Support Vector Machines: Theory, Algorithms, and Applications

Mohammed Jabardi

Abstract—**Support Vector Machines, or SVMs, are a strong group of supervised learning models that are commonly used for tasks like regression and classification. SVMs are based on the theory of statistical learning and try to find the best hyperplane that maximizes the gap between different classes. This makes it easier to apply to new data. Since kernel functions are used with SVMs, they are more flexible and can handle both linear and nonlinear situations well. Even though they have a strong theoretical base, they still face problems in the real world, like being hard to code and difficult to tune parameters, especially for big datasets. Recent improvements, like scalable solvers and estimated kernel methods, have made them a lot more useful. This essay talks about SVM theory, its main algorithms, and how it is used in the real world. It shows how it is used in bioinformatics, banking, and image processing, among other areas.**

Index Terms—**Support Vector Machine, Classification, Regression, Machine Learning, Hyperplanes, Kernel Functions.**

I. Introduction

Support Vector Machine (SV), is one of the simplest and most refined classification methods in machine learning. Unlike neural networks, SVMs can work with very small datasets and are not inclined to overfitting. The SVM is used to classify each object by representing points in an N-dimensional space and the coordinates of these points, which are usually called features. [2]

SVM perform the classification procedure by drawing a hyperplane that is a line in 2 or 3 dimensional in a plane such a way that all points of one category are on one side of the hyperplane and all points of other categories are on the other side. If there are multiple hyperplanes, SVM try to find the one that best separate the two categories, in the sense that maximizes the distance to points in either category [3-4]. This distance called the Margin and all the points fall exactly on the margin are called the Supporting Vectors. To find the hyperplane in the first place the SVM requires for training set or set of points that already labeled with the correct category, this is why SVM is said to be supervised learning algorithm. In the background SVM solve a convex optimization problem that maximize this margin and where constraints say that points for each category should be fall in the correct side of the hyperplane. [5-6]

While it's mainly used for binary classification, SVM can also handle multiclass problems by using strategies like one-vs-all (comparing one class against all others) or one-vs-one (building a classifier for each pair of classes [7]. Originally presented by Vapnik, SVMs are well-known for their kernel-based approach to classification and regression tasks [8–10]. In data mining, pattern recognition, and machine learning, their remarkable generalization capacity, optimal solutions, and discriminative power have attracted plenty of attention. Originally presented by Vapnik, SVM is well-known for its kernel-based method of handling regression and classification problems [8–10]. The data mining, pattern recognition, and machine learning groups in recent years have shown great interest in its exceptional generalization capacities, optimal solutions, and discriminative capability [11]. To maximize the separation margin in a high-dimensional feature space, SVMs optimize decision functions directly from training data [12–18]. This strategy not only minimizes training data errors but also improves generalization abilities. The support vector machine algorithm or SVM it looks at the extremes of the data sets and draws a decision boundary also known as a hyperplane near the extreme points in the data set so essentially the support vector machine algorithm is a frontier which best segregates the two classes. [19-20].

M. Jabardi is with Department of Software, College of Education, University of Kufa, Najaf, Iraq. (E-mail: mohammed.naji@uokufa.edu.iq)

DOI: 10.36244/ICJ.2025.1.8

A. *SVM Historical Perspective and their Evolution.*

Vladimir Vapnik and Alexey Chervonenkis's landmark paper, "A Theory of Learning from General Examples" (1964), laid the foundation for statistical learning theory. It emphasized minimizing generalization error rather than training error, a principle central to SVMs [21]. Vapnik and Boser further developed SVMs in their 1992 paper, "Pattern Recognition Using an Insensitive Loss Function," where they detailed classical SVM algorithms for binary classification and introduced the concept of support vectors [22]. The development of kernel methods by Christopher J.C. Burges in "A Tutorial on Support Vector Machines for Pattern Recognition" (1998) enabled SVMs to effectively handle non-linearly separable data [23].

B. *Application of SVN*

The late 1990s and early 2000s saw the development of robust SVM software libraries like LIBSVM and SMO, making them readily accessible to practitioners. This accessibility sparked a surge in SVM applications across various fields, including [24-25]:

- Text classification: Spam filtering, sentiment analysis, topic modeling.
- Image classification and object detection: Handwritten digit recognition, object detection, image segmentation.
- Bioinformatics and computational biology: Gene classification, protein analysis, disease prediction.
- Financial forecasting: Stock market prediction, credit risk assessment.

- Healthcare and medical diagnosis
- Cybersecurity and intrusion detection
- Environmental science and climate modeling.

C. SVMs Advantages and Disadvantages [26-28]:

Advantages:

- Particularly successful in high-dimensional environments.
- Memory effective: Makes only use of a subset of the training points.
- flexible: the kernel method helps here.
- Scalability and efficiency: advances in large-scale SVMs and mass dataset handling optimizing techniques.
- Multi-class and multi-label classification stretches SVMs beyond binary classification to address intricate data structures.
- Designs new kernel functions and investigates adaptive kernels automatically that learn from data automatically in kernel learning and adaptation.
- This method minimizes generalization error and improves performance with fresh data by concentrating on margin maximizing and capacity control.
- Robustness: Greater robustness results from larger margins lowering susceptibility to data changes or noise. learning models.

