Enhancing SVM Performance Accuracy for Diabetes Diagnosis Using an Improved Ant Colony Optimization Based Support Vector Machine

Balogun Kayode Nuren[1]; Badru, Rahmon Ariyo[2]; Waheed Azeez Ajani[3]; Akinmuda Oluseye Ayobami[4]

[1,4]Department of Computer Science, Lead City University, Ibadan
[3]Department of Mathematics & Lead City University Ibadan, Oyo State, Nigeria
[2]Department of Information and Communication Technology, Cooperative Information Network (COPINE), National Space Research and Development Agency(NASRDA), Obafemi Awolowo University, Ile-Ife, Osun State

Publication Date: 2025/09/10

Abstract: Accurate diagnosis of diabetes is crucial for effective management and improved patient outcomes. Traditional Support Vector Machine (SVM) classifiers often struggle with accuracy due to parameter optimization challenges and unbalanced datasets. These challenges were addressed by developing an improved pheromone update technique for Ant Colony Optimization ACO-optimized SVM classifier. To achieve the aforementioned, the research generated a Hybrid Adaptive Pheromone Update Technique (HAPUT), Dynamic Exploration-Exploitation Balance (DEEB) and Pheromone Influence Factor (PIF). Subsequently, the parameters, BoxConstraint and KernelScale of the Support Vector Machine (SVM) classifier were optimized using an Ant Colony Optimization (ACO) approach in which HAPUT was used as the ACO pheromone update technique. Hence, each ant selects SVM parameters based on pheromone levels. The model developed was run in MATLAB codes using the PIMA Indian Dataset (PID) which composed of 268 diabetic and 500 non-diabetic samples. The dataset was split into 80/20 for training and validation. Thus, the accuracy of ACO-optimized SVM for default and improved pheromone update were compared.The comparative analysis shows that SVM has the optimum performance with accuracy, precision and recall of 79.13%, 69.388 % and 50.746%, respectively; while ACO optimized with SVM has the optimal accuracy and precision of 83.0435 % and 80.9524 %. Moreso, the results of the ACO-optimized SVM with a Default Pheromone Update Technique (DPUT) and ACO-optimized SVM with an Improved Pheromone Update Technique (IPUT) shows that IPUT reflected higher performance of 86.520 %, 81.130 % and 67.187 % for accuracy, precision and recall, respectively. This outcome is still optimal when compared to results from related studies. In conclusion, the model developed converges to the best combination of SVM parameters, BoxConstraint (C) and KernelScale, which yields the highest classification accuracy.

How to Cite: Balogun Kayode Nuren; Badru, Rahmon Ariyo; Waheed Azeez Ajani; Akinmuda Oluseye Ayobami (2025) Enhancing SVM Performance Accuracy for Diabetes Diagnosis Using an Improved Ant Colony Optimization Based Support Vector Machine. *International Journal of Innovative Science and Research Technology*, 10(8), 2804-2818. https://doi.org/10.38124/ijisrt/25aug1486

I. INTRODUCTION

Diabetes is a chronic metabolic disorder characterized by elevated blood glucose levels due to insufficient insulin production (type 1 diabetes) or ineffective insulin utilization (type 2 diabetes (Addissouky *et al.*, 2024; Taylor, 2024) . This condition leads to significant complications affecting cardiovascular, ocular, renal, and neurological systems (Dórea et al., 2024). In 2019, diabetes was responsible for approximately 1.5 million deaths globally, with nearly 48% occurring before the age of 70 (WHO, 2020). Between 2000 and 2019, global age-standardized mortality rates due to diabetes increased by 3%, with a more pronounced 13% rise in low- and middle-income countries (Khaltaev & Axelrod, 2021). Although a cure remains unavailable, early diagnosis remains critical in managing diabetes and preventing complications, with individuals diagnosed with pre-diabetes able to take measures to delay or avoid type 2 diabetes onset. The etiology of type 1 diabetes remains incompletely understood, though family history is a recognized risk factor (Pociot & Lernmark, 2016). In contrast, type 2 diabetes risk factors are better characterized, including pre-diabetes, overweight or obesity, age over 45 years, familial diabetes history, and physical inactivity (Xia et al., 2021). Early identification of at-risk individuals enhances opportunities for intervention and disease management.

Machine learning applications have expanded rapidly in medical diagnostics, providing tools to detect complex patterns in heterogeneous clinical data (Ahmed et al., 2020). Established algorithms such as Support Vector Machines (SVM), decision trees, random forests, and neural networks have been extensively employed in diabetes classification tasks (Kumari & Chitra, 2013). However, challenges such as hyperparameter optimization and class imbalance reduce traditional SVM classifier accuracy, motivating exploration of metaheuristic optimization algorithms like Ant Colony Optimization (ACO) to improve diagnostic performance.

