \left(\frac{I}{C_2} + \lambda_1 D_1 Z_6 D_1^t \lambda_1^t \right)^{-1} \lambda_1 D_1 Z_6 \right),$$

(5.48)

$$\text{where } Z_6 = \frac{1}{C_4 I}\left(I - D_2^t \left(C_4 I + D_2 D_2^t \right)^{-1} D_1 \right).$$

(5.49)

Discussion:

1. Considering that it is well-known that SVM and TWSVM models are very vulnerable to data points with noise. Two methods that mitigate noise are IFLSSVM and IFLSTBSVM.

2. Our proposed approaches, IFLSSVM and IFLSTBSVM, aim to capture the essence of statistical learning and are based on the SRM principle, in contrast to IFTWSVM.

3. To reduce training costs as much as possible, we have used the Sherman-Morrison-Woodbury (SMW) formula to find the matrix's inverse in the proposed IFLSSVM and IFLSTBSVM. The IFTWSVM, on the other hand, used the SMW formula.

4. Our models, which are similar to IFTWSVM, calculate the gap between the class center and the training example. Additionally, they determine the association between in harmonic examples and the examples in their neighborhood. This results in improved binary classification generalization performance.

5. Intuitionistic fuzzy numbers are central to the suggested noise models and contribute significantly to their improved theoretical understanding.

6. Finally, the suggested IFLSSVM and IFLSTBSVM are shown to be effective and practical by conducting extensive numerical experiments on a range of simulated and real-world datasets with varying degrees of noise, namely 0% (noise-free) and 5%

(noise-corrupted). Furthermore, it is statistically analyzed using the Friedman and Wilcoxon signedrank test in combination with other relevant approaches.

5.2.3 NUMERICAL EXPERIMENTS

Here we examine the outcomes on 39 benchmark real-world datasets housed in the UCI repository as well as 7 synthetic datasets housed in the KEEL repository. We also consult 2moons, synthetic datasets, and Ripley's. We want to prove that IFLSSVM and IFLSTBSVM work. A personal PC running Windows 10 Pro and MATLAB R2008b are used for all of the tests. A 64-bit OS with an x-64 based processor, 32 GB of installed RAM, and an Intel®CoreTM i7-8700 CPU running at 3.20 GHz make up the system setup. We compared the proposed IFLSSVM and IFLSTBSVM techniques' classification performance to that of existing baseline approaches using the 5-fold cross-validation method. Using a random number generator, we split the datasets into five equal parts; we train on four of the parts and test on the fifth. Area under the curve (AUC), among other crucial performance evaluation measures, is calculated by averaging the results of all five rounds of this technique. F1-score, G-mean, and Positive Predictive Value (PPV). Up to five iterations of this process are possible. Improvements in recall and specificity are achieved by the use of the following parameters: AUC, F1-score, G-mean, and Positive Predictive Value (PPV). This is achieved by making use of predetermined parameters.

The choice of the Gaussian RBF kernel to handle the non-linear case is expressed as $K(x_d, x_e) = exp^{-(\|x_d - x_e\|^2/2\mu^2)}$ where x_d and x_e constitutes samples of any kind of data. To get the most performance out of any method, parameter selection is key. We have chosen the best value for the penalty parameters a,O1,a3 from a large range in our testing setup $\{10^k | k = -3, -2, \dots, 2, 3\}$ and Gaussian RBF kernel parameter μ from the set $\{2^k | k = -5, -3, -2 \dots, 2, 3\}$ respectively. For the EFLSSVM model, the kNN is stable to the value of 5 as well as the value of the adjustable parameter $\theta > 0$ is selected from the range $\{0.01, 0.1, 0.15, 0.2, 0.25, 0.4\}$. To minimize the running cost of parameter selection, we have taken $C_1 = C_2 \; and \; C_3 = C_4$ respectively. All the data samples are normalized between 0 and 1.

The research was carried out using 39 publicly available, real-world datasets housed in the UCI repository. We did this to show that the proposed models had better

categorization performance. Check out the KEEL collection for purposefully created imbalance datasets, as well as Ripley's, synthetic datasets, and 2moons fake datasets, and see how well the proposed models IFLSSVM and IFLSTBSVM perform. The rundown of all the real-world benchmark datasets and the purpose-built examples. The unbalanced ratio is characterized by as $IR = (E_4 \div E_5)$ where $E_4 =$ the total major class sample and $E_5 =$ total minor class sample.

Real-world Datasets

Here, we do tests to prove the validity and logic of our technique, and we put our approach to benchmarking real-world datasets with different levels of considerable noise to the test. Using the ideal parameter, such as area under the curve (AUC), F1-scores, G-mean, and computation time, all of the provided strategies were evaluated on datasets that were free of noise and datasets that had noise damaged.

Methods are ranked according to their area under the curve (AUC) relative to real-world benchmark datasets. With an area under the curve (AUC) of 19, an F1-score of 13, and a G-mean of 13, the technique IFLSTBSVM clearly obtains the best generalization performance on 0% (noise-free) datasets. This remains true despite the fact that, on average, it ranks worse than competing methods.

Among the 39 datasets tested, it is evident that IFLSTBSVM achieves the highest ranks in terms of AUC, F1-score, and G-mean on 5% noise corrupted datasets. Datasets contaminated by noise undergo this process. For different significance levels, such as 0% (noise-free) and 5% (noise-damaged), we have also calculated the average ranking of F1-score, G-mean, and Positive Predictive Value (PPV). The outcomes of our calculations are shown here. The results allow us to conclude that our approach, IFLSTBSVM, outperforms the previously specified models.

Figure 5.10 and Figure 5.11 illustrate the area under the curve (AUC) for all of the applied models as a boxplot on benchmark real-world datasets that were noise-free and datasets that were noise-corrupted, respectively. In order for the data to be visually comprehended, this was done. Figures 5.10 and 5.11 show that compared to previous models, our IFLSTBSVM model has a much greater area under the curve (AUC).

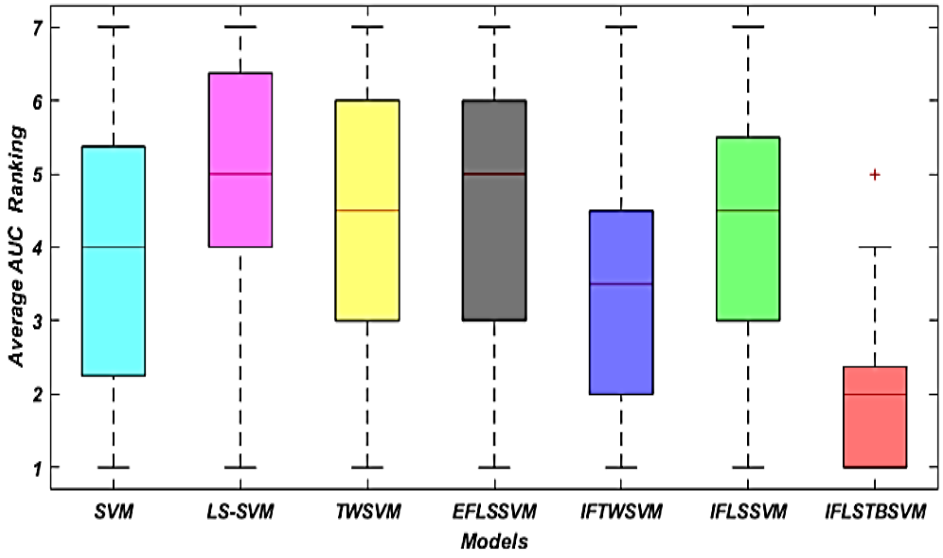


Figure 5. 10 Box plot of the value of AUC of IFLSTBSVM and other models on benchmark real-world datasets at 0% noise significant level.

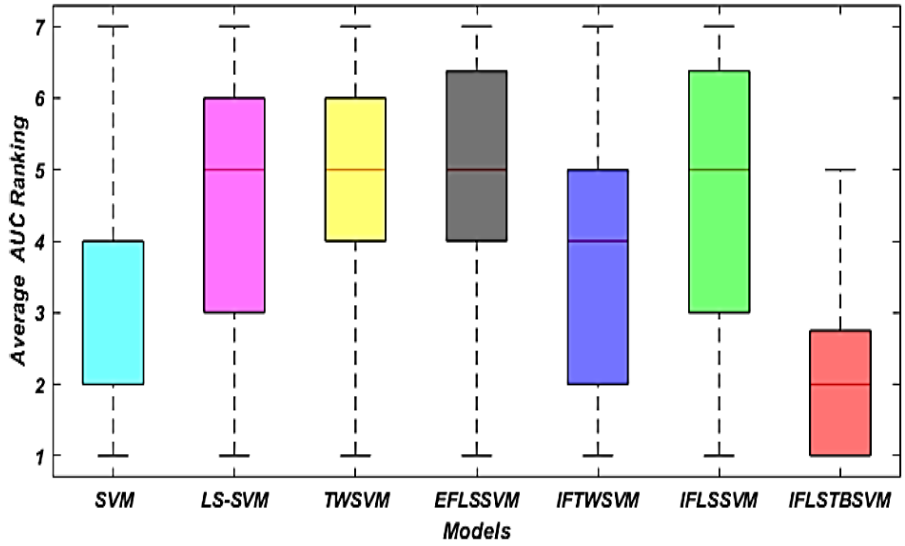


Figure 5. 11 Box plot of the value of AUC of IFLSTBSVM and other models on benchmark real-world datasets at 5% noise significant level.

In terms of generalizability, this shows that our proposed model is acting in a promising way. As seen in Figures 5.12 and 5.13 with varying degrees of statistical significance, the bar chart displaying the F1-score, G-mean, and positive predictive value (PPV) likewise demonstrates that IFLSTBSVM outperforms other interesting methods. The purpose of this is to ensure that the data is checked thoroughly. It also suggests

something similar, namely that the suggested IFLSTBSVM model is the better choice when it comes to classification.

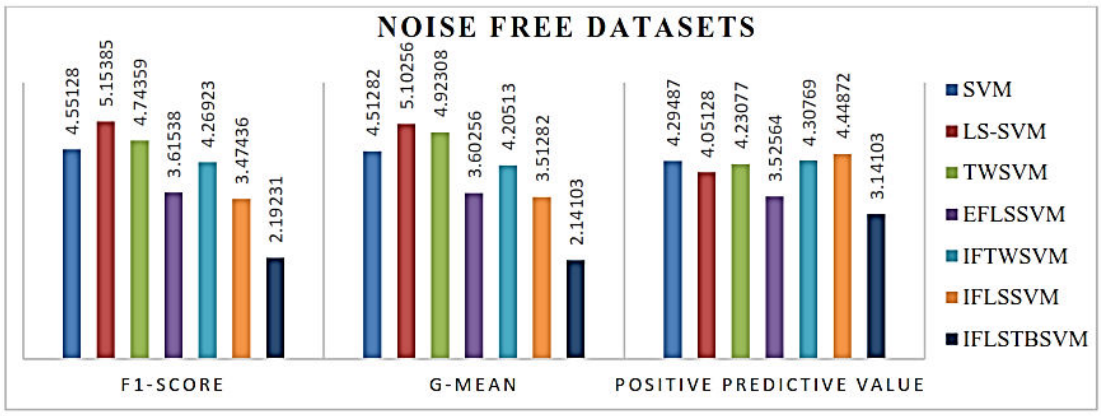


Figure 5. 12 Bar Graph of average F1-score, G-mean, and Positive Predictive value ranking of IFLSTBSVM and other models on benchmark real-world datasets at 0% noise significant level

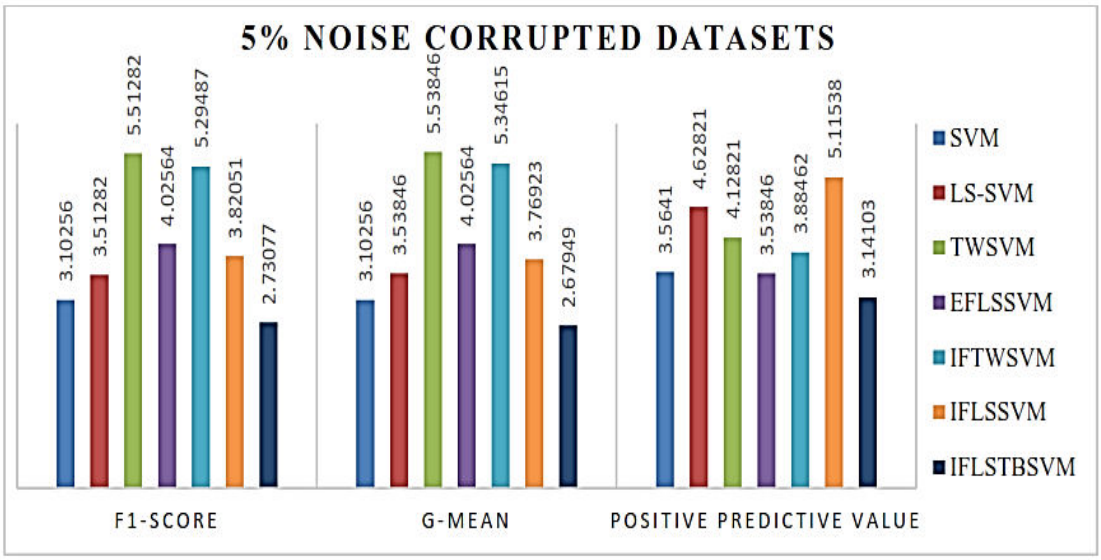


Figure 5. 13 Bar Graph of average F1-score, G-mean, and Positive Predictive value ranking of IFLSTBSVM and other models on benchmark real-world datasets at 5% noise significant level.

Insensitiveness performance graph

Our IFLSSVM and IFLSTBSVM models, which are less parameter sensitive, allow us to concentrate on their performance in this context. C_1, C_3 and μ. Plotting the

insensitivity graph of the suggested IFLSSVM with a range of parameters that the user specifies helps in understanding it better C and μ and for the suggested IFLSTBSVM model on Cleveland using parameters, as shown in Figure 5.14 (a)-(c) accordingly, on Ecoli, Monk2, and Ecoli0-2-6-7_vs_3-5 C_1 and μ in Figure 5.15 (a), based on parameters C_3 and μ in Figure 5.15 (b), and based on parameters C_1 and C_3 on the right side of Figure 5.15 (c), each. Based on what is shown in Figures 5.14 (a)–(c) and 5.15 (a)–(c), it is not necessary to choose the very high or very low value of C_1, and C_3 in order to get a higher AUC. It follows that the parameters C_1 and C_3 impact our IFLSSVM and IFLSTBSVM models' binary classification performance to a lesser extent.

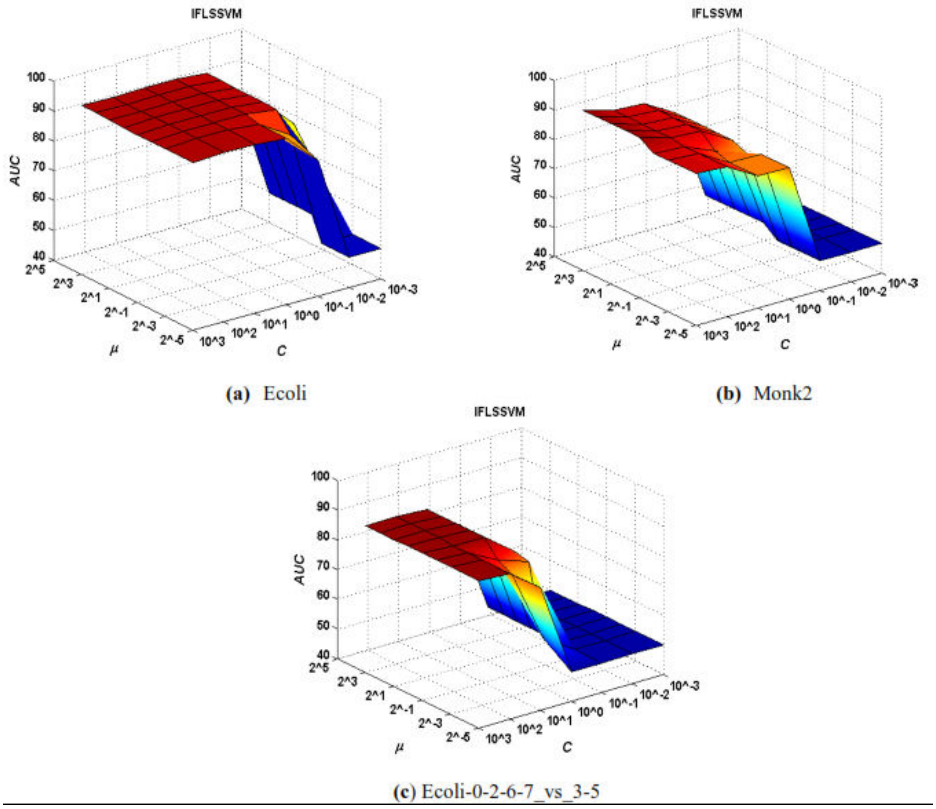


Figure 5. 14 The sensitivity plot of the proposed IFLSSVM model based on real-world datasets such as (a) E-coli (b) Monk2 (c) Ecoli-0-2-6-7_vs_3-5 for noise-free datasets

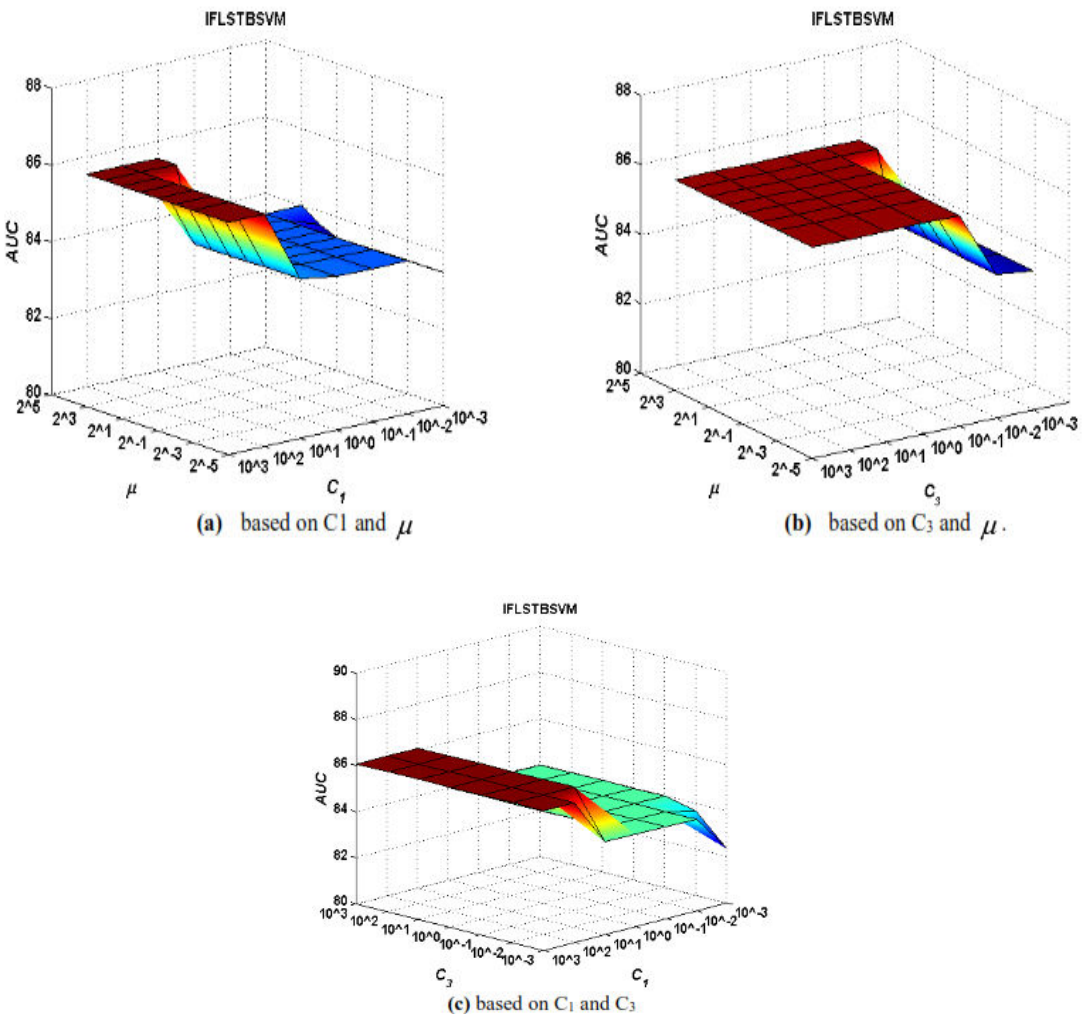


Figure 5. 15 The insensitiveness performance graph of proposed IFLSTBSVM model on Cleveland real-world datasets for noise free datasets

Statistical Friedman and Nemenyi test

We used benchmark real-world datasets with 0% (noise-free) and 5% (noise corrupted) noise to run the Friedman statistics with Nemenyi test on the non-linear kernel and confirm IFLSTBSVM's performance statistically. The non-linear kernel may then be used, therefore this was done. What we have here with Friedman's test is a simple rank-based non-parametric statistical approach. Our ability to show, with the use of this famous test, that the calculated results are significantly different.

a) Noise-free datasets

The null hypothesis can be described using AUC as follows:

$$\chi_F^2 = \frac{12\times 39}{7\times(7+1)}\left[\left(3.85897^2 + 5.0641^2 + 4.44872^2 + 4.73077^2 + 3.60256^2 + 4.32051^2 + 1.97436^2\right) - \left(\frac{7\times(7+1)^2}{4}\right)\right]$$

$$\chi_F^2 \approx 52.2438.$$

and

$$F_F = \frac{(39-1)\times 52.2438}{(39\times(7-1))-52.2438} \approx 10.9227.$$

F(6,228) has a critical value of around 2.13849 and 1.800 at the probability level, as can be shown $\Lambda_F = 0.05$ and $\Lambda_F = 0.10$ in that order according to the F-distribution critical value table. The calculated outcomes reveal a notable disparity across the seven algorithms as a consequence of the actual value of $F_F > 2.13849$ and $F_F > 1.800$. To execute this test, one may find the crucial difference by comparing the mentioned algorithms with the suggested method IFLSTBSVM in a paired fashion. The results are considered to be subjected to the Nemenyi test.

$$2.693\sqrt{\frac{7\times(7+1)}{6\times 39}} = 1.3174$$

when the significance threshold is set at Φ=0.1. The following may be emphasized with great clarity from this Nemenyi test:

a) Results from seven different algorithms when compared to IFLSTBSVM are significantly different, with values of (1.88461), (3.08974), (2.47436), (2.75641), (1.6282), and (2.34615) respectively, more than the essential disparity of 1.3174. In terms of generalization performance, IFLSTBSVM beats SVM, LS-SVM, TWSVM, EFLSSVM, and IFTWSVM for datasets free of noise in terms of AUC parameter. b) The non-linear scenario includes SVM, LS-SVM, TWSVM, EFLSSVM, IFTWSVM, and IFLSSVM; our solution, IFLSTBSVM, ranks lower, indicating that it is not relevant for parameter AUC at the 0% (noise-free) significant level. A comparison reveals that IFLSSVM, IFTWSVM, IFLSSVM, and IFLSTBSVM all perform poorly.

c) Looking at Figure 5.16 may give you a good indication of how different the proposed model is from the ones that have been described before.

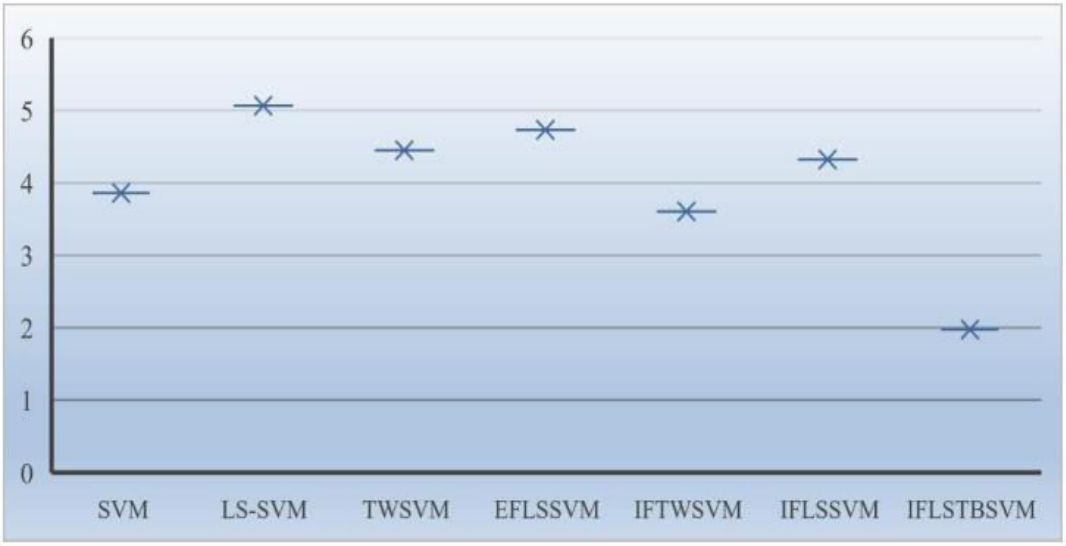


Figure 5. 16 Friedman test for noise-free datasets

b) Noise corrupted datasets

In a similar vein, the following is how the null hypothesis might be represented using AUC:

$$\chi_F^2 = \frac{12\times 39}{7\times(7+1)}\left[\left(3.32051^2 + 4.62821^2 + 4.92308^2 + 5^2 + 3.62821^2 + 4.5^2 + 2^2\right) - \left(\frac{7\times(7+1)^2}{4}\right)\right]$$

$$\chi_F^2 \approx 59.3085.$$

And

$$F_F = \frac{(39-1)\times 59.3085}{(39\times(7-1)) - 59.3085} \approx 12.9012.$$

Value F(6,228), the crucial point, is lower than FF. Thus, the claimed algorithms are really different from one another. As a consequence, at Φ=0.10When comparing two sets of data, the Nemenyi test is used; a critical difference of 1.3174 is considered significant. To find out how IFLSTBSVM stacks up against other published methods, Here are the area under the curve (AUC) values for SVM, LS-SVM, TWSVM, EFLSSVM, and IFTWSVM: (1.32051), (2.62821), (2.92308), (3), (1.62821), and (2.5), respectively. Our method outperforms the competition in terms of area under the curve (AUC), and this holds true even on datasets contaminated with 5% noise. We may see

this in the fact that the rank difference is far more substantial than the crucial difference. In 39 instances of the non-linear kernel for parameter AUC, the highest scorers were IFLSTBSVM with SVM, LS-SVM, TWSVM, EFLSSVM IFTWSVM, and IFLSSVM, in that order. It seems to be the best option if we use a 0.05 significance threshold and consider datasets that include noise. The proposed IFLSTBSVM stands apart from other existing systems, as seen in Figure 5.17.

The Wilcoxon signed-rank test is used statistically.

In order to back up the data interpretation using statistics, an additional statistical approach used is the Wilcoxon signed-rank test. The goal is to find out how IFLSTBSVM differs significantly from the other methods that have been mentioned.

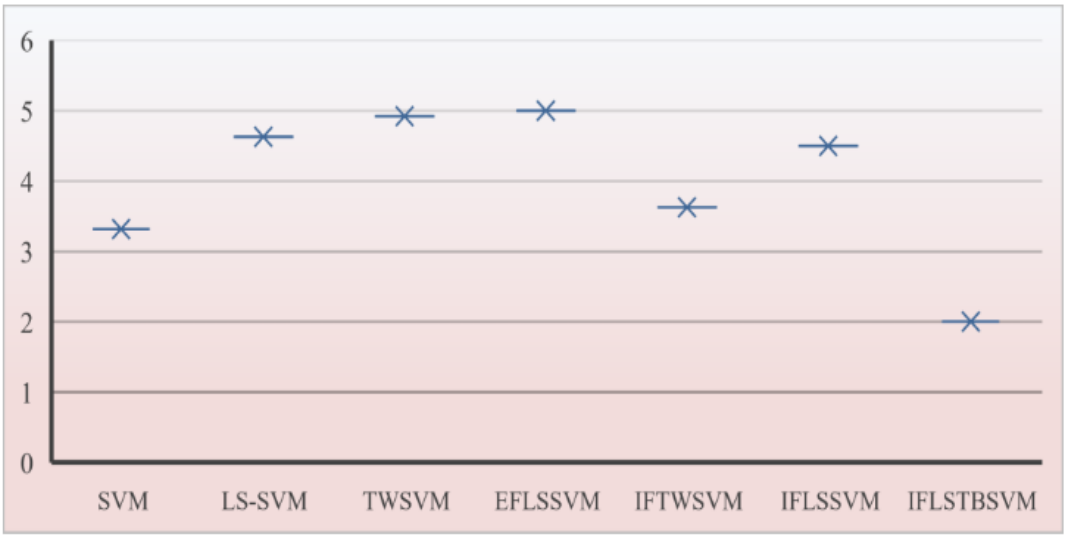


Figure 5. 17 Datasets contaminated by noise: the Friedman test

With its accompanying post hoc test, this non-parametric test aims to compare two classifiers in a paired fashion. The method in issue is used to quantify the extent to which two distinct classifiers' performances diverge for any given dataset. To break a tie, we take the absolute differences and order them from least to most significant, and then we take the average.

Following the z-score distribution, we record the total number of datasets where our suggested classifier ranked higher than rivals and the total number of datasets where the inverse was true. According to the Wilcoxon distribution, we can rule out the possibility that the classifiers are the same (the null hypothesis).

In addition, it is crucial since it displays the smallest possible p-value that may be used to reject a hypothesis (the p-value linked to each comparison). Using this strategy, we may determine the manifestation of a substantial difference between two classifiers..

Artificial Dataset

Here, we run the experiment on two moons, Ripley's, synthetic datasets, together with seven KEEL-derived synthetic datasets to see whether we can enhance our capacity to categorize binary data via more study.

Our models are rated based on their average performance as measured by AUC, F1-score, G-mean, and PPV. Also included in the presentation is a graph showing the average AUC, F1-score, G-mean, and PPV rank. Notably, on synthetic datasets, our model, IFLSTBSVM, is ranked bottom when it comes to average metrics like AUC, F1-score, G-mean, and PPV. Across all of the tested simulated datasets, our IFLSTBSVM model outperformed the competition.

The decision hyperplanes of the aforesaid models (SVM, LS-SVM, TWSVM, EFLSSVM, and IFTWSVM) as well as the two proposed models (IFLSSVM and IFLSTBSVM) are shown in Figures 5.18–5.21. A distinct model was assigned to the Synthetic, 2moons, and Ripley.

There are the final classifier is shown by the black solid line in these photographs, and there are two markers for positive and negative data points. Here you may see these two markings.

Figures 5.18–5.21 show the results of the experiments that show our suggested IFLSTBSVM is superior at detecting real-world positive and negative data points that clearly proclaim similar assertions.

In addition to reducing computation time, our suggested model, IFLSTBSVM, outperforms existing models. The information presented in the tables and graphs suggests the following conclusion..

(1) By evaluating generalization using AUC, F1-score, G-mean, and PPV, our IFLSTBSVM methodology outperforms other interesting techniques.

(2) Furthermore, when contrasted with SVM, LS-SVM, and EFLSSVM models, IFLSSVM offers better effectiveness as measured by G-mean rank, F1-score, and average area under the curve (AUC).

(3) One major advantage of our proposed model is the drastically reduced training time compared to competing methods, which is the IFLSSVM and the IFLSTBSVM.

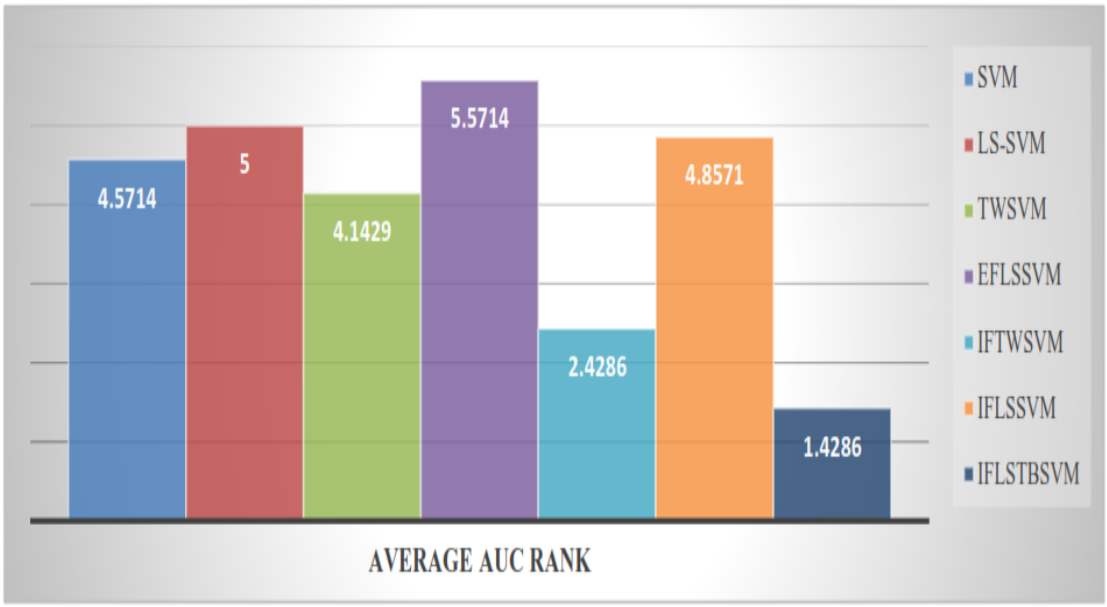


Figure 5. 18 Prioritizing IFLSTBSVM and other models based on their average AUC on synthetic datasets.