Disadvantages:

- Not appropriate for big datasets: Long training times can be problematic.
- Sensitive to kernel and hyperparameter decisions: These can greatly affect results.
- Difficult Interpretability: Complicating knowledge are complex elements including high-dimensional decision limits and kernel modifications.
- Handling big datasets might result in major computational expenses and increase training times, hence computationally demanding.
- Correct parameter settings define performance; inadequate calibration can produce less than ideal results.
- Lack probabilistic outcomes: procedures like Platt scaling are required for probabilities; SVMs mostly produce binary classifications without direct probability estimations.
- Understanding complex models is challenging: Particularly with nonlinear kernels, intricate decision boundaries complicate models for interpretation.
- Scalability problems: Memory and computing restrictions can make training on very big datasets unworkable.

II. Recent Review Articles and Surveys on SVMs

TABLE I
THE SUMMARY OF THE LITERATURE REVIEW

Reference	Description
[25-28]	Introduces machine learning, focusing on supervised learning and SVMs. Explores SVM capabilities, applications, and future prospects.
[29]	SVM for two-class classification, kernels, and penalty functions, furthermore covering multiclass methods, one-class SVDD, and Support Vector Regression for handling outliers and non-linear data.
[30-31]	Provides an overview of SVM applications, challenges, and emerging trends, highlighting their utility in various fields.
[32]	Comprehensive review of SVMs, covering fundamental concepts, kernel methods, optimization algorithms, and applications.
[33]	Surveys SVM applications in bioinformatics, healthcare, finance, image processing, and natural language processing.
[34]	Focuses on interpretable SVMs, covering techniques like rule extraction, feature importance analysis, and model-agnostic methods.
[35]	Reviews challenges and solutions for large-scale SVM training, including stochastic gradient descent and distributed computing.
[36]	Explores hybrid models combining SVMs with deep learning to improve performance and address individual limitations.
[37]	Highlights emerging SVM applications in bioinformatics, healthcare, finance, and natural language processing.
[38]	Develops an automated facial expression recognition system using SVM, MLP, and KNN classifiers with HOG and PCA for feature extraction.
[39]	Examines linear SVM classification, focusing on solvers, improvements, empirical findings, and future research directions.
[40]	Demonstrates SVM's effectiveness in predicting Alzheimer's disease using MRI data, emphasizing its potential in medical diagnosis.
[41]	Proposes a bagged ensemble SVM technique for speech emotion recognition, contributing to Human-Computer Interaction research.
[42]	Introduces an SVM-based intrusion detection framework with naive Bayes feature embedding, improving network security.

[43]	Presents a U-Net-based method for melanoma classification in dermoscopy images, using segmentation, feature extraction, and SVM.
[44]	Combines deep neural networks (DNN) and multiclass SVMs for classification, using K-means clustering for feature extraction.
[45]	Enhances SVM classification capabilities by incorporating dynamic graph learning and self-paced learning.
[46]	Highlights SVM's role in interpreting neuroimaging data for brain disorder research and precision psychiatry.
[47]	Proposes a CNN-SVM hybrid model for diagnosing faults in rotating machinery, improving early-stage fault detection.
[48]	Combines deep learning and SVM for identifying and predicting rice leaf diseases.
[49]	Introduces a method for detecting malaria parasites using deep neural networks and SVM with transfer learning.
[50]	Explores SVM's role in image classification, discussing its evolution, variants, and applications.
[51]	Uses PSO, GA, and Grid Search to optimize SVM parameters for risk assessment in railway transportation systems.
[52]	Addresses factors affecting SVM performance in classifying nonlinearly separable problems, providing insights for future research.
[53]	Proposes a deep learning method for breast cancer detection using mammography, combining DNN and multiclass SVM.
[54]	Introduces DeepSVM-fold, a computational predictor for protein fold recognition, offering improved accuracy over existing methods.

III. METHODOLOGY

One of the most well-known supervised learning methods is Support Vector Machine (SVM). Its main job is to sort things into groups, but it can also help with error problems in machine learning [24]. In n-dimensional space, the SVM algorithm tries to find the best line or decision boundary that splits it into classes. This makes it easy to put new data points into the right category. A hyperplane is the name for this best border. SVM finds the most important extreme points or vectors for defining this hyperplane [25–26].

The Core Concepts [27-30]

Support Vector Machines (SVMs) aim to identify a hyperplane with the largest possible margin, resulting in a robust classification model. The mathematical method maximizes the squared norm of the weight vector under constraints guaranteeing class separation. Lagrange multipliers help to simplify this optimization issue. Crucially important data points defining the decision-making range are support vectors. Then, depending on their feature vectors, a decision function groups newly occurring data points.

Important notes:

Hyperplanes are data point classification boundary. Points on opposing sides of the hyperplane fall into several categories. The number of features determines the hyperplane's dimension; for instance, a hyperplane with two features is a line and with three it becomes a plane.

Margins: The margin is the distance between the hyperplane and the closest data points, known as support vectors. This distance can be mathematically expressed as: $2/(\|w\|)$. The Euclidean norm of the weight vector w is denoted as $\|w\|$.