➢ *Statement of Research Problem*

Accurate diagnosis of diabetes is crucial for effective management and improved patient outcomes. Traditional Support Vector Machine (SVM) classifiers often struggle with accuracy due to parameter optimization challenges and unbalanced datasets, such as the PIMA diabetes dataset (Mavrovouniotis et al., 2023; Manakkadu & Dutta, 2024). These challenges highlight the need for advanced meta-heuristic optimization algorithms, such as the Ant Colony Optimization (ACO) algorithm. Developing improved pheromone update techniques for ACO-optimized SVM classifiers could address these challenges and enhance classification accuracy, providing a solution to the pressing need for more accurate diabetes diagnoses. Hence, the need for this study.

This study therefore aims to enhance the accuracy of diabetes diagnosis by developing an improved pheromone update technique for an ACO-optimized SVM classifier.ormance metrics.

II. LITERATURE REVIEW

➢ *Diabetes Mellitus*

Diabetes mellitus is a chronic metabolic disorder marked by high blood glucose due to impaired insulin secretion or action. It includes three main types: Type 1 diabetes (autoimmune destruction of pancreatic beta cells), Type 2 diabetes (insulin resistance and deficiency), and gestational diabetes (Addissouky *et al.*, 2024; Taylor, 2024). Globally, diabetes prevalence is rapidly increasing, causing significant health burdens such as cardiovascular disease and kidney failure. Diagnosis relies on blood glucose and HbA1c measurements, but these require invasive sampling (Aisha et al., 2024). Emerging non-invasive techniques like photoplethysmography (PPG) combined with machine learning show promise, potentially enhancing early diabetes detection without blood tests (Alghlayini et al., 2025).

The global prevalence of diabetes has been rising at an alarming rate, with significant implications for public health systems worldwide. As of 2021, approximately 537 million adults aged 20-79 years were living with diabetes globally, a number projected to rise to 643 million by 2030 and 783 million by 2045 (Yameny, 2024; Soomro & Jabbar, 2024). This escalating prevalence underscores the urgent need for effective diagnostic and management strategies to mitigate the associated morbidity and mortality. Diabetes is a leading cause of cardiovascular diseases, kidney failure, blindness, and lower-limb amputations, contributing significantly to healthcare costs and economic burden (Ma et al., 2022).

The pathogenesis of diabetes involves a complex interplay of genetic, environmental, and lifestyle factors. In T1D, autoimmune destruction of pancreatic beta cells leads to absolute insulin deficiency. Conversely, T2D is primarily driven by insulin resistance, where peripheral tissues fail to respond adequately to insulin, coupled with an eventual decline in beta-cell function. Risk factors for T2D include obesity, physical inactivity, unhealthy diet, advancing age, and a family history of diabetes (Serbis et al., 2023). Emerging evidence also implicates factors such as sleep duration in the development of metabolic disorders, including diabetes. Both short and long sleep durations were associated with an increased risk of metabolic syndrome, a cluster of conditions that elevate the risk for diabetes and cardiovascular diseases. The diagnosis of diabetes traditionally relies on the measurement of blood glucose levels and glycated hemoglobin (HbA1c) (Rahman et al., 2023). The American Diabetes Association (ADA) criteria for diagnosing diabetes include:

- Fasting Plasma Glucose (FPG) ≥ 126 mg/dL (7.0 mmol/L)
- 2-hour Plasma Glucose ≥ 200 mg/dL (11.1 mmol/L) during an Oral Glucose Tolerance Test (OGTT)
- HbA1c ≥ 6.5% (48 mmol/mol)
- In a patient with classic symptoms of hyperglycemia, a random plasma glucose ≥ 200 mg/dL (11.1 mmol/L)

While these methods are well-established, they may not detect early dysglycemia or predict the progression to diabetes effectively. Moreover, they require blood samples and laboratory analysis, which may not be feasible in all settings. Recent advancements in technology have introduced non-invasive methods for diabetes detection. Photoplethysmography (PPG), an optical technique that measures blood volume changes in the microvascular bed of tissue, has been explored for this purpose. A study developed a machine learning-based approach using PPG signal features to classify diabetic and non-diabetic individuals. Their model achieved an Area Under the Curve (AUC) of 79.2%, indicating the potential of PPG signals in non-invasive diabetes detection (Oliveira et al., 2023).