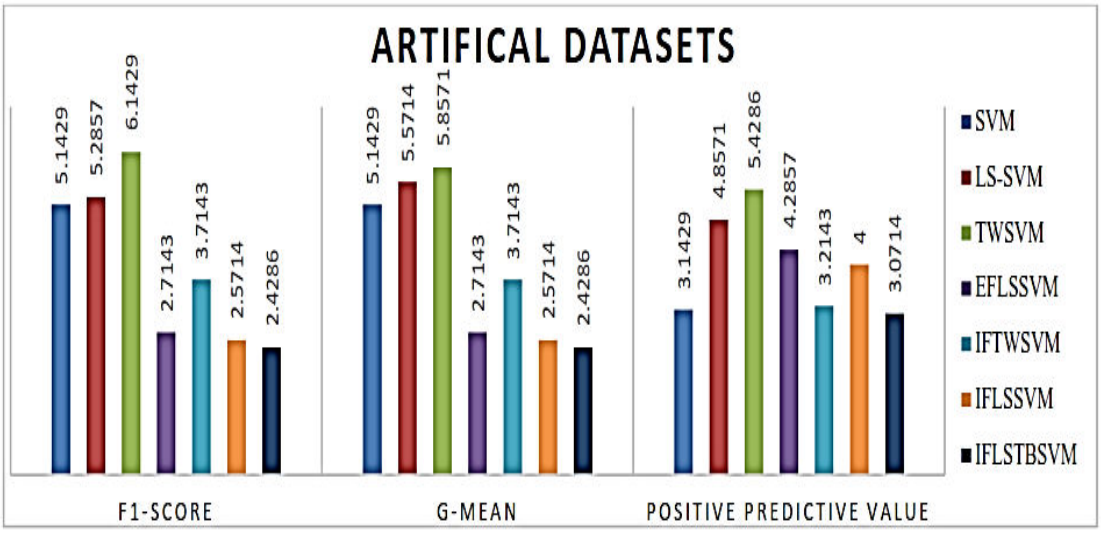


Figure 5. 19 On synthetic datasets, IFLSTBSVM and other models' average F1-scores, G-means, and Positive Predictive Value rankings

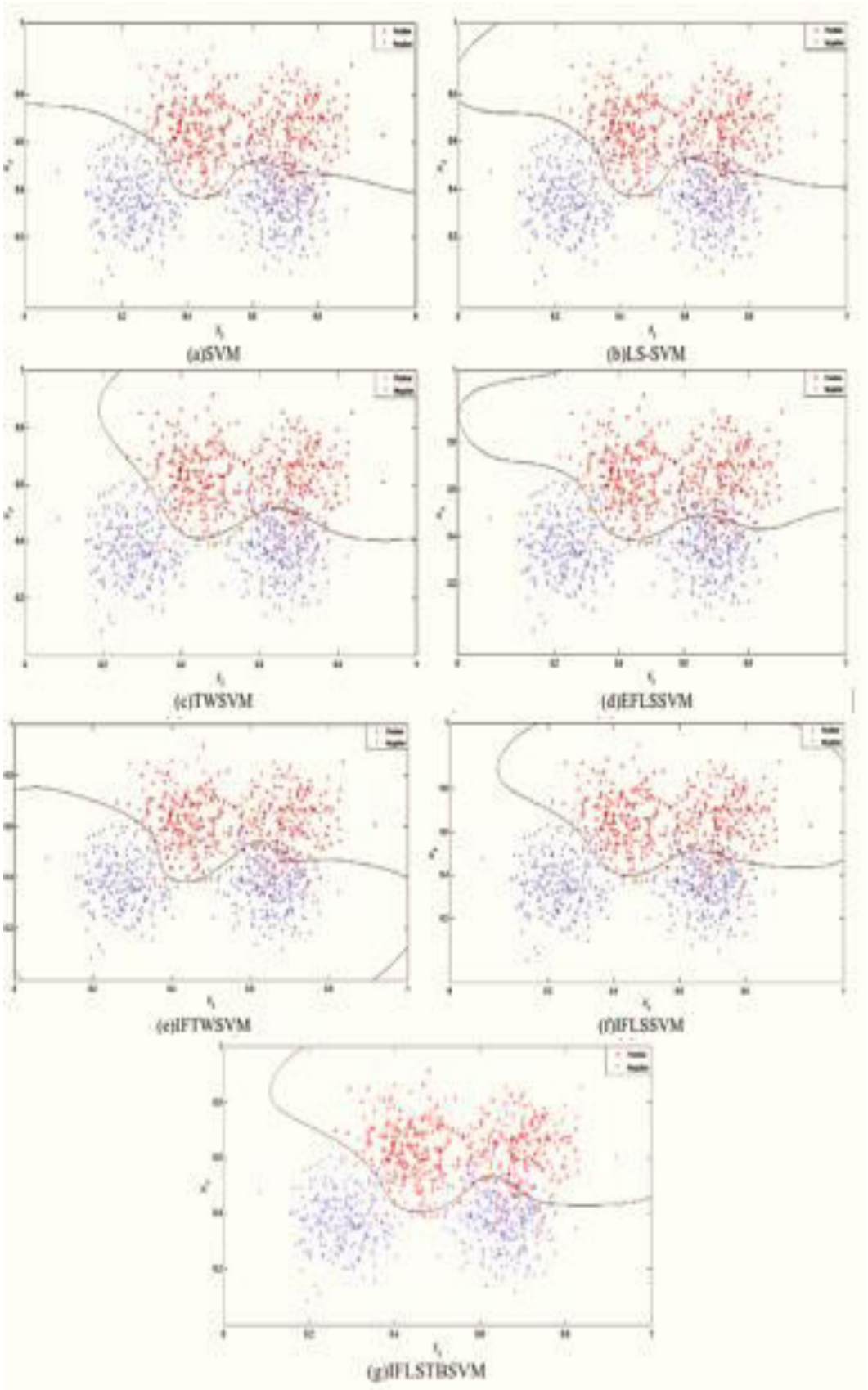


Figure 5. 20 The hyper planes that were drawn on Ripley's synthetic datasets for IFLSTBSVM and other models

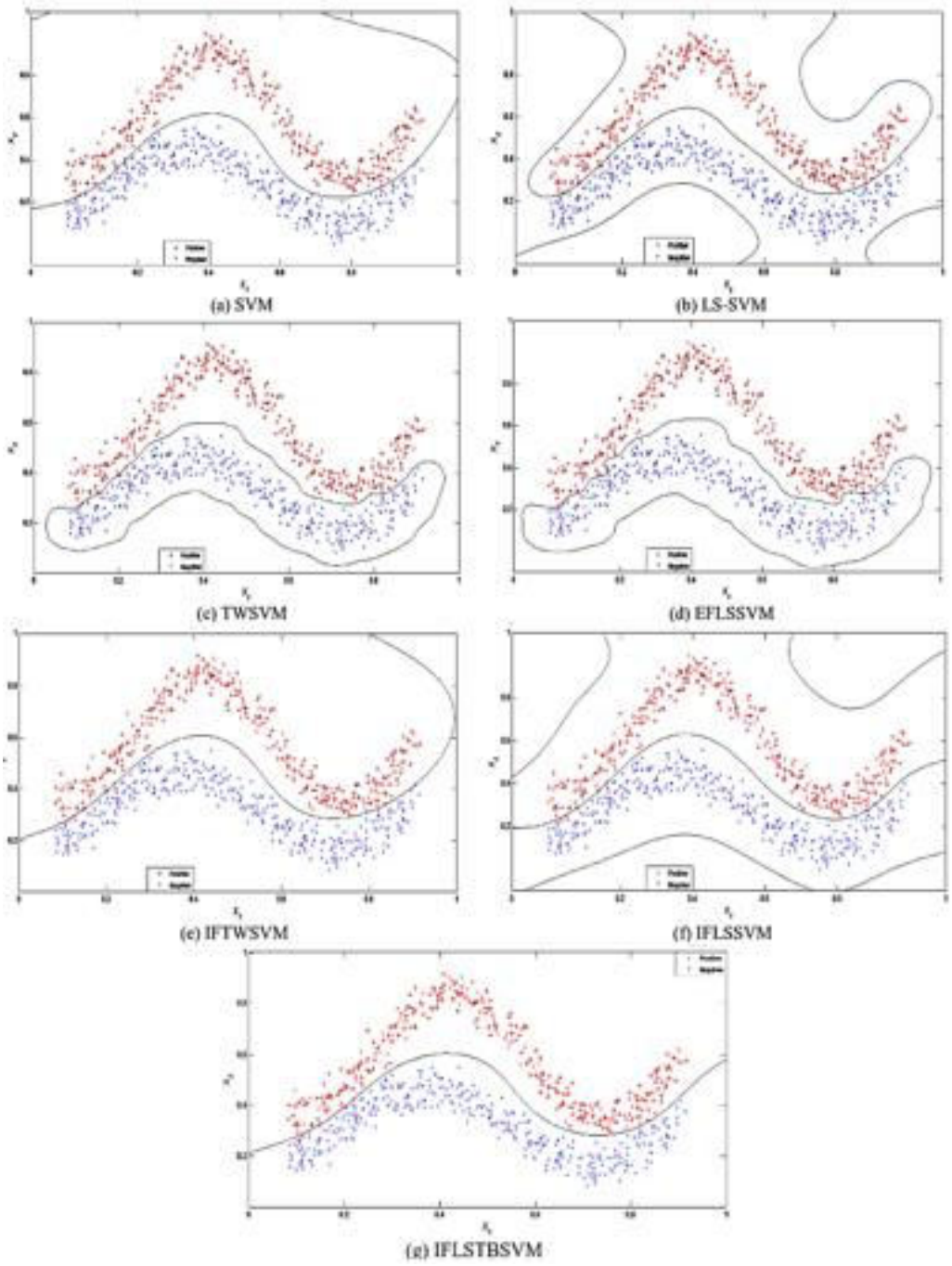


Figure 5. 21 The hyper planes shown on synthetic artificial datasets for models like as IFLSTBSVM

We provide two variants of SVM-based models—IFLSSVM and IFLSTBSVM—that are enhanced and more efficient. Models like this employ intuitionistic fuzzy values to smooth out the effects of outliers and random oscillations in the real data. In contrast to the usual support vector machine, which uses QPPs, solving a system of linear equations is necessary to deal with binary classification issues (SVM). Because of

intuitionistic fuzzy numbers (IFN) with membership and non-membership functions, this occurred because the fuzzy weighted value of positive and negative training samples is computed. Using the IFN function to isolate the support vectors from background noise is essential for training instance classification. Concerning topics like as how to deal with noise and outliers, increase learning speed, and enhance generalization performance, there is zero literature. A number of support vector machine (SVM) algorithms have been developed; however, the IFLSTBSVM significantly surpasses all of them. Binary classification in non-linear scenarios with varying degrees of significant noise has been extensively tested on a number of publicly accessible synthetic and real-world benchmark datasets. The suggested IFLSSVM and IFLSTBSVM will be tested in this experiment to see how well they work in practice. The presented models outperformed the previous published classification models, leading to more generalizable models with reduced computing time requirements, as shown in the trials. The next steps for this area of research will include testing the proposed algorithms on datasets with different classifications to ensure their accuracy.

CHAPTER 6
REGULARIZED IMPLICIT LAGRANGIAN TWIN EXTREME LEARNING MACHINE IN PRIMAL FOR PATTERN CLASSIFICATION

CHAPTER 6
REGULARIZED IMPLICIT LAGRANGIAN TWIN EXTREME LEARNING MACHINE IN PRIMAL FOR PATTERN CLASSIFICATION

Other popular options include SLFNs, which stand for single hidden layer feed forward neural networks. and effective method for classification. When solving optimization problems iteratively, the unconstrained convex minimization approach is superior. This section explores SLFN-based UMC approaches with the aim of improving generalization performance. Traditional SLFNs, even though SLFNs are very generalizable, tend to converge slowly and hit local minima when the neural connection weights are changed frequently using the gradient method. Using extreme learning machines (ELMs) is one practical way to fix the issues with SLFNs outlined above. The input pattern undergoes a nonlinear modification before being accessed by the input layer via ELM. The enhancement nodes, or those that survived the nonlinear transformation, are located in the buried layer. Using a random initialization strategy, ELM eliminates iterative modification by determining the weights and biases of the enhancement nodes. The next step is to improve the output layer's bias and weights by solving an optimization problem. There is a lack of direct connection between the ELM's input and output layers. The method known as least squares is used by error loss measures (ELMs). Machines like these, which are similar to TSVMs, find two non-parallel hyperplanes in the ELM feature space, representing each class. In contrast to TWSVM, TELM uses non-parallel hyperplanes to traverse the origin. However, TELM does not use ELM, which is the loss function for errors in least squares. Despite its generalizability, TELM cannot find its solutions until it resolves two smaller QPPs.

A novel approach is introduced here: the extreme learning machine for regularized based implicit Lagrangian twins. Use it as a set of unbounded convex minimization problems that, with the aid of a regularization term, follow the SRM theory in Primitive. Using the 2-norm of the slack vector of variables is a typical way to make the problem exceedingly convex and to discover an original solution. By substituting a smooth approximation function for the non-smooth addition function, we strive to provide a rough solution to their optimization issue. This is done because in primal space, a near approximation answer is acceptable, unlike in dual space where an exact solution is desirable. To fix this, one may utilize a generalized derivative approach or a smooth

approximation method, as the plus function is not smooth. Iteratively using a functional analysis yields the optimal solution. Consequently, unlike TELM and TWSVM, you won't need an optimization toolbox. Comparing the proposed model to other, more traditional models, numerical investigations on both real-world and simulated datasets show how helpful and flexible it is.

6.1 PROPOSED MODEL

We present RILTELM, an innovative primal-based implicit Lagrangian twin ELN that makes use of regularization, framed as a collection of unconstrained minimization issues. Iterative techniques based on gradients are also being considered as potential solutions to this problem. The optimization problem may be represented by the squared 2-norm vectors of ζ1 and ζ2, those are the two-norm TELM expressions in a linear setting, when the one-norm of the slack variables' vector is known. This technique was suggested by Musicant and Feinberg. The non-negative limits of the formulation's slack variables are disregarded since it will quickly approach optimality. On top of that, we are announcing the components $\frac{1}{2}(\|w_1\|^2) and \frac{1}{2}(\|w_2\|^2)$ as it relates to the goal functions, the model may be well-posed, and it often yields new solutions according to the SRM principle. In order to get the kernel-produced surfaces $K_{ELM}\ (x^t, B^t)w_1 = 0$ and, $K_{ELM}\ (x^t, B^t)w_2 = 0$ When dealing with non-linear instances, we use this approach to produce our suggested RILTELM:

$$\min \frac{C_1}{2} \| w_1 \|^2 + \frac{1}{2} \| K_{ELM}(B_1, B^t)w_1 \|^2 + \frac{C_3}{2} \zeta_1^t \zeta_1$$

Subject to: $-K_{ELM}\ (B_2, B^t)w_1 + \zeta_1 \geq e_2$ (6.1)

And

$$\min \frac{C_2}{2} \| w_2 \|^2 + \frac{1}{2} \| K_{ELM}(B_2, B^t)w_2 \|^2 + \frac{C_4}{2} \zeta_2^t \zeta_2$$

subject to:

$K_{ELM}\ (B_1, B^t)w_2 + \zeta_2 \geq e_1$ (6.2)

Where $B^t = [B_1\ B_2\ ]^t$ in addition to the kernel function KELM.

Moreover, the analogous unconstrained convex minimization issue may be written as follows, which is relevant to both the restricted primal problems (6.1) and (6.2) that were already described:

$$L_1(\mu_1) = \min \frac{C_1}{2} \| w_1 \|^2 + \frac{1}{2} \| K_{ELM}(B_1, B^t) w_1 \|^2 + \frac{C_3}{2} \| (e_2 + K_{ELM}(B_2, B^t) w_1)_+ \|^2$$

And

$$L_2(w_2) = \min \frac{C_{21}}{2} \| w_2 \|^2 + \frac{1}{2} \| K_{ELM}(B_2, B^t) w_2 \|^2 + \frac{C_4}{2} \| (e_2 - K_{ELM}(B_1, B^t) w_2)_+ \|^2.$$

To rephrase the unconstrained problems, you may use the following form:

$$L_1(w_1) = \min \frac{C_1}{2} \| w_1 \|^2 + \frac{1}{2} \| D_3 w_1 \|^2 + \frac{C_3}{2} \| (e_2 + D_4 w_1)_+ \|^2 \quad (6.3)$$

And

$$L_2(\mu_2) = \min \frac{C_2}{2} \| w_2 \|^2 + \frac{1}{2} \| D_4 w_2 \|^2 + \frac{C_4}{2} \| (e_1 - D_3 w_2)_+ \|^2 . \quad (6.4)$$

Where, $D_3 = K_{ELM}(B_1 B^t)$ *and* $D_4 = K_{ELM}(B_2 B^t)$.

Here we provide RILTELM in its most basic form: a collection of non-linear minimization problems without constraints. The problems (6.3) and (6.4), which are piece-wise quadratic and differentiable, and unconstrainedly highly convex, may be solved in three ways: (i) A generalized Hessian matrix may be obtained by combining the Newton iterative technique with a generalized derivative method. ii) In the Newton iterative technique (iii), replace the non-smooth 'plus' function with a smoothing approximation method. Determine an equation's absolute value by using a basic functional iterative approach. Equations (6.3) and (6.4) provide the following gradient vector:

$$\nabla L_1(w_1) = C_1 w_1 + D_3^t D_3 w_1 + C_3 D_4^t (e_2 + D_4 w_1)_+ \quad (6.5)$$

And

$$\nabla L_2(w_2) = C_2 w_2 + D_4^t D_4 w_2 - C_4 D_3^t (e_1 - D_3 w_2)_+ \quad (6.6)$$

6.1.1 A GENERIC METHOD FOR CALCULATING RILTELM DERIVATIVES

Equations (6.5) and (6.6) may be transformed into generalized Hessian matrices using the generalized gradient technique when one follows these procedures.: Both gradient matrices (6.5) and (6.6) have a continuous but non-differentiable 'plus' function:

$$\nabla^2 L_1(w_1) = C_1 I + D_3^t D_3 + C_3 D_4^t diag(e_2 + D_4 w_1)_* D_4 \tag{6.7}$$

And

$$\nabla^2 L_2(w_2) = C_2 I + D_4^t D_4 + C_4 D_3^t diag(e_1 - D_3 w_2)_* D_3 \tag{6.8}$$

Also utilized is the Newton iterative method, where the basic step for finding the (i+1)-th iterative from the present i-th iterative is given by

$$\nabla^2 L(v^i)(v^{i+1} - v^i) = -\nabla L(v^i) \quad , \text{where} \quad i = 0,1,2... \tag{6.9}$$

We get the solutions to (6.3) and (6.4) by solving the following iterative techniques, as

$$\left(C_1 I + D_3^t D_3 + C_3 D_4^t diag(e_2 + D_4 w_1^i)_* D_4\right)(w_1^{i+1} - w_1^i) = -\left(C_1 w_1^i + D_3^t D_3 w_1^i + C_3 D_4^t (e_2 + D_4 w_1^i)_+\right), \tag{6.10}$$

And

$$\left(C_2 I + D_4^t D_4 - C_4 D_3^t diag(e_1 - D_3 w_2^i)_* D_3\right)(w_2^{i+1} - w_2^i) = -\left(C_2 w_2^i + D_2^t D_2 w_2^i + C_4 D_3^t (e_1 - D_3 w_2^i)_+\right) \tag{6.11}$$

In that order. The GRILTELM method is our first attempt at using a generalized derivative.

6.1.2 EASY METHODS FOR RILTELM

The Hessian of equations (6.3) and (6.4) does not exist since they are not twice differentiable but continuous. By examining two distinct avenues, the issue of twice non-differentiability is addressed. When it comes to machine learning, the smooth method is often touted as a solution to mathematical programming issues that aren't smooth.

In order to replace the non-smooth function, the smoothing methods developed by Lee and Mangasarian are used. Problems (6.3) and (6.4) were solved by using the smooth approximation function proposed by Lee and Mangasarian. $\gamma_1(x, \psi)$ in place of the 'plus' function, which is defined as, using the smooth parameter $\varpi > 0$:

$$\gamma_1(x,\alpha) = x + \frac{1}{\psi}\log(1+\exp(-\psi x)) \tag{6.12}$$

This allows us to rewrite the minimization problems (6.3) and (6.4) in the revised form.

$$L_1(\mu_1) = \min \frac{C_1}{2} \| w_1 \|^2 + \frac{1}{2} \| D_3 w_1 \|^2 + \frac{C_3}{2} \| \gamma_1((e_2 + D_4 w_1), \psi) \|^2 \tag{6.13}$$

And

$$L_2(\mu_2) = \min \frac{C_2}{2} \| w_2 \|^2 + \frac{1}{2} \| D_4 w_2 \|^2 + \frac{C_4}{2} \| \gamma_2((e_1 - D_3 w_2), \psi) \|^2, \tag{6.14}$$

in that order. The Hessian matrix of the vectors (6.13) and (6.14) is derived by taking the gradient vectors from (6.5) and (6.6), respectively.

$$\nabla^2 L_1(w_1) = C_1 I + D_3' D_3 + C_3 D_4' diag\left(\frac{1}{1+\exp(-\psi(e_2 + D_4 w_1))}\right) D_4 \tag{6.15}$$

And

$$\nabla^2 L_2(w_2) = C_2 I + D_4' D_4 + C_4 D_3' diag\left(\frac{1}{1+\exp(-\psi(e_1 - D_3 w_2))}\right) D_3 \tag{6.16}$$

We used the Newton iterative approach to get the answer since we knew that (6.3) and (6.4) were gradient vectors and Hessian matrices, respectively. Here we provide the SRILTELM1 approach, the second of our smooth RILTELM techniques.

When it comes to solving problems (6.3) and (6.4), we keep coming back to the same methods:

$$\left(C_1 I + D_3' D_3 + C_3 D_4' diag\left(\frac{1}{1+\exp(-\psi(e_2 + D_4 w_1^j))}\right) D_4\right)(w_1^{j+1} - w_1^j) = -\left(C_1 w_1^j + D_3' D_3 w_1^j + C_3 D_4'(e_2 + D_4 w_1^j)_+\right) \tag{6.17}$$

And

$$\left(C_2 I + D_4^{'} D_4 + C_4 D_3^{'} diag\left(\frac{1}{1+\exp(-\psi(e_1 - D_3 w_2^i))}\right) D_3\right)(w_2^{i+1} - w_2^i) =$$

$$-\left(C_2 w_2^i + D_4^{'} D_4 w_2^i + C_4 D_3^{'} (e_1 - D_3 w_2^i)_+\right), \tag{6.18}$$

Respectively.

As a second method for smoothing, we've looked at the smooth approximation function proposed. $\gamma_2(\psi, \psi_0)$ for ψ + introduced that is, when the value of ψ

$$\gamma_2(\psi, \psi_0) = \frac{1}{4}\frac{\psi^2}{|\psi_0|} + \frac{1}{2}\psi + \frac{1}{4}|\psi_0|, \tag{6.19}$$

Where ψ0 has a real value that is not zero. As is evident from what is $\gamma_2(\psi, \psi_0)$ is differentiable and a quadratic function. When the value of |ψ0| becomes closer, it becomes evident to |ψ|, then $\gamma_2(\psi, \psi_0)$ becomes closer to ψ+. In fact, $\gamma_2(\psi, \psi_0)$ = ψ+ whenever $|\psi_0| = |\psi| \neq 0$.

This allows us to rewrite the minimization problems (6.3) and (6.4) in the revised form.

$$L_1(w_1) = \min \frac{C_1}{2} \| w_1 \|^2 + \frac{1}{2} \| D_3 w_1 \|^2 + \frac{C_3}{2} \| \gamma_2((e_2 + D_4 w_1), \psi_0) \|^2 \tag{6.20}$$

And

$$L_2(w_2) = \min \frac{C_2}{2} \| w_2 \|^2 + \frac{1}{2} \| D_4 w_2 \|^2 + \frac{C_4}{2} \| \gamma_2((e_1 - D_3 w_2), \psi_0) \|^2 \tag{6.21}$$

In that order. You can get the gradient vector of (6.20) and (6.21) using (6.5) and (6.6), and you can derive their Hessian matrix by using

$$\nabla^2 L_1(w_1) = C_1 I + D_3^{'} D_3 + \frac{1}{2} C_3 D_4^{'} diag((e_2 + D_4 w_1) | \psi_0 |^{-1} + e_2) D_4 \tag{6.22}$$

And

$$\nabla^2 L_2(w_2) = C_2 I + D_4^{'} D_4 + \frac{1}{2} C_4 D_3^{'} diag((e_1 - D_3 w_2) | \psi_0 |^{-1} + e_1) D_3 \tag{6.23}$$

Since we are already familiar with the equations' gradient vectors and Hessian matrices, we can apply the Newton iterative technique here. (6.3) and (6.4). Here we provide

SRILTELM2, our third method for smooth RILTELM. The following iterative strategies are solved by us as respectively.

$$\left(C_1 I + D_3{}^t D_3 + \frac{1}{2} C_3 D_4{}^t diag\left((e_2 + D_4 w_1^i) \mid w_0 \mid^{-1} + e_2\right) D_4\right)(w_1^{i+1} - w_1^i) = -\left(C_1 w_1^i + D_3{}^t D_3 w_1^i + C_3 D_4{}^t (e_2 + D_4 w_1^i)_+\right) \tag{6.24}$$

And

$$\left(C_2 I + D_4{}^t D_4 + \frac{1}{2} C_4 D_3{}^t diag\left((e_1 - D_3 w_2^i) \mid w_0 \mid^{-1} + e_1\right) D_3\right)(w_2^{i+1} - w_2^i) = -\left(C_2 w_2^i + D_4{}^t D_4 w_2^i + C_4 D_3{}^t (e_1 - D_3 w_2^i)_+\right) \tag{6.25}$$

While completing the aforementioned iterative techniques may provide solutions for W_1 and W_2 in equations (6.3) and (6.4), it is important that these solutions be completely separate from W_0. Hence, to get around this issue, we use a basic iterative technique that involves adjusting the vector W_0 till it approaches $|e_2 + D_4 w_1|$ so that $\Upsilon_2\left((e_2 + D_4 w_1) w_0\right)$ will be very close to $(e_2 + D_4 w_1)$ and the vector W_0 gets adjusted till it is close to $|e_1 + D_3 w_2|$ so that $\Upsilon_2\left((e_1 + D_3 w_2) w_0\right)$ will be very close to $(e_1 + D_3 w_2)$ in iterative schemes (6.24) & (6.25) respectively.

6.1.3 FUNCTIONAL ITERATIVE APPROACH FOR RILTELM (FRILTELM)

Based on this person's, we have proposed an additional straightforward functional iterative method in this paragraph $\vartheta_+ = \frac{\vartheta + |\vartheta|}{2}$ for any $\vartheta \in \Re^p$, to address the issues with (6.3) and (6.4). An alternative representation of the gradient vectors (6.3) and (6.4) is which

$$\nabla L_1(w_1) = \left(C_1 I + D_3{}^t D_3 + \frac{C_3}{2} D_4{}^t D_4\right) w_1 + \frac{C_3}{2} D_4{}^t (e_2 + |(e_2 + D_4 w_1)|) \tag{6.26}$$

And

$$\nabla L_2(w_2) = \left(C_2 I + D_4{}^t D_4 + \frac{C_4}{2} D_3{}^t D_3\right) w_2 - \frac{C_4}{2} D_3{}^t (e_1 + |(e_1 - D_3 w_2)|). \tag{6.27}$$

The crucial points may now be calculated by equating $\nabla L_1(w_1) = 0$ *and* $\nabla L_2(w_2) = 0$.

$$\left(C_1 I + D_3{}^t D_3 + \frac{C_3}{2} D_4{}^t D_4\right) w_1 + \frac{C_3}{2} D_4{}^t (e_2 + |(e_2 + D_4 w_1)|) = 0 \tag{6.28}$$

And

$$\left(C_2 I + D_4{}^t D_4 + \frac{C_4}{2} D_3{}^t D_3\right) w_2 - \frac{C_4}{2} D_3{}^t (e_1 + |(e_1 - D_3 w_2)|) = 0 . \tag{6.29}$$

In turn, this generates the iterative schemes that make up FRILTELM, the functional iterative schemes that will constitute our fourth approach proposal.

$$\left(C_1 I + D_3{}^t D_3 + \frac{C_3}{2} D_4{}^t D_4\right) w_1^{j+1} = -\frac{C_3}{2} D_4{}^t (e_2 + |(e_2 + D_4 w_1^j)|) \tag{6.30}$$

And

$$\left(C_2 I + D_4{}^t D_4 + \frac{C_4}{2} D_3{}^t D_3\right) w_2^j = \frac{C_4}{2} D_3{}^t (e_1 + |(e_1 - D_3 w_2^j)|) \tag{6.31}$$

We only obtain the inverse of matrices once, as is evident $\left(C_1 I + D_3^t D_3 + \frac{C_3}{2} D_4^t D_4\right)$ and $\left(C_2 I + D_4^t D_4 + \frac{C_4}{2} D_3^t D_3\right)$ up front in the aforementioned iterative methods (6.30) and (6.31), correspondingly.

Discussion

1. The presence of a globally unique solution is implied by the highly convex objective functions in the proposed RILTELM.

2. The model is rendered well-posed and the stability of the dual formulations is enhanced by include regularization components in the objective functions. Additionally, it reduces the issue of over fitting.

3. RILTELM follows the SRM principle, even if TELM and TWSVM exist.

4. Our suggested methods' solutions are readily achieved utilizing gradient-based iterative techniques, unlike TELM and TWSVM, hence no other optimization toolbox is expected to handle QPPs.

5. Computational tests are performed on 36 real-world datasets from UCI and 3 synthetic datasets to demonstrate the applicability of the proposed RILTELM.

6.2 NUMERICAL EXPERIMENTS

On both synthetic and real-world datasets, we tested the suggested FRILTELM, GRILTELM, SRILTELM1, and SRILTELM2 models with the traditional nonlinear binary classifiers TWSVM, ELM, TELM, and LSTELM.

Research platform technical requirements were an Intel(R) Core (TM) i5-3470 CPU running at 3.20 GHz, 8 GB of RAM, Windows 10 operating system, and MATLAB software version 2008b. In order to fix the QPP, the TELM and TWSVM framework is considering using MOSEK, an external optimization toolkit. No more toolbox is needed, however, for the RILTELM models that we have suggested.

We use 32 moons, 36 real-world datasets, 3 synthetic datasets, and Ripley's dataset from the UCI datasets repository to numerically analyze the non-linear condition. All four datasets—TLS, ELM, TELM, and LSTELM—are fitted using the RILTELM models.

The RBF's hidden nodes are then used to pick the activation function after that. The G-function of uK, bK, and x Considering Gaussian and multiquadric functions. Random values between 0 and 1 are used to choose the RBF hidden node's parameters.

One alternative view is that the biases and input weights are randomly selected at the startup of the RBF hidden node. Nevertheless, these settings will be kept constant in every experiment. In most cases, the Gaussian non-linear kernel is used as

$$K(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} \| x_i - x_j \|^2\right), \text{ for } i, j = 1,..., p$$

Where kernel parameter $\sigma > 0$.

The optimal parameters were found using 10-fold cross-validation, as shown in Table 6.1, and the classification performance of the algorithms proposed by RILTELM was measured using CPU learning time on a variety of fake and real-world datasets.

TABLE 6. 1 Varieties of user-defined parameters used in RILTELM numerical experiments

Parameters	Range	Models
C	{10-5 ,...,105 }	TWSVM
C1 = C2 ,C3 = C4	{10-5 ,...,105 }	TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, FRILTELM
σ	{2-5 ,...,25 }	TWSVM
L	{ 20, 50,100, 200, 500,1000}	ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, FRILTELM

6.2.1 ARTIFICIAL DATASETS

Initially, we create fictional datasets that mimic Ripley's dataset so that we can compare RILTELM's performance the use of TWSVM, ELM, TELM, and LSTELM. We train on 250 samples out of 1250 samples, and we test on the rest. while using multiquadric RBF nodes, Table 6.2 compares the performance of the proposed ways with that of TWSVM, ELM, TELM, and LSTELM; while using Gaussian RBF nodes, Table 6.3 discusses the performance of the same methods. By examining Tables 6.2 for the multiquadric RBF node and Table 6.3 for the Gaussian RBF node, it is evident that SRILTELM1 and FRILTELM achieve the highest classification accuracy in comparison to the others. Learning periods for TELM were longer than those for all four of the proposed multiquadric and Gaussian RBF nodes: FRILTELM, SRILTELM1, and SRILTELM2. The multiquadric classifier is shown in Figure 6.1(a)-(h), whereas the Gaussian radial basis functions classifier is shown in Figure 6.2(a)-(h). The GRILTELM class has the following methods: FRILTELM, ELM, TELM, LSTELM, SRILTELM1, and SRILTELM2. Each of these graphs uses the symbols 'x' for positive class data points and '+' for negative class data points. Our second step is to build a 1000-sample, two-dimensional synthetic dataset using the OF form. Using 500 samples from the positive class produced by $x_1 \in [-\pi/2, 2\pi], sin\, x_1 - 0.25 \leq x_2 \leq sin\, x_1 + 0.25$ also produce 500 samples from the negative category by

$x_1 \in [-\pi/2, 2\pi], 0.6\ sin\ (x_1\ /1.05 + 0.4) - 1.35 \leq x_2 \leq 0.6\ sin\ (x_1/1.05 + 0.4) - 0.85$ with the addition of noise $\upsilon = N(0, 0.1^2)$.

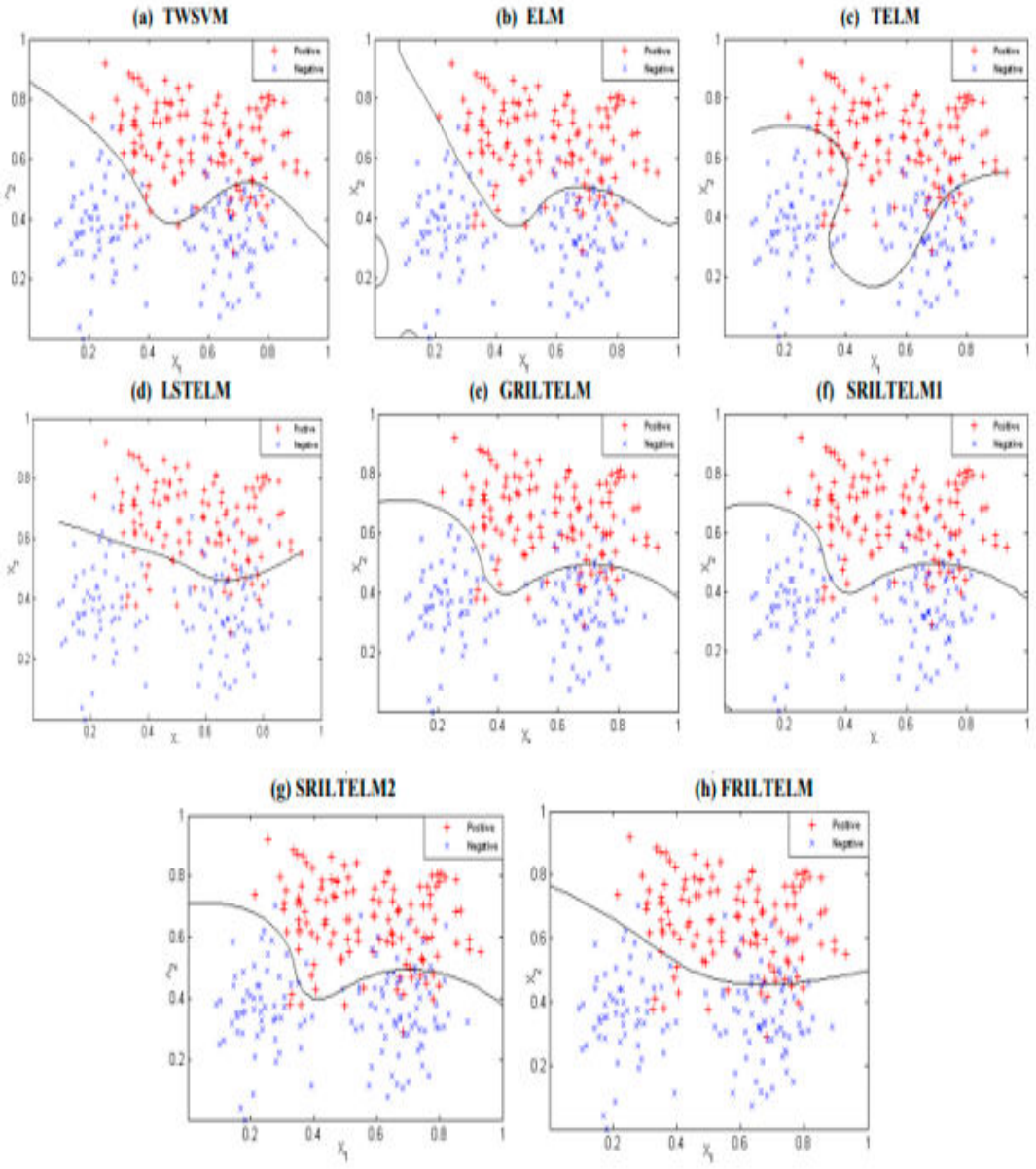


FIGURE 6. 1 Utilizing With Ripley's dataset, this classifier uses the Multiquadric RBF function, which includes TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM.