Maximizing Margins: SVMs strive to find the hyperplane with the widest margin, enhancing the classifier's generalization capabilities.

Regularization: This technique helps prevent overfitting in SVMs by introducing a penalty term in the objective function, which encourages the model to prefer simpler decision boundaries over complex ones that perfectly fit the training data.

Support Vectors: These are data points close to the hyperplane that significantly influence its position and orientation and are essential for constructing the SVM

Support Vectors: These are data points close to the hyperplane that significantly influence its position and orientation and are essential for constructing the SVM. Altering these points would shift the hyperplane, as

$\vec{x}.\vec{w} = c$ (the point lies on the decision boundary)
$\vec{x}.\vec{w} > c$ (positive samples)
$\vec{x}.\vec{w} < c$ (negative samples)

illustrated in Figure 1.

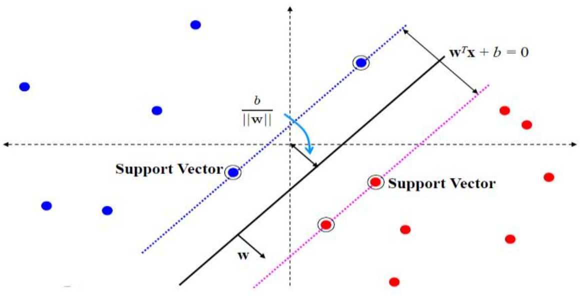

Fig. 1. Support vectors

Kernel Functions: These functions transform the input data into a higher-dimensional feature space, making it easier for a linear classifier to separate the data. Kernels help capture complex, nonlinear patterns like curves and circles. Common types include linear, polynomial, radial basis function (RBF), and sigmoid. There are two different categories that are classified using a decision boundary or hyperplane as shown in Figure. 2.

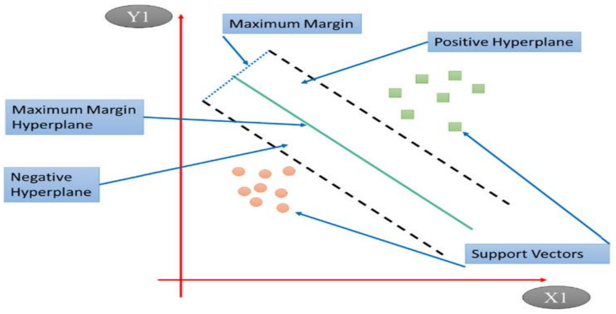

Fig. 2. The hyperplanes used to classify data points [30]

Consider a random point X, and determine whether it is above or below the hyperplane, or on it, as shown in Figure 3. First, represent X as a vector. Then, construct a vector (w) perpendicular to the hyperplane. Suppose C is the distance from the origin to the decision boundary along (w). Project X onto (w) through a dot product. If the dot product exceeds C, X is above the plane; if less, below; if equal, on the decision boundary [31].

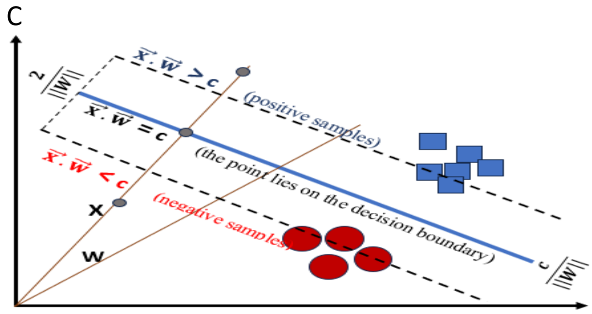

Fig. 3. The core concept of the SVM algorithm

A. *Types of Support Vector Machine*

Based on the nature of the decision boundary, Support Vector Machines (SVM) can be divided into two main parts as shown in Figure 1:
Support Vector Machines (SVM) can be categorized into two main types based on the nature of the decision boundary

1. Linear SVM:
   This type is applicable when the data is perfectly linearly separable. This means that the data points can be divided into two classes using a single straight line in a two-dimensional space, as shown in Figure 4.A.
2. Non-Linear SVM:
   When the data is not linearly separable, a Non-Linear SVM is used. This occurs when the data points cannot be separated into two classes by a straight line. In such cases, advanced techniques like the kernel trick are employed to classify the data. Most real-world applications involve non-linearly separable data, hence the kernel trick is commonly used, as depicted in Figure 4.B.