➢ *Machine Learning*

Machine learning, a branch of artificial intelligence, focuses on creating algorithms that enable computers to learn from data and make predictions or decisions without explicit programming Mehrotra, 2019). It encompasses supervised learning, where models are trained on labeled data; unsupervised learning, which identifies patterns in unlabeled data; and reinforcement learning, which involves learning through rewards in dynamic environments. Supervised learning, the emphasis of this study, includes both regression and classification tasks using algorithms such as logistic regression, decision trees, random forests, support vector machines (SVM), and neural networks (Morales & Escalante, 2022). This research specifically explores an Ant Colony Optimization (ACO)-optimized

SVM to enhance diabetes diagnosis accuracy, combining optimization techniques with traditional classifiers to improve predictive performance.

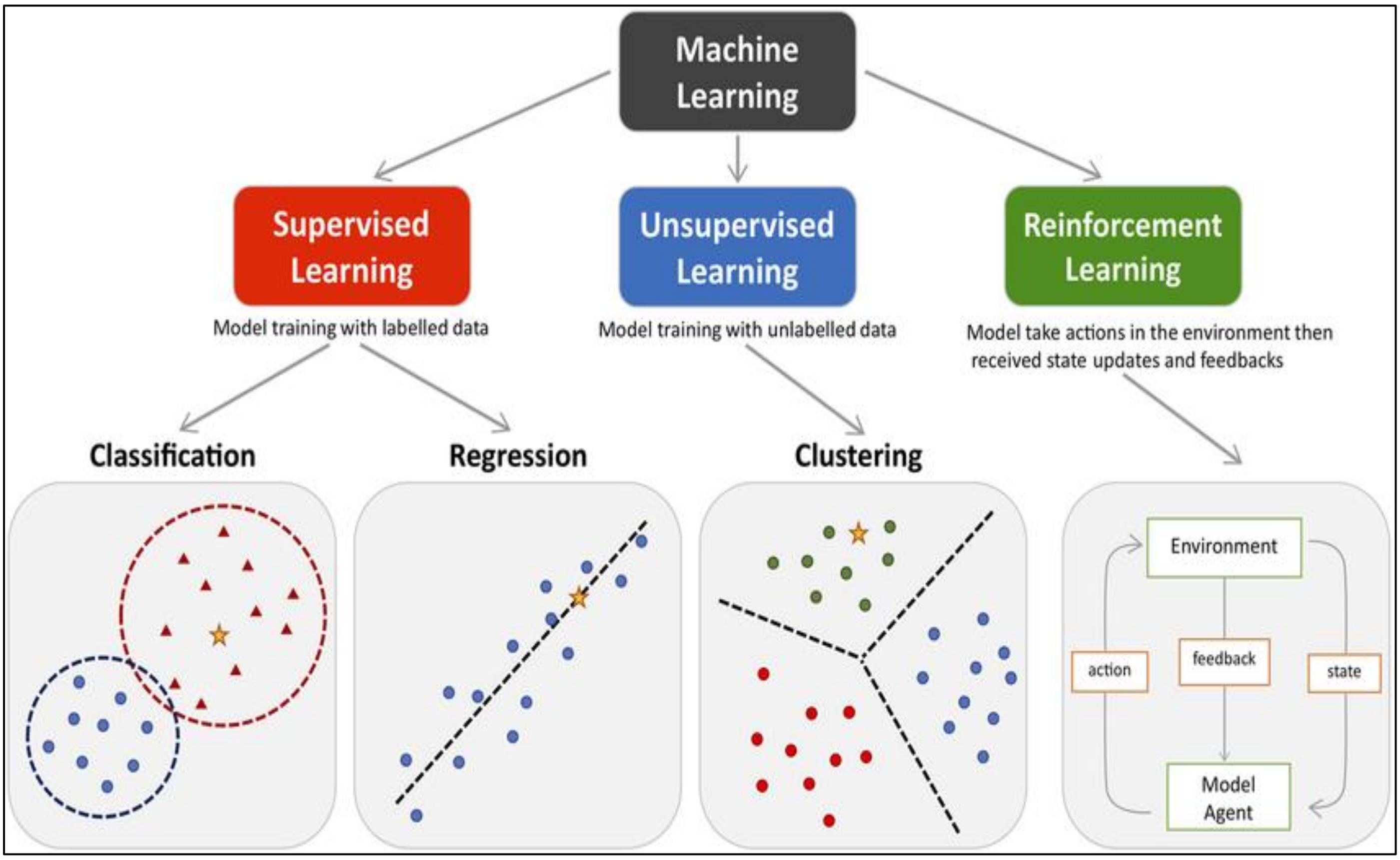

Fig 1 Types of Machine Learning (Thanmaikora, 2023).