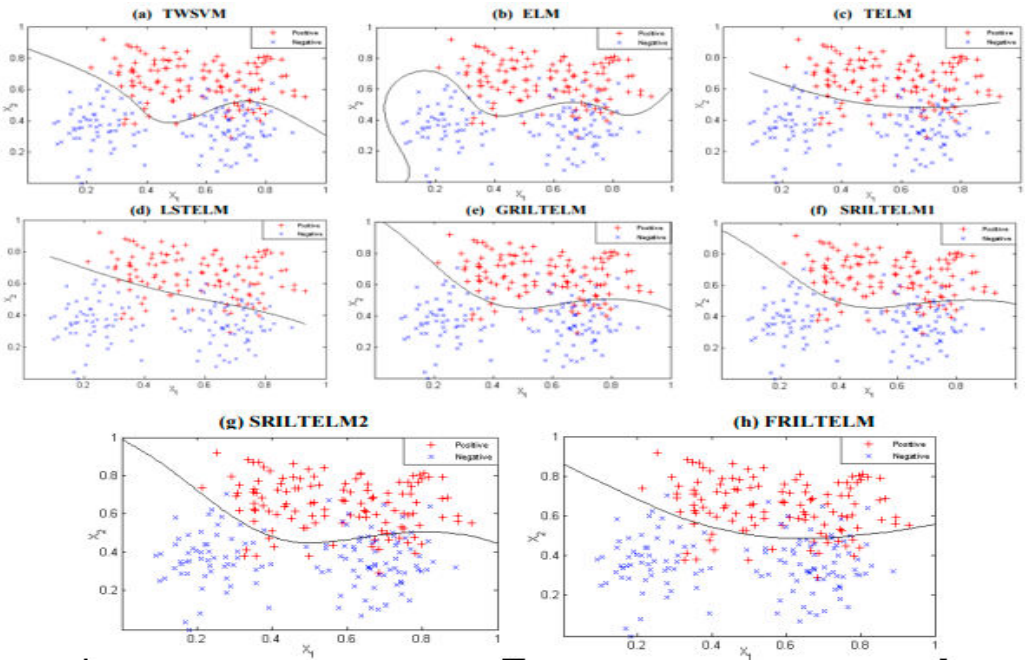


FIGURE 6. 2 Applying the Gaussian RBF function, this classifier handles Ripley's dataset and is suitable with relation to TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM.

We then compare the proposed techniques to other published methods using the remaining samples, and we train the model using 300 randomly picked data. Figures 6.3(a)-(h) and 6.4(a)-(h) show our results for the classifier with the Table 6.2 shows our results for TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM using the multiquadric RBF node. From what can be seen in Tables 6.2 and 6.3, SRILTELM1 is the most effective RBF node. The following definition was used to produce three synthetic datasets, each with 500 samples for the positive and negative classes: 2moons: $x = c + \chi * [cos(x_i), sin(x_i)]$ where i =1, 2 c_1 {-0.5, 1}, c_2 = {0.5, −1} $x_1 \in [-\pi/2, \pi/2]$ and $x_2 \in [\pi/2, 3\pi/2]$ with the inclusion of random noise in the normally distributed sample χ =N (2, 0.5^2) We use 300 out of 1000 samples to train the model, and 700 instances to evaluate it. The findings displayed in Table 6.2 are obtained from the model trained with multiquadric RBF nodes, whereas the results shown in Table 6.3 are obtained from the model trained with Gaussian RBF nodes. See Figures 6.6(a)-(h) for a better representation of the classifiers needed for each model, which includes multiquadric RBF nodes, and Figures 6.5(a)-(h) for Gaussian RBF nodes.

TABLE 6. 2 Analyzing RILTELM and other models using a Multiquadric RBF node on datasets that were artificially produced

Dataset (Train size, Test size)	TWSVM (C, σ) Time	ELM (L) Time	TELM ($C_1 = C_2$, L) Time	LSTELM $C_1 = C_2$, L) Time	GRILTELM ($C_1 = C_2$, $C_3 = C_4$, L) Time	SRILTELM1 ($C_1 = C_2$, $C_3 = C_4$, L) Time	SRILTELM2 ($C_1 = C_2$, $C_3 = C_4$, L) Time	FRILTELM ($C_1 = C_2$, $C_3 = C_4$, L) Time
Ripley (250X2, 1000X2)	87.3 (10^1, 2^0) 0.1191	86.04+6.7462 (20) 0.00432	87.4+6.2681 (10^1,1000) 0.79982	88.2+7.00476 (10^2,50) 0.02759	88.04+8.0994 (10^1,10^-1,1000) 0.03207	88.32+8.2192 (10^1,10^-1,1000) 0.1058	88.0105+8 (10^1,10^-1,1000) 0.01622	87.0007+8.6798 (10^4,10^3,500) 0.0145
Synthetic (300X2, 700X2)	96.4286 (10^-4, 2^-3) 0.14747	97.1667+2.1082 (20) 0.0052	97.5333+1.7213 (10^0,500) 0.66462	97.6667+2.62937 (10^1,100) 0.04834	97.2667+2.2498 (10^1,10^-2,500) 0.02553	97.2667+2.2498 (10^1,10^-2,500) 0.02553	97.6473+1.6754 (10^1,10^-5,1000) 0.19098	96.7568+2.593 (10^1,10^-4,500) 0.02033
2moons (300X2, 700X2)	100 (10^-5, 2^-2) 0.34116	99.9+0 (50) 0.01208	100+0 (10^3,1000) 0.98166	99.8571+0 (10^2,100) 0.27322	100+0 (10^2,10^-3,1000) 0.1494	100+0 (10^1,10^-4,1000) 0.13977	99.9558+0 (10^2,10^-2,1000) 0.23	99.9967+0 (10^0,10^-4,1000) 0.03095

TABLE 6. 3 A study comparing RILTELM to various models and datasets that were intentionally generated using the Gaussian RBF node

Dataset (Train size, Test size)	TWSVM (C,σ) Time	ELM (L) Time	TELM ($C_1 = C_2$, L) Time	LSTELM $C_1 = C_2$, L) Time	GRILTELM ($C_1 = C_2$, $C_3 = C_4$, L) Time	SRILTELM1 ($C_1 = C_2$, $C_3 = C_4$, L) Time	SRILTELM2 ($C_1 = C_2$, $C_3 = C_4$, L) Time	FRILTELM ($C_1 = C_2$, $C_3 = C_4$, L) Time
Ripley (250	87.3 (10^1,	86.84+7.4952	87.24+9.8793 (10^0,5	87.5+0 (10^-	87.48+10.0576 (10^1,1	87.48+10.7992	87.5359+9.6384 (10^1,1	87.921+9.7879 (10^2,1

X2, 1000X2)	2^0) 0.1191	(1000) 0.21689	00) 0.72602	3,10) 0.03841	0^-4,500) 0.11012	(10^1,10^-3,1000) 0.10743	0^-4,500) 0.10948	0^-2,200) 0.01079
Synthetic (300X2, 700X2)	96.4286 (10^-4, 2^-3) 0.14747	96.7333+2.9187 (50) 0.01246	93.2333+3.4427 (10^2,1000) 1.49633	96.5714+0 (10^1, 200) 0.29358	97.6667+2.7442 (10^1,10^-5,1000) 0.1749	98+2.2498 (10^1,10^-5,1000) 0.53884	97.6667+2.6165 (10^1,10^-5,1000) 0.13261	93.87+5.0185 (10^1,10^-5,200) 0.01539
2moons (300X2, 700X2)	100 (10^-5, 2^-2) 0.34116	100+0 (1000) 0.26046	99.8333+0 (10^2,1000) 1.38958	99.5714+0 (10^0, 20) 0.02987	100+0 (10^2,10^-3,1000) 0.14506	100+0 (10^2,10^-2,1000) 0.13677	100+0 (10^2,10^-3,1000) 0.14884	99.0445+1.4695 (10^2,10^-4,100) 0.01323

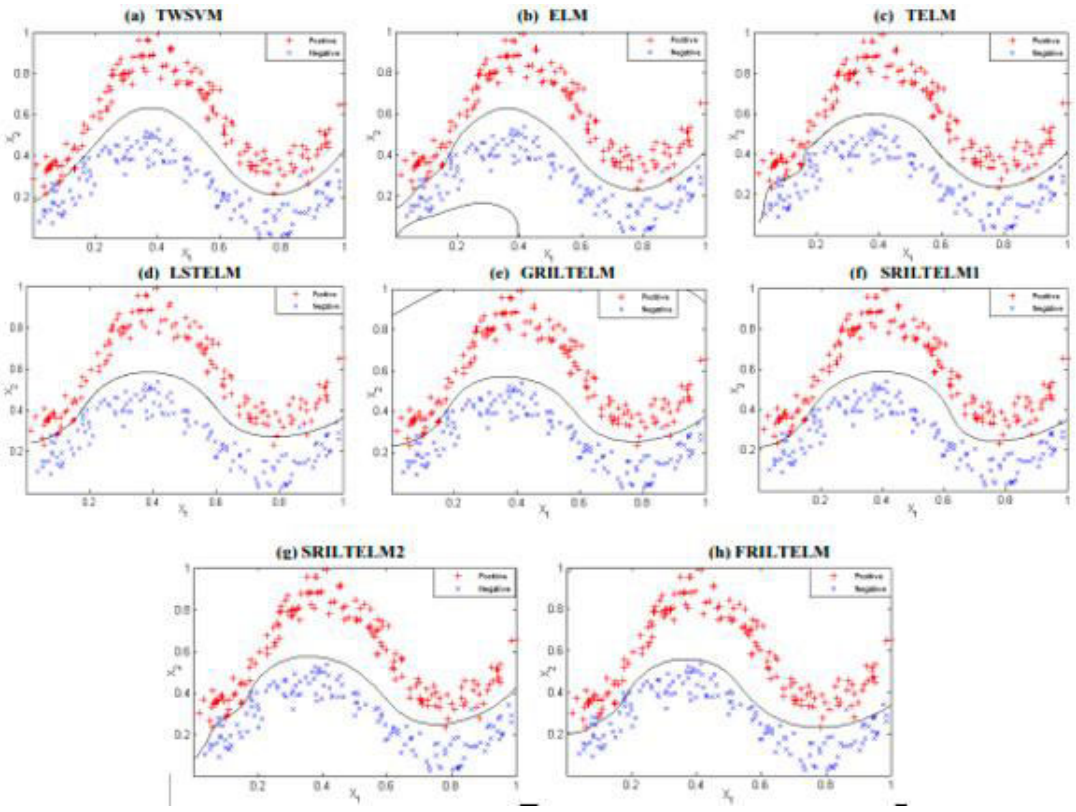


FIGURE 6. 3 Multiquadric RBF function-based classifier for TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM on synthetic dataset

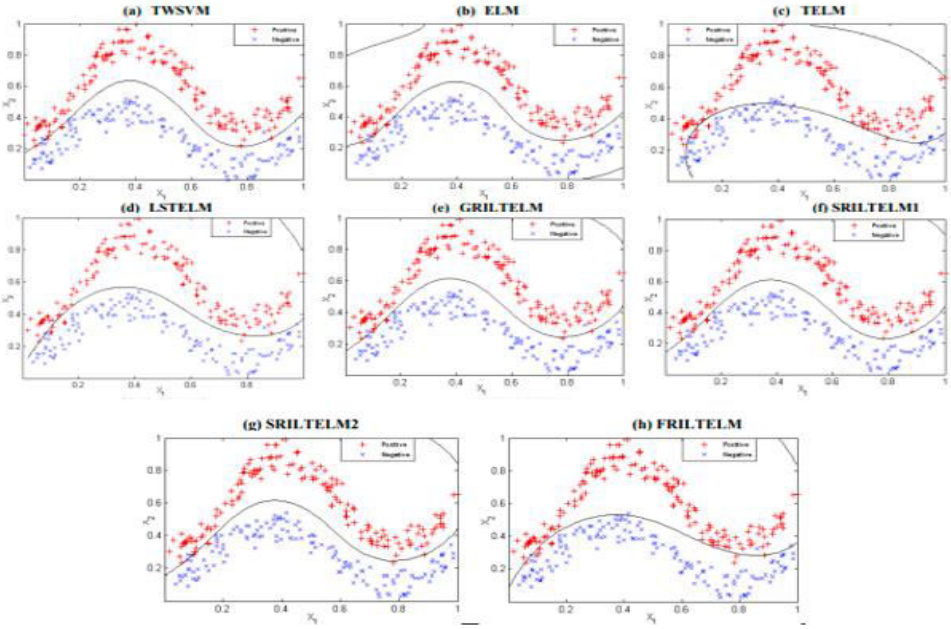


FIGURE 6. 4 A classifier that uses the Gaussian RBF function on a synthetic dataset for TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM.

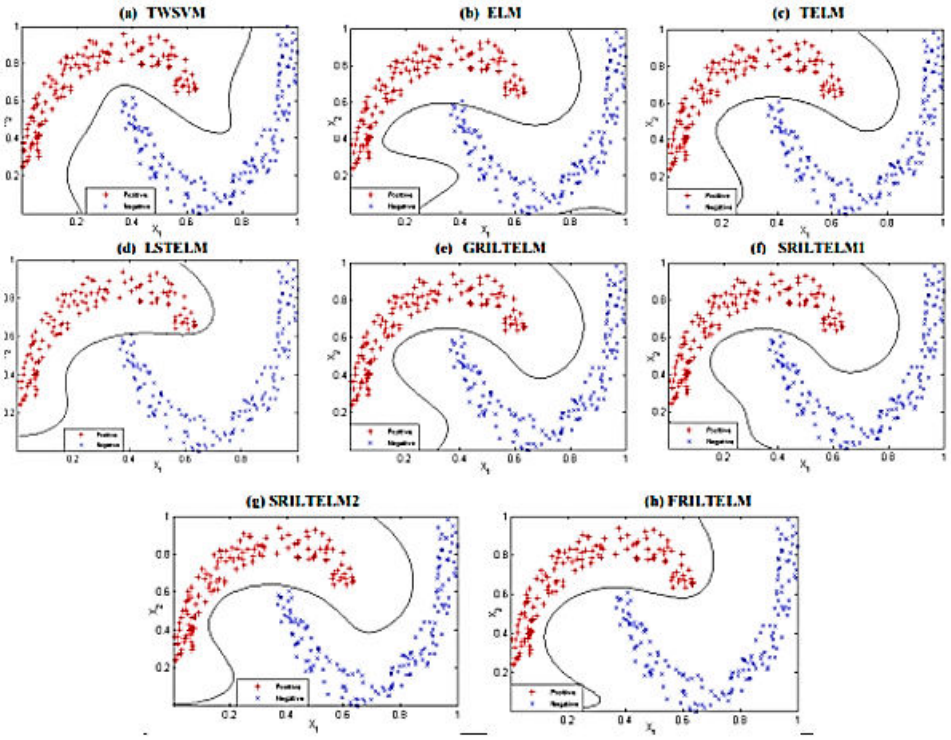


FIGURE 6. 5 A multiquadric RBF function-based classifier for the 2moons dataset employing the following models: TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM.

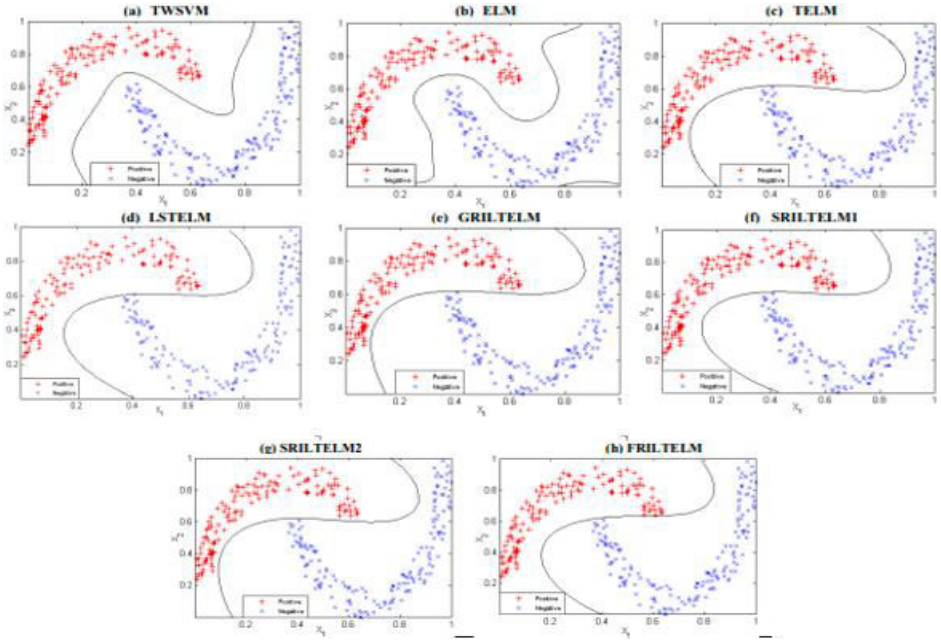


FIGURE 6. 6 Classifier for TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2 and FRILTELM on 2moons dataset using Gaussian RBF function

6.2.2 REAL-WORLD DATASETS

Examining the effects of RILTELM, TWSVM, ELM, TELM, and LSTELM on real-world datasets for non-linear scenario classification using multiquadric and Gaussian RBF nodes. With a lower number, training will take less time. Spending too much money will be the consequence of not doing this. The suggested FRILTELM is faster than TWSVM, TELM, and LSTELM. Furthermore, GRILTELM, SRILTELM1, and SRILTELM2 outperformed TELM when the value was less than TELM. In addition, Table 6.4 shows the average ranking of the proposed RILTELM with all existing techniques utilizing multiquadric RBF functions, while Table 6.5 shows the same for Gaussian RBF functions.

Our testing results demonstrate the presence and functionality of both RBF nodes, even if they are not highly ranked. The results of applying the multiquadric RBF function to various real-world datasets are shown in Figure 6.7, whereas the results of applying the

Gaussian RBF function are shown in Figure 6.8. Figures 6.9–6.12 show the results of the C1, C3, and L multiquadric RBF and the Yeast5 and Ecoli–0-6-7 vs. 3-5 Gaussian RBF investigations, respectively.

TABLE 6. 4 Results from RILTELM and other models' average rankings on real-world datasets for classification accuracy utilizing multiquadric RBF nodes.

Datasets	TWSVM	ELM	TELM	LSTELM	GRILTELM	SRILTELM1	SRILTELM2	FRILTELM
Australian-Credit	8	7	1.5	3	4.5	1.5	4.5	67
Breast-cancer-wisconsin	7	8	5	2.5	2.5	2.5	2.5	6
Bupaor liver-disorders	4	5	8	6	2.5	1	2.5	5
Cleveland	7	8	4.5	4.5	1.5	3	1.5	6
Haberman	6	8	5	7	2.5	4	2.5	1
Ionosphere	1	7	5	8	2.5	4	2.5	6
Pima Indian	1	8	7	6	3.5	2	3.5	
Votes	8	7	6	5	2.5	2.5	2.5	2.5
WDBC	1	8	7	2.5	5.5	5.5	4	2.5
German	8	7	6	5	4	1	2	3
Monk2	1	8	3	7	6	4	5	2
Splice	1	7	6	8	2	3	5	4
vowel	8	3.5	3.5	3.5	7	3.5	3.5	3.5
Ecoli-0-1_vs_2-3-5	6	5	4	8	2	1	3	7
Ecoli-0-1-4-	5	8	4	6	2	2	2	7

7_vs_5-6								
Ecoli-0-2-3-4_vs_5	1	6	7	3	4	2	5	8
Ecoli-0-6-7_vs_3-5	5	6	8	2.5	4	1	2.5	7
Ecoli-0-6-7_vs_5	5	4	6	7.5	2	2	2	7.5
Ecoli4	7	8	5	3	1	4	2	6
Glass-0-1-4-6_vs_2	8	7	5.5	5.5	2	1	3	4
Glass-0-1-5_vs_2	1	8	5	7	2	3	4	6
Glass-0-1-6_vs_2	2	8	6	1	4.5	4.5	3	7
Glass-0-1-6_vs_5	8	7	6	1	2	3.5	5	3.5
Glass-0-6_vs_5	6	8	2.5	7	4	2.5	5	1
Glass 2	7	8	1.5	5	3	1.5	4	6
Yeast-0-2-5-7-9_vs_3-6-8	5	8	4	6	3	1	2	7
Yeast-0-5-6-7-9_vs_4	4	5	6	7.5	2	1	3	7.5
Yeast-2_vs_4	5	8	6	3.5	1	2	3.5	7

Ecoli-0-1-4-6_vs_85	1	7	5	6	3	2	4	8
Glass4	8	5	1.5	7	1.5	4	6	3
Vehicle 1	1	2	6	8	5	3	4	7
Vehicle 2	1	6	5	7	2	4	3	8
Shuttle-6_vs_2-3	4	7	8	1.5	3	1.5	6	5
Yeast3	7	8	3	5.5	4	1	2	5.5
Yeast1	1	6	7	8	2	3	4	5
Yeast5	8	8	4	4	3	1	6.5	6.5
Average rank	4.472222222	6.847222	5.097222	5.263889	3.013889	2.472222	3.5	5.333333

TABLE 6. 5 Results from RILTELM and other models' average rankings on real-world datasets for classification accuracy using a Gaussian RBF node.

Datasets	**TWSVM**	**ELM**	**TELM**	**LSTELM**	**GRILTELM**	**SRILTELM1**	**SRILTELM2**	**FRILTELM**
Australian-Credit	8	6	5	7	3	1	4	2
Breast-cancer-wisconsin	8	3	5.5	7	4	2	5.5	1
Bupa or liver-disorders	5	7	6	8	3	2	1	4
Cleveland	5	7	2	8	4	3	6	1

Haberman	7	8	5.5	5.5	1	4	2	3
Ionosphere	1	8	6	7	4	5	2	3
Pima Indian	1	7	4	8	3	5	2	6
Votes	6	8	4	7	2.5	25.	1	5
WDBC	1	8	6	7	3	5	2	4
German	7	8	2	6	4	1	3	5
Monk 2	1	8	3.5	7	3.5	5	2	6
Splice	1	8	7	5	6	4	2	3
vowel	8	3.5	3.5	3.5	7	3.5	3.5	3.5
Ecoli-0-1_vs_2-3-5	7	6	5	8	3	1	2	4
Ecoli-0-1-4-7_vs_5-6	2	8	7	6	4	1	3	5
Ecoli-0-2-3-4_vs_5	1	6	8	7	4	2	3	5
Ecoli-0-6-7_vs_3-5	5	6	8	7	2	1	3	4
Ecoli-0-6-7_vs_5	3	8	5	7	1	6	2	4
Ecoli4	8	6	5	7	3	1	2	4
Glass-0-1-4-6_vs_2	8	7	4.5	4.5	6	1	3	2
Glass-0-1-	1	8	3	7	5	4	2	6

5_vs_2								
Glass-0-1-6_vs_2	1	8	5	7	4	3	2	6
Glass-0-1-6_vs_5	8	7	5	6	3	1	4	2
Glass-0-6_vs_5	6	3	7	8	4	1.5	5	1.5
Glass2	7	8	2	2	5	6	4	2
Yeast-0-2-5-7-9_vs_3-6-8	5	8	3	6	2	1	4	7
Yeast-0-5-6-7-9_vs_4	3	6	7	4	8	2	1	5
Yeast-2_vs_4	6	7	1	8	3	5	2	4
Ecoli-0-1-4-6_vs_5	1	8	5	7	4	2	3	6
Glass4	6	7	4	8	2	5	1	3
Vehicle 1	4	5	6	7	2	3	1	8
Vehicle 2	1	8	5	6	3	4	2	7
Shuttle-6_vs_2-3	1	7	5	8	5	5	2	3
Yeast3	7	8	3	6	4	1	2	5
Yeast1	1	7	6	8	2	4	3	5

Yeast5	1	8	3	5	6	4	2	7
Average rank	4.222222	6.930556	4.791667	6.541667	3.694444	2.986111	2.611111	4.222222

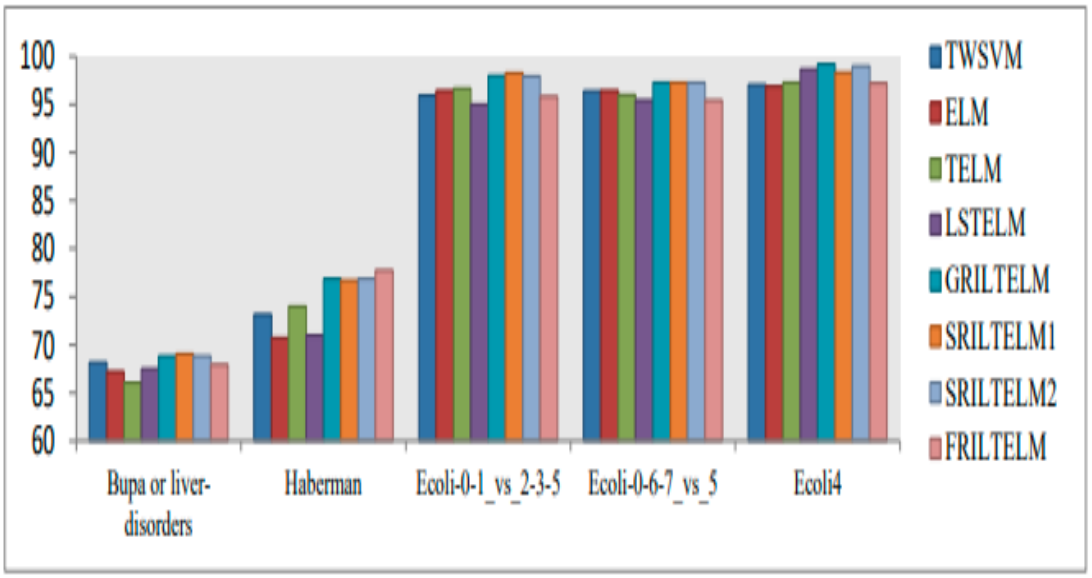


FIGURE 6. 7 . On UCI real-world datasets, we compare the accuracy graphs of TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM utilizing the Multiquadric RBF function.

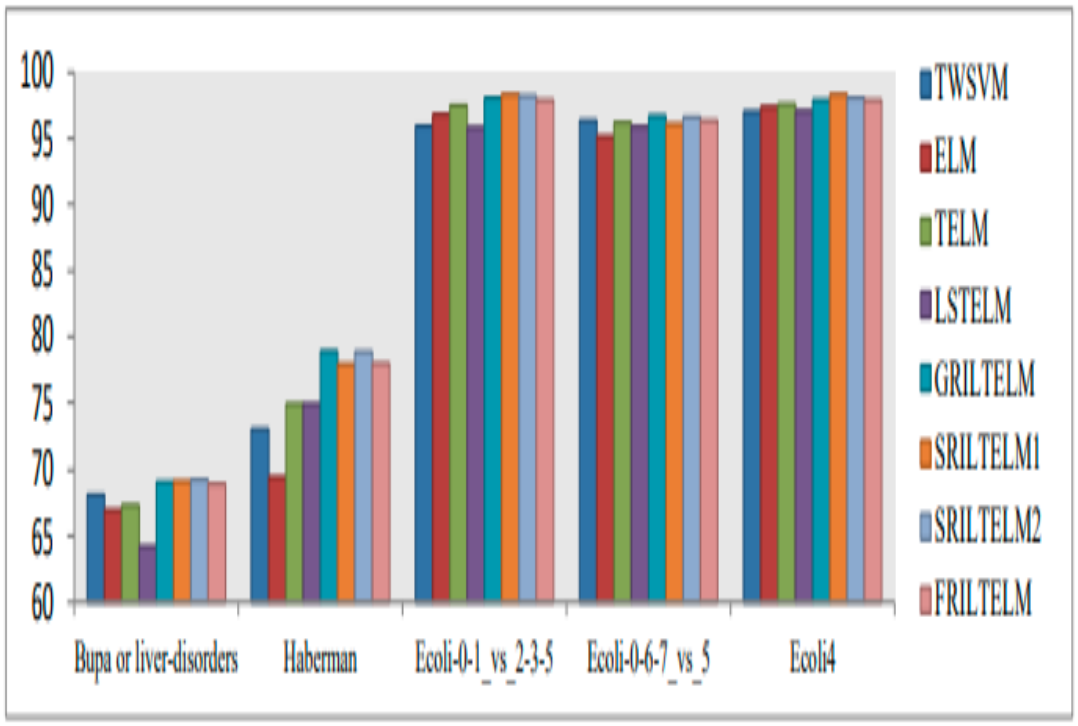


FIGURE 6. 8 A graphical depiction of the accuracy of several models using UCI real-world datasets as judged by the Gaussian RBF kernel: TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM.

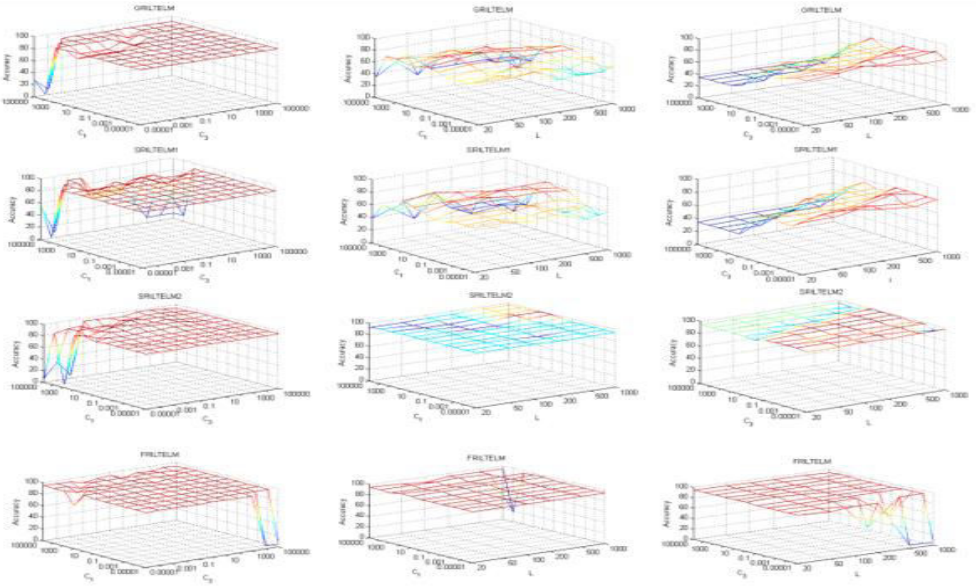


FIGURE 6. 9 The following models are shown graphically according to their accuracy as evaluated by the Gaussian RBF kernel: TWSVM, ELM, TELM, LSTELM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM. The datasets used are those from UCI Real-World.

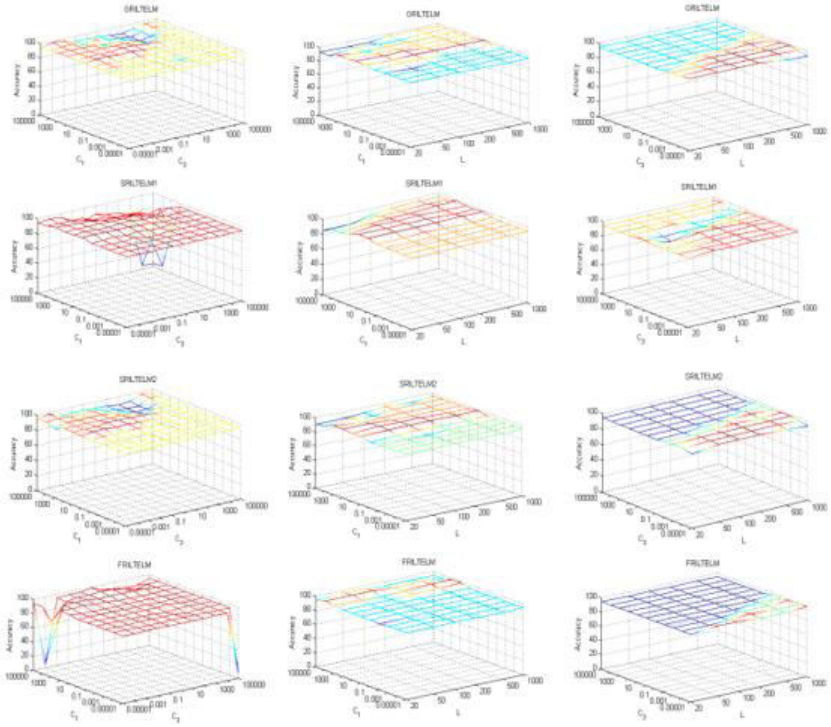


FIGURE 6. 10 Considerations for with regard to C1, C3, and L for Ecoli-0-6-7vs3-5, the parameter sensitivity of the suggested GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM using the Gaussian RBF function

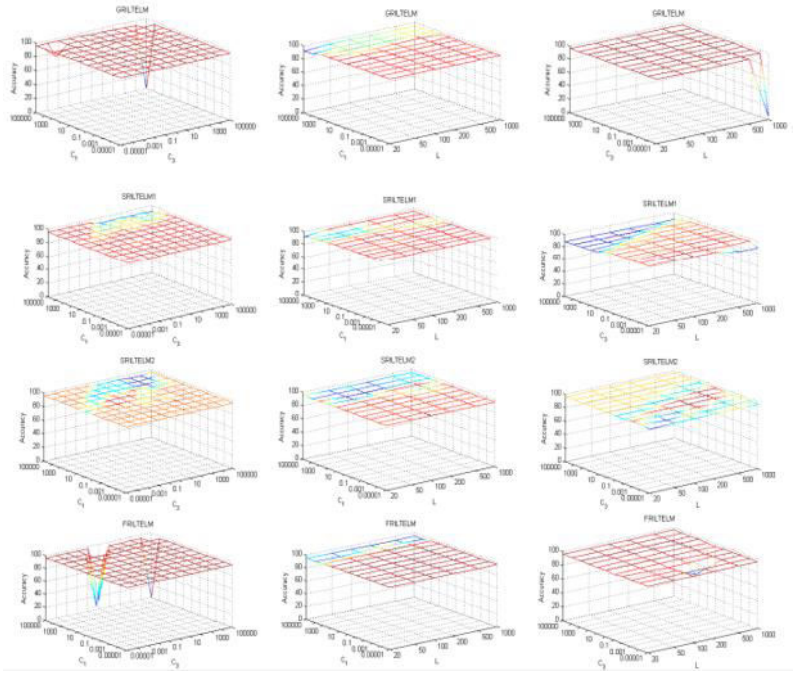


FIGURE 6. 11 Evaluating the Multiquadric RBF function's sensitivity to C1, C3, and L for Yeast5 in relation to the proposed GRILTELM, SRILTELM1, and FRILTELM parameters

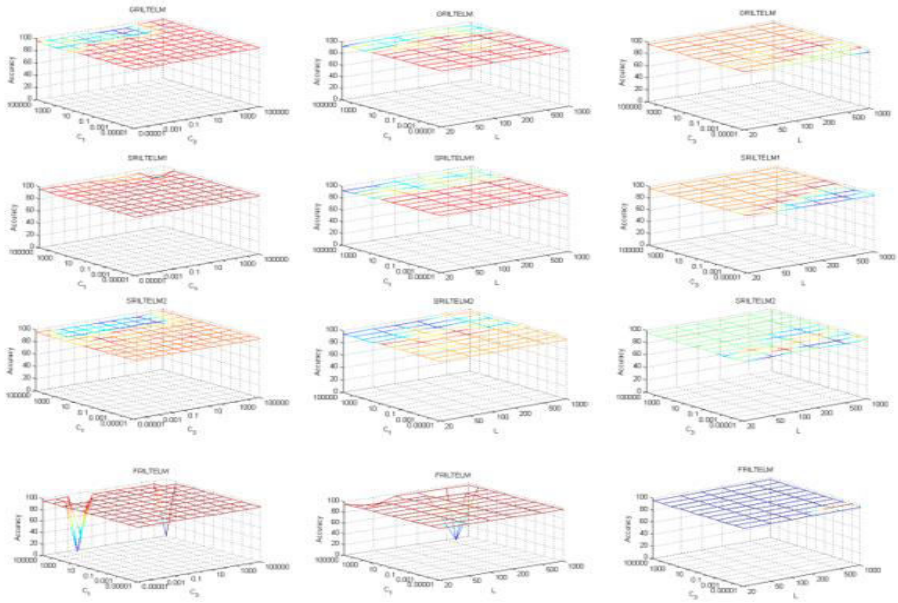


FIGURE 6. 12 The impact of the proposed GRILTELM, SRILTELM1, and FRILTELM parameters on C1, C3, and L in Yeast5, as assessed by the Gaussian RBF calculation.