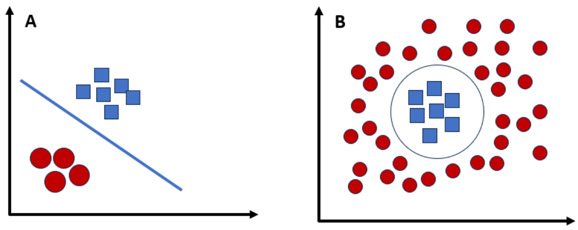

Fig.4. **A:** Linearly Separable Data **B:** Non-Linearly Separable Data

B. *Mathematical Formulation*

1. **Linear SVM**:
   This type is applicable when the data is perfectly linearly separable. This means that the data points can be divided into two classes using a single straight line in a two-dimensional space, as shown in Figure 4.A.
   1.1. Data: a dataset of points ({Xi}, {Yi}), where Xi is an n-dimensional vector representing the features of a data point and Yi is the class label (+1 or -1).
   1.2. Hyperplane Equation: The hyperplane is defined as:
   $W_T \times X + b = 0$ .... (1)
   Where W is a weight vector normal (perpendicular) to the hyperplane X is an input vector b is the bias term
   1.3. Constraints: For a data point to be correctly classified, we need:
   $Yi\ (W_T \times X_i + b) \geq 1$ for $Y_i = +1$ ...(2)
   $Yi\ (W_T \times X_i + b) \leq -1$ for $Y_i = -1$ ...(3)
   1.4. Optimization Problem: Maximizing the margin is equivalent to minimizing the following objective function: $W = \frac{||W^2||}{2}$ (subject to the constraints above). This is a quadratic optimization problem, usually solved using techniques like Lagrange multipliers.

2. **Non-Linear SVMs (The Kernel functions)**:
   The popular kernel types that we can use transform the data into high dimensional feature space are polynomial kernel, radial basis function RPF or RBF kernel and sigmoid kernel [32]. Choosing the correct kernel is a non- trivial task and may depend on specific task at hand no matter which kernel we choose, just we need to tune the kernel parameters to get good performance from a classifier. A popular parameter tuning technique includes k-fold cross-validation. Some of the most common kernel functions for support vector machines include:
   2.1. The Linear Kernel:
   The linear kernel, or dot product kernel, is the simplest function. It calculates the dot product of input feature vectors in the original input space. Mathematically, it is expressed as in equation (4).
   $K(xi, xj) = xiT \times xj$ .... (4)

Advantages:

- Efficiency: The linear kernel excels in computational efficiency, involving only a simple dot product operation, making it suitable for high-dimensional data where other kernels might become computationally expensive.
- Interpretability: It offers the highest level of interpretability among kernel functions. The decision boundary learned by the SVM is a hyperplane in the original feature space, and the weights assigned to each feature reveal their relative importance for classification.
- No Hyperparameter Tuning: Unlike most other kernels, the linear kernel requires no hyperparameter tuning, making it easier to use and reducing the risk of overfitting due to poorly chosen hyperparameter.

Limitations:

- Limited to Linearly Separable Data: Its main limitation is that it can only handle data that is already linearly separable in the original feature space. If the data exhibits complex, non-linear relationships, the linear kernel will not effectively learn a separation boundary.
- Less Flexible: Due to its simplicity, the linear kernel is less flexible in modeling complex non-linear patterns compared to kernels like polynomial or RBF.

2.2. Polynomial Kernel:

The polynomial kernel calculates the similarity between two vectors by raising the dot product of the original vectors to a given power d, adding non-linearity to the decision boundary. Mathematically, it is expressed as in equation (5).

$K(xi, xj) = (xiT \times xj + 1)d$ .... (5)

2.3. Radial Basis Function (RBF) Kernel:

A lot of users work with the RBF kernel, which is also known as the Gaussian kernel. The Gaussian distribution is used to measure the distance between two vectors in the feature space to find out how close they are. This is helpful when there isn't a clear line between the input data. It can be written mathematically as shown in equation (6). If you change the hyperparameter gamma, it changes how wide the Gaussian distribution is. Radial Basis Function (RBF) kernel is a popular and flexible choice for SVMs that work with data that is not linear.

$K(x_i, x_j) = e^{(-gamma \, \| K(Xi, Xj) \|2)}$ ……. (6)

Advantages:

- Useful for Non-Linear Data: The RBF kernel is great at turning data into a higher-dimensional feature space by using a Gaussian function. This change makes it possible for SVMs to see complicated, non-linear connections between traits that weren't possible in the original space.
- Stability: The RBF kernel is less likely to suffer from the curse of dimensionality than the polynomial kernel. It works well with data that has a lot of dimensions and doesn't have the same risk of overfitting.
- Fewer Hyperparameters: The RBF kernel only has one hyperparameter, called gamma. However, it is not as sensitive to setting hyperparameters as the degree parameter of the polynomial kernel. This can make the process of choosing a model easier.

Limitations:

- Interpretability: Like most non-linear kernels, the RBF kernel sacrifices some interpretability compared to the linear kernel. The decision boundary becomes less intuitive in the original feature space.
- Computational Cost: Although generally more efficient than the polynomial kernel for high dimensions, the RBF kernel can still be computationally expensive, especially for very large datasets.
- Hyperparameter Tuning: While less sensitive than the polynomial kernel, the RBF kernel's performance still depends on finding the optimal gamma value. Careful hyperparameter tuning is essential.

2.4. Optimal Kernel Selection:

An Empirical Methodology The effectiveness of SVMs relies on carefully choosing the right kernel function that suits the specific challenge. Here's an analysis of typical options and their practical applications:

1- The linear kernel:

Situation: Consider classifying emails as either spam or legitimate. Possible features include word frequency and the presence of spam keywords. The linear kernel is a good initial choice because the relationship between these features is likely linear (a higher frequency of spam keywords often indicates spam). This model is computationally efficient and interpretable, with the decision boundary being a straight line in the original feature space.

2- The polynomial kernel:

This is used when assessing handwritten digits for recognition. Pixels can serve as features, but defining a clear decision boundary to separate different digits (such as 6 and 8) with a linear plane is often challenging. A low-degree polynomial kernel, such as quadratic, can introduce non-linearity, enabling the SVM to more accurately capture the curved features of some digits. However, using a high-degree polynomial kernel may lead to overfitting, underscoring the need for careful parameter tuning.

2.5. The Radial Basis Function (RBF)

Situation: Categorize photographs featuring various species of animals, such as cats, dogs, and birds. Pixel intensities and local features are used. However, it should be noted that the relationships between them exhibit a high degree of nonlinearity. The RBF kernel performs exceptionally well in this context. It turns input points into a space with an unlimited number of dimensions. This lets the SVM find complex spatial patterns, like the edges and textures that are unique to each species. The parameter (σ) controls how smooth the decision limit is. A higher π number makes the transition smoother, which could group animals that look alike, like all cats, even if they are in different positions. A lower number of σ allows for more complex differences, which could make it easier to tell the difference between different breeds.

2.6. SVMs Constraints and Difficulties [25-27]:

- It can be hard to work with very large datasets because SVMs can require a lot of processing power and memory, especially when working with very large datasets that are high dimensional.
- Choosing the Kernel Function: Picking the correct kernel function and its values has a big impact on how well SVM works. Finding the best kernel and tweaking its settings, on the other hand, can be hard and needs specialized knowledge.
- Noise Sensitivity: SVMs have great dataset outlier and noise sensitivity, overfitting can result from this sensitivity upsetting the decision limit. Sometimes preprocessing methods like eliminating outliers and lowering noise are required.
- Binary Focus: SVMs are mostly meant for binary classification tasks, so multiclassification becomes difficult. Usually, they are adapted for multiclassification utilizing one-vs- one or one-vs- all approaches. These techniques, however, can have problems with scalability and lower performance, hence stressing SVMs' shortcomings.
- Lack of Interpretability: SVMs generate a black-box model that makes it challenging to grasp the learned decision bounds and the fundamental data linkages. In sectors such banking and healthcare, interpretability is absolutely vital.
- Unbalanced Datasets: SVMs may perform badly in imbalanced datasets, in which case cases across classes vary significantly. To solve this, one could need different evaluation measures, resampling, or class weighting.

Despite these challenges, SVMs remain a robust and widely used approach in machine learning for tasks like classification, regression, and anomaly detection. Researchers continue to explore ways to overcome these obstacles and improve the efficiency and scalability of SVMs in various fields.

2.7. Why SVMs Are Computationally Expensive

The following causes make the SVMs computationally expensive:

1. **Kernel Computations:** SVMs calculate the kernel matrix when employing non-linear kernels; this structure has a size of (n × n), so it is not viable for high n.
2. **Dense Data Visualizations:** The computational cost rises significantly for extremely dimensional or sparse data such text data.
3. **Memory Requirement:** Large datasets are blocked by storing the intermediate optimization variables and kernel matrix using a lot of RAM.
4. **Time Complexity:** The time complexity of solving the quadratic optimization problem in SVMs is typically within range O(n^2) and O(n^3), where n is the number of training samples.

2.8. SVM and large dataset.

Using SVM with large datasets can be challenging due to computational complexity and memory requirements. The SGD-based SVMs, GPU acceleration, parallelization, Dimensionality reduction, and feature selection methods are utilized to overcome the scalability of SVMs for large datasets. The summary of these methods is summarized in Table II.

TABLE II
METHODS FOR OVERCOMING THE SCALABILITY OF SVMS FOR LARGE DATASETS

Methods	Approach	Advantages	Trade-offs
SGD-Based SVMs	Incremental optimization using small batches.	scalability and efficiency.	Noisy updates, suboptimal solutions
Parallelization	Allocate training across multi-processors	Operates large datasets efficiently	Demands distributed computing resources
GPU Acceleration	Use GPUs to parallelize computations.	Substantial speed-up for huge datasets	Needs GPUs and technical libraries.
Dimensionality reduction and feature selection	Decrease feature number.	Deflates computational cost.	Loss of information

2.9. Types of SVM Based on Functionality

SVM techniques can be functionally categorized based on their type and purpose. Below are the main types of SVM functions as shown in Table III.