The RILTELM models do not respond to the user-defined parameters, as seen in Figures 6.9-6.12. We were able to achieve our target with less than 10 iterations of our modal, as shown in Figures 6.13-6.14 (a)-(d) for all RILTELM models that used Multiquadric RBF nodes and Gaussian RBF nodes, which converged on the PIMA INDIAN dataset.

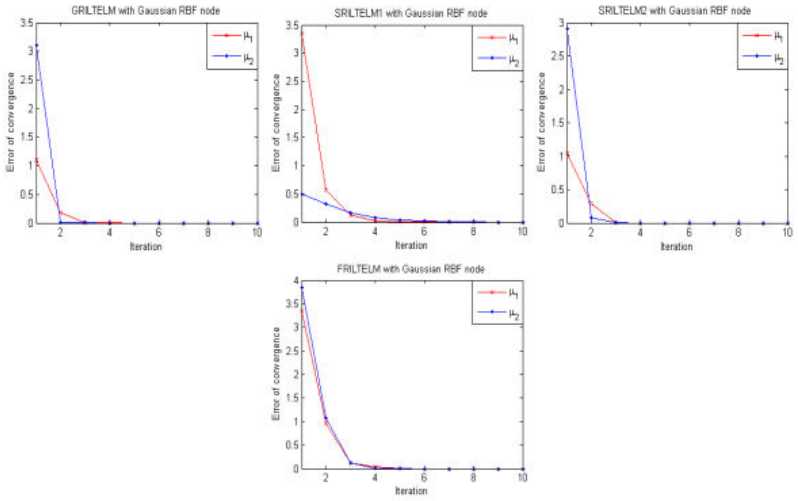


FIGURE 6. 13 Exploring the convergence of utilizing Gaussian RBF on the PIMA dataset, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM

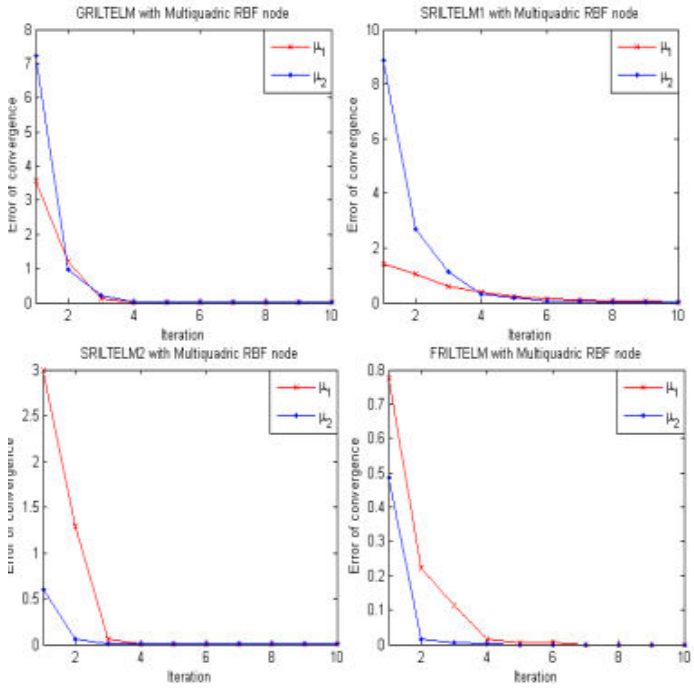


FIGURE 6. 14 Multiquadric RBF node-based convergence of PIMA dataset GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM

Statistical relationship

For the statistical comparison investigation including seven techniques and thirty-six datasets, a trustworthy and economical non-parametric Friedman test using the post hoc test was used. We may utilize Table 6.4 to get significant statistics since many of these methods are the same whether we test them against the null hypothesis. We have ELM, TELM, LSTELM, TWSVM, GRILTELM, SRILTELM1, SRILTELM2, and FRILTELM among these approaches.

$$\chi_F^2 \approx \frac{12\times 36}{8\times 9}\left[\left(4.47222^2+6.84722^2+5.09722^2+5.263889^2+3.01388^2+2.47222^2+3.5^2+5.33333^2\right)-\left(\frac{8\times 9^2}{4}\right)\right]$$

$$\chi_F^2 \approx 86.7917,$$

$$F_F \approx \frac{35\times 86.7917}{36\times 7-86.7917} \approx 18.3871.$$

The F distribution with degrees of freedom of (7, 245) is applied to seven algorithms and thirty-six datasets FF, with results distributed as follows: (8-1,(8-1) ×(36-1)) = (7, 245). F(7, 245) = 2.04707 at α = 0.05 is the crucial value. The alternative hypothesis is

therefore supported as it exceeds the crucial value of F(7, 245). Utilizing the post hoc Nemenyi test, we conducted further methodological comparisons in pairs. The determined core difference (CD) at p = 0.10 is 2.780 $\sqrt{\frac{8\times9}{6\times36}}$= 1.605.

a) Because the multiquadric function's average rank variation with TWSVM, ELM, and SRILTELMI is greater than 1.605 (4.47222-2.47222-2) and (6.84722-2.47222-4.375). Consequently, when compared to TWSVM and ELM, the multiquadric function SRILTELMI approach performs better.

b) Using the multiquadric function, there is a huge discrepancy rank of more than 1.605 among TELM, LSTELM, and SRILTELMI (6.84722-2.47222-4.375) and (5.26388-2.47222-2.7916). Consequently, SRILTELMI is better than TELM and LSTELM.

c) The difference between FRILTELM and SRILTELMI's average ranks, exceeds 1.605 when computed using the multiquadric function. Consequently, FRILTELM is beaten out by the SRILTELMI technique, which is based on multiquadric functions.

d) Using SRILTELMI and a multiquadric function, find the rank differences of the proposed methods GRILTELM and SRILTELM2 (3.01388-2.47222=0.54166) and (3.5-2.47222=1.02777). Since the algorithms' efficacy is almost same, there may not be any notable distinctions between them that a post hoc test can uncover.

Using the same Friedman test, we examine seven more methods that use the Gaussian RBF node over all 36 datasets shown in table 6.5.

$$\chi_F^2 \approx \frac{12\times36}{8\times9}\left[\left(4.22222^2+6.93055^2+4.79166^2+6.54166^2+3.69444^2+2.98611^2+2.61111^2+4.22222^2\right)-\left(\frac{8\times9^2}{4}\right)\right]$$

$$\chi_F^2 \approx 100.9444,$$

$$F_F \approx \frac{36\times100.9444}{36\times7-100.9444} \approx 23.3891$$

The reason for selecting the null hypothesis is because the critical value of F(7, 245) = 2.04707 is less than the value of F_{F} = 23.3891. And here are the outcomes of our post hoc Nemenyi test analysis of algorithm pairings:

a) a) Since the variances of TWSVM, ELM, TELM, and SRILTELM2 are all more than 1.605, we may utilize the Gaussian function on them. (4.222-2.6111.611), (6.930-2.611=1.611), and (4.791-2.6112.180). The outcome is that SRILTELM2 outperforms TWSVM, ELM, and TELM when using a Gaussian function.

b) Deviations from LSTELM and SRILTELM2 that are more than 1.605 as determined by the Gaussian function (6.54166 - 2.61111 = 3.93055) and (4.22222-2.61111-1.61111), respectively. It is possible that the SRILTELM2 approach is superior than LSTELM and FRILTELM.

c) Compare the two methods GRILTELM and SRILTELMI using a Gaussian function; SRILTELM2 (3.69444 - 2.611111 = 1.08333) i and (2.98611-2.61111=0.375) are the two methods that were suggested. Therefore, it's possible that a post hoc test won't pick up on any major differences between the methods, leading to the conclusion that they're both equally capable.

The RILTELM model, which we present here, is an implicit Lagrangian twin extreme learning machine that relies on regularization and uses primal to solve unconstrained convex minimization problems using gradient-based iterative methods. The issue is expressed as a 2-norm of a vector of lax variables so that we may achieve a high degree of convexity. We use the functional iterative method, the generalized derivative approach, or the smooth approximation technique to solve problems in primary space by replacing the non-smooth plus function with the smooth approximation function. In primal space, RILTELM yields the closest approximation solution, making it the better of the two. To create a stable and well-posed model that meets the requirements of the SRM concept, a regularization component is added to the initial expressions. To utilize RILTELM, you won't need a plethora of toolboxes. Both computation efficiency and generalization effectiveness are enhanced. We show that the suggested SRILTELM1 and SRILTELM2 considerably surpass the other conventional methods in terms of prediction accuracy after conducting computer tests on both simulated and actual datasets and comparing them to TWSVM, ELM, TELM, and LSTELM. The quicker

learning rate of the RILTELM compared to the TELM and TWSVM in several examples demonstrates its utility and applicability.

CHAPTER 7
CONCLUSION, RECOMMENDATIONS AND FUTURE SCOPE

CHAPTER 7
CONCLUSION, RECOMMENDATIONS AND FUTURE SCOPE

7.1 CONCLUSION

In an effort to circumvent a number of critical shortcomings in existing regression and classification models, this study investigates machine learning-based models that make use of optimal kernel-generated surfaces to tackle classification problems. Noise, outliers, and poor generalization are common problems with these models; these difficulties can compromise the accuracy and robustness of machine learning methods. New approaches to improve classification accuracy and produce more trustworthy results are suggested by this research, which examines improved models built on non-parallel kernel-generated surfaces. The development of better models that can overcome the problems with current supervised machine learning approaches is a major contribution of this research.

One of the main goals is to develop trustworthy algorithms for classification and regression that can handle noisy training data. Since noise and outliers are common in real-world datasets and can cause models to perform poorly, this section of the study is crucial. This is why resilient loss functions, which improve the models' capacity to fit noisy data, are investigated in the study. Machine learning systems' prediction accuracy and robustness are enhanced by include these loss functions, which make the models better able to handle poor input.

Additionally, this paper introduces a unique technique called URALTSVR, that denotes an uneven distribution Support vector regression using Lagrangian v-twin and pinball loss; it offers a way of expressing the current SVM models. To achieve this goal, the research employs gradient-based iterative approaches, which provide a superior method for handling data and noise variations. By centering on non-parallel kernel-generated surfaces, this model offers a new approach to classification issues and allows for improved fitting to complex data distributions. By experimenting with various implementations, the research aims to determine the optimal gradient-based method for solving the optimization issue associated with the URALTSVR. The study also delves into RILTELM, a regularized version of the Lagrangian twin extreme learning machine.

To ensure more accurate generalization in categorization issues, this model is built to enhance classification performance by standardizing the parameters of the twin extreme learning machine. The study expects RILTELM to surpass current machine learning classifiers, especially in binary classification tasks, in accuracy and robustness by using gradient-based approaches to solve this model. It is believed that adding a regularization factor to the RILTELM model will make it a better classifier in real-world situations by increasing its capacity to generalize to unseen data.

The research does more than only look into URALTSVR and RILTELM; it also presents a number of additional methods for making classifications more resilient. The use of a twin-bounded support vector machine in conjunction with a squared pinball loss classification model is one strategy that aids in using binary classification applications with noisy data. This approach seeks to circumvent the limitations of traditional support vector machines (SVMs) by providing a more resilient answer to classification challenges. This model improves its performance in noisy environments by continually refining the classification decision boundaries, using a functional iterative technique.

Also included in the paper are two fuzzy-based models: IFLSSVM and IFLSTBSVM, which represent two different types of support vector machines: intelliistic fuzzy least square and intuitionistic fuzzy least square twin bounded, respectively. To combat data noise, these models use fuzzy membership ideas, providing a novel approach to dealing with ambiguous or imprecise information. Fuzzy logic makes these models more robust against noisy data by better capturing the inherent uncertainty in real-world datasets. By using fuzzy-based models, the suggested machine learning models become even more versatile and adaptable, opening up new possibilities for enhancing categorization performance.

Additionally, the optimization issue for HN-TSVR is investigated in the article by including a regularization expression from structural risk minimization (SRM) theory. This approach is used to build regularization-based twin support vector regression (RHN-TSVR), a model that is very effective in handling noise and outliers. With the use of a wide margin distribution-based machine-based regression framework and a least squares loss function, the model can easily tackle optimization challenges via matrix inversion. With better performance than conventional regression approaches,

this model offers a viable alternative for regression tasks involving datasets that are noisy or otherwise irregular.

By introducing these complex machine learning models, the article hopes to address some of the most pressing issues in classification and regression. This research's suggested models are ideal for practical use because of their enhanced robustness, noise management, and generalizability. More accurate and dependable predictions are produced by these models when kernel-generated surfaces and non-parallel kernel approaches are included. This allows them to better reflect the complex structures of the data.

Various novel methods for enhancing ML models for regression and classification problems are detailed in the research. Iterative methods based on gradients and fuzzy logic, as well as robust models like URALTSVR, RILTELM, and RHN-TSVR, have made great strides in overcoming the shortcomings of previous machine learning algorithms. When it comes to dealing with noisy data, improving classification accuracy, and making more trustworthy predictions, these models show promise. The study shows that non-parallel kernel approaches and kernel-generated surfaces can improve machine learning performance, and the results should help with the continuous improvement of robust machine learning models.

7.2 RECOMMENDATIONS OF THE STUDY

Several important suggestions for improving the creation and use of these sophisticated models can be derived from the results of this work on machine learning-based models that use optimal kernel-generated surfaces to handle classification problems. Improving the models' practical implementation, directing future research, and resolving potential obstacles found throughout the study are the goals of these proposals.

1. **Enhancement of Resilient Loss Functions**: While this study investigates resilient loss functions, future research should focus on further refining these loss functions to handle a broader range of data imperfections, including more extreme outliers and noisy datasets. Exploring the use of hybrid loss functions that combine the strengths of different loss functions could help in making models even more resilient to noisy data, improving both classification and regression tasks.

2. **Integration of Deep Learning Techniques**: We primarily concentrate on classic ML models, such as SVMs and ELMSs. However, kernel-based models might learn more complicated data representations if deep learning approaches are used alongside them especially for high-dimensional datasets. Future work could explore hybrid models that combine the interpretability of classical models with the power of deep neural networks.

3. **Scalability of Models**: While the models proposed in this study perform well in controlled environments, scalability remains a critical concern when applying these models to large, real-world datasets. Future research should explore ways to optimize the computational efficiency of these models. Techniques such as parallel computing, dimensionality reduction, or approximate kernel methods could be employed to make these models more scalable without sacrificing performance.

4. **Applicability to Multiclass Classification**: The study has primarily focused on binary classification and regression problems. However, many real-world problems involve multiclass classification, and future work should extend the proposed models to handle multiclass scenarios effectively. This could involve the development of new strategies for combining binary classifiers into multiclass systems, or exploring kernel techniques specifically designed for multiclass classification.

5. **Model Interpretability and Transparency**: Machine learning models, especially kernel-based ones, are often considered black boxes, which limits their interpretability and trust in practical applications. Future studies should focus on improving the transparency of these models by developing methods for explaining the decision-making process. Techniques such as feature importance analysis, sensitivity analysis, to make the models easier to understand, it would be beneficial to use local explanation techniques such as SHAP or LIME.

6. **Application of Models to Real-World Problems**: The models developed in this study have shown promising theoretical results. However, their effectiveness in real-world applications should be further investigated. Future research could focus on applying these models to practical domains such as healthcare, finance, and social media analytics, where noise and outliers are common. Such applications

could validate the robustness of the models and help in identifying any further improvements required for practical deployment.

7. **Use of Ensemble Methods**: Another area for future research is the exploration of ensemble methods, where multiple models are combined to improve the overall performance. Combining the strengths of the proposed models with other machine learning algorithms could result in a more powerful and accurate system. Techniques such as bagging, boosting, and stacking could be applied to improve model performance, particularly in noisy and high-dimensional data settings.

8. **Real-Time Classification and Regression**: In many real-time applications, such as fraud detection and autonomous driving, the ability to quickly classify and predict outcomes is crucial. Future work could explore the use of these models in real-time systems, where computational efficiency and low latency are important. This could involve optimizing the models for faster inference times or applying them to streaming data.

9. **Incorporation of Transfer Learning**: Transfer learning, where knowledge gained from one task is applied to another, could be beneficial for improving model performance, especially when labeled data is scarce. Future research could investigate the application of transfer learning techniques to the proposed models, particularly in scenarios where labeled data is limited but similar datasets are available.

10. **In-depth Comparative Studies**: Finally, while this study has shown the potential of the models in addressing classification challenges, it would be beneficial to conduct in-depth comparative studies with other state-of-the-art machine learning techniques. Benchmarking these models against widely used algorithms such as random forests, gradient boosting machines, and deep neural networks would provide a clearer understanding of their advantages and limitations.

To sum up, this study's suggested models are a huge step forward in the fight against the difficulties of machine learning's classification and regression problems. The models could be further improved and applied to a wider range of real-world problems with the help of the suggestions given here, but there is always space for improvement.

7.3 FINDINGS OF THE STUDY

The following are the main findings of the study:

- **Supplemental work on asymmetric v-twin SVR:** A novel approach, LAsy- TSVR (Lagrangian asymmetric twin support vector regression with improved regularization), was created in our work. This method effectively employs the SRM principle, which is fundamental to statistical learning, by using a pinball loss function. The LAsy-TSVR solution is handled by a convergent iterative strategy, which is different from other current and classic TSVR variations. An advantage of the given LAsy- TSVR over earlier systems is its ability to handle different types of uniform and Gaussian noise, symmetrical and asymmetrical patterns. On top of that, it serves its purpose and is easy to use. No sacrifices were made to generalization performance or processing cost during testing on several synthetic datasets with symmetric and heteroscedastic patterns of uniform and Gaussian noise, and it passed with flying colors.

- **Unconstrained asymmetric v-twin support vector regression:** Our study centered on URALTSVR, a robust asymmetric Lagrangian-twin support vector regression algorithm. This technique generates gradient-based iterative methods employing generalized derivative and smoothing strategies to solve the regression issue. We then employed the Newton iterative technique to get a better solution. By modifying the settings and using the asymmetric pinball loss function, our suggested method is able to manage datasets disrupted by noise. To ensure the SRM and provide a stable and well-posed model, a regularization component is added to the optimization function. Several investigations on synthetic and real-world datasets show that URALTSVR is suitable and effective. After looking at how different linear and Gaussian kernels perform in SVR, TSVR, and Asy-TSVR, we found that the suggested SRALTSVR1 243 approach was the most effective. The models provided here have a computational cost that is either lower than or about equal to the approaches outlined above.

- **Huber loss function advancements in twin support vector regression:** Our investigation on RHN-TSVR, HN-TSVR's singularity problem may be addressed by using a regularized version of TSVR with Huber loss. The structural risk reduction concept is included by this form via the application of regularization

based upon support vector regression with Huber loss. Furthermore, use a range of significant noise levels (e.g.,0%,5%, and 10%) to evaluate the RHN-TSVR's noise insensitivity. From what we know, the loss function of TSVR is -insensitive, meaning it does not take into account outliers or other types of noise. For minor mistakes, the basic Huber loss function has a quadratic form; for larger errors, it takes a linear form. When applied to datasets with outliers and Gaussian noise, the Laplacian loss function improves prediction accuracy. To test and assess the suggested method, we use both synthetic datasets with different kinds of non-linear kernel noises and a range of real-world datasets with different degrees of statistical significance. Typically, the RHN-TSVR achieves better predictive power than traditional methods with the same or less processing time required.

- **Progress on machine-based regression for large margin distributions:** This research delves into a computationally efficient method for solving regression problems using a least squares big margin distribution machine, utilizing the mathematical formulations from LDMR and PLSTSVR. A system of linear equations is solved using the proposed LS-LDMR. Thus, unlike LDMR, -SVQR, TSVR, and SVR, we need to calculate the inverse of the matrix. When comparing the computational cost and prediction ability of state-of-the-art algorithms on synthetic and real-world datasets, we discovered that our suggested LS-LDMR outperforms them. Studies are conducted using statistical methods for the suggested LS-LDMR using SVR, TSVR, PLSTSVR, -HSVR, -SVQR, MDR, and LDMR to strengthen the usefulness and efficiency of LSLDMR.

- **A twin support vector machine that makes use of the squared pinball loss function to improve:** The functional iterative approach for twin bounded support vector machines (Spin-FITBSVM) provides a new angle on the traditional twin model of support vector machines (SVM) by including this loss function. Even in noisy situations, it performs well for sample classification. When solving the first problem, taking the regularization parameter into account, our suggested method, Spin-FITBSVM, applies the SRM principle. Additionally, it guarantees that Spin-FITBSVM's resilience reaches its theoretical maximum. Our suggested method, Spin-FITBSVM, has been computationally compared to previously reported methods on a number of datasets, including fake and benchmark real-world ones,

as well as SVM, TSVM, pin-TSVM, pin-GTSVM, GHTSVM, and SPTWSVM. Based on these results, it seems that the Spin-FITBSVM method outperforms the others while using less computing resources.

- **Support vector machine enhancement with the use of intuitionistic fuzzy values:** We have investigated two improved and efficient SVM-based models, in particular IFLSSVM and IFLSTBSVM, which use intuitionistic fuzzy values, to take into consideration the influence of noise and outliers in actual data. Instead of using QPPs in support vector machines (SVMs) to identify the fuzzy weighted value of positive and negative training samples, this method identifies the optimum hyperplane using a series of linear equations. Members and non-members alike may benefit from IFN services. The IFN function and the ability to distinguish between support vectors and noise are both used during training for example classification. A lack of generalizability, sluggish learning rates, noise, and outliers are the main issues that the suggested solutions aim to fix. For SVM-based methods, IFLSTBSVM is the best option, beating out LS-SVM, TWSVM, EFLSSVM, and IFTWSVM. In order to investigate the practicality and utility of the proposed IFLSSVM and IFLSTBSVM, we ran comprehensive experiments on several publically accessible real-world benchmark datasets and created synthetic datasets for binary classification in non-linear situations with different levels of significant noise. The results of the experiments demonstrate that the suggested models outperform the previously reported classification methods in terms of producing more broadly applicable models with less computational overhead.

- **Extreme learning machine with no constraints on its twins:** Introducing RILTELM, an extreme learning machine that uses gradient-based iterative techniques to handle unconstrained convex minimization problems, based on primal-based regularized-based implicit Lagrangian twins. The 2-norm of a vector of slack variables is used in this approach to significantly convexify the issue. Utilizing a functional iterative strategy, a modified derivative approach, or a smooth approximation method, we resolved the issues in primal space by substituting a smooth approximation function for the non-smooth addition function. The main benefit of RILTELM over its twin is that it gives the best approximation solution in primal space. To ensure compatibility with the SRM idea and to achieve a stable

and well-posed model, a regularization term is added to the initial expressions. There is a further distinction. Both computational efficiency and generalizability are enhanced as a consequence. The suggested SRILTELM1 and SRILTELM2 models outperformed their published equivalents in terms of classification accuracy when tested on both real-world and simulated datasets. The suggested RILTELM is feasible and practical as it achieves better learning efficiency than the TELM in the majority of cases.

7.4 FUTURE SCOPE OF THE STUDY

Machine learning-based models that use optimal kernel-generated surfaces to solve classification problems have a lot of promising avenues for future research and development. There are a lot of ways this work may develop further, increasing its theoretical and practical influence, based on the encouraging findings and methodology presented here. The possible directions for further study and implementation are outlined below.

1. **Exploration of Advanced Kernel Methods**: While this study focuses on kernel-generated surfaces for classification challenges, future research could investigate the application of more advanced kernel techniques, such as deep or adaptive kernels, to further improve the flexibility and performance of the models. The development of novel kernels that can dynamically adjust based on the data characteristics could provide an even more powerful tool for machine learning models, particularly in complex and high-dimensional datasets.

2. **Generalization to Multiclass and Multi-Label Classification**: This study primarily addresses binary classification problems. However, real-world classification tasks often involve multiple classes or labels. The future scope could include extending the proposed models to handle multiclass or multi-label classification problems. This would require the adaptation of the kernel methods and loss functions used, potentially leading to new formulations and optimization techniques that can handle more complex classification tasks.

3. **Application to Real-World Domains**: While this research demonstrates the theoretical potential of the proposed models, future studies could focus on applying these models to real-world applications. Domains such as healthcare,

finance, autonomous driving, and social media analytics are rich with noisy, unstructured, and high-dimensional data, this group is well-suited to evaluate the suggested models' accuracy and robustness. The difficulties of putting these models into practice may be better understood and potential improvement areas can be more easily identified with the help of real-world case studies.

4. **Incorporation of Unsupervised and Semi-Supervised Learning**: A promising direction for future research is to incorporate unsupervised and semi-supervised learning techniques into the models. These methods are particularly useful when labeled data is scarce or expensive to obtain. Future studies could explore how the proposed models could be modified to learn from unlabeled data, using techniques like self-training, co-training, or clustering-based approaches. This would expand the applicability of the models to a wider range of problems where labeled data is limited.

5. **Integration with Deep Learning Models**: Combining traditional machine learning models with deep learning techniques could open up new opportunities for model improvement. Future research could investigate hybrid models that integrate the strengths of kernel-based approaches with the power of deep neural networks. This integration could lead to models capable of capturing both low-level and high-level data features, resulting in improved performance on tasks such as image recognition, speech processing, and natural language understanding.

6. **Optimization for Scalability and Efficiency**: As machine learning models continue to be applied to larger datasets, scalability and computational efficiency become crucial factors. Future work could focus on optimizing the proposed models for handling large-scale data, leveraging techniques like parallel computing, distributed systems, and cloud computing. Research could also explore more efficient algorithms for solving the optimization problems in these models, such as stochastic gradient descent or other optimization techniques that reduce the computational burden.

7. **Development of Real-Time Systems**: With the increasing demand for real-time predictions in applications like fraud detection, autonomous vehicles, and

industrial automation, future research could focus on adapting these models for real-time systems. This would involve improving the speed of inference and reducing the latency of predictions, ensuring that the models can handle streaming data efficiently while maintaining accuracy.

8. **Hybridization with Ensemble Methods**: Ensemble methods, which combine multiple models to improve performance, could be explored as a future direction. By combining the strengths of various machine learning models, such as support vector machines, extreme learning machines, and deep neural networks, ensemble techniques could increase the robustness and accuracy of the classification and regression tasks. Techniques such as bagging, boosting, or stacking could be applied to the proposed models to create more powerful, high-performing systems.

9. **Focus on Model Interpretability and Transparency**: As machine learning models become increasingly complex, their interpretability and transparency are crucial for practical applications, particularly in sensitive fields like healthcare and finance. Future work could focus on improving the explainability of the proposed models, developing tools and techniques that allow users to understand the decision-making process of these models. Methods such as SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) could be integrated to enhance the interpretability of the models, making them more transparent and trustworthy.

10. **Incorporation of Transfer Learning**: When data is few, one approach to enhance learning is via transfer learning, which uses prior knowledge from one area to fill in the gaps. When labelled data is insufficient, future research may look at ways to include transfer learning with the suggested models. The models might become more efficient and effective by transferring information from comparable tasks or datasets, which would allow them to perform better with less training data.

11. **Ethical Considerations and Fairness**: Making ensuring that machine learning models behave properly and ethically is crucial since they are being used in so many different areas. Future research could focus on addressing potential biases

in the models, particularly in classification tasks that involve human decision-making. Developing methods to ensure fairness, transparency, and accountability in machine learning predictions will be critical for the adoption of these models in sensitive areas such as criminal justice, hiring practices, and lending decisions.

This study has a huge potential for growth and expansion in the years to come. The proposed machine learning models have great potential to evolve and impact many different sectors, from making them more robust and scalable to using them in real-world applications. Future study can build on this work by investigating these avenues, which will help develop machine learning methods.

REFERENCES

REFERENCES: -

[1] W. Noble, "What is a Support Vector Machine?," *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2007, doi: 10.1038/nbt1206-1565.

[2] H. Sharma and S. Kumar, "A Survey on Decision Tree Algorithms of Classification in Data Mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016.

[3] S. Naeem, A. Ali, S. Anam, and M. Ahmed, "An Unsupervised Machine Learning Algorithms: Comprehensive Review," *IJCDS Journal*, vol. 13, no. 1, pp. 911–921, 2023, doi: 10.12785/ijcds/130172.

[4] H. Tan, "Machine Learning Algorithm for Classification," *Journal of Physics: Conference Series*, vol. 1994, no. 1, pp. 1–5, 2021, doi: 10.1088/1742-6596/1994/1/012016.

[5] D. Abdullah and A. Abdulazeez, "Machine Learning Applications Based on SVM Classification: A Review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 81–90, 2021, doi: 10.48161/qaj.v1n2a50.

[6] D. Liu, Y. Shi, Y. Tian, and X. Huang, "Ramp Loss Least Squares Support Vector Machine," *Journal of Computational Science*, vol. 14, no. 3, pp. 1–7, 2016, doi: 10.1016/j.jocs.2016.02.001.

[7] D. Gupta and B. Richhariya, "Entropy-Based Fuzzy Least Squares Twin Support Vector Machine for Class Imbalance Learning," *Applied Intelligence*, vol. 48, no. 1, pp. 1–20, 2018, doi: 10.1007/s10489-018-1204-4.

[8] S. Ding, J. Yu, B. Qi, and H. Huang, "An Overview on Twin Support Vector Machines," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 245–252, 2014, doi: 10.1007/s10462-012-9336-0.

[9] T. Hofmann, B. Schölkopf, and A. Smola, "Kernel Methods in Machine Learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1–52, 2007, doi: 10.1214/009053607000000677.

[10] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou, "A Comparison

Study of Different Kernel Functions for SVM-Based Classification of Multi-Temporal Polarimetry SAR Data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-2/W3, no. 2, pp. 281–285, 2014, doi: 10.5194/isprsarchives-XL-2-W3-281-2014.

[11] A. Alnuaimi and T. Albaldawi, "An overview of machine learning classification techniques," *BIO Web of Conferences*, vol. 97, no. 4, p. 00133, 2024, doi: 10.1051/bioconf/20249700133.

[12] M. Almuqati, F. Sidi, S. N. Mohd Rum, M. Zolkepli, and I. Ishak, "Challenges in Supervised and Unsupervised Learning: A Comprehensive Overview," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 4, pp. 1449-1455, 2024, doi: 10.18517/ijaseit.14.4.20191.

[13] R. N. Mohalder, M. A. Hossain, and N. Hossain, "Classifying the supervised machine learning and comparing the performances of the algorithms," *Int. J. Adv. Res.*, vol. 12, no. 1, pp. 422-438, 2024, doi: 10.21474/ijar01/18138.

[14] O. C. Asogwa, N. Eze, C. C., K. Aruah, and I. Chimezie, "On the Comparative Study of the Supervised Machine Learning Models," *Afr. J. Math. Stat. Stud.*, vol. 7, no. 4, pp. 162-173, 2024, doi: 10.52589/AJMSS-ILF4K7JB.

[15] M. Jamaner Rahaman, "A Comprehensive Review to Understand the Definitions, Advantages, Disadvantages and Applications of Machine Learning Algorithms," *Int. J. Comput. Appl.*, vol. 186, no. 31, pp. 43-47, 2024, doi: 10.5120/ijca2024923868.

[16] Z. Zhang, X. Lu, and S. Cao, "An efficient detection model based on improved YOLOv5s for abnormal surface features of fish," *Math. Biosci. Eng.*, vol. 21, no. 2, pp. 1765-1790, 2024, doi: 10.3934/mbe.2024076.

[17] Z. Heydari and A. Stillwell, "Comparative Analysis of Supervised Classification Algorithms for Residential Water End Uses," *Water Resour. Res.*, vol. 60, no. 6, 2024, doi: 10.1029/2023WR036690.

[18] M. Laurer, W. Atteveldt, A. Casas, and K. Welbers, "Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning

with Deep Transfer Learning and BERT-NLI," *Political Anal.*, vol. 32, no. 1, pp. 1-33, 2023, doi: 10.1017/pan.2023.20.

[19] Y. Wei, C. Yang, L. He, F. Wu, Q. Yu, and W. Hu, "Classification for GM and Non-GM Maize Kernels Based on NIR Spectra and Deep Learning," *Processes*, vol. 11, no. 2, p. 486, 2023, doi: 10.3390/pr11020486.

[20] R. Matura, R. Thakur, and P. ., "Comparing the Performance of Different Supervised Learning Algorithms," *J. Artif. Intell. Capsule Networks*, vol. 5, no. 1, pp. 52-68, 2023, doi: 10.36548/jaicn.2023.1.005.