TABLE III
TYPES OF SVM BASED ON FUNCTIONALITY

Function	Objective	Approach	Key Idea	Use Case Example
Binary Classification	Separate two classes using a hyperplane that maximizes the margin.	Optimize hyperplane $w \cdot x + b = 0$ to maximize margin. Use kernel functions (e.g., linear, polynomial, RBF) for nonlinear boundaries. Minimize $\frac{1}{2}\|\boldsymbol{w}\|^2$ subject to constraints.	Maximize margin between two classes.	Spam detection, medical diagnosis.
Multiclass Classification	Classify data into more than two classes.	**OneVsOne**: Train $\boldsymbol{k} * \frac{\boldsymbol{k-1}}{\boldsymbol{2}}$ binary classifiers for pairwise comparisons. **OneVsRest**: Train K binary classifiers, one per class. **Native Multiclass SVM**: Directly optimize multiclass objective.	Extend binary classification to multiple classes.	Handwritten digit recognition, image classification.
Multilabel Classification	Assign multiple labels to each data point.	Binary Relevance: Train a binary classifier for each label. Classifier Chains: Use predictions from previous classifiers as features. Adapted Algorithms: Modify loss function for multilabel tasks.	Assign multiple labels to a single instance.	Stock price prediction, housing price estimation.
Regression (SVR)	Predict continuous values .	Find function $f(x) = w \cdot x + b$ that deviates from true values by at most ϵ. Use slack variables for errors outside ϵ - tube. Apply kernel functions for nonlinear relationships.	Predict continuous values with a margin of tolerance.	Large-scale datasets, real-time applications.

IV. SVM AND SOME COMMON MACHINE LEARNING TECHNIQUES

Support Vector Machines (SVMs) provide a robust and interpretable classification technique that excels in processing high-dimensional data. Nevertheless, it is worth noting that alternative algorithms such as Logistic Regression, Decision Trees, Random Forests, and Neural Networks may be more appropriate for addressing the particular problem and data attributes, as indicated in Table IV. Assessing various algorithms on a dataset is crucial for making a well-informed conclusion regarding the most efficient performance of a particular activity.

SVMs are renowned for their effectiveness in high-dimensional domains and their ability to mitigate some issues caused by the curse of dimensionality.

TABLE IV
SVM AND SOME COMMON MACHINE LEARNING TECHNIQUES

	Strength	Limitation
SVM	**Non-linearity:** This can be efficiently addressed by employing kernel functions to handle non-linear data. **High dimensionality**: Effective in feature spaces with a large number of dimensions. **Resistant to extreme values**: Logistic regression is more vulnerable to outliers in the data than other statistical methods. **Enhanced computational efficiency**: SVMs exhibit slower training times, particularly when dealing with extensive datasets.	**The interpretability** of the model is challenging due to the intricate decision boundaries in a high-dimensional space, particularly when employing kernels. **Cost of computation**: Training SVMs can be computationally demanding, particularly when dealing with extensive datasets using kernel approaches. **Parameter tuning** poses a significant challenge when selecting the appropriate kernel function and its corresponding hyperparameter.
Logistic Regression	Its computing efficiency during training and its ability to effectively handle datasets are characterized by a high number of features and a relatively small number of data points. Offers interpretability by providing insights into the relationship between features and the target variable through the model coefficients. The calibration process allows for the straightforward determination of probability estimates pertaining to class membership.	Restricted to linear data: Optimal for data that can be separated linearly. May exhibit suboptimal performance when dealing with intricate, non-linear associations. The model's performance can be considerably affected by outliers present in the data.
Random forest	The ensemble nature of random forests generally leads to good accuracy in a wide range of classification and regression applications. By integrating numerous decision trees, variation can be decreased and generalization enhanced. Random forests have shown success in handling datasets with many different qualities. Using a random feature selection approach at every decision tree split help to prevent overfitting. Though not as clearly understandable as single decision trees, random forests allow one to obtain feature relevance scores to identify the variables most likely to contribute to the predictions of the model. Interpretability: Although not as easily comprehensible as individual decision trees, it is possible to get feature importance scores from random forests in order to ascertain the attributes that provide the greatest contributions to the model's predictions.	The interpretability of random forests may be comparatively lower than that of simpler models such as logistic regression, mostly due to the intricate ensemble of decision trees involved. Training random forests can have a significant computational cost, especially considering large datasets. The operation involves training several decision trees. Blackbox nature: While feature importance provides some understanding, grasping the complex internal mechanics of the ensemble can prove challenging.
Artificial Neural Networks	(ANNs) have the potential to learn intricate patterns and correlations from extensive datasets, rendering them well-suited for jobs that challenge conventional algorithms. Non-linearity: In contrast to less complex models such as logistic regression, artificial neural networks (ANNs) can accurately capture non-linear associations between features and the target variable. Feature extraction: ANNs can autonomously acquire pertinent features from unprocessed data, preventing manual feature engineering in some scenarios. Well-trained ANNs are effective in real-world prediction challenges because they can generalize well to unknown data.	**Black box nature**: Artificial neural networks' (ANNs') inner mechanisms can be complex and difficult to understand. This can have a disadvantage when interpretability is of great relevance. **Computational cost**: Training ANNs—especially large and deep ones—may be time-consuming and expensive. This work calls for large datasets and significant computational resources. ANNs are highly dependent on both the quality and quantity of data available. Insufficient, noisy, or biased training data can lead to poor performance. **Overfitting** occurs when artificial neural networks (ANNs) are not adequately regularized, leading to worse performance on unseen data.