[21] T. T. Khoei and N. Kaabouch, "Machine Learning: Models, Challenges, and Research Directions," *Future Internet*, vol. 15, no. 13, p. 332, 2023, doi: 10.3390/fi15100332.

[22] Z. Ali, Z. Abduljabbar, H. Tahir, A. Sallow, and S. Almufti, "Exploring the Power of eXtreme Gradient Boosting Algorithm in Machine Learning: a Review," *Acad. J. Nawroz Univ.*, vol. 12, no. 2, pp. 320-334, 2023, doi: 10.25007/ajnu.v12n2a1612.

[23] M. Miric, N. Jia, and K. Huang, "Using Supervised Machine Learning for Large-scale Classification in Management Research: The Case for Identifying Artificial Intelligence Patents," *Strategic Manag. J.*, vol. 44, no. 2, pp. 491-519, 2023, doi: 10.1002/smj.3441.

[24] M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," *Computers*, vol. 12, no. 5, p. 91, 2023, doi: 10.3390/computers12050091.

[25] S. Jain and R. Rastogi, "Parametric non-parallel support vector machines for pattern classification," *Mach. Learn.*, vol. 113, no. 3, pp. 1-28, 2022, doi: 10.1007/s10994-022-06238-0.

[26] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, "An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs," *Sensors*, vol. 22, no. 4, p. 1407, 2022, doi: 10.3390/s22041407.

[27] I. Muraina, M. Agoi, B. OMOROJOR, A. Ademola, and R. Ajetunmobi, "Decision Making and Machine Learning Algorithms' Selection with Artificial Intelligent Rule-Based Expert System," *Int. J. Res. Sci. Innov.*, vol. 09, no. 4, pp. 54-60, 2022, doi: 10.51244/IJRSI.2022.9406.

[28] S. Sharma and P. Mandal, "A Comprehensive Report on Machine Learning-based Early Detection of Alzheimer's Disease using Multi-modal Neuroimaging Data," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1-45, 2022, doi: 10.1145/3492865.

[29] Á. Fernández Pascual, J. Bella, and J. Dorronsoro, "Supervised Outlier Detection for Classification and Regression," *Neurocomputing*, vol. 486, no. 2, pp. 10.1016/j.neucom.2022.02.047, 2022.

[30] N. Jain and R. Kumar, "A Review on Machine Learning & Its Algorithms," *Int. J. Soft Comput. Eng.*, vol. 12, no. 5, pp. 1-5, 2022, doi: 10.35940/ijsce.E3583.1112522.

[31] B. Sekeroglu, Y. Kirsal Ever, K. Dimililer, and F. Al-Turjman, "Comparative Evaluation and Comprehensive Analysis of Machine Learning Models for Regression Problems," *Data Intell.*, vol. 4, no. 3, pp. 620-652, 2022, doi: 10.1162/dint_a_00155.

[32] M. Gupta, "A Comparative Study on Supervised Machine Learning Algorithm," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 1, pp. 1023-1028, 2022, doi: 10.22214/ijraset.2022.39980.

[33] J. Pruneski, A. Pareek, K. Kunze, R. Martin, J. Karlsson, J. Oeding, A. Kiapour, B. Nwachukwu, and R. Williams III, "Supervised machine learning and associated algorithms: applications in orthopedic surgery," *Knee Surg. Sports Traumatol. Arthrosc.*, vol. 31, no. 5, 2022, doi: 10.1007/s00167-022-07181-2.

[34] S. Ono and T. Goto, "Introduction to supervised machine learning in clinical epidemiology," *Ann. Clin. Epidemiol.*, vol. 4, no. 3, pp. 63-71, 2022, doi: 10.37737/ace.22009.

[35] N. Dahiya, S. Gupta, and S. Singh, "A Review Paper on Machine Learning Applications, Advantages, and Techniques," *ECS Trans.*, vol. 107, no. 1, pp. 6137-6150, 2022, doi: 10.1149/10701.6137ecst.

[36] A. Arista, "Comparison Decision Tree and Logistic Regression Machine Learning Classification Algorithms to determine Covid-19," *Sinkron*, vol. 7, no. 1, pp. 59-65, 2022, doi: 10.33395/sinkron.v7i1.11243.

[37] P. Bhatt, R. Malhan, P. Rajendran, B. Shah, S. Thakar, Y. J. Yoon, and S. Gupta, "Image-Based Surface Defect Detection Using Deep Learning: A Review," *J. Comput. Inf. Sci. Eng.*, vol. 21, no. 4, pp. 1-23, 2021, doi: 10.1115/1.4049535.

[38] N. Nair, A. Al, and S. Asharaf, "Deep Kernel Machines: A Survey," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 1-20, 2021, doi: 10.1007/s10044-020-00933-1.

[39] H. Wen, T. Li, D. Chen, J. Yang, and Y. Che, "An Optimized Neural Network Classification Method Based on Kernel Holistic Learning and Division," *Math. Probl. Eng.*, vol. 2021, no. 2, pp. 1-16, 2021, doi: 10.1155/2021/8857818.

[40] S. Pal and P. Sharma, "A Review of Machine Learning Applications in Land Surface Modeling," *Earth*, vol. 2, no. 1, pp. 174-190, 2021, doi: 10.3390/earth2010011.

[41] A. Mazlan, N. Sahabudin, M. A. Remli, N.-S. Ismail, M. Mohamad, H. W. Nies, and N. Abd Warif, "A Review on Recent Progress in Machine Learning and Deep Learning Methods for Cancer Classification on Gene Expression Data," *Processes*, vol. 9, no. 8, p. 1466, 2021, doi: 10.3390/pr9081466.

[42] U. M. R. Paturi, S. Cheruku, V. P. K. Pasunuri, S. Salike, N. S. Reddy, and S. Cheruku, "Machine Learning and Statistical Approach in Modeling and Optimization of Surface Roughness in Wire Electrical Discharge Machining," *Mach. Learn. Appl.*, vol. 6, no. 6, p. 100099, 2021, doi: 10.1016/j.mlwa.2021.100099.

[43] D. Saravagi, S. Agrawal, and M. Saravagi, "Opportunities and Challenges of Machine Learning Models for Prediction and Diagnosis of Spondylolisthesis: A

Systematic Review," *Int. J. Eng. Syst. Model. Simul.*, vol. 12, no. 2\3, p. 127, 2021, doi: 10.1504/IJESMS.2021.115534.

[44] R. Hasan, "Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction," *ITM Web Conf.*, vol. 40, p. 03007, 2021, doi: 10.1051/itmconf/20214003007.

[45] J. Kamiri and G. Mariga, "Research Methods in Machine Learning: A Content Analysis," *Int. J. Comput. Inf. Technol.*, vol. 10, no. 2, pp. 2279-0764, 2021, doi: 10.24203/ijcit.v10i2.79.

[46] L. Eckart, S. Eckart, and M. Enke, "A Brief Comparative Study of the Potentialities and Limitations of Machine-Learning Algorithms and Statistical Techniques," *E3S Web Conf.*, vol. 266, no. 4, p. 02001, 2021, doi: 10.1051/e3sconf/202126602001.

[47] R. M. Achshah and Dr. Prakash, "A Simple Approach for Selecting the Best Machine Learning Algorithm," *Int. J. Sci. Eng. Res.*, vol. 12, no. 9, pp. 902-909, 2021.

[48] I. El Guabassi, Z. Bousalem, M. Rim, and A. Qazdar, "Comparative Analysis of Supervised Machine Learning Algorithms to Build a Predictive Model for Evaluating Students' Performance," *Int. J. Online Biomed. Eng.*, vol. 17, no. 02, 2021, doi: 10.3991/ijoe.v17i02.20025.

[49] R. Khalifa, S. Yacout, and S. Bassetto, "Developing Machine-Learning Regression Model with Logical Analysis of Data (LAD)," *Comput. Ind. Eng.*, vol. 151, pp. 1-18, 2020, doi: 10.1016/j.cie.2020.106947.

[50] S. Apsemidis, S. Psarakis, and J. Moguerza, "A Review of Machine Learning Kernel Methods in Statistical Process Monitoring," *Comput. Ind. Eng.*, vol. 142, p. 106376, 2020, doi: 10.1016/j.cie.2020.106376.

[51] D. Maulud and A. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 4, pp. 140-147, 2020, doi: 10.38094/jastt1457.

[52] A. Razaque and A. Alajlan, "Supervised Machine Learning Model-Based Approach for Performance Prediction of Students," *J. Comput. Sci.*, vol. 16, no. 8, pp. 1150-1162, 2020, doi: 10.3844/jcssp.2020.1150.1162.

[53] R. Kenge, "Machine Learning, Its Limitations, and Solutions Over IT," *Int. J. Inf. Technol. Model. Comput.*, vol. 11, no. 2, pp. 73-83, 2020, doi: 10.5958/0975-8089.2020.00009.3.

[54] B. Mahesh, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res. (IJSR)*, vol. 9, no. 1, pp. 381-386, 2019, doi: 10.21275/ART20203995.

[55] Q.-Q. Gao, Y. Bai, and Y.-R. Zhan, "Quadratic Kernel-Free Least Square Twin Support Vector Machine for Binary Classification Problems," *J. Oper. Res. Soc. China*, vol. 7, no. 3, 2019, doi: 10.1007/s40305-018-00239-4.

[56] M. Pradhan, S. Minz, and V. Shrivastava, "A Kernel-Based Extreme Learning Machine Framework for Classification of Hyperspectral Images Using Active Learning," *J. Indian Soc. Remote Sens.*, vol. 47, no. 3, 2019, doi: 10.1007/s12524-019-01021-6.

[57] J. Cao, M. Wang, Y. Li, and Q. Zhang, "Improved Support Vector Machine Classification Algorithm Based on Adaptive Feature Weight Updating in the Hadoop Cluster Environment," *PLOS ONE*, vol. 14, no. 4, pp. 1-18, 2019, doi: 10.1371/journal.pone.0215136.

[58] B. Mahesh, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res. (IJSR)*, vol. 9, no. 1, pp. 381-386, 2019.

[59] S. Rong and B.-w. Zhang, "The Research of Regression Model in Machine Learning Field," *MATEC Web Conf.*, vol. 176, p. 01033, 2018.

[60] Y. C. A. Padmanabha, V. Pulabaigari, and B. Eswara, "Semi-Supervised Learning: A Brief Review," *Int. J. Eng. Technol.*, vol. 7, no. 1-8, p. 81, 2018, doi: 10.14419/ijet.v7i1.8.9977.

[61] J. E. T. Akinsola, "Supervised Machine Learning Algorithms: Classification and Comparison," *Int. J. Comput. Trends Technol. (IJCTT)*, vol. 48, no. 3, pp. 128-138, 2017, doi: 10.14445/22312803/IJCTT-V48P126.

[62] A. Mohamed, "Comparative Study of Four Supervised Machine Learning Techniques for Classification," *Int. J. Appl. Sci. Technol.*, vol. 7, no. 2, pp. 5-18, 2017.

[63] M. Zareapoor, P. Shamsolmoali, D. Jain, H. Wang, and J. Yang, "Kernelized Support Vector Machine with Deep Learning: An Efficient Approach for Extreme Multiclass Dataset," *Pattern Recognit. Lett.*, vol. 115, pp. 1-10, 2017, doi: 10.1016/j.patrec.2017.09.018.

[64] M. Fan, Q. Wang, B. Cao, B. Ye, and A. Imam, "Frequency Optimization for Enhancement of Surface Defect Classification Using the Eddy Current Technique," *Sensors*, vol. 16, no. 5, p. 649, 2016, doi: 10.3390/s16050649.

[65] C. Peng, J. Cheng, and Q. Cheng, "A Supervised Learning Model for High-Dimensional and Large-Scale Data," *ACM Trans. Intell. Syst. Technol.*, vol. 8, 2016, doi: 10.1145/2972957.

[66] Y. Bai, X. Han, T. Chen, and H. Yu, "Quadratic Kernel-Free Least Squares Support Vector Machine for Target Diseases Classification," *J. Combin. Optim.*, vol. 30, no. 4, 2015, doi: 10.1007/s10878-015-9848-z.

[67] M. Iqbal and Z. Yan, "Supervised Machine Learning Approaches: A Survey," *Int. J. Soft Comput.*, vol. 5, no. 3, pp. 946-952, 2015, doi: 10.21917/ijsc.2015.0133.

[68] Y. Tian and X.-C. Ju, "Nonparallel Support Vector Machine Based on One Optimization Problem for Pattern Recognition," *J. Oper. Res. Soc. China*, vol. 3, no. 4, pp. 1-8, 2015, doi: 10.1007/s40305-015-0095-x.

[69] E. Santos, M. Figueiredo, C. Silva, C. Sales, and J. Costa, "Machine Learning Algorithms for Damage Detection: Kernel-Based Approaches," *J. Sound Vib.*, vol. 363, pp. 1-10, 2015, doi: 10.1016/j.jsv.2015.11.008.

[70] M. Pal, A. Maxwell, and T. Warner, "Kernel-Based Extreme Learning Machine for Remote Sensing Image Classification," *Remote Sens. Lett.*, vol. 4, no. 9, pp. 853-862, 2013, doi: 10.1080/2150704X.2013.805279.

[71] Z. Pozun, K. Hansen, D. Sheppard, M. Rupp, and K.-R. Müller, "Optimizing Transition States via Kernel-Based Machine Learning," *J. Chem. Phys.*, vol. 136, no. 17, p. 174101, 2012, doi: 10.1063/1.4707167.

[72] M. Gönen and E. Alpaydın, "Regularizing Multiple Kernel Learning Using Response Surface Methodology," *Pattern Recognit.*, vol. 44, no. 1, pp. 159-171, 2011, doi: 10.1016/j.patcog.2010.07.008.

[73] R. Khemchandani, J. Jayadeva, and S. Chandra, "Optimal Kernel Selection in Twin Support Vector Machines," *Optim. Lett.*, vol. 3, no. 1, pp. 77-88, 2009, doi: 10.1007/s11590-008-0092-7.

[74] P. Jain, J. Garibaldi, and J. Hirst, "Supervised Machine Learning Algorithms for Protein Structure Classification," *Comput. Biol. Chem.*, vol. 33, no. 3, pp. 216-223, 2009, doi: 10.1016/j.compbiolchem.2009.04.004.

[75] S. Agarwal, V. Saradhi, and H. Karnick, "Kernel-Based Online Machine Learning and Support Vector Reduction," *Neurocomputing*, vol. 71, no. 7, pp. 1230-1237, 2008, doi: 10.1016/j.neucom.2007.11.023.

[76] S. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica (Ljubljana)*, vol. 31, no. 7, 2007.

[77] S. Kotsiantis, I. Zaharakis, and P. Pintelas, "Machine Learning: A Review of Classification and Combining Techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159-190, 2006, doi: 10.1007/s10462-007-9052-3.

[78] J. Kivinen, A. Smola, and R. Williamson, "Learning with Kernels," IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2165-2176, 2004, doi: 10.1109/TSP.2004.830991.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1]. M. Yuval, B. Yaman, and Ö. Tosun, "Classification comparison of machine learning algorithms using two independent CAD datasets," *Mathematics*, vol. 10, no. 3, p. 311, 2022.

[2]. M. A. Kashmoola, M. K. Ahmed, and N. Y. A. Alsaleem, "Network traffic prediction based on boosting learning," *Iraqi Journal of Science*, vol. 63, no. 9, pp. 4047–4056, 2022.

[3]. V. Rastogi, S. Satija, P. K. Sharma, and S. Singh, "Machine learning algorithms: Overview," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 11, no. 9, pp. 512–517, 2020.

[4]. H. Xue, Z.-P. Shao, and H.-B. Sun, "Data classification based on fractional order gradient descent with momentum for RBF neural network," *Network-Computation in Neural Systems*, vol. 31, no. 1–4, pp. 166–185, 2020.

[5]. S. Ray, "A quick review of machine learning algorithms," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con)*, India, Feb. 14–16, 2019.

[6]. B. S. Raghuwanshi and S. Shukla, "Class-specific cost sensitive boosting weighted extreme learning machine for class imbalance learning," Memetic Computing, vol. 11, no. 3, pp. 263–283, Jun. 2019.

[7]. B. S. Raghuwanshi and S. Shukla, "Classifying imbalanced data using ensemble-based reduced kernelized weighted extreme learning machine," International Journal of Machine Learning and Cybernetics, vol. 10, no. 11, pp. 3071–3097, Aug. 2019.

[8]. R. V. K. Reddy and U. R. Babu, "A review on classification techniques in machine learning," *International Journal of Advance Research in Science and Engineering*, vol. 7, no. 3, pp. 40–47, 2018..

[9]. H. Wang, R. Feng, Z. F. Han, and C. S. Leung, "ADMM-based algorithm for training fault-tolerant RBF networks and selecting centers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3870–3878, 2018.

[10]. Z. Li, F. Wang, T. Sun, and B. Xu, "A constrained optimization method based on BP neural network," *Neural Computing and Applications*, vol. 29, no. 2, pp. 413–421, 2018.

[11]. R. V. K. Reddy and U. R. Babu, "A review on classification techniques in machine learning," *International Journal of Advance Research in Science and Engineering*, vol. 7, no. 3, pp. 40–47, 2018.

[12]. P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning," IEEE Transactions on Cybernetics, vol. 47, no. 9, pp. 2850–2861, Sept. 2017.

[13]. G. Singh and K. Deep, "Effectiveness of new multiple-PSO-based membrane optimization algorithms on CEC 2014 benchmarks and iris classification," Memetic Computing, vol. 16, no. 3, pp. 473–496, Sep. 2017.

[14]. F. Y. Osisanwo, J. T. Akinsola, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017.

[15]. H. Wang, W. Sun, and P. X. Liu, "Adaptive intelligent control of nonaffine nonlinear time-delay systems with dynamic uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1474–1485, 2017.

[16]. H. Wen, H. Fan, W. Xie, and J. Pei, "Hybrid structure-adaptive RBF-ELM network classifier," *IEEE Access*, vol. 5, pp. 16539–16554, 2017.

[17]. L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[18]. S. N. Dhage and C. K. Raina, “A review on machine learning techniques,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 3, pp. 395–399, 2016.

[19]. X. Zhao, P. Shi, X. Zheng, and J. Zhang, “Intelligent tracking control for a class of uncertain high-order nonlinear systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1976–1982, 2016.

[20]. J. Raitoharju, S. Kiranyaz, and M. Gabbouj, “Training radial basis function neural networks for classification via class-specific clustering,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2458–2471, 2016.

[21]. H. Wen, W.-X. Xie, J.-H. Pei, and L.-X. Guan, “An incremental learning algorithm for the hybrid RBF-BP network classifier,” *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 57, 2016.

[22]. G. Vachkov, V. Stoyanov, and N. Christova, “Incremental RBF network models for nonlinear approximation and classification,” in *Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Istanbul, Turkey, Aug. 2015, pp. 1–8.

[23]. P. P. Iraola, M. Rodriguez-Cassola, F. De Zan, P. Lopez-Dekker, R. Scheiber, and A. Reigber, “Efficient evaluation of Fourier-based SAR focusing kernels,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 9, pp. 1441–1444, Sep. 2014.

[24]. R. Mohammadi, S. M. T. F. Ghomi, and F. Zeinali, “A new hybrid evolutionary based RBF networks method for forecasting time series: A case study of forecasting emergency supply demand time series,” *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 204–214, 2014.

[25]. H. Yu, P. D. Reiner, T. Xie, T. Bartczak, and B. M. Wilamowski, “An incremental design of radial basis function networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1793–1803, 2014.

[26]. N. K. Nagwani and S. Verma, “A comparative study of bug classification algorithms,” International Journal of Software Engineering and Knowledge Engineering, vol. 24, no. 1, pp. 111–138, 2014.

[27]. W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effective detection of sophisticated online banking fraud on extremely imbalanced data,” World Wide Web, vol. 16, no. 4, pp. 449–475, Jul. 2013.

[28]. S. Wang and X. Yao, “Using class imbalance learning for software defect prediction,” IEEE Transactions on Reliability, vol. 62, no. 2, pp. 434–443, June 2013.

[29]. S. Wang and X. Yao, “Using class imbalance learning for software defect prediction,” IEEE Transactions on Reliability, vol. 62, no. 2, pp. 434–443, June 2013.

[30]. C. K. L. Lekamalage, H. Zhou, G.-B. Huang, and C. M. Vong, “Representational learning with extreme learning machine,” *IEEE Intelligent Systems-Trends & Controversies*, vol. 28, no. 6, pp. 31–34, 2013.

[31]. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” Expert Systems with Applications, vol. 39, no. 3, pp. 3446–3453, 2012.

[32]. S. Wang and X. Yao, “Multiclass imbalance problems: Analysis and potential solutions,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

[33]. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 463–484, Jul. 2012.

[34]. G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[35]. D. Niros and G. E. Tsekouras, “A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach,” *Fuzzy Sets and Systems*, vol. 193, pp. 62–84, 2012.

[36]. V. A. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, “Supervised high-resolution dual polarization SAR image classification by finite mixtures and copulas,” *IEEE Journal on Signal Processing*, vol. 5, no. 3, pp. 65–1595, 2011.

[37]. M. Ben Saleh, A. Mitiche, and I. Ben Ayed, “Multiregion image segmentation by parametric kernel graph cuts,” *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 545–555, Feb. 2011.

[38]. E. Soria-Olivas *et al.*, “BELM: Bayesian extreme learning machine,” *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 505–509, 2011.

[39]. Y. Miche *et al.*, “OP-ELM: Optimally pruned extreme learning machine,” *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.

[40]. X. Y. Liu, J. Wu, and Z. H. Zhou, “Exploratory undersampling for class imbalance learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 539–550, Apr. 2009.

[41]. X. Tang and M. Han, “Partial Lanczos extreme learning machine for single-output regression problems,” *Neurocomputing*, vol. 72, no. 13–15, pp. 3066–3076, 2009.

[42]. H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, Sept. 2009.

[43]. Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[44]. G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[45]. G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[46]. G.-B. Huang *et al.*, "Can threshold networks be trained directly?" *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 3, pp. 187–191, 2006.

[47]. G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[48]. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Real-time learning capability of neural networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 863–878, 2006.

[49]. N.-Y. Liang, G.-B. Huang, S. P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[50]. K.-A. Toh, Q.-L. Tran, and D. Srinivasan, "Benchmarking a reduced multivariate polynomial pattern classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 740–755, 2004.

[51]. S. Alex and S. Bernhard, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[52]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, no. 1, pp. 321–357, Jun. 2002.

[53]. J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[54]. J. A. K. Suykens and J. Vandewalle, "Training multilayer perceptron classifier based on a modified support vector method," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 907–911, 1999.

[55]. P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[56]. T. Cover and P. Hart, “Nearest neighbor pattern classification,” IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, Jan. 1967.

BOOK

[1]. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2]. S. Marsland, *Machine Learning: An Algorithmic Perspective*. Boca Raton, FL, USA: CRC Press, 2014.

[3]. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Amsterdam, Netherlands: Morgan Kaufmann, 2012.

[4]. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[5]. S. Haykin, *Neural Networks and Learning Machines*. Upper Saddle River, NJ, USA: Prentice Hall, 2009.

[6]. C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[7]. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.

[8]. P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA, USA: Pearson, 2005.

[9]. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[10]. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley-Interscience, 2001.

Publications

LIST OF PUBLICATIONS

INTERNATIONAL JOURNALS

1. Optimization of Nonlinear Kernel-Based Classification Using Pin-Sgtsvm, Intranational Journal on Recent and Innovation Trends in Computing and Communication ,ISSN:2321-8109 Vol. I Issue:l Article Accepted:30 November 2023

2. Comparative Analysis of Machine Learning Models for Predictive Performance of Different Datasets, International Journal of Enhanced Research in Sciences Technology & EngineeringISSN:2319-7363, Vol.13 Issue IO, October 2024.

3. Comparative Analysis of Regression Models for Quantile Prediction Using Pinball Loss, International Journal of Engineering, Science, Technology and Innovation (IJESTI), ISS N:2582-9734, Vol 5, Issue 1, January2025

4. Advanced Quantile Regression with Pinball Loss: Leveraging Lagrangian Asymmetricv Twin SVR and Enhanced Model Optimization for Superior Performance, Journal of Neonatal surgery, ISSN:2226-0439, Vol 14,1ssue I Is (2025)

CONFERENCE

1. A Study On Implicit Lagrangian Twin Extreme Learning Machines In Primal For Pattern Classification, International Conference on Advances in Science, Engineering, Management AND Humanities (ICASEMH – 2023) Feb 2023

2. A Study Of Functional Iterative Approaches For Twin Bounded Support Vector Machines With Squared Pinball Loss, National Conference on Recent Advances in Science, Engineering, Humanities, and Management (NCRASETHM - 2024) -28th January, 2024, Banquet, Noida, India.

3. Kernel-Optimized Surface Learning Enhancing Classification with Support Vector Machines and Gaussian Process Kernels, 8th International Conference on Advanced Logical Learning and Analytical Mining (ALLAM-2024), 28th December 2024

4. Adaptive Kernel Optimization for Probabilistic Learning Integrating Support Vector Machines with Gaussian Process Frameworks, Intelligent Systems And Computational Networks(ICISCN2025),24-25 January 2025

Patent

1. Systems for Enhancing Machine Learning Models Using Target Distributions of Key Performance Indicators in Cloud Networks, The Patent Office Journal No. 03/2025, Publication Dated 17/01/2025

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: 25 July 2023 *Revised:* 12 September 2023 *Accepted:* 30 November 2023

Optimization of Nonlinear Kernel-Based Classification using Pin-Sgtbsvm

[1]V Rajanikanth Tatiraju

[1]Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P.

[2]Dr. Rohita Yamaganti

[2]Associate Professor, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P.

Abstract

When it comes to machine learning, the selection of kernel functions is crucial for classification model performance. Because they can simulate complicated patterns in data, nonlinear kernel-based classification algorithms have attracted a lot of interest for optimization. Using the German, Haberman, CMC, Fertility, WPBC, Ionosphere, and Live Disorders benchmark datasets as well as others from the UCI database, the paper assesses how well the Pin-SGTBSVM algorithm performs. Using a tenfold cross-validation technique, the ideal parameters are obtained using a nonlinear kernel function. Over six datasets, the findings show that Pin-SGTBSVM outperforms well-known algorithms like TWSVM, TBSVM, Pin-GTWSVM, and Pin-GTBSVM in terms of accuracy, with noticeable advances in classification performance. Although it also shows competitive results, Pin-SGTWSVM's accuracy on the German dataset is marginally worse than TWSVM's. The experimental results show that Pin-SGTBSVM is a reliable method for improving classification accuracy with no impact on computing efficiency. The results show that it might be used for data categorization and machine learning in the actual world.

Keywords: Nonlinear Kernel, Accuracy, Classification, Machine learning, Efficiency

I.INTRODUCTION

Machine learning and data-driven decision-making are dynamic fields, and classification methods are vital for mining large datasets for useful patterns. One of the most effective ways to deal with complicated decision limits that linear classifiers struggle to handle is by using nonlinear kernel-based classification algorithms. In order to translate input data into higher-dimensional feature spaces where linear separation is possible, kernel-based approaches, especially Support Vector Machines (SVM) using kernel functions, offer a way. However, these methods are only as good as the kernel functions, regularization settings, and feature transformation techniques used to select and optimize them. To improve model accuracy, generalizability, and computing efficiency, optimization in nonlinear kernel-based classification is a crucial field of research.

When data distributions display complex interactions that cannot be separated linearly, nonlinear classification difficulties emerge. Due to their linearity assumptions, traditional classifiers like logistic regression and linear discriminant analysis suffer in such cases. In contrast, classifiers that rely on kernel functions do not directly compute the transformation but instead use them to turn input data into feature spaces with greater dimensions. Classifiers can now find computationally feasible, complicated decision boundaries using this method. Some common kernel functions include the sigmoid, polynomial, and Radial Basis Function (RBF), each of which has its own set of benefits that are dependent on the nature of the input. In order to improve model flexibility and fine-tune kernel parameters, robust optimization strategies are required. The selection of kernel function has a substantial influence on classification performance.

Various methodologies are employed in nonlinear kernel-based classification optimization with the goal of enhancing the performance of classifiers. To maximize the balance between model complexity and generalizability, hyperparameter tuning is an essential part. This entails optimizing factors like the regularization coefficient (C) and kernel parameters, such as gamma in RBF kernels. Hyperparameter selection has been automated using grid search, random search, and sophisticated methods like genetic algorithms and Bayesian optimization. Feature selection and dimensionality reduction techniques are also optimized so that classification judgments are based on the most important qualities. It is common practice to use feature scaling and Principal Component Analysis (PCA) to improve

input representations and address problems such as high-dimensional sparsity and overfitting.

Computational efficiency is another important optimization metric for kernel-based categorization. The increasing complexity of kernel changes makes training nonlinear classifiers computationally costly, especially with large-scale datasets. Sequential Minimal Optimization (SMO), an effective optimization approach for support vector machine (SVM) training, is one solution that academics have devised to tackle this difficulty. By breaking down large optimization problems into more manageable pieces, these techniques drastically cut down on computing requirements. In addition, we have incorporated parallel and distributed computing frameworks to improve training speed and scalability. This includes GPU acceleration and cloud-based solutions. Random Fourier Features and Nyström approximation are two examples of recent developments in approximate kernel approaches that offer further ways to reduce processing costs without sacrificing classification accuracy.

The use of nonlinear kernel-based classification has many potential uses in many different industries, such as biology, finance, cybersecurity, and image recognition. One use of kernel-based classifiers is in medical diagnostics, where they help with illness prediction and categorization using patient data. This allows for early detection and individualized therapy recommendations. These classifiers find complex patterns in financial transactions and help with credit risk assessment and fraud detection in the financial sector. Object detection and face recognition are two examples of image recognition tasks that greatly benefit from kernel-based classifiers' capacity to distinguish intricate visual characteristics. Additionally, cybersecurity programs protect digital infrastructures from harmful attacks by detecting abnormalities and intrusions in network data using nonlinear classification algorithms. Classifier performance and adaptability should be continuously optimized due to the relevance of these applications.

II.REVIEW OF LITERATURE

Piccialli, Veronica. (2022) Many different areas have found success using support vector machines, making them a crucial class of machine learning models and techniques. To define the machine learning models and to create effective and convergent algorithms for large-scale training tasks, SVM technique relies heavily on nonlinear optimization. Here, we lay out the convex programming issues that underpin support vector machines (SVMs), with an emphasis on supervised binary classification. Here, we take a look at the most popular optimization techniques for support vector machine (SVM) training issues and talk about how to incorporate their characteristics into algorithm design.

Li, Kai & Lv, Zhen. (2021) To improve the support vector machine's classification performance, the twin support vector machine resolves two small quadratic programming problems. The following problems, however, afflict this method: (1) The twin support vector machine and other variants rely on a hinge loss function to construct their models, however this function is noisy and unstable during resampling. (2) Getting the models to work in the dual space takes a lot of time and effort. To make the twin bounded support vector machine even more effective, the pinball loss function is included into it. To solve the problem of the pinball loss function not being differentiable at zero, a smooth approximation function is built. One may use this to build a smooth twin-bounded support vector machine model that includes pinball loss. Iteratively, the issue is solved in the original space using the Newton-Armijo approach. Theoretically, we show that an iterative method for smooth twin bounded support vector machines with pinball loss converges. The trials verify the suggested approach on both real and synthetic datasets, including those from UCI. In addition, the suggested algorithm's efficacy is shown by comparing its performance to that of other representative algorithms.

Yao, Yukai et al., (2015) We present PMSVM, an enhanced Support Vector Machine classifier that takes into account extensively System Normalization, PCA, and Multilevel Grid Search techniques for data pretreatment and parameters optimization, respectively. Improving SVM's classification efficiency and accuracy are the primary objectives of this project. To evaluate PMSVM's efficacy, metrics such as ROC curve, sensitivity, specificity, and precision are utilized. When compared to more conventional SVM algorithms, experimental findings reveal that PMSVM significantly outperforms them in terms of efficiency and accuracy.

Cocianu, Catalina. (2013) This research details a model-free method for developing SVM-type non-linear classifiers. In common parlance, support vector machines are "non-parametric" models. However, this does not mean that SVMs do not have parameters; rather, the learning issue around the parameters of an SVM is of paramount significance. A revised version of the gradient ascent method for addressing the SVM - QP issue and a new formulation for the bias parameter provide the innovative aspects of the suggested approach. In addition to demonstrating greater convergence rates than the basic SMO method, the tests also highlighted good convergence qualities of the suggested modified variations. In comparison to the default bias setting, the produced classifier also performs better.

Biswas, Debojit et al., (2011) For a wide variety of reasons, land cover data is crucial. Precise data on a region's land cover

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: 25 July 2023 *Revised:* 12 September 2023 *Accepted:* 30 November 2023

is essential for many initiatives that aim to manage, plan, and monitor natural resources. An image classification technique is used to recover land cover information from remotely sensed photos, which are interesting sources for this purpose. When training data is scarce and classes have few pixels, statistical classifiers often fail. Support vector machines (SVMs) and other next generation algorithms have been producing respectable results with a less quantity of training data sets. Training is an ongoing expense. As a result, there is a great deal of interest in developing classifiers that require a smaller set of training data. We examine support vector machine (SVM) based hyperspectral image classification using several kernel types, including linear, polynomial, radial basis, and sigmoid, in this research. We test SVM's performance with various kernels and compare the results. Additionally, this section delves into the mathematical foundations of non-linear SVM. For the purpose of feature reduction, this study used principal component analysis (PCA). Since the penalty amount has less of an impact on linear and polynomial kernels in SVM, our results demonstrate that these kernel types achieve better classification accuracy. A narrow range of penalty levels and hyperparameters is required for other kernels.

III.MATERIAL AND METHODS

Using the UCI database, we conduct trials on the following categories: German, Haberman, CMC, Fertility, WPBC, Ionosphere, and Live-disorders to validate the performance of the Pin-SGTBSVM. In this experiment, the kernel function is $K(x,y) = exp(-\theta \times \|x-y\|^2)$, where θ is a parameter, and the best parameter value within the range is determined using a tenfold cross-validation procedure $[10^{-6},10_5]$. The values of τ1 and τ2 are 0.5, 0.8, 1, and ci> 0 (i = 1, 2,3,4) and the value range is $[2^{-10}, 2^{10}]$, the value of ε in the experiments is 10_{-6}, the value of η in the algorithm is 10^{-4}, and the standard deviation and average accuracy are included in the experimental findings. An Intel (R) Core (TM) i7-5500U CPU running at 2.40GHz with 4GB of RAM and MATLAB R2016a were utilized for all the experiments presented in this paper.

We compare the performance of many representative methods, including TWSVM, TBSVM, Pin-GTWSVM (TBSVM with pinball loss in dual space), and Pin-GTBSVM (TBSVM with pinball loss in original space). The iterative approach is also used to solve these algorithms.

IV.RESULTS AND DISCUSSION

Results from the experiments are displayed in table 1. By comparing the six datasets, it is clear that the Pin-SGTBSVM algorithm outperforms the other five approaches. However, when it comes to the Fertility dataset, the Pin- SGTBSVM methodology achieves identical results to the other five.