V. Conclusion

Support Vector Machine sometimes known as SVM is an example of a typical form of supervised learning algorithm that was designed expressly for classification problem. The basic objective is to locate the hyperplane that most effectively divides data points into those belonging to distinct classes while simultaneously increasing the margin. The margin is the distance that separates the hyperplane from the data points that are closest to it, which are referred to as the support vectors. SVM is one of the most fundamental approaches to machine learning, which are renowned for their robust theoretical underpinning and their capacity to generate appropriate decision boundaries for categorization. Because they make use of kernel functions, they are able to effectively manage complex and non-linear data, which enables them to be adapted to a wide variety of applications, including text classification and picture recognition.

References

[1] Abdullah DM, Abdulazeez AM. Machine learning applications based on SVM classification a review. Qubahan Academic Journal. 2021 Apr 28;1(2):81-90. DOI: 10.48161/qaj.v1n2a50

[2] Hussain SM, Jilani MN, Haq MU, Shaikh MS. A Machine Learning Approach to Arabic Phoneme Classification through Ensemble Techniques. In 2024 5th International Conference on Advancements in Computational Sciences (ICACS) 2024 Feb 19 (pp. 1–7). IEEE. Artificial Neural Networks Random forest Logistic Regression

[3] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). DOI: 10.1007/s42979-021-00592-x

[4] Prasad SC, Anagha P, Balasundaram S. Robust pinball twin bounded support vector machine for data classification. Neural Processing Letters. 2023 Apr; 55(2):1131-53.

[5] Otchere DA, Ganat TO, Gholami R, Ridha S. Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models. Journal of Petroleum Science and Engineering. 2021 May 1; 200:108182.

[6] Zulfiqar M, Kamran M, Rasheed MB, Alquthami T, Milyani AH. Hyperparameter optimization of support vector machine using adaptive differential evolution for electricity load forecasting. Energy Reports. 2022 Nov 1; 8:13333-52.

[7] Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet of Things and Cyber-Physical Systems. 2024 Jan 3.

[8] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1998.

[9] Roushangar K, Ghasempour R. Supporting vector machines. In Handbook of Hydro informatics 2023 Jan 1 (pp. 411–422). Elsevier.

[10] Roy A, Chakraborty S. Support vector machine in structural reliability analysis: A review. Reliability Engineering & System Safety. 2023 May 1; 233:109126.

[13] B.-Y. Sun, D.-S. Huang, H.-T. Fang, Lidar signal denoising using least-squares support vector machine, IEEE Signal Process. Lett. 12 (Feb 2005) 101–104.

[14] P. Chen, B. Wang, H.-S. Wong, D.-S. Huang, Prediction of protein b-factors using multi-class bounded SVM, Protein Peptide Lett. 14 (Feb 2007) 185–190.

[15] X. Liang, L. Zhu, D.-S. Huang, Multi-task ranking SVM for image segmentation, Neurocomputing 247 (Jul 2017) 126–136.

[16] J. Cervantes, F. García Lamont, A. López-Chau, L. Rodríguez Mazahua, J. Sergio Ruíz, Data selection based on decision tree for SVM classification on large data sets, Appl. Soft Comput. J. (2015).

[17] V. A. Naik, A. A. Desai, online handwritten gujarati character recognition using svm, mlp, and k-nn, in: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1–6.

[18] J. L. Raheja, A. Mishra, A. Chaudhary, Indian sign language recognition using svm, Pattern Recognin. Image Anal. 26 (Apr 2016) 434–441.

[19] Hamel LH. Knowledge discovery with support vector machines. John Wiley & Sons; 2011 Sep 20.

[20] Brereton RG, Lloyd GR. Support vector machines for classification and regression. Analyst. 2010; 135(2):230-67.

[21] Vapnik, V. N., & Chervonenkis, A. Y. (1964). A theory of learning from general examples. Proceedings of the 3rd Annual ACM Conference on Theory of Computing (pp. 151–160). ACM.)

[22] Vapnik VN. An overview of statistical learning theory. IEEE transactions on neural networks. 1999 Sep;10(5):988-99.

[23] Burges CJ. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery. 1998 Jun;2(2):121-67.

[24] Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing. 2020 Sep 30; 408:189-215.

[25] Soman KP, Loganathan R, Ajay V. Machine learning with SVM and other kernel methods. PHI Learning Pvt. Ltd.; 2009 Feb 2.

[26] Ghosh S, Dasgupta A, Swetapadma A. A study on support vector machine based linear and non-linear pattern classification. In2019 International Conference on Intelligent Sustainable Systems (ICISS) 2019 Feb 21 (pp. 24–28). IEEE.

[27] Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing. 2020 Sep 30; 408:189-215. DOI: 10.1016/j.neucom.2019.10.118

[28] Steinwart I, Christmann A. Support vector machines. Springer Science & Business Media; 2008 Sep 15.

[29] Hubert K, Elisha B. Support Vector Machines (SVM): Explaining SVM and its application in regression tasks for sales forecasting. 2023

[30] Ishfaq K, Sana M, Ashraf WM. Artificial intelligence – built analysis framework for the manufacturing sector: performance optimization of wire electric discharge machining system. The International Journal of Advanced Manufacturing Technology. 2023 Oct; 128(11-12):5025-39.

[31] Cortes, C., Vapnik, V. Support-vector networks. Mach Learn 20, 273–297 (1995). DOI: 10.1007/BF00994018

[32] Hofmann T, Schölkopf B, Smola AJ. A review of kernel methods in machine learning. Mac-Planck-Institute Technical Report. 2006 Dec 14; 156.

[33] radhan A. Support vector machine-a survey. International Journal of Emerging Technology and Advanced Engineering. 2012 Aug; 2(8):82-5.

[34] Martin-Barragan B, Lillo R, Romo J. Interpretable support vector machines for functional data. European Journal of Operational Research. 2014 Jan 1; 232(1):146-55.

[35] Menon AK. Large-scale support vector machines: algorithms and theory. Research Exam, University of California, San Diego. 2009 Feb 27; 117.

[36] Gamal M, Abbas H, Sadek R. Hybrid approach for improving intrusion detection based on deep learning and machine learning techniques. InProceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020) 2020 (pp. 225–236). Springer International Publishing.

[37] Byvatov E, Schneider G. Support vector machine applications in bioinformatics. Applied bioinformatics. 2003 Jan 1;2(2):67-77.

[38] Dino HI, Abdulrazzaq MB. Facial expression classification based on SVM, KNN and MLP classifiers. In2019 International Conference on Advanced Science and Engineering (ICOASE) 2019 Apr 2 (pp. 70–75). IEEE.

[39] Chauhan VK, Dahiya K, Sharma A. Problem formulations and solvers in linear SVM: a review. Artificial Intelligence Review. 2019 Aug 15; 52(2):803- 55.

[40] Battineni G, Chintalapudi N, Amenta F. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). Informatics in Medicine Unlocked. 2019 Jan 1; 16:100200.

[41] Bhavan A, Chauhan P, Shah RR. Bagged support vector machines for emotion recognition from speech. Knowledge-Based Systems. 2019 Nov 15; 184:104886.

[42] Gu J, Lu S. An effective intrusion detection approach using SVM with naïve Bayes feature embedding. Computers & Security. 2021 Apr 1; 103:102158.

[43] Seeja RD, Suresh A. Deep learning based skin lesion segmentation and classification of melanoma using support vector machine (SVM). Asian Pacific journal of cancer prevention: APJCP. 2019; 20(5):1555.

[44] Patil RS, Biradar N. Automated mammogram breast cancer detection using the optimized combination of convolutional and recurrent neural network. Evolutionary intelligence. 2021 Dec; 14:1459-74.

[45] Hu R, Zhu X, Zhu Y, Gan J. Robust SVM with adaptive graph learning. World Wide Web. 2020 May; 23:1945-68.

[46] Pisner DA, Schnyer DM. Support vector machine. In Machine learning 2020 Jan 1 (pp. 101-121). Academic Press.

[47] Gong W, Chen H, Zhang Z, Zhang M, Wang R, Guan C, Wang Q. A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. Sensors. 2019 Apr 9; 19(7):1693.

[48] Jiang F, Lu Y, Chen Y, Cai D, Li G. Image recognition of four rice leaf diseases based on deep learning and support vector machine. Computers and Electronics in Agriculture. 2020 Dec 1; 179:105824.

[49] Vijayalakshmi A. Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications. 2020 Jun; 79:15297-317.

[50] Chandra MA, Bedi SS. Survey on SVM and their application in image classification. International Journal of Information Technology. 2021 Oct; 13:1-1.

[51] Huang W, Liu H, Zhang Y, Mi R, Tong C, Xiao W, Shuai B. Railway dangerous goods transportation system risk identification: Comparisons among SVM, PSO-SVM, GA-SVM and GS-SVM. Applied Soft Computing. 2021 Sep 1; 109:107541.

[52] Roman I, Santana R, Mendiburu A, Lozano JA. In-depth analysis of SVM kernel learning and its components. Neural Computing and Applications. 2021 Jun; 33(12):6575-94.

[53] Kaur P, Singh G, Kaur P. Intellectual detection and validation of automated mammogram breast cancer images by multi-class SVM using deep learning classification. Informatics in Medicine Unlocked. 2019 Jan 1; 16:100151.

[54] Liu B, Li CC, and Yan K. DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. Briefings in bioinformatics. 2020 Sep; 21(5):1733-41.

Mohammed Jabardi received the B.Sc. degree in Computer Science from the University of Technology, Baghdad, Iraq, in 1990, the M.Sc. degree in Computer Science from Jamia Hamdard University, New Delhi, India, in 2014, and the Ph.D. degree in Information Technology from the University of Babylon, Babylon, Iraq, in 2020.

He is currently an Assistant Professor with the Department of Software, College of Education, University of Kufa, Najaf, Iraq. His research interests include data mining, machine learning, and artificial intelligence applications. He has over six years of experience in scientific and academic research at the Information Technology Research Center, University of Kufa. Since 2017, he has been teaching courses on AI applications, programming, compiler construction, data mining, and machine learning for both undergraduate and master's students. He has also been supervising master's students since 2021.

Dr. Jabardi has served as a scientific reviewer for numerous Scopus-indexed journals, including AIJ, IJCSM, IJEECS, BEEI, and IJECE, and has contributed to the organization of academic conferences such as ICECCME 2023.