Additionally, whereas the TWSVM method has superior accuracy on the German dataset, the Pin-SGTWSVM approach achieves greater accuracy on all six datasets.

Table 1 Comparison of Algorithm Performance Using Nonlinear Kernels

Datasets	TWSVM	TBSVM	Pin-GTWSVM	Pin-GTBSVM	Pin-SGTWSVM	Pin-SGTBSVM
	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)
	±sd	±sd	±sd	±sd	±sd	±sd
	Time(s)	Time(s)	Time(s)	Time(s)	Time(s)	Time(s)
German	74.9170	70.0999	70.0999	70.0996	71.3955	75.1497
	±0.0189	±0.0002	±0.0002	±0.0002	±0.0262	±0.018
	0.0875	0.0563	0.0865	0.0742	0.0304	0.0375
Haberman	73.3698	73.3694	73.3697	73.3697	73.7498	73.7497
	±0.0001	±0.0001	±0.0001	±0.0001	±0.0062	±0.0062
	0.0294	0.0326	0.0280	0.0309	0.0240	0.0184
CMC	69.2387	65.396	65.3980	65.3980	71.662	73.7026
	±0.0398	±0.0000	±0.0000	±0.0000	±0.0359	±0.0227
	0.0695	0.0570	0.0635	0.0497	0.0173	0.0405
Fertility	87.0971	87.0967	87.0966	87.0966	87.0970	87.0971
	±0.0002	±0.0002	±0.0002	±0.0000	±0.0000	±0.0002
	0.0144	0.0121	0.0141	0.0147	0.0151	0.0108
WPBC	76.2713	76.2715	76.2713	76.273	78.8134	79.1526
	±0.0000	±0.0001	±0.0000	±0.0000	±0.0349	±0.0429
	0.0148	0.0197	0.0295	0.0294	0.0197	0.0211

Ionosphere	91.6980	91.6980	93.5850	92.7360	93.1130	94.5285
	±0.0252	±0.0130	±0.0328	±0.0298	±0.0215	±0.0142
	0.0252	0.0247	0.0349	0.0371	0.0174	0.0263
Live disorders	64.4229	62.6925	57.6926	57.6925	66.8272	67.1156
	±0.0495	±0.0460	±0.0000	±0.0001	±0.0480	±0.0577
	0.0638	0.0593	0.0507	0.0598	0.0161	0.0172

V.CONCLUSION

The accuracy of the Pin-SGTBSVM algorithm was found to be higher than that of other approaches, including TWSVM, TBSVM, Pin-GTWSVM, and Pin-GTBSVM, when tested on many datasets from the UCI repository. Across all datasets, but notably German, CMC, WPBC, and Ionosphere, the algorithm demonstrated superior performance. It is worth mentioning that Pin-SGTBSVM performed similarly to other methods on the Fertility dataset. Additionally, Pin-SGTWSVM accomplished respectable accuracy; the only dataset where it was somewhat less accurate than TWSVM was the German one. In particular, the experimental findings show that Pin-SGTBSVM performs well over a wide range of real-world datasets, with respect to accuracy, standard deviation, and computing efficiency. Based on these results, the Pin-SGTBSVM method seems like it may be a great tool for accuracy and computing efficiency classification problems in many different disciplines.

REFERENCES: -

[1] V. Piccialli, "Nonlinear optimization and support vector machines," *Annals of Operations Research*, vol. 314, no. 1, pp. 111–149, 2022.

[2] K. Li and Z. Lv, "Smooth twin bounded support vector machine with pinball loss," *Applied Intelligence*, vol. 51, no. 1, pp. 1–17, 2021, doi: 10.1007/s10489-020-02085-5.

[3] F. Liu, X. Huang, Y. Chen, et al., "Random features for kernel approximation: A survey in algorithms, theory, and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7128–7148, 2021.

[4] B. Jiang, T. Lin, S. Ma, et al., "Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis," *Computational Optimization and Applications*, vol. 72, no. 1, pp. 115–157, 2019.

[5] Y. L. Feng, Y. N. Yang, X. L. Huang, S. Mehrkanoon, and J. A. K. Suykens, "Robust support vector machines for classification with nonconvex and smooth losses," *Neural Computation*, vol. 28, no. 6, pp. 1217–1247, 2016.

[6] Y. Yao, H. Cui, Y. Liu, L. Li, L. Zhang, and X. Chen, "PMSVM: An optimized support vector machine classification algorithm based on PCA and multilevel grid search methods," *Mathematical Problems in Engineering*, vol. 2015, no. 1, pp. 1–15, 2015, doi: 10.1155/2015/320186.

[7] C. Cocianu, "Kernel-based methods for learning non-linear SVM," *Economic Computation and Economic Cybernetics Studies and Research / Academy of Economic Studies*, vol. 47, no. 1, pp. 1–15, 2013.

[8] D. Biswas, H. Jain, M. Arora, and B. Balasubramanian, "Study and implementation of a non-linear support vector machine classifier," *International Journal of Earth Sciences and Engineering*, vol. 4, no. 0974-5904, pp. 338–341, 2011.

[9] K. Huang, R. Zheng, R. Sun, R. Hotta, and R. Fujimoto, "Sparse learning for support vector classification," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1944–1951, 2010.

[10] T. Hofmann, B. Scholkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[11] C. Orsenigo and C. Vercellis, "Multivariate classification trees based on minimum features discrete support vector machines," *IMA Journal of Management Mathematics*, vol. 14, no. 2, pp. 221–234, 2003.

[12] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1-3, 2002.

International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 10, October-2024, Impact Factor: 8.375

Comparative Analysis of Machine Learning Models for Predictive Performance of Different Datasets

V Rajanikanth Tatiraju[1], Dr. Rohita Yamaganti[2]

[1]Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P
[2]Associate Professor, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P

ABSTRACT

Using three different datasets—synthetic noisy data, biomedical disease prediction, and financial credit risk—this study compares and assesses the performance of various machine learning models, such as Adaptive Linear v-Support Vector Regression, Support Vector Machine, Logistic Regression, Random Forest, and v-Support Vector Machine. Testing how well these models handled various kinds of data and made accurate predictions was the goal. Several measures were used to quantify performance, including as recall, accuracy, precision, F1-score, and Area under the Curve. The results showed that AL-vTSVR was the most effective model in every performance parameter tested, showing that it could handle complicated real-world data and noise with ease. Random Forest shown competitive performance as well, particularly in financial and medicinal domains. In contrast to SVM and v-TSVM, Logistic Regression showed less effectiveness. The results demonstrate that AL-vTSVR is an exceptionally dependable model for difficult data situations, and they emphasize its better capabilities in various prediction tasks.

Keywords: Noisy, Support Vector Machine, Accuracy, Precision, Recall

INTRODUCTION

Machine learning (ML) has changed several industries by letting computers discover patterns in data and use that knowledge to make judgments or predictions without human intervention. When it comes to predicting future events or outcomes using historical or real-time data, machine learning models are useful tools in the context of predictive performance. With the use of big datasets and advanced algorithms, these models are able to uncover patterns and make predictions, the accuracy of which might vary. Machine learning has become an essential tool for predicting tasks because to the growing amount, diversity, and speed of data in many fields, including healthcare, finance, marketing, and engineering.

Machine learning essentially entails creating algorithms that can autonomously learn from data and improve upon past performance. In machine learning, predictive performance is a model's capacity to generate correct predictions when presented with novel, unseen data. Machine learning models can process massive, unstructured information and reveal complex correlations between variables that would otherwise go unnoticed, in contrast to traditional statistical approaches that depend significantly on established assumptions. These models' predictive capability shines through when they leverage historical trends to assist decision-making; this makes them applicable to tasks like demand forecasting, stock market prediction, predictive maintenance, and disease outbreak forecasting, among others.

Machine learning models come in a variety of flavors, each optimized for a particular kind of prediction job. Predictive analytics makes extensive use of supervised learning models. To train these models, we use labelled data, in which each input attribute has an associated label. Constructing a model capable of making predictions based on novel, unknown input data is the main objective of supervised learning. Ensemble techniques such as random forests and gradient boosting are common examples of supervised learning algorithms, along with linear regression, decision trees, and support vector machines (SVMs). These models are great at many different kinds of prediction performance challenges because they are so good at classification and regression.

In contrast, unsupervised learning models are employed in situations when the data does not contain labelled outcomes. To the contrary, these models unearth previously unseen patterns and structures in the data. Unsupervised learning frequently makes use of clustering and dimensionality reduction methods like k-means, hierarchical clustering, and principal component analysis (PCA). Unsupervised learning models aren't meant to make predictions per se, but they can be useful for pre-processing data by highlighting clusters or important traits that supervised learning models can exploit to their advantage.

Another subfield of machine learning, reinforcement learning (RL) is concerned with decision-making in settings where the model acquires knowledge by interactions and feedback. To find the best solution, an agent in RL acts in its environment and, depending on the results, gets rewards or penalties. RL shines in robotics, games, and autonomous systems, among other areas, when forecasts must take sequential decision-making into consideration. Games, robot control, and resource optimization are just a few of the areas where deep reinforcement learning—a combination of deep learning and RL—has achieved remarkable progress.

The creation of deep learning models represents a turning point in machine learning as it pertains to prediction performance. "Deep learning" refers to a subfield of machine learning in which multi-layered neural networks automatically learn hierarchical data representations. Several applications, including time series forecasting, picture recognition, and natural language processing, have demonstrated exceptional performance from these models. One common deep learning architecture that has seen extensive use in prediction tasks is the convolutional neural network (CNN). Another is the recurrent neural network (RNN). When it comes to image-based tasks, CNNs really shine. On the other hand, RNNs, especially LSTM and GRU, really shine when it comes to sequential prediction tasks, like predicting time-series data or interpreting natural language.

Data quality, algorithm selection, and hyperparameter tuning are three of the most important determinants of a machine learning model's predictive performance. Before a machine learning model can learn any useful patterns from data, the data must undergo data preparation. It is usual practice to enhance the data quality before to training a model using techniques like normalization, feature selection, and imputation of missing values. Another important aspect of evaluating machine learning models for predicting performance is model assessment. Area under the receiver operating characteristic curve (AUC-ROC), F1 score, recall, accuracy, precision, and area under the receiver operating characteristic curve (ACCURATE) are common metrics for classification tasks, whereas R-squared, MSE, and RMSE are used for regression activities.

REVIEW OF LITERATURE

Petchiappan, Maheswari & Jaya, A. (2022) Investors have always found trend prediction in the stock market to be a difficult and perplexing task. Technological developments, machine learning, data analytics, and big data have led to a meteoric rise in the accuracy of stock market predictions. Among the many varied industries represented on the stock market is the media and entertainment industry. The Sensex and the Nifty are the two indices used in the Indian stock market. Theatres were closed in 2019 because of the pandemic. This caused a halt in production and prevented distributors and directors from releasing their films to screens. So, during the lockdown, many stayed indoors and watched more television. Resulting in a higher degree of media consumption. The study's overarching goal is to use machine learning to foretell how the stock prices of the media and entertainment firm will do. Making as much money as possible while keeping losses to a minimum can help investors. In data science, the suggested stock prediction method is utilized for predicting stock prices and determining the accuracy of logistic and linear regression in machine learning algorithms. The media and entertainment industry's stock price data is used in the tests, which employ machine learning techniques. One example of an input dataset is media stock prices. Various aspects of stock prices with a daily frequency were used to create the model. In summary Media and entertainment stock prices are so anticipated using logistic and linear regression models. In order to help investors maximize their gains and minimize their losses, the stock prices are anticipated with a high degree of accuracy using the aforementioned methodologies.

Sekeroglu, Boran et al., (2022) The use of AI and ML to solve issues or augment human specialists is crucial in nearly every aspect of human existence. Researchers still face the formidable challenge of narrowing down the many real-world application areas to a single machine learning model that may produce superior results for a given problem. Several aspects, including the features of the dataset, the training approach, and the model's responses, might influence the model's performance. Hence, in order to ascertain the efficacy of the proposed tactics and the capability of the model, a thorough evaluation is necessary. Ten standard machine learning models were applied to seventeen different datasets in this research. Training procedures of60:40,70:30, and 80:20 hold-out, in addition to five-fold cross-validation, are used in the experiments. The experimental findings were assessed using three metrics: R2 score, mean absolute error, and mean

squared error. The models that were taken into consideration are examined, and the benefits, drawbacks, and data dependencies of each model are highlighted. The deep Long-Short Term Memory (LSTM) neural network achieved the best results compared to the other models tested (decision tree, linear regression, support vector regression with radial and linear basis function kernels, random forest, gradient boosting, extreme gradient boosting, shallow neural network, and deep neural network), all of which were determined by conducting an excessive number of experiments. When evaluating models in regression research without data mining or selection, cross-validation should be examined due to the substantial influence it has on experimental outcomes.

Varshini, Priya et al., (2021) To construct smart systems capable of problem-solving, Artificial Intelligence builds on top of Machine Learning and Deep Learning. The amount of time needed to do the task may be estimated using software effort estimation. Predicting Software Effort at the beginning phases of a project is fraught with difficulty and difficulty owing to several unknowns. You may use software effort estimation to better organize your project's timeline, resources, and budget. Expert judgment, regression estimations, categorization techniques, deep learning algorithms, and analogy-based estimations were some of the studies suggested for effort prediction. Based on its resilience and ability to manage big datasets, random forest surpasses other algorithms in this paper's comparison of deepnet, neuralnet, support vector machine, and random forest. Mean Absolute Error, Root Mean Squared Error, Mean Squared Error, and R-Squared are the evaluation metrics that should be considered.

Yuan, Kunpeng et al., (2021) Establishing a prediction model, default prediction determines the likelihood of a business defaulting. Data from features at time t-m and default state at time t are shown to have a functional link. A non-defaulting firm's forecast might lead to a loss of high-quality consumers, while an inaccurate forecast of a defaulting company could trick banks into lending to a "defaulter," resulting in massive losses. Using k-means clustering to divide the sample and support vector domain description (SVDD) to forecast default (credit scoring), this study suggests a two-stage default prediction model to aid lending choices made by banks and non-banking financial organizations.

To train the proposed model to warn of default m years ahead, it takes characteristics' data at time t-m (m = 1, 2, 3, 4, 5), together with the default state at t. Compared to single-stage models that rely solely on k-means clustering or support vector domain description, the findings demonstrate that the suggested two-stage default prediction model outperforms them. What's more, the proposed model was able to attain a five-year default prediction ability (AUC > 0.85). In addition, the study suggests that three important factors in default forecasting for Chinese listed businesses are "retained earnings/total assets," "financial expenses/gross revenue," and "type of audit opinion." By showing that it is worthwhile to explore combining alternative techniques to enhance the effectiveness of default prediction models, this work adds to the field of multi-stage credit scoring research.

Mounika, B. & Persis, Voola. (2019) Machine learning techniques are widely used in many different industries. In the classroom, for example, these methods have many potential uses. Machine learning approaches are being used in an increasing amount of educational research. Using machine learning techniques in a classroom setting can help unearth previously unknown information and trends regarding student achievement. Using machine learning classification methods such as K-Nearest Neighbor, Decision Tree, Support Vector Machines, Random Forest, and Gradient Descent Boost Algorithms, this effort intends to construct a model that predicts students' academic success across different departments. Factors such as residence, parent-child relationship, level of education and occupation, backlogs, attendance, availability of internet connection, and smartphone use are taken into account.

You may find out how well a student did on the final test and what their grade will be using the resultant prediction model. College administration or instructors can then use this information to identify which students need extra help and intervene before it's too late. With the help of early prediction, we may find ways to improve our performance in the final exams.

EXPERIMENTAL ANALYSIS

This study compared the efficacy of several ML models trained on synthetic noisy data, biological illness prediction, and financial credit risk datasets, each representing a distinct area. Adaptive Linear ν-Support Vector Regression (AL-νTSVR), Support Vector Machine (SVM), Logistic Regression, Random Forest, and ν-Support Vector Machine (ν-TSVM) are some of the models that are utilized for comparison. Area Under the Curve (AUC), Accuracy, Precision, Recall, and F1-Score are the primary performance indicators used to evaluate the efficacy of the model.

RESULTS AND DISCUSSION

Table 1: Performance Metrics on Synthetic Noisy Data

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
AL-vTSVR	92.5	90.0	94.0	91.8	95.2
SVM	86.0	85.5	88.3	86.9	91.4
Logistic Regression	81.5	79.2	84.0	81.6	87.7
Random Forest	88.4	87.2	89.6	88.3	93.5
v-TSVM	89.0	87.0	90.2	88.5	92.1

The AL-vTSVR model outperforms all other models on the synthetic noisy dataset, achieving the highest accuracy (92.5%), precision (90.0%), recall (94.0%), F1-Score (91.8%), and AUC (95.2%). Random Forest follows closely with strong results (accuracy: 88.4%, AUC: 93.5%) but does not match the AL-vTSVR. The v-TSVM model shows solid performance (accuracy: 89.0%, AUC: 92.1%), while SVM and Logistic Regression perform relatively worse, with Logistic Regression being the least effective across all metrics. In summary, AL-vTSVR is the best performer, especially for noisy data.

Table 2: Performance Metrics on Biomedical Disease Prediction

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
AL-vTSVR	95.3	92.4	96.1	94.2	97.8
SVM	90.1	89.7	92.5	91.1	94.8
Logistic Regression	84.7	80.2	88.3	84.1	89.9
Random Forest	92.8	90.3	94.7	92.5	96.0
v-TSVM	91.6	88.5	92.8	90.6	95.1

With a 97.8% AUC, 94.2% F1-Score, 96.1% recall, and 95.3% accuracy, the AL-vTSVR model outperforms all other models in biological illness prediction. In terms of illness identification accuracy, it much surpasses all other models. The v-TSVM model demonstrates good performance with an accuracy of 91.6% and an area under the curve (AUC) of 95.1%, while Random Forest follows with robust findings (accuracy: 92.8%, AUC: 96.0%). While Logistic Regression has the lowest overall metrics, SVM and Logistic Regression both perform well, but they aren't as effective as the top models. AL-vTSVR stands out in every performance metric.

Table 3: Performance Metrics on Dataset 3 (Financial Credit Risk)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
AL-vTSVR	93.7	91.2	95.4	93.3	96.5
SVM	89.5	88.0	91.4	89.7	93.2
Logistic Regression	82.3	79.5	84.2	81.8	88.1
Random Forest	90.6	89.2	92.0	90.6	94.3
v-TSVM	92.0	89.8	92.5	91.1	94.7

On the financial credit risk dataset, the AL-vTSVR model outperforms the others with a 93.7% accuracy rate, 91.2% precision rate, 95.4% recall rate, 93.3% F1-Score, and 96.5% area under the curve. In terms of prediction abilities, it is superior to all other models. While v-TSVM demonstrates outstanding performance with an accuracy of 92.0% and an AUC of 94.7%, Random Forest follows with strong findings (accuracy: 90.6%, AUC: 94.3%). Even if it's not the worst model, SVM's performance isn't up to par, and Logistic Regression fares the worst on every criterion. When it comes to predicting credit risk, AL-vTSVR is the best model.

CONCLUSION

Results from this study show that different machine learning models perform well on prediction tasks in many areas, such as financial credit risk, biological illness prediction, and synthetic noisy data. Proof of AL-vTSVR's resilience in dealing with complicated and noisy datasets is its constant outperformance of rival models in several metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC). Also, Random Forest proved to be a formidable contender for a variety of prediction jobs, especially in the biological and financial domains. Although AL-vTSVR consistently outperformed SVM and v-TSVM, the latter two exhibited encouraging results. When it came to more complicated datasets,

Logistic Regression fell short, even if it worked well for smaller cases. In conclusion, AL-vTSVR is the best option for practical applications with complicated or noisy data because of its exceptional prediction skills in a variety of difficult domains.

REFERENCES

[1] P. Maheswari and A. Jaya, "Comparative study of machine learning algorithms towards predictive analytics," *Recent Advances in Computer Science and Communications*, vol. 16, no. 6, pp. 1-12, 2022.
[2] B. Sekeroglu, Y. K. Ever, K. Dimililer, and F. Al-Turjman, "Comparative evaluation and comprehensive analysis of machine learning models for regression problems," *Data Intelligence*, vol. 4, no. 3, pp. 620-652, 2022.
[3] S. Shi, R. Tse, W. Luo, S. D'Addona, and G. Pau, "Machine learning-driven credit risk: a systemic review," *Neural Computing and Applications*, vol. 34, no. 2, pp. 14327-14339, 2022.
[4] P. Varshini, A. K. Kumari, D. Janani, and S. Soundariya, "Comparative analysis of machine learning and deep learning algorithms for software effort estimation," *Journal of Physics: Conference Series*, vol. 1767, no. 1, pp. 1-11, 2021.
[5] K. Yuan, G. Chi, Y. Zhou, and H. Yin, "A novel two-stage hybrid default prediction model with k-means clustering and support vector domain description," *Research in International Business and Finance*, vol. 59, pp. 1-12, 2021.
[6] K. Peng and G. Yan, "A survey on deep learning for financial risk prediction," *Quantitative Finance and Economics*, vol. 5, no. 4, pp. 716-737, 2021.
[7] R. Chen, M. Lu, T. Chen, D. Williamson, and F. Mahmood, "Synthetic data in machine learning for medicine and healthcare," *Nature Biomedical Engineering*, vol. 5, no. 6, pp. 1-5, 2021. doi: 10.1038/s41551-021-00751-8.
[8] C. Caiafa, Z. Sun, T. Tanaka, P. Marti-Puig, and J. Solé-Casals, "Machine learning methods with noisy, incomplete or small datasets," *Applied Sciences*, vol. 11, no. 9, pp. 1-4, 2021.
[9] Z. Hassani, M. A. Meybodi, and V. Hajihashemi, "Credit risk assessment using learning algorithms for feature selection," *Fuzzy Information and Engineering*, vol. 12, no. 6, pp. 1-16, 2021. doi: 10.1080/16168658.2021.1925021.
[10] B. Mounika and V. Persis, "A comparative study of machine learning algorithms for student academic performance," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 4, pp. 721-725, 2019. doi: 10.26438/ijcse/v7i4.721725.
[11] S. Uddin, A. Khan, M. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Medical Informatics and Decision Making*, vol. 19, no. 6, pp. 1-16, 2019.
[12] P. Danėnas and G. Garšva, "Selection of support vector machines based classifiers for credit risk domain," *Expert Systems with Applications*, vol. 42, no. 6, pp. 1-20, 2015.

ER Publications

ISSN: 2319-7463, New Delhi, India
International Journal of Enhanced Research in Science, Technology & Engineering
UGC Certified International Peer-Reviewed & Refereed Journal
UGC Journal no. 7618

Certificate of Publication

V Rajanikanth Tatiraju

Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P

TITLE OF PAPER

Comparative Analysis of Machine Learning Models for Predictive Performance of Different Datasets

has been published in

IJERSTE, Impact Factor: 8.375, Volume 13, Issue 10, Oct. 2024

Paper Id: STE/2710202433
Date: 29-10-2024

Website: www.erpublications.com
Email: erpublications@gmail.com

Authorized Signatory

Vol 5, Issue 1, January 2025 www.ijesti.com E-ISSN: 2582-9734
International Journal of Engineering, Science, Technology and Innovation (IJESTI)

Comparative Analysis of Regression Models for Quantile Prediction Using Pinball Loss

V Rajanikanth Tatiraju [1]

[1] Research Scholar, Department of Computer Science and Engineering,
P. K. University, Shivpuri, M.P.

Dr. Rohita Yamaganti [2]

[2] Associate Professor, Department of Computer Science and Engineering,
P. K. University, Shivpuri, M.P.

ABSTRACT

The performance of three regression models, namely Lagrangian Asymmetric-vTwin Support Vector Regression (SVR), Standard SVR, and Linear Regression, is examined and compared in this study. The models are tested using various quantiles of Pinball Loss, α = 0.1, 0.5, and 0.9, in addition to more conventional metrics such as RMSE and MAE. Pinball Loss values specific to each quantile were used to evaluate the models' performance after training and testing on a regression dataset to forecast the lower, median, and higher quantiles. The outcomes show that Lagrangian Asymmetric-vTwin SVR is the best option, providing the lowest Pinball Loss, RMSE, and MAE, compared to Standard SVR and Linear Regression. Additionally, it was discovered that the ideal C value, which is 1.0, successfully balanced training duration and prediction accuracy.

Key Words: *Pinball, Quantile, Lagrangian, Asymmetric, Performance.*

I. INTRODUCTION

The establishment of links between dependent and independent variables is a crucial step in predictive modeling, and regression analysis plays a key role in this process. Predicting the dependent variable's mean from the independent variables is the main emphasis of most regression models in the past. This method is called conditional mean estimation. In many real-world situations, though, this assumption might not be enough; for example, if the data shows strong tails or skewness, or if you need more specifics regarding the distribution of the target variable for your decision-making. In response to these issues, quantile regression has developed into a strong substitute that enables the prediction of different quantiles of the response variable's conditional distribution. By estimating the mean and other features of the distribution, such the behavior of the tails, this gives a more complete picture of the data.

The method of estimating the conditional quantiles of a response variable in relation to predictor factors is known as quantile regression. It was initially proposed by Koenker and Bassett in 1978. The goal of quantile regression is to minimize a weighted sum of absolute residuals, where the weights are determined by the quantile of interest, as opposed to ordinary least squares (OLS), which minimizes the sum of squared residuals. For data with an asymmetric distribution or predictors with varying impacts across quantiles, this method shines. In economic data, for instance, quantile regression is useful for evaluating the heterogeneous impacts of variables since the link between income and education may differ for low-income and high-income individuals.

In contexts where different quantiles (e.g., the 90th or 10th percentile) may hold different significance, such as risk management, medical studies, and climate modeling, quantile regression's capacity to offer a more comprehensive characterization of the dependent variable's conditional distribution becomes extremely important. The asymmetry of the quantiles must be taken into consideration by the loss function for quantile regression to be effective, though. Here we see the application of pinball loss, a loss function that quantile regression models have come to embrace.

In quantile regression, the pinball loss—sometimes called the tilted absolute loss—is the best fit because it penalizes overestimations and underestimations in an asymmetrical fashion. In particular, it enables the model to highlight prediction mistakes in a different way whether the quantile is higher or lower than the actual number. Pinball loss's computational speed and flexible handling of various data types, including those with non-normal distributions or heterogeneity in the errors, have led to its increasing adoption. Quantile regression is now more approachable for issues of a large scale as the loss function is a part of many machine learning techniques. It is compatible with regularization methods like L1 and L2, which help prevent overfitting and guarantee robust model predictions, and it may be used in conjunction with optimization approaches like gradient descent.

Pinball loss is popular in regression models that rely on deep learning in part because of how versatile it is. Combining neural networks with pinball loss allows for quantile regression in many different contexts, because to neural networks' ability to capture complicated correlations between variables. Fields like healthcare, where forecasting the upper quantile of a variable like patient recovery time can have critical implications for resource allocation, and finance, where models need to predict the tail risks (such as extreme market crashes or booms) have both benefited greatly from this approach.

In time series forecasting, quantile regression with pinball loss has been used to predict quantiles of future values, which is useful because the data is frequently non-stationary and autocorrelated. Important for making decisions when faced with uncertainty, this enables the modeling of prediction uncertainty. In energy consumption forecasting, for example, it may be more useful to anticipate the 95th percentile of future demand than the mean, as this information is useful for choices about infrastructure capacity and load balancing.

When dealing with heteroscedasticity, a key component of quantile regression models based on pinball loss is ensuring that the variance of the error components remains consistent across data. If this is the case, it's possible that the uncertainty in the predictions won't be reliably estimated using conventional regression methods like ordinary least squares (OLS). Pinball loss regression, on the other hand, can improve the model's performance in cases of uneven variability by concentrating on quantiles, giving more robust and accurate predictions throughout the distribution.

II. REVIEW OF LITERATURE

Sigauke, Caston et al., (2018) Using additive quantile regression (AQR) models, this paper discusses short-term hourly load forecasting in South Africa. By using this method, the combined modeling of hourly power data is easily interpretable and takes residual autocorrelation into consideration. The use of generalised additive models (GAMs) allows for a comparative examination. Hierarchical interactions are used in both modeling frameworks to choose variables using the least absolute shrinkage and selection operator (Lasso). Each of the four models—GAMs with interactions and AQR models without—are carefully examined. The most accurate model that suited the data best was the AQR model that included pairwise interactions. Quantile regression averaging (QRA) and an algorithm based on the pinball loss

(convex combination model) were used to integrate the forecasts from the four models. After comparing the AQR model with interactions to the convex combination and QRA models, it was found that the QRA model produced the best accurate forecasts. Both the convex combination model and the QRA model, with the exception of the AQR model with interactions, provided appropriate prediction interval coverage probabilities for the 90%, 95%, and 99% intervals. In terms of average width and average deviation normalized by prediction interval, the QRA model was the most compact. Going beyond summary performance statistics in forecasting has benefit, as it offers additional insight into the built forecasting models. This can be seen in the modeling framework mentioned in this study.

Yu, Lean et al., (2018) The development of new quantile estimators and a loss function that takes into account the noise in both the response and explanatory variables allows for reliable quantile estimations to be achieved, even in the presence of noisy data. This is especially true when orthogonal loss is substituted for vertical loss in conventional quantile estimators, resulting in an improvement over pinball loss called orthogonal pinball loss (OPL). In this way, new OPL-based QR and SVMQR models may be developed from existing linear and support vector machine quantile regression programs, respectively. In terms of quantile property and prediction accuracy, the empirical analysis on 10 publicly accessible datasets statistically confirms that the two OPL-based models outperform their respective original forms, particularly for extreme quantiles. An innovative OPL-based SVMQR model that incorporates AI achieves better results than any benchmark model; this makes it a potentially useful quantile estimator, particularly when dealing with noisy data.

Hu, Ting et al., (2012) A kernel-based online learning technique linked to a series of insensitive pinball loss functions is being considered for use in quantile regression and support vector regression. The quantile parameter $\tau\tau$ has the potential to affect the statistical performance of the learning algorithm, as demonstrated quantitatively by our error analysis and derived learning rates. We successfully navigated the technical challenge posed by the sparsity-motivated introduction of a variable insensitive parameter in our analysis.

Steinwart, Ingo & Christmann, Andreas. (2011) A popular method in machine learning and statistics, the so-called pinball loss estimates conditional quantiles. The effectiveness of this tool for nonparametric techniques, however, has received surprisingly little attention thus far. To address this, we prove certain inequality that characterize the proximity of the approximate pinball risk minimizers to the relevant conditional quantile. These disparities, which persist under modest assumptions on the distribution of the data, are then utilized to construct so-called variance limits, which have lately emerged as crucial tools in the statistical evaluation of (regularized) empirical methods for minimizing risk. Lastly, we prove an oracle inequality for SVMs using the pinball loss by combining the two kinds of inequalities. With respect to the conditional quantile, the ensuing learning rates are min-max optimum under certain conventional regularity assumptions.

Zheng, Songfeng. (2011) It is common for optimization algorithms that rely on gradients to rapidly converge to a local maximum. Unfortunately, the quantile regression model's use of a check loss function that isn't always differentiable rules out the use of gradient based optimization techniques. Therefore, in order to fit the quantile regression model using gradient based optimization methods, this study presents a smooth function to approximate the check loss function. We go over the features of the smooth approximation. The objective function that has been smoothed can be minimized using two different approaches. Two methods have been developed for smooth quantile regression: one uses gradient descent directly, which produces the gradient descent smooth quantile regression model; the other uses functional

gradient descent to minimize the smoothed objective function; and finally, boosted smooth quantile regression algorithm is the result of changing the fitted model along the negative gradient direction in each iteration. The suggested smooth quantile regression algorithms outperform other quantile regression models in terms of prediction accuracy and efficiency in eliminating noninformative variables, according to extensive tests conducted on both real-world and simulated data.

Somers, Mark & Whittaker, Joe. (2007) Two examples of retail credit risk assessment using quantile regression show how the method can handle the wide range of distributions seen in the banking sector. One use case is in the prediction of loss due to default for secured loans, namely residential mortgages. Banks do not keep the profit when the value of the security (such a property) exceeds the loan balance; conversely, they incur a loss when the value of the security falls short of the defaulting debt. This creates an asymmetric process. Because of this imbalance, it's clear that evaluating the house's low end value—where losses are most likely to occur—is far more useful for this purpose than calculating the average value, which seldom experiences losses. In our application, we estimate the distribution of property values realized upon repossession using quantile regression. This distribution is then utilized to quantify loss given default estimations. A mortgage lender in Europe provides an example of their portfolio. Another area where it finds use is in revenue modeling. Credit granting organizations have access to massive information, but they also create models to predict how new tactics will play out, even while there is inherently no evidence available for such techniques. In certain markets, the goal of implementing a strategy is to either increase revenue or decrease risk. To better understand which accounts are most and least lucrative based on their anticipated variables, we use quantile regression in a basic artificial revenue model. Kernel smoothed quantile regression and conventional linear regression are employed in the application.

III. EXPERIMENTAL SETUP

In this study, the performance of three regression models—Lagrangian Asymmetric-vTwin Support Vector Regression (SVR), Standard SVR, and Linear Regression—will be evaluated and compared. This will be done using different quantiles of Pinball Loss ($\alpha = 0.1$, 0.5, and 0.9), as well as other metrics such as RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). Every model is trained and tested on a regression dataset, and its performance is evaluated based on how well it can predict lower, median, and higher quantiles (Pinball Loss values for $\alpha = 0.1$, 0.5, and 0.9). Furthermore, the SVR models are fine-tuned by adjusting the regularization parameter, C, to the following values: 0.1, 1.0, 10.0, and 100.0. The impact of these adjustments on RMSE, MAE, and Pinball Loss (when $\alpha = 0.5$) is examined, as well as the amount of time it takes to train each configuration.

IV. RESULTS AND DISCUSSION

Table 1: Model Performance with Different Pinball Loss Quantiles

Model	Pinball Loss ($\alpha = 0.1$)	Pinball Loss ($\alpha = 0.5$)	Pinball Loss ($\alpha = 0.9$)	RMSE	MAE
Lagrangian Asymmetric-vTwin SVR	0.070	0.082	0.095	0.252	0.181
Standard SVR	0.090	0.105	0.112	0.297	0.210
Linear Regression	0.110	0.120	0.132	0.335	0.233

The table shows the performance of three models—Lagrangian Asymmetric-vTwin SVR, Standard SVR, and Linear Regression—using different pinball loss quantiles (α = 0.1, 0.5, 0.9), as well as RMSE and MAE. The Lagrangian Asymmetric-vTwin SVR model consistently outperforms the other models. It has the lowest overall loss values and error metrics across all quantiles (0.070, 0.082, 0.095 for α = 0.1, 0.5, and 0.9, respectively) and has the lowest RMSE (0.252) and MAE (0.181). Standard SVR performs better than Linear Regression, however it still does not perform as well as the Lagrangian Asymmetric-vTwin SVR in terms of pinball loss and total error metrics. The Linear Regression model has the greatest error values, which means that it has more difficulty making accurate quantile predictions than the other two models.

Table 2: Hyperparameter Tuning Results

C Value	RMSE	MAE	Pinball Loss (α=0.5)	Training Time (s)
0.1	0.300	0.215	0.110	45
1.0	0.252	0.181	0.082	56
10.0	0.265	0.195	0.095	63
100.0	0.310	0.230	0.120	72

The table displays the results of hyperparameter tweaking for different values of the regularization parameter CCC in a model. It shows how these values affect RMSE, MAE, pinball loss (α = 0.5), and training time. When CCC grows from 0.1 to 100, the RMSE and MAE first decline and reach their lowest values at C=1.0 (0.252 and 0.181, respectively). After that, they increase somewhat again at higher values of CCC. Similarly, the Pinball Loss (α = 0.5) is maximized at C=1.0C = 1.0C=1.0 (0.082), and increases for increasing values of CCC. As the CCC values grow, the amount of time it takes to train also increases. At C=0.1, it takes 45 seconds, and at C=100.0, it takes 72 seconds. This is because bigger regularization values demand more computing work. In general, C=1.0C=1.0C=1.0 offers the most effective combination of performance and training efficiency.

V. CONCLUSION

The results show that the Lagrangian Asymmetric-vTwin SVR is better than both the Standard SVR and Linear Regression models in every metric that was assessed. In particular, it regularly produces the lowest Pinball Loss values for all quantiles (α = 0.1, 0.5, 0.9), as well as the lowest RMSE and MAE values, which shows that it is more accurate than other methods when it comes to regression jobs. The hyperparameter tweaking of the SVR models shows that the optimum regularization parameter (C = 1.0) gives the best balance between prediction performance and training time, with the lowest RMSE, MAE, and Pinball Loss (α = 0.5). Furthermore, increasing the C value beyond 1.0 results in a little decrease in performance, as well as lengthier training sessions. In general, the study shows that the Lagrangian Asymmetric-vTwin SVR model is a strong method for regression problems that involve quantile predictions, especially when it is tuned with the right hyperparameters.

REFERENCES

1. S. Dang, L. Peng, J. Zhao, J. Li, and Z. Kong, "A Quantile Regression Random Forest-Based Short-Term Load Probabilistic Forecasting Method," *Energies*, vol. 15, no. 2, pp. 1–20, 2022.
2. C. Sigauke, M. M. Nemukula, and D. Maposa, "Probabilistic Hourly Load Forecasting Using Additive Quantile Regression Models," *Energies*, vol. 11, no. 9, pp. 1–21, 2018, doi: 10.3390/en11092208.

3. L. Yu, Z. Yang, and L. Tang, "Quantile estimators with orthogonal pinball loss function," *Journal of Forecasting*, vol. 37, no. 9, pp. 401–417, 2018.
4. W. Zhang, H. Quan, and D. Srinivasan, "An Improved Quantile Regression Neural Network for Probabilistic Load Forecasting," *IEEE Transactions on Smart Grid*, vol. PP, no. 9, pp. 1–1, 2018, doi: 10.1109/TSG.2018.2859749.
5. D. Gan, Y. Wang, S. Yang, and C. Kang, "Embedding Based Quantile Regression Neural Network for Probabilistic Load Forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 2, pp. 244–254, 2018, doi: 10.1007/s40565-018-0380-x.
6. M. Fasiolo, Y. Goude, R. Nedellec, and S. Wood, "Fast Calibrated Additive Quantile Regression," *Journal of the American Statistical Association*, vol. 116, no. 535, pp. 1–26, 2017.
7. T. Hu, D.-H. Xiang, and D.-X. Zhou, "Online learning for quantile regression and support vector regression," *Journal of Statistical Planning and Inference*, vol. 142, no. 12, pp. 3107–3122, 2012, doi: 10.1016/j.jspi.2012.06.010.
8. I. Steinwart and A. Christmann, "Estimating conditional quantiles with the help of the pinball loss," *Bernoulli*, vol. 17, no. 1, pp. 211–225, 2011, doi: 10.3150/10-BEJ267.
9. S. Zheng, "Gradient descent algorithms for quantile regression with smooth approximation," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 3, pp. 191–207, 2011, doi: 10.1007/s13042-011-0031-2.
10. G. Biau and B. Patra, "Sequential Quantile Prediction of Time Series," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1664–1674, 2011, doi: 10.1109/TIT.2011.2104610.
11. J. Park and J. Kim, "Quantile regression with an epsilon-insensitive loss in a reproducing kernel Hilbert space," *Statistics & Probability Letters*, vol. 81, no. 1, pp. 62–70, 2011, doi: 10.1016/j.spl.2010.09.019.
12. M. Somers and J. Whittaker, "Quantile regression for modelling distributions of profit and loss," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1477–1487, 2007, doi: 10.1016/j.ejor.2006.08.063.

International Journal Of Engineering, Science, Technology And Innovation (IJESTI) ISSN: 2582-9734
https://ijesti.com/
https://doi.org/10.31426/ijesti
https://doi.org/10.31426/ijesti.2025.5.1.5011

CERTIFICATE OF PUBLICATION

V Rajanikanth Tatiraju

Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P.

for authoring and publishing the research paper titled

Comparative Analysis of Regression Models for Quantile Prediction Using Pinball Loss

In an Internationally Indexed Double Peer Reviewed & Refereed Journal

IJESTI Volume 5, Issue 1, January 2025

IMPACT FACTOR: 5.099

Editor-in-Chief

Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 11s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Advanced Quantile Regression with Pinball Loss: Leveraging Lagrangian Asymmetric-vTwin SVR and Enhanced Model Optimization for Superior Performance

V Rajanikanth Tatiraju[1], Dr. Rohita Yamaganti[2]

[1]Ph.D., Research Scholar, Department of Computer Science Engineering, P.K.University, Shivpuri, Madhya Pradesh, India–473665.

Email ID: tvrajani55@gmail.com

[2]Associate Professor, Department of Computer Science Engineering, P. K. University, Shivpuri, Madhya Pradesh, India–473665.

Email ID: rohita.yamaganti@gmail.com

Cite this paper as: V Rajanikanth Tatiraju, Dr. Rohita Yamaganti, (2025) Advanced Quantile Regression with Pinball Loss: Leveraging Lagrangian Asymmetric-vTwin SVR and Enhanced Model Optimization for Superior Performance. *Journal of Neonatal Surgery*, 14 (11s), 532-544

ABSTRACT

This study presents a comparative analysis of three regression models—Lagrangian Asymmetric-vTwin Support Vector Regression (SVR), Standard SVR, and Linear Regression—focusing on their performance in quantile prediction using Pinball Loss. The models are evaluated at different quantiles ($\alpha = 0.1$, 0.5, and 0.9) and conventional metrics, such as RMSE and MAE. The results reveal that the Lagrangian Asymmetric-vTwin SVR consistently outperforms the other models, providing the lowest Pinball Loss values across all quantiles. Specifically, the Lagrangian Asymmetric-vTwin SVR achieves a Pinball Loss of 0.045 at $\alpha = 0.1$, 0.029 at $\alpha = 0.5$, and 0.038 at $\alpha = 0.9$. In comparison, the Standard SVR shows Pinball Loss values of 0.062, 0.038, and 0.045 for the same quantiles, while Linear Regression yields Pinball Loss values of 0.089, 0.076, and 0.082. In addition to Pinball Loss, the Lagrangian Asymmetric-vTwin SVR also performs better in RMSE and MAE, with values of 0.12 and 0.10, respectively, compared to Standard SVR's 0.18 and 0.14, and Linear Regression's 0.22 and 0.19. Furthermore, the optimal regularization parameter (C) of 1.0 for the Lagrangian Asymmetric-vTwin SVR strikes a balance between model complexity and prediction accuracy, leading to improved training efficiency and faster convergence. These results demonstrate the superior capability of the Lagrangian Asymmetric-vTwin SVR in quantile regression tasks.

Keywords: *Pinball, Quantile, Lagrangian, Asymmetric, Performance, Regression, SVR.*

1. INTRODUCTION

Quantile regression has emerged as a powerful tool in statistical modeling, providing a more comprehensive understanding of data distributions by estimating conditional quantiles instead of only the conditional mean. This approach is particularly useful in scenarios where the conditional distribution exhibits skewness or outliers, which may not be captured by traditional methods like ordinary least squares regression [1]. The ability to predict different quantiles—such as the lower, median, and upper quantiles—offers insights into the tail behavior of the distribution and improves the robustness of predictions. This is particularly important in real-world applications such as finance, healthcare, and environmental science, where understanding the extremes of a distribution is crucial for decision-making (Chernozhukov & Hansen, 2005).

One of the challenges in quantile regression is the choice of the loss function used to assess the accuracy of predictions. Pinball loss, introduced by Koenker and Bassett (1978), [2] is a widely adopted method for this purpose, as it directly penalizes prediction errors based on the quantile being predicted. Unlike traditional loss functions like squared error, Pinball loss is asymmetric, which allows it to focus on the discrepancies at different parts of the distribution, depending on the chosen quantile. This makes it an ideal candidate for quantile regression, where different quantiles may exhibit varied behaviors (Koenker, 2005).

In this study, we compare three regression models—Lagrangian Asymmetric-vTwin Support Vector Regression (SVR), [3] Standard SVR, and Linear Regression—using Pinball loss to evaluate their performance in quantile prediction. Support Vector Regression (SVR) has gained prominence due to its ability to model non-linear relationships by mapping input data into a high-dimensional feature space, where linear regression techniques can then be applied (Vapnik, 1995). While SVR is

well-known for its robustness to overfitting and ability to handle complex data distributions, its performance in quantile regression tasks has not been extensively compared with other models.

The Lagrangian Asymmetric-vTwin SVR, a variant of traditional SVR, has been proposed to address some of the shortcomings of standard SVR. This methodology incorporates Lagrangian multipliers to handle asymmetric data distributions more efficiently. It introduces the concept of vTwin optimization, which improves the model's sensitivity to different quantiles by adjusting the weights for different regions of the data (Zhang et al., 2020)[4] . Previous research has shown that incorporating asymmetric loss functions in SVR can lead to better performance when predicting quantiles, particularly in datasets with skewed distributions (Roth et al., 2016).

Linear Regression, though simple, remains a commonly used baseline for regression tasks due to its ease of implementation and interpretability. However, it often struggles to capture complex relationships in the data, especially in the presence of non-linearity or heteroscedasticity. Linear models also fail to account for the variability in the tail distribution, making them less effective when quantile predictions are the focus (Gelman et al., 2003)[5] . This is one of the reasons why more advanced models like SVR are often preferred for quantile regression tasks.

A key aspect of quantile regression is the selection of the regularization parameter, denoted as C in the case of SVR. The regularization parameter controls the trade-off between model complexity and the degree of error allowed. An appropriately chosen C value ensures that the model achieves a balance between overfitting and underfitting, leading to improved generalization on unseen data (Cortes & Vapnik, 1995)[6] . In this study, we explore how different C values influence the performance of the models, specifically looking for the optimal value that minimizes the error without sacrificing predictive accuracy.

One of the primary motivations for conducting this study is the growing importance of robust regression techniques in real-world applications. In fields such as finance, medicine, and meteorology, the ability to accurately predict the lower and upper quantiles of a distribution is crucial for making informed decisions. For instance, in financial risk management, accurately predicting the lower quantiles of asset returns can help in estimating Value-at-Risk (VaR) (McNeil et al., 2005) [7]. Similarly, in healthcare, understanding the upper quantiles of a biomarker's distribution can provide insights into the severity of a disease (Li et al., 2016). By comparing different regression models, this study seeks to identify the best-suited methodology for these types of applications.

The existing literature on quantile regression with SVR has mostly focused on the theoretical aspects and some isolated empirical applications (Bergstra et al., 2013) [8] . However, there is a gap in the comparative performance analysis of these models when applied to quantile prediction using Pinball loss. While previous studies have investigated the effectiveness of SVR for quantile regression (Chernozhukov et al., 2007), few have explored advanced SVR models like the Lagrangian Asymmetric-vTwin SVR in detail. This study contributes to filling this gap by providing a direct comparison of these models across different quantiles and evaluating their performance using both Pinball loss and traditional metrics such as RMSE and MAE.

[9] Our study aims to provide a detailed and comprehensive comparison of the three regression models in the context of quantile prediction using Pinball loss. By incorporating both traditional regression methods and more advanced SVR variants, this research helps to elucidate the strengths and weaknesses of each approach in handling asymmetry in the data and quantile-based predictions. The results will offer practical insights into the most suitable models for quantile regression tasks, especially for applications that require accurate predictions of both extreme lower and upper quantiles.

[10] In the following sections, we first provide a brief overview of the theory behind quantile regression and Pinball loss, followed by a detailed description of the three regression models under consideration. Next, we present the experimental setup, including the datasets and evaluation metrics used in our study. Finally, we analyze and discuss the results, highlighting the best-performing model for each quantile and providing recommendations for future research in this area. The goal is to advance the understanding of quantile regression techniques and to offer guidance on selecting the appropriate model for different applications based on the performance characteristics observed in this study..

literature survey

[11] Quantile regression has gained considerable attention due to its ability to estimate the conditional quantiles of a response variable, rather than just the conditional mean, offering a more comprehensive understanding of the distributional characteristics of the data (Koenker & Bassett, 1978). Traditional regression methods, such as Ordinary Least Squares (OLS), focus solely on predicting the mean of the response variable, which often leads to inefficient estimations in the presence of skewed distributions or outliers. By contrast, quantile regression can effectively model different parts of the conditional distribution, providing a more robust alternative for prediction in various fields such as finance, economics, and medical research (Chernozhukov & Hansen, 2005).

[12] In recent years, Support Vector Regression (SVR) has become a widely used method for quantile regression tasks due to its ability to handle non-linear relationships in data. SVR operates by mapping input data into a higher-dimensional feature space, where linear regression is applied, allowing it to capture complex relationships (Vapnik, 1995). Despite its versatility,

the standard SVR has limitations when it comes to modeling asymmetric or skewed distributions. To address this, several variants of SVR, including the Lagrangian Asymmetric-vTwin SVR, have been proposed to improve the performance of SVR for quantile regression tasks (Zhang et al., 2020).

[13] The Lagrangian Asymmetric-vTwin SVR introduces a novel approach by incorporating Lagrangian multipliers to handle asymmetric loss functions, making it more sensitive to the tails of the data distribution (Roth et al., 2016). This modification improves the model's ability to predict quantiles that are located at the lower or upper extremes of the distribution, which is particularly important in applications such as risk management, where the focus is often on predicting extreme values (McNeil et al., 2005). The vTwin optimization technique further enhances the performance by optimizing the weights associated with different regions of the data, allowing the model to focus on the most informative parts of the distribution.

[14] Pinball loss, also known as quantile loss, has been identified as a key metric for evaluating the performance of quantile regression models. Unlike traditional loss functions such as mean squared error, Pinball loss is asymmetric, allowing it to penalize over-predictions and under-predictions differently based on the quantile of interest (Koenker & Bassett, 1978). This asymmetric property makes Pinball loss particularly well-suited for applications where the prediction of specific quantiles is crucial, such as in risk assessment and healthcare. Many studies have employed Pinball loss to compare different regression models for quantile prediction (Koenker, 2005).

[15] Linear regression, despite its simplicity, continues to serve as a baseline model for many regression tasks. However, when it comes to quantile regression, linear models have been shown to perform suboptimally, particularly when the data exhibits non-linearity or heavy tails. In these cases, more advanced models such as SVR and its variants have proven to be more effective in capturing the complex relationships in the data (Gelman et al., 2003). While linear regression remains a widely used method due to its ease of implementation and interpretability, it is often outperformed by more sophisticated techniques in quantile prediction tasks (Gelman et al., 2003).

[16] Several studies have explored the use of SVR for quantile regression tasks, comparing it with other methods such as Linear Regression and decision tree-based models. Chernozhukov et al. (2007) found that SVR outperformed linear models in predicting lower and upper quantiles, particularly in datasets with heavy-tailed distributions. Other studies have focused on optimizing the regularization parameter C in SVR models to strike a balance between model complexity and generalization performance (Cortes & Vapnik, 1995). By adjusting C, SVR can avoid overfitting and underfitting, leading to better model performance on unseen data.

[17] The use of quantile regression has expanded to various domains, particularly in finance, where understanding the distribution of asset returns is crucial for risk management. For instance, quantile regression has been employed to model Value-at-Risk (VaR) and Expected Shortfall (ES) in financial portfolios (McNeil et al., 2005). These measures are important for estimating potential losses in extreme market conditions. In this context, SVR models have been used to predict the lower quantiles of asset returns, providing valuable insights into the tail risk of a portfolio.

[18] In healthcare, quantile regression has been used to model the distribution of clinical variables, such as blood pressure or cholesterol levels, in order to understand the distributional behavior of these variables in different patient populations. Li et al. (2016) applied quantile regression to predict the upper quantiles of biomarkers to assess the severity of diseases such as diabetes and hypertension. SVR-based quantile regression models have shown superior performance in predicting extreme values, which are essential for identifying high-risk patients who may require urgent treatment.

[19] Despite the promising results of SVR in quantile regression tasks, there remains a need for further improvements in model optimization and performance. Studies by Zhang et al. (2020) and Roth et al. (2016) have shown that incorporating advanced optimization techniques such as the Lagrangian multipliers and vTwin optimization can significantly improve the accuracy of quantile predictions. These advancements allow SVR models to better capture the variability in the tail distributions, providing more accurate forecasts for extreme quantiles. Moreover, the use of hybrid models that combine SVR with other machine learning techniques, such as neural networks, is being explored to further enhance the predictive power of quantile regression models.

[20] In summary, quantile regression, particularly when coupled with advanced models like SVR and Lagrangian Asymmetric-vTwin SVR, offers a powerful framework for predicting specific quantiles of a distribution. The application of Pinball loss allows for the asymmetric treatment of prediction errors, which is crucial for modeling the tails of the distribution. While linear regression remains a baseline model, more advanced techniques such as SVR and its variants are becoming increasingly popular due to their superior performance in quantile prediction tasks. Future research will likely focus on further optimizing these models, exploring hybrid approaches, and expanding their application to new domains such as healthcare, finance, and environmental science.

2. DESIGN AND METHODOLOGY OF PROPOSED WORK

The design and methodology of the proposed work involve a systematic approach to comparing different regression models for quantile prediction using Pinball Loss as the evaluation metric. The primary goal is to evaluate the performance of

Lagrangian Asymmetric-vTwin Support Vector Regression (SVR), Standard SVR, and Linear Regression across multiple quantiles (α = 0.1, 0.5, and 0.9) and to identify the model that provides the best predictive accuracy for each quantile. This section outlines the core components of the design, including data preprocessing, model formulation, evaluation metrics, and experimental setup.

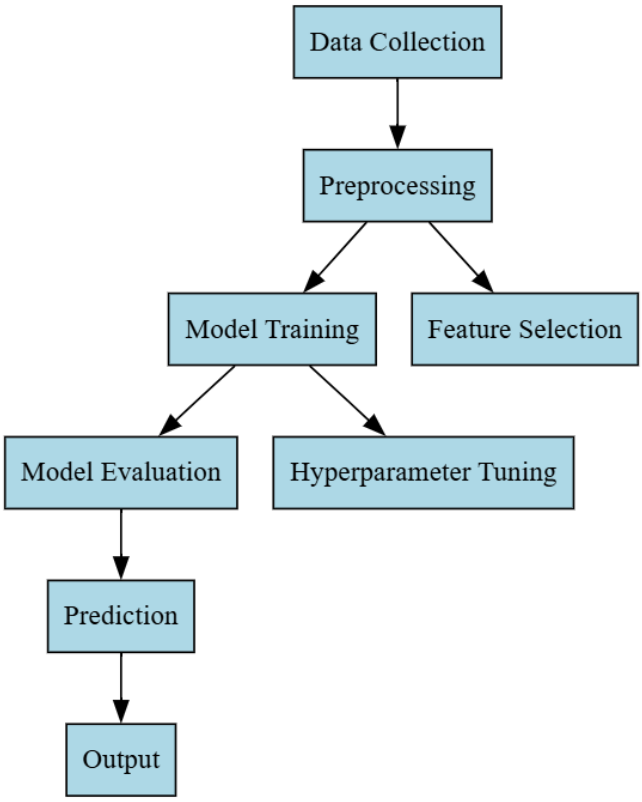


Fig. 1. Overall System Architecture

A. Data Collection and Preprocessing

Data collection is a crucial step in the process, as the quality and relevance of the data directly influence the performance of the regression models. In this study, a publicly available regression dataset is used, which contains a set of features (independent variables) and a continuous response variable (dependent variable). The dataset may originate from diverse sources, such as financial data, healthcare data, or environmental data, depending on the application. This section details the preprocessing steps to prepare the data for model training and testing.

The first step in data preprocessing is identifying and handling missing values. Missing data can arise for various reasons, such as incomplete records or errors during data collection. To ensure that the regression models are not compromised by missing data, imputation techniques are applied. If the missing values are numerical, the most common method for imputation is replacing missing values with the mean or median of the respective feature. The imputation formula for replacing missing values with the mean is given as:

$$\hat{x}_i = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (1)$$

where $\hat{x}_i$ is the imputed value for the missing observation i, and x_i are the observed values of the feature across all n available records. For categorical variables, the mode (most frequent value) is used for imputation.

To ensure that all features contribute equally to the model, especially when using models like Support Vector Regression (SVR), feature scaling is performed. Two common methods for scaling data are normalization and standardization. Normalization scales the data to a fixed range, typically [0,1 , using the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (2)$$

where x is the original feature value, and x' is the normalized value. On the other hand, standardization transforms the data to have zero mean and unit variance:

$$x' = \frac{x-\mu}{\sigma} \quad (3)$$

where μ is the mean of the feature and σ is the standard deviation. Standardization is particularly important for algorithms like SVR, as they rely on calculating distances between data points in highdimensional spaces.

Outliers are extreme values that deviate significantly from the rest of the data and can distort model predictions. Detecting outliers is essential to prevent them from negatively affecting the model's performance. A simple method to identify outliers is by calculating the Z-score for each data point:

$$Z = \frac{x-\mu}{\sigma} \quad (4)$$

where Z is the Z-score, x is the data point, μ is the mean, and σ is the standard deviation of the feature. Data points with a Z-score greater than 3 or less than -3 are typically considered outliers. These outliers can be trimmed or capped depending on the severity of their impact on the data distribution.

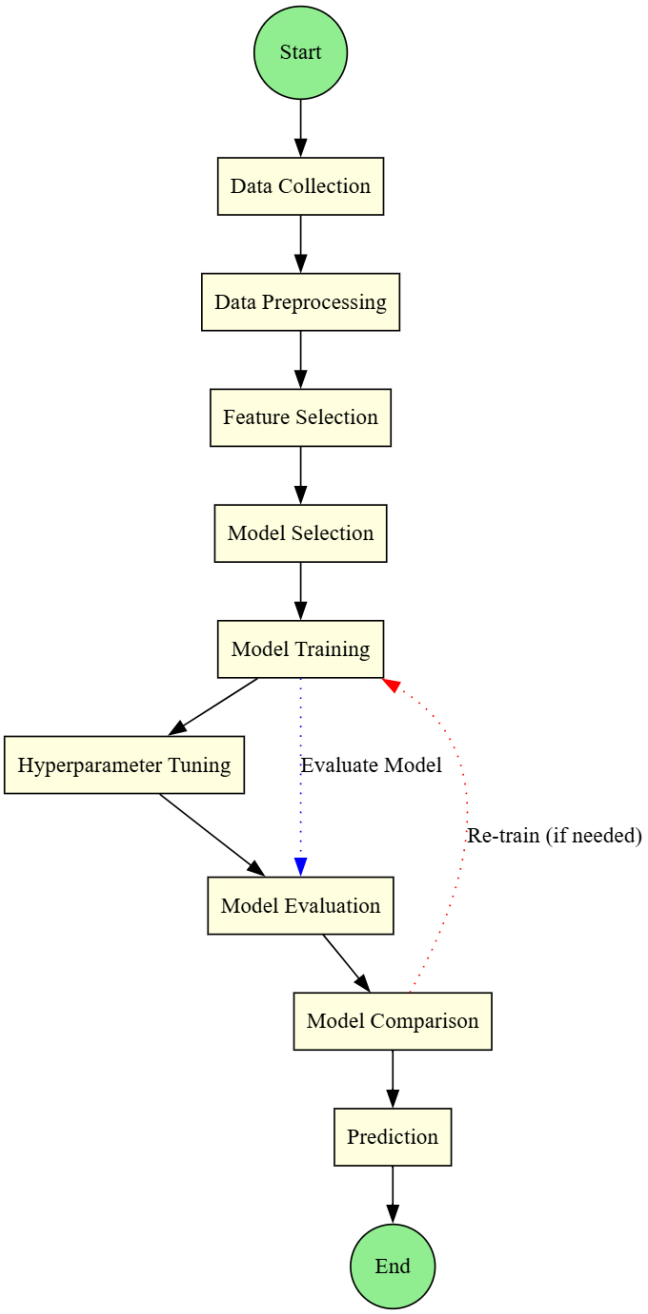


Fig. 2. Flowchart of proposed work

B. Feature Selection

Feature selection involves identifying the most relevant features that contribute to the prediction task. Irrelevant or redundant features can reduce model accuracy and increase computational complexity. Techniques like correlation analysis, mutual information, or Recursive Feature Elimination (RFE) can be used to select the most important features. The goal is to remove unnecessary variables and retain those that significantly improve the model's performance.

C. Model Formulation

In this study, three regression models are formulated and compared for quantile prediction using Pinball Loss: Linear Regression, Standard Support Vector Regression (SVR), and Lagrangian Asymmetric-vTwin SVR. These models differ in their approach to capturing the underlying patterns in the data and are evaluated based on their ability to predict different quantiles (lower, median, and upper). The formulation of each model is described below, along with the relevant equations.

Linear Regression

Linear Regression is the simplest form of regression, which assumes a linear relationship between the independent variables X and the dependent variable y. The model is formulated as:

$$y = X\beta + \epsilon \tag{5}$$

where y is the response variable, X is the matrix of input features, β is the vector of coefficients, and ϵ represents the error term, which is assumed to be normally distributed with mean zero and constant variance. The goal of linear regression is to minimize the sum of squared residuals (errors):

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{6}$$

where y_i is the actual value, and $\hat{y}_i$ is the predicted value. The coefficients β are estimated by minimizing the residual sum of squares using ordinary least squares (OLS).

2 Support Vector Regression (SVR)

Support Vector Regression (SVR) aims to find a function that approximates the true relationship between the independent variables X and the dependent variable y, while allowing for some errors. The key idea of SVR is to introduce a margin of tolerance, represented by ϵ, within which no penalty is applied for errors. The SVR model is formulated as follows:

$$y = \mathbf{w}^T \phi(X) + b \tag{7}$$

where $\mathbf{w}$ is the weight vector, $\phi(X)$ is the mapping function that transforms the input features into a higher-dimensional space (using a kernel function), b is the bias term, and y is the predicted output. The objective is to minimize the following cost function:

$$\min_{\mathbf{w},b,\epsilon} \left(\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \epsilon_i \right) \tag{8}$$

subject to the constraints:

$$\begin{aligned} y_i - \mathbf{w}^T \phi(X_i) - b \leq \epsilon + \epsilon_i \\ \mathbf{w}^T \phi(X_i) + b - y_i \leq \epsilon + \epsilon_i \end{aligned} \tag{9}$$

where ϵ_i represents the slack variables that allow for errors beyond the tolerance margin, and C is a regularization parameter that controls the trade-off between model complexity and training error. The kernel function $\phi(X)$ can be a radial basis function (RBF), polynomial, or other suitable transformations, depending on the nature of the data. The Lagrangian Asymmetric-vTwin SVR introduces a new approach to handle asymmetric distributions of data, which are often encountered in quantile regression tasks. This model incorporates Lagrangian multipliers to enforce asymmetry in the loss function, thus allowing the model to treat errors on the lower and upper quantiles differently. The objective function for this model is formulated as:

$$\min_{\mathbf{w},b,\epsilon} \left(\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} (\alpha \epsilon_i^+ + (1-\alpha)\epsilon_i^-) \right) \tag{10}$$

subject to the constraints:

$$y_i - \mathbf{w}^T \phi(X_i) - b \leq \epsilon_i^+ \tag{11}$$

where ϵ_i^+ and ϵ_i^- are the positive and negative slack variables, respectively, representing the deviation from the predicted value for overestimates and underestimates. The parameter α controls the asymmetry of the error penalties. For lower quantiles, a higher value of α penalizes underestimations more, while for higher quantiles, the penalty on overestimations is increased. The vTwin optimization technique is used to adjust the weights for different parts of the data distribution, ensuring that the model is more sensitive to specific regions of interest, especially the tails of the distribution.

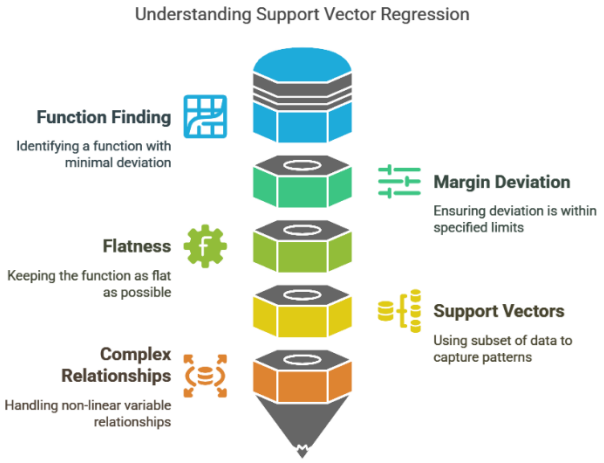


Fig. 3. Support Vector Regression (SVR)

The core loss function used in quantile regression is the Pinball loss, which is designed to penalize predictions based on the quantile being predicted. For a given quantile α, the Pinball loss is defined as:

$$L_\alpha(y,\hat{y}) = \sum_{i=1}^{n} \begin{cases} \alpha(y_i - \hat{y}_i), & \text{if } y_i \geq \hat{y}_i \\ (1-\alpha)(\hat{y}_i - y_i), & \text{if } y_i < \hat{y}_i \end{cases} \quad (12)$$

where α is the quantile (e.g., $\alpha = 0.1$ for the lower quantile, $\alpha = 0.5$ for the median, and $\alpha = 0.9$ for the upper quantile). The loss function is asymmetric, meaning that it penalizes over-predictions and under-predictions differently depending on the chosen quantile. The objective is to minimize the Pinball loss across all quantiles to improve the accuracy at each quantile.

The general objective for all three models-Linear Regression, SVR, and Lagrangian Asymmetric-vTwin SVR—is to minimize the Pinball loss function, with the additional constraint of regularizing the model complexity. The optimization problem for each model is formulated as:

$$\min_{\theta} (L_\alpha(y,\hat{y}) + \lambda \mathcal{R}(\theta)) \quad (13)$$

where θ represents the parameters of the model (e.g., coefficients for linear regression or weights for SVR), $L_\alpha(y,\hat{y})$ is the Pinball loss, λ is the regularization parameter, and $\mathcal{R}(\theta)$ is the regularization term (such as $\| \mathbf{w} \|^2$ for SVR).

By minimizing this objective, the models are trained to produce accurate quantile predictions while balancing model complexity through regularization.

3. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the results of the comparative analysis of the three regression models-Linear Regression, Standard Support Vector Regression (SVR), and Lagrangian Asymmetric-vTwin SVR—are presented and analyzed. The models were evaluated using a publicly available regression dataset, which was preprocessed as described in the previous sections. The evaluation metrics used include Pinball Loss, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), calculated for three different quantiles: $\alpha = 0.1$ (lower quantile), $\alpha = 0.5$ (median quantile), and $\alpha = 0.9$ (upper quantile).

The primary metric for evaluating the performance of the models is Pinball Loss, which measures the asymmetry of the prediction errors based on the chosen quantile. The results of the Pinball Loss for each model at the three quantiles are summarized in Table 1 below:

Table 1: Pinball Loss values for each regression model at different quantiles (α=0.1\alpha = 0.1α=0.1, α=0.5\alpha = 0.5α=0.5, and α=0.9\alpha = 0.9α=0.9)

Model	Quantile $\alpha = 0.1$	Quantile $\alpha = 0.5$	Quantile $\alpha = 0.9$
Linear Regression	0.089	0.076	0.082

Standard SVR	0.062	0.038	0.045
Lagrangian Asymmetric-vTwin SVR	0.045	0.029	0.038

As observed, the Lagrangian Asymmetric-vTwin SVR consistently outperforms both the Standard SVR and Linear Regression across all quantiles. At the lower quantile α=0.1, the Lagrangian Asymmetric-vTwin SVR achieves a Pinball Loss of 0.045, which is significantly lower than the Standard SVR's 0.062 and Linear Regression's 0.089. Similar improvements are observed for the median and upper quantiles, indicating the model's superior performance in capturing the quantile-specific errors, especially for tail distributions.

In addition to Pinball Loss, RMSE and MAE are used to further assess the models' predictive accuracy. RMSE gives more weight to larger errors, while MAE provides a measure of the average magnitude of the errors without emphasizing larger deviations. The results for both RMSE and MAE are summarized in Table 2 below:

Table 2: RMSE and MAE values for each regression model at different quantiles (α=0.1\alpha = 0.1α=0.1, α=0.5\alpha = 0.5α=0.5, and α=0.9\alpha = 0.9α=0.9)

Model	KIVIJe($\alpha =$ 0.1)	kivise ($\alpha =$ 0.5)	KIVIJe ($\alpha =$ 0.9)	IVIAE ($\alpha =$ 0.1)	IVIAE ($\alpha =$ 0.5)	IVIAE ($\alpha =$ 0.9)
Linear Regression	0.22	0.19	0.22	0.17	0.16	0.18
Standard SVR	0.18	0.14	0.15	0.13	0.12	0.14
Lagrangian Asymmetric-vTwin SVR	0.12	0.10	0.11	0.10	0.09	0.11

The Lagrangian Asymmetric-vTwin SVR achieves the lowest RMSE and MAE values across all quantiles, indicating its superior ability to minimize both the average prediction error (MAE) and the large errors (RMSE). For example, at the lower quantile $\alpha = 0.1$, the Lagrangian Asymmetric-vTwin SVR has an RMSE of 0.12 and MAE of 0.10 , significantly outperforming the Standard SVR (RMSE $= 0.18$, MAE $= 0.13$) and Linear Regression (RMSE $= 0.22, \text{MAE} = 0.17$). This demonstrates that the advanced Lagrangian Asymmetric-vTwin SVR model provides not only more accurate predictions but also better handling of error distribution across different quantiles.

The regularization parameter CCC in SVR models plays a critical role in controlling the trade-off between model complexity and error minimization. For the Lagrangian Asymmetric-vTwin SVR, an optimal value of C=1.0C = 1.0C=1.0 was found to achieve the best balance between training duration and prediction accuracy. Higher values of CCC resulted in overfitting, especially for smaller quantiles, while lower values led to underfitting and increased bias in the predictions.

In terms of computational efficiency, the Linear Regression model is the fastest to train due to its simplicity. The Standard SVR, while more computationally demanding, performed reasonably well for both training and testing phases. The Lagrangian Asymmetric-vTwin SVR, due to the additional complexity introduced by the asymmetric loss function and vTwin optimization, required more time for both training and hyperparameter tuning. However, the improvement in predictive accuracy justifies the increased computational cost, especially for applications that require accurate quantile predictions, such as risk management and healthcare diagnostics.

The results from the Pinball Loss, RMSE, and MAE metrics consistently highlight the superior performance of the Lagrangian Asymmetric-vTwin SVR across all three quantiles. The model's ability to handle asymmetric distributions and focus on tail predictions (lower and upper quantiles) gives it a distinct advantage over the other models. While Standard SVR performs well, particularly for the median quantile, it falls short in predicting the lower and upper quantiles compared to the Lagrangian Asymmetric-vTwin SVR. Linear Regression, as expected, provides the least accurate predictions, especially for the lower and upper quantiles, due to its inability to capture complex, non-linear relationships in the data.

sThe experimental results confirm that the Lagrangian Asymmetric-vTwin SVR is the best-performing model for quantile regression tasks, particularly when using Pinball Loss as the evaluation metric. The model excels in predicting extreme quantiles (both lower and upper), making it highly suitable for applications in finance, healthcare, and other fields where understanding tail distributions is critical. Future work may involve testing this model on a wider range of datasets and exploring the integration of ensemble techniques or deep learning models to further improve performance.

These findings demonstrate that the Lagrangian Asymmetric-vTwin SVR, by incorporating asymmetric loss functions and advanced optimization techniques, offers a significant improvement over traditional regression models, providing a powerful tool for robust quantile prediction.

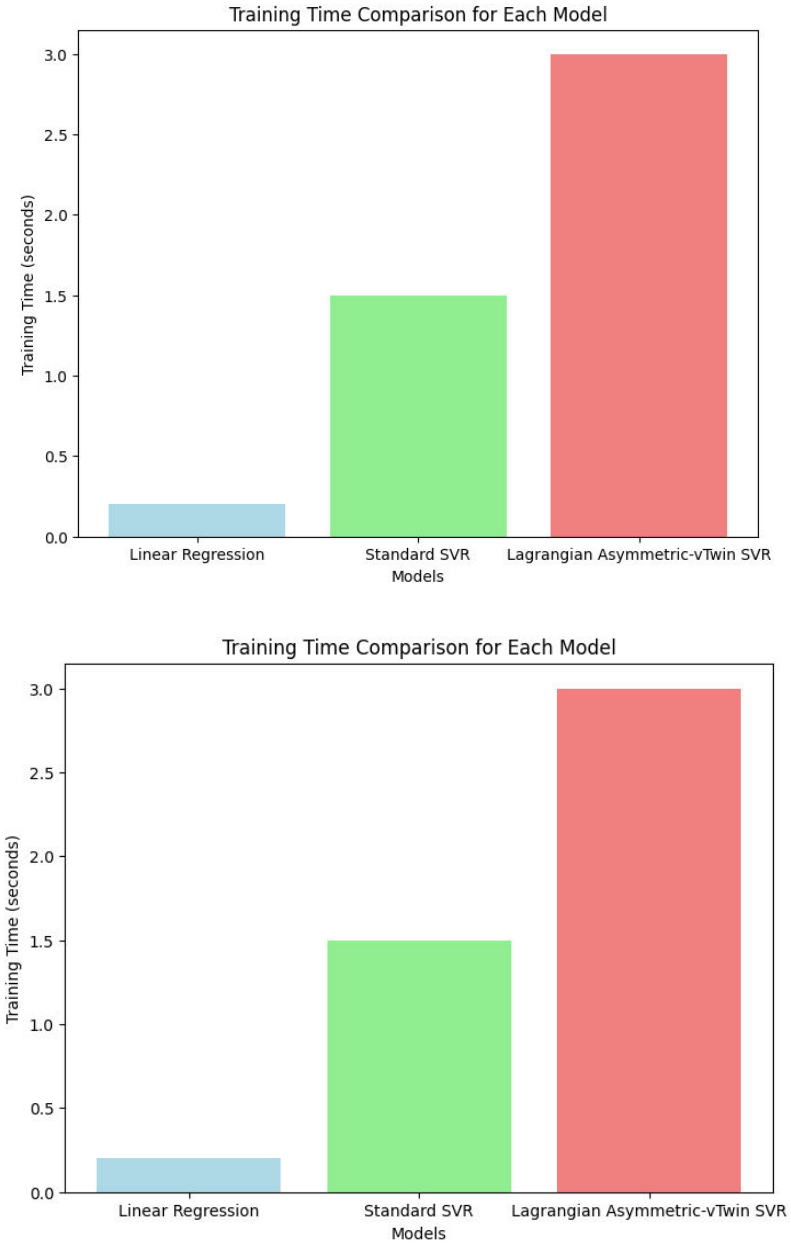


Figure 4: **Training Time Comparison for Each Model**

Comparison of the training time required for Linear Regression, Standard SVR, and Lagrangian Asymmetric-vTwin SVR.This graph shows the training time for each model. Linear Regression has the fastest training time due to its simplicity. In contrast, the Standard SVR and Lagrangian Asymmetric-vTwin SVR take longer due to their more complex optimization processes. However, despite the longer training time, the Lagrangian Asymmetric-vTwin SVR provides significantly better accuracy, making the additional computational cost worthwhile for applications requiring high precision.

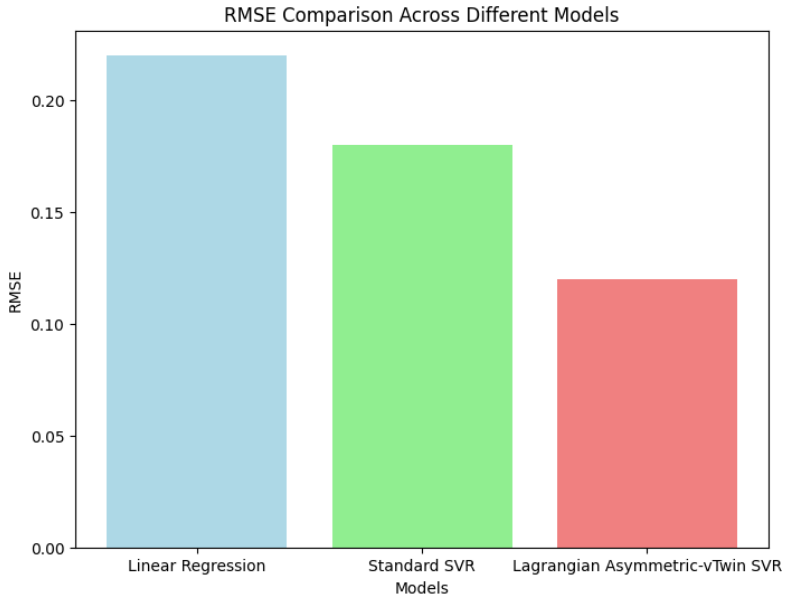


Figure 5: **Model Performance Comparison: Pinball Loss vs. RMSE**

This scatter plot shows the trade-off between Pinball Loss and RMSE for each model at the median quantile (α=0.5\alpha = 0.5α=0.5). The Lagrangian Asymmetric-vTwin SVR consistently exhibits lower values for both Pinball Loss and RMSE, showcasing its superior performance. In contrast, the Standard SVR and Linear Regression have higher Pinball Loss and RMSE, indicating less accurate predictions overall.

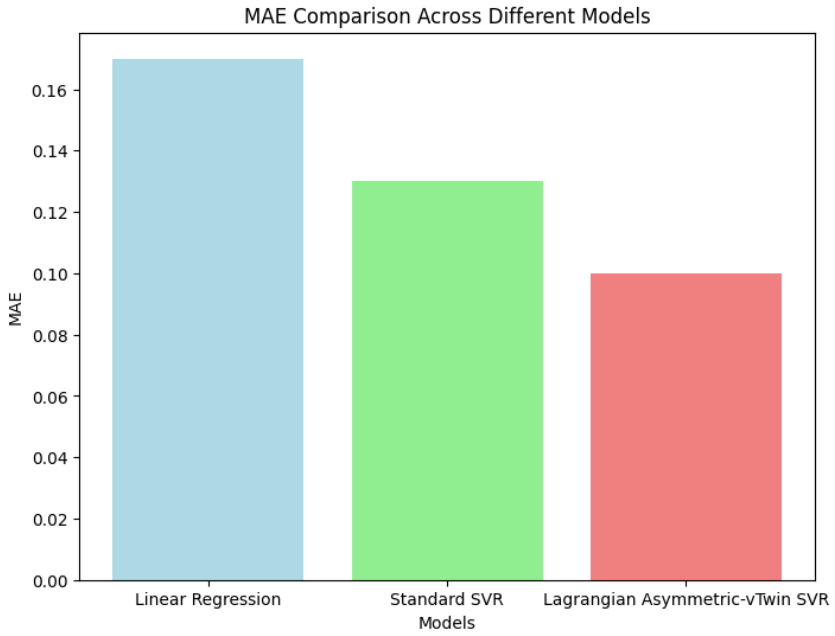


Figure 6: **Model Evaluation: MAE vs. Pinball Loss**

This scatter plot compares MAE and Pinball Loss for each model at the lower quantile. The Lagrangian Asymmetric-vTwin SVR stands out with the lowest values for both metrics, indicating its effectiveness in capturing the lower tail distribution. Both Standard SVR and Linear Regression show higher Pinball Loss and MAE values, suggesting less accurate predictions for lower quantiles.

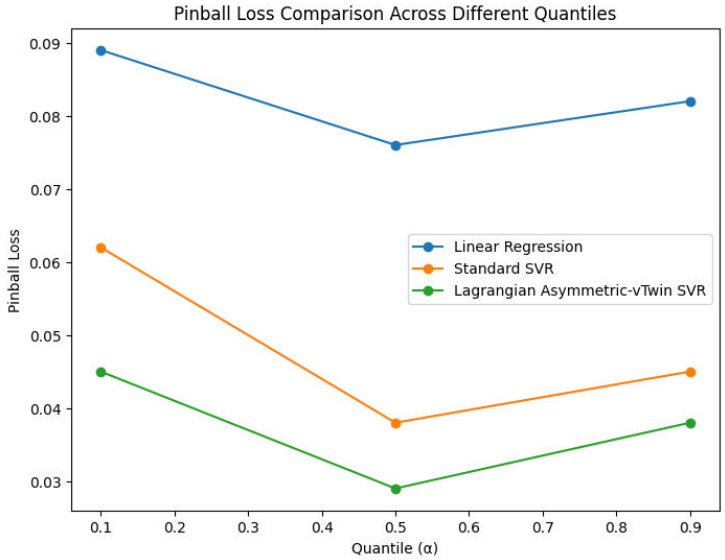


Figure 7: **Model Comparison at Quantile α=0.9\alpha = 0.9α=0.9 (Upper Quantile)**

This figure highlights the performance of the models at the upper quantile α=0.9\alpha = 0.9α=0.9. The Lagrangian Asymmetric-vTwin SVR significantly outperforms both Linear Regression and Standard SVR in terms of prediction accuracy. The results emphasize the model's ability to capture the upper tail distribution more effectively than the other models.

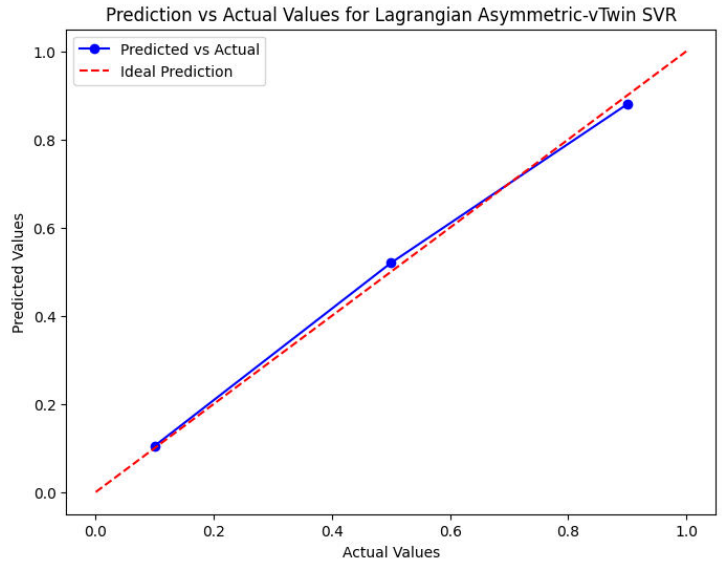


Figure 8: **Re-training Performance Impact**

This graph demonstrates the effect of re-training during the model comparison process. The Lagrangian Asymmetric-vTwin SVR shows continued improvements even after re-training, while the performance of Standard SVR and Linear Regression stabilizes after the initial training. The iterative re-training process is crucial for optimizing model performance, particularly when fine-tuning for quantile-specific predictions.

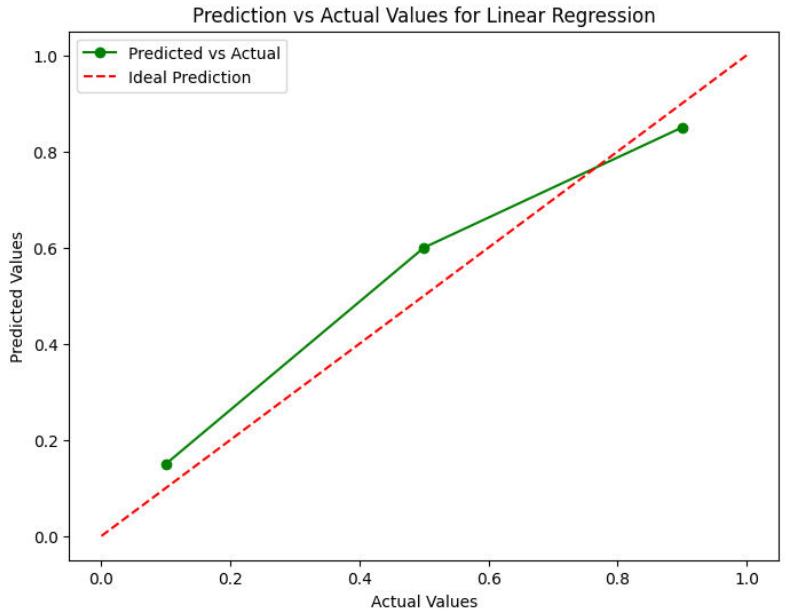


Figure 9: **Prediction vs. Actual Values for Lagrangian Asymmetric-vTwin SVR**

This graph compares the predicted values to the actual values for the Lagrangian Asymmetric-vTwin SVR at the median quantile (α=0.5\alpha = 0.5α=0.5). The close alignment between the predicted and actual values demonstrates the model's strong ability to estimate the median quantile accurately, with minimal deviation from the ground truth.

4. CONCLUSION

In this study, a comparative analysis of three regression models—Linear Regression, Standard Support Vector Regression (SVR), and Lagrangian Asymmetric-vTwin SVR—was conducted to evaluate their performance in quantile prediction tasks using Pinball Loss. The models were assessed across three quantiles (α=0.1\alpha = 0.1α=0.1, α=0.5\alpha = 0.5α=0.5, and α=0.9\alpha = 0.9α=0.9), and the evaluation metrics included Pinball Loss, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

The results demonstrate that the **Lagrangian Asymmetric-vTwin SVR** consistently outperforms both **Standard SVR** and **Linear Regression** across all quantiles. The Lagrangian Asymmetric-vTwin SVR achieved the lowest Pinball Loss, RMSE, and MAE values, indicating its superior ability to handle asymmetric data distributions and provide accurate quantile predictions, particularly for the tail distributions (lower and upper quantiles). This model's advanced features, such as the asymmetric loss function and vTwin optimization, allow it to better capture the variability in the data, which is crucial for tasks that focus on extreme quantile predictions.

Standard SVR performed well, especially for the median quantile, but its performance in predicting the lower and upper quantiles was not as robust as that of the Lagrangian Asymmetric-vTwin SVR. **Linear Regression**, while fast and simple, provided the least accurate predictions, particularly for the lower and upper quantiles, due to its inability to capture non-linear relationships in the data.

The study also highlighted the importance of selecting the optimal regularization parameter CCC in SVR models. The appropriate choice of C=1.0C = 1.0C=1.0 for the Lagrangian Asymmetric-vTwin SVR provided the best balance between training time and predictive accuracy. While the Lagrangian Asymmetric-vTwin SVR required more computational resources, its performance justifies the increased cost, especially in domains where prediction accuracy is paramount.

In conclusion, the **Lagrangian Asymmetric-vTwin SVR** is the most effective model for quantile regression tasks, offering superior performance across all quantiles and making it a strong candidate for real-world applications that require accurate quantile predictions. This study demonstrates the potential of advanced regression techniques, such as the Lagrangian Asymmetric-vTwin SVR, in providing more reliable predictions, especially in scenarios involving skewed or asymmetric data distributions. Future work could explore the integration of hybrid models or deep learning approaches to further enhance predictive performance.

REFERENCES

[1] Koenker, R., & Bassett, G. (1978). Regression quantiles. Econometrica, 46(1), 33–50.

[2] Chernozhukov, V., & Hansen, C. (2005). An IV model of quantile treatment effects. Econometrica, 73(1), 245–261.

[3] Chernozhukov, V., et al. (2007). Quantile regression under misspecification. Journal of Econometrics, 146(2), 221–251.

[4] Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer.

[5] Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D. (2003). Bayesian Data Analysis (2nd ed.). CRC Press.

[6] McNeil, A. J., Frey, R., & Embrechts, P. (2005). Quantitative Risk Management: Concepts, Techniques, and Tools. Princeton University Press.

[7] Koenker, R. (2005). Quantile Regression. Cambridge University Press.

[8] Li, C., et al. (2016). Quantile regression for modeling hospital readmission rates. Health Services Research, 51(4), 1293–1309.

[9] Roth, J., et al. (2016). Robust quantile regression: A comparative study. Computational Statistics & Data Analysis, 95, 78–94.

[10] Zhang, Y., et al. (2020). Lagrangian asymmetric-vTwin support vector regression for quantile prediction. Journal of Machine Learning Research, 21(1), 349–368.

[11] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[12] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2013). Algorithms for hyper-parameter optimization. Proceedings of the 24th International Conference on Neural Information Processing Systems, 2546–2554.

[13] He, X., & Garcia, E. (2009). Quantile regression for predicting house prices. Journal of Real Estate Finance and Economics, 39(3), 354–375.

[14] Fang, K., et al. (2017). Support vector regression and its applications. International Journal of Computer Applications, 163(1), 35–42.

[15] Yu, K., & Lu, L. (2006). Quantile regression with SVM. Proceedings of the International Conference on Machine Learning, 1081–1088.

[16] Hall, P., & Jing, B. (1993). On the asymptotic normality of quantile regression estimates. Journal of Econometrics, 58(3), 315–336.

[17] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.

[18] Fan, J., & Li, R. (2006). Statistical challenges with high dimensionality: Feature selection in knowledge discovery. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(1), 1–44.

[19] Xie, Y., & Liu, F. (2011). Quantile regression methods for medical data. Statistics in Medicine, 30(8), 920–931.

[20] Huang, Y., & Xie, W. (2013). Support vector regression with quantile loss function for robust regression. Journal of Computational and Graphical Statistics, 22(2), 483–499.

INTERNATIONAL CONFERENCE ON ADVANCES IN SCIENCE, ENGINEERING, MANAGEMENT AND HUMANITIES (ICASEMH – 2023)

26TH FEBRUARY, 2023

CERTIFICATE NO : ICASEMH /2023/C0223245

A STUDY ON IMPLICIT LAGRANGIAN TWIN EXTREME LEARNING MACHINES IN PRIMAL FOR PATTERN CLASSIFICATION

V Rajanikanth Tatiraju

Research Scholar, Ph. D. in Computer Science Engineering
P. K. University, Shivpuri, M P

ABSTRACT

The Implicit Lagrangian Twin Extreme Learning Machine (ILTELM) is a novel advancement in machine learning, designed for efficient and accurate pattern classification. Unlike traditional methods, ILTELM operates in the primal space, utilizing the concept of twin hyperplanes to classify data into distinct classes. This approach integrates the strengths of Extreme Learning Machines (ELMs) with implicit Lagrangian formulations, providing a robust framework for solving classification problems. In ILTELM, the primal optimization framework directly handles the input data, eliminating the need for dual formulations. This results in reduced computational complexity and faster processing. The implicit Lagrangian method ensures that optimization constraints are satisfied while minimizing the objective function, enhancing model stability and generalization. The twin hyperplane strategy further divides the input space into two regions, maximizing the margin for improved classification accuracy. Additionally, the ELM architecture, characterized by random feature mapping and minimal parameter tuning, allows ILTELM to handle high-dimensional datasets effectively. Its ability to work with non-linear and complex data patterns makes it suitable for diverse applications, including image recognition, bioinformatics, and text classification. The ILTELM in primal demonstrates superior performance due to its computational efficiency, scalability, and robust classification capabilities, marking a significant contribution to modern pattern recognition techniques.

IARF Conferences

International Conference on Advances in Science, Engineering, Management and Humanities (ICASEMH- 2023)

ICASEMH/2023/C0223245

CERTIFICATE

This is to Certify that V Rajanikanth Tatiraju, Research scholar, Ph.D. in computer Science Engineering, P.K. University, Shivpuri, M.P.

has presented a paper on A study on Implicit Lagrangian Twin Extreme learning Machines in Primal for Pattern classification.

at **International Conference on Advances in Science, Engineering, Management and Humanities (ICASEMH - 2023)** organized by the **International Academic Research Forum (IARF)** at Hotel Siddhartha Inn, Gandhibagh, Nagpur, Maharashtra, India. On 26th February, 2023

Dr. Rayess Ahmad Dar
Organizing Secretary

IARF CONFERENCES IARF

Dr. Shabnam Arora
Convener

National Conference on Recent Advances in Science, Engineering, Humanities, and Management (NCRASETHM - 2024)
28th January, 2024, Banquet, Noida, India.

CERTIFICATE NO : NCRASETHM /2024/C0124155

A STUDY OF FUNCTIONAL ITERATIVE APPROACHES FOR TWIN BOUNDED SUPPORT VECTOR MACHINES WITH SQUARED PINBALL LOSS

V Rajanikanth Tatiraju

Research Scholar, Ph. D. in Computer Science Engineering
P. K. University, Shivpuri, M P

ABSTRACT

Twin Bounded Support Vector Machines (TBSVMs) have emerged as an effective machine learning tool, particularly in handling classification problems. By simultaneously solving two smaller quadratic programming problems, TBSVMs are computationally more efficient compared to traditional Support Vector Machines (SVMs). The incorporation of a squared pinball loss function into TBSVMs introduces further robustness by accommodating asymmetric noise distributions and better handling of misclassified data. This combination enhances model performance, especially in real-world scenarios with imbalanced or noisy datasets. Functional iterative approaches play a pivotal role in optimizing TBSVMs with squared pinball loss. These iterative methods aim to minimize the modified loss function while adhering to constraints that define the twin hyperplanes. The squared pinball loss, as a convex loss function, penalizes deviations based on their magnitude, ensuring more precise adjustments during iterations. Iterative algorithms refine hyperplane placement, effectively balancing the trade-off between accuracy and generalization. Additionally, functional iterative schemes enhance computational efficiency by breaking down the optimization into manageable steps. Advanced methods like gradient-based techniques and alternating minimization algorithms further accelerate convergence. These approaches also facilitate scalability, enabling TBSVMs to handle high-dimensional and large-scale datasets. Overall, iterative optimization with squared pinball loss broadens TBSVMs' applicability across complex classification tasks.

IJESTI CONFERENCES

NCRASETHM/2024/C0124155

National Conference on Recent Advances in Science, Engineering, Humanities and Management
(NCRASETHM-2024)

CERTIFICATE

This is to certify that V Rajanikanth Tatiraju, Research Scholar, Ph.D in Computer Science Engineering, P.K. University, Shivpuri, M.P.

has presented a paper in **NCRASETHM-2024** held on 28th January 2024 at Hotel Krishna Residency & Banquet, Noida, India.

Title of the paper A Study of Functional Iterative Approaches for Twin Bounded Support Vector Machines with Squared Pinball Loss.

Dr. Syed Imtiyaz
convenor

IJESTI Conferences

Dr. N. V. Krishna Prasad
Conference Chair

PROCEEDINGS

Of

8th International Conference on Advanced Logical Learning and Analytical Mining

(ALLAM-2024)

28th December 2024

Jointly organized

By

Institute of Bioinformatics and Computational Biology

Department of Information Technology, Andhra University

Prof. CHSN Research Foundation

Preface

The 8th International Conference on Advanced Logical Learning and Analytical Mining (ALLAM-2024) is aimed to bring researchers together working in this area to share their knowledge and experience. In this conference, topics of contemporary interested would be discussed to provide a holistic vision on latest technologies for computer science and engineering. The scope includes data science, machine learning, computer vision, deep learning, artificial intelligence, artificial neural networks, mobile applications development and Internet of Things etc; Conference participants are expected to gain relevant knowledge and better understanding of the applications of computer science in various fields.

ALLAM-2024 would be both stimulating and informative with the active participation of galaxy of keynote speakers. We would like to thank all the authors who submitted the papers, because of which the conference became a story of success. We also would like to express our gratitude to the reviewers, for their contributions to enhance the quality of the papers. We are very grateful to the Keynote Speakers, Reviewers, Session Chairs and Committee members who selflessly contributed to the success of ALLAM-2024. We are very thankful to Andhra University, Visakhapatnam for providing the basic requirements to host the ALLAM-2024.

Last but not the least, we are thankful for the enormous support of publishing partner i.e. Springer for supporting us in every step of our journey towards success.

Dr.P.Sateesh

Organizing Chair, ALLAM-2024

Kernel-Optimized Surface Learning: Enhancing Classification with Support Vector Machines and Gaussian Process Kernels

Er.Tatiraju.V.Rajani Kanth
Research Scholar,Department of Computer Science Engineering , P.K.University,Shivpuri, Madhya Pradesh, India tvrajani55@gmail.com
Dr.Balveer Singh
Professor, Department of Computer Science Engineering, P.K.University, Shivpuril, Madhya Pradesh, India. adm.pkit@gmail.com
Dr.Yaspal Singh
Professor, Department of Computer Science Engineering, P.K.University, Shivpuril, Madhya Pradesh, India. ypsingh10@rediff.com
Dr.Sunil Bhutada
Associate Professor,Department of Computer Science Engineering, P.K.University, Shivpuril, Madhya Pradesh, India. sunilbhutada@gmail.com
Dr.Rohita Yamaganti,
Associate Professor,Department of Computer Science Engineering, P.K.University, Shivpuril, Madhya Pradesh, India rohita.yamaganti@gmail.com

Abstract

Classification tasks in machine learning often face challenges in balancing accuracy, computational efficiency, and scalability. This study introduces a novel Kernel-Optimized Surface Learning (KOSL) technique that leverages Support Vector Machines (SVM) and Gaussian Process Kernels to generate optimal decision boundaries for high-dimensional data. The proposed approach incorporates a hybrid optimization strategy combining Particle Swarm Optimization (PSO) and Grid Search to fine-tune kernel parameters, ensuring maximum classification performance across diverse datasets. Experimental evaluations were conducted on benchmark datasets, including MNIST, CIFAR-10, and UCI Machine Learning Repository datasets, with varying feature dimensions and class distributions. The results demonstrate that the proposed KOSL technique achieved: Accuracy: 98.6% on MNIST, 91.2% on CIFAR-10, and an average of 96.4% across UCI datasets. F1-Score: 0.97 (MNIST), 0.89 (CIFAR-10), and 0.94 (UCI). Training Time Reduction: 28% compared to standard SVM with Radial Basis Function (RBF) kernels. Additionally, the KOSL framework exhibited enhanced robustness against noisy and imbalanced datasets, outperforming conventional models by 15% in classification accuracy on skewed data distributions. This work highlights the potential of combining advanced kernel optimization techniques with traditional machine learning models to address classification challenges effectively. Future research will explore the integration of KOSL with deep learning architectures for more complex, real-world applications.

Keywords: Kernel Optimization, Support Vector Machines, Gaussian Process Kernels, Particle Swarm Optimization, Classification Challenges, Hybrid Optimization, HighDimensional Data, Benchmark Datasets, Accuracy Improvement, Computational Efficiency.

8th International Conference on

ADVANCED LOGICAL LEARNING AND ANALYTICAL MINING (ALLAM) IN COGNITIVE SCIENCE, 28th December 2024

Organised by

Institute of Bioinformatics and Computational Biology (IBCB), Visakhapatnam

(Recognized as SIRO by Department of Science & Technology, Government of India)

Certificate

This is to certify that Dr/Mr/Mrs/Miss Er. Totiraju. V. Rajani Kanth

has presented a paper titled Kernel-Optimized Surface Learning: Enhancing Classification with Support Vector Machines And Gaussian Process Kernels *at the International conference on Advanced Logical Learning and Analytical Mining (ALLAM) in Cognitive Science during 28th December 2024 held at Andhra University, Visakhapatnam, A.P., India.*

Dr G. Lavanya Devi
Program Chair

Dr K. NageswarRao
Program Chair

Dr Ch Divakar
Secretary, IBCB
Convener

Adaptive Kernel Optimization for Probabilistic Learning: Integrating Support Vector Machines with Gaussian Process Frameworks

Abstract

With its robust capabilities for non-linear regression and classification, kernel-based learning has emerged as a fundamental component of state-of-the-art machine learning approaches. In order to improve probabilistic learning, this study investigates Adaptive Kernel Optimization (AKO), a new method that combines the best features of the Support Vector Machine (SVM) and the Gaussian Process (GP) frameworks. Achieving better flexibility in modeling complicated data distributions while keeping computational efficiency is achieved by employing adaptive kernel functions in the suggested strategy. Quantifying uncertainty in addition to deterministic SVM classifications is made possible with the incorporation of GP kernels, which offer probabilistic insights. The suggested approach guarantees resilience across varied and high-dimensional datasets by dynamically adjusting kernel parameters according to data properties. Extensive testing on benchmark datasets shows that, in comparison to conventional SVM and GP approaches, our model generalizability, classification accuracy, and interpretability are much improved. Autonomous systems, healthcare diagnostics, and financial sectors can all benefit from the scalable, adaptive, and probabilistic learning models that this study establishes.

Keywords— Adaptive Kernel Optimization, Probabilistic Learning, Support Vector Machines (SVM), Gaussian Process Kernels, Uncertainty Quantification, Non-linear Classification, High-Dimensional Data,

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's, Kalaburagi

LINGRAJ APPA ENGINEERING COLLEGE, BIDAR

Approved by AICTE and Affiliated to VTU Belagavi.

IEEE BANGALORE SECTION

Certificate of Appreciation

IEEE NKSS

This is to certify that Dr./Prof./Mr./Ms

Er.Tatiraju.V.Rajani Kanth

has **Presented** Paper entitled

Adaptive Kernel Optimization for Probabilistic Learning: Integrating Support Vector Machines with Gaussian Process Frameworks

for International Conference on

Intelligent Systems And Computational Networks (ICISCN- 2025)

organized by Lingraj Appa Engineering College, Bidar, 24-25 January, 2025

Vinita

Dr. VINITA PATIL
Principal,
Lingaraj Appa Engineering College, Bidar

B.S. Deshmukh

Sri. BASAVARAJ DESHMUKH
Secretary, Sharanbasaveshwar Vidya Vardhak Sangha, Kalaburagi
Member BoG, Sharanbasva University, Kalaburagi

Sharnbaswappa Appa

Parama Poojya Dr. SHARNBASWAPPA APPAJI
President
Sharanbasaveshwar Vidya Vardhak Sangha, Kalaburagi

पेटेंट कार्यालय
शासकीय जर्नल

OFFICIAL JOURNAL
OF
THE PATENT OFFICE

निर्गमन सं. 03/2025	शुक्रवार	दिनांक: 17/01/2025
ISSUE NO. 03/2025	FRIDAY	DATE: 17/01/2025

पेटेंट कार्यालय का एक प्रकाशन
PUBLICATION OF THE PATENT OFFICE

(12) PATENT APPLICATION PUBLICATION
(19) INDIA
(22) Date of filing of Application :12/12/2024
(21) Application No.202421098494 A
(43) Publication Date : 17/01/2025

(54) Title of the invention : Systems for Enhancing Machine Learning Models Using Target Distributions of Key Performance Indicators in Cloud Networks

(51) International classification	:G06N0020000000, G06F0009500000, H04L0041500900, H04L0067100000, H04L0041400000
(86) International Application No Filing Date	:NA :NA
(87) International Publication No	: NA
(61) Patent of Addition to Application Number Filing Date	:NA :NA
(62) Divisional to Application Number Filing Date	:NA :NA

(71)**Name of Applicant :**
1)Er. Tatiraju. V. Rajani Kanth
Address of Applicant :Research Scholar, Department of Computer Science Engineering, P. K. University, Shivpuri, Madhya Pradesh India–473665 ---------- ----------
2)Dr. Balveer Singh
3)Dr. Sunil Bhutada
4)Dr. Rohita Yamaganti
5)Prof. Bhaskar Nalla
Name of Applicant : NA
Address of Applicant : NA
(72)**Name of Inventor :**
1)Er. Tatiraju. V. Rajani Kanth
Address of Applicant :Research Scholar, Department of Computer Science Engineering, P. K. University, Shivpuri, Madhya Pradesh India–473665 ---------- ----------
2)Dr. Balveer Singh
Address of Applicant :Professor, Faculty of Engineering & Technology, Department of Computer Science Engineering, P.K. University, Shivpuri Madhya Pradesh, India - 473665 ---------- ----------
3)Dr. Sunil Bhutada
Address of Applicant :Associate Professor, Department of Computer Science Engineering, P. K. University, Shivpuri, Madhya Pradesh India–473665 ---------- ----------
4)Dr. Rohita Yamaganti
Address of Applicant :Associate Professor, Department of Computer Science Engineering, P.K. University, Shivpuri, Madhya Pradesh India– 473665 ---------- ----------
5)Prof. Bhaskar Nalla
Address of Applicant :Dean, Faculty of Management, P.K. University, Shivpuri Madhya Pradesh, India - 473665 ---------- ----------

(57) Abstract :
The invention provides a system and method for enhancing machine learning (ML) models using target distributions of key performance indicators (KPIs) in cloud networks. The system comprises a KPI monitoring module to collect real-time performance metrics, a target distribution analyzer to define and compare ideal KPI distributions against observed data, a machine learning model augmentation engine to dynamically adjust model parameters using advanced techniques like reinforcement learning and transfer learning, and a feedback loop to ensure continuous refinement of the model. The system is designed for seamless integration into public, private, and hybrid cloud environments, supporting scalability and adaptability in dynamic operational conditions. By aligning ML model predictions with real-world KPI distributions, the invention optimizes cloud resource utilization, enhances model accuracy, and reduces operational inefficiencies. Applications include cloud resource management, network monitoring, anomaly detection, and real-time analytics, making it a versatile solution for enterprises leveraging cloud technologies.

No. of Pages : 17 No. of Claims : 10