- *Supervised Learning Algorithm*

Supervised Learning Algorithm is the primary emphasis of this study. With supervised learning, data from previously completed predictions are used to train a model for future predictions. Supervised learning can be broken down into two categories, regression and classification, depending on whether the output value is continuous or discrete (El Mrabe et al., 2021). Regression (both linear and logistic), Naive Bayes, K-nearest neighbour, random forest, support vector machines, decision trees, linear classifiers, and neural networks are all examples of supervised learning algorithms (Josephine, 2021). Improved pheromone update technique for diabetes diagnosis can be built via a wide variety of Machine Learning algorithms such as Logistic regression, decision trees, random forests, support vector machines (SVM), artificial neural networks (ANN), and gradient boosting techniques like XGBoost and LightGBM. However, this study will focus on ACO-optimized SVM.

- *SVM*

Support Vector Machine (SVM) is a versatile machine learning algorithm widely used for classification and regression tasks (Saravanan et al., 2023). It represents data points in a multi-dimensional space and identifies an optimal hyperplane that distinctly separates different classes. SVM uses kernel functions such as linear, polynomial, and radial basis function (RBF) to transform non-linearly separable data into higher dimensions where it becomes linearly separable (Aiad et al., 2021). This kernel trick enhances SVM's ability to handle complex datasets effectively. SVM aims to maximize the margin between classes while minimizing classification errors, which is formulated as a convex optimization problem ensuring global optimality (Pirouz & Pirouz, 2023). Despite its strengths, SVM can be computationally intensive on large datasets, and its performance heavily depends on the choice of kernel and hyperparameter tuning. Key SVM hyperparameters include the kernel type, regularization parameter (C), gamma (for RBF kernel), and class weights to handle imbalanced datasets (Tsai & Chang, 2023). SVM demonstrates robustness against outliers and is widely applied in image recognition, text classification, and bioinformatics, though interpretability can be a challenge with complex kernels.

➢ *Ant Colony Optimization (ACO)*

ACO is a probabilistic technique inspired by the foraging behavior of ants, particularly their ability to find the shortest path between their colony and a food source (Abdulghani & Abdulghan, 2024**)**. Ants deposit pheromones on paths they traverse, and the concentration of pheromones influences the probability of other ants following the same path. Over time, shorter paths accumulate higher pheromone concentrations, guiding the colony toward optimal solutions (Abdulghani & Abdulghan, 2024). The integration of ACO with SVM involves using the ACO algorithm to search for optimal SVM parameters, thereby enhancing the model's performance. This approach has been explored in various studies, each proposing different methodologies to achieve effective parameter optimization (Manakkadu & Dutta, 2024)· Traditional ACO algorithms are designed for discrete optimization problems. However, SVM parameter optimization involves continuous variables. A continuous Ant Colony Optimization (ACOR) algorithm for optimizing SVM parameters without discretizing continuous values (Zhou et al., 2023). Their approach demonstrated improved classification accuracy and computational efficiency compared to grid search methods.

- *Improved Ant Colony Optimization (IACO) for SVM Parameter Optimization*

Improved Ant Colony Optimization (IACO) is a technique to optimize SVM parameters. The method was evaluated on benchmark datasets and showed enhanced performance compared to standard ACO-SVM algorithms (Rongali & Yalavarthi, 2015).

- *ACO-Based Instance Selection for SVM Speed Optimization*

Beyond parameter optimization, ACO has been applied to instance selection to improve SVM training speed. In a work that proposed an ACO-based instance selection algorithm that identifies boundary instances, which are crucial for defining the decision surface in SVMs. By selecting these instances, the algorithm reduces training time without significantly affecting classification accuracy (Akinyelu et al., 2020). In the context of SVM parameter optimization, each ant in the ACO algorithm represents a potential solution, i.e., a set of parameters (C, γ)[73]. The quality of each solution is evaluated based on the classification accuracy of the SVM model with those parameters. The probability $\boldsymbol{P_{ij}}$ of an ant moving from parameter value i to j is influenced by the pheromone concentration $\boldsymbol{\tau_{ij}}$ and a heuristic value $\boldsymbol{\eta_{ij}}$ (Al-Shourbaji et al., 2022).

$$P_{ij}=\frac{(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}}{\sum_{k\epsilon allowed}(\tau_{ik})^{\alpha}.(\eta_{ik})^{\beta}} \quad (1)$$

Where:

- ✓ Α and β are parameters that control the influence of pheromone concentration and heuristic information, respectively.
- ✓ η_{ij} is typically defined as the inverse of the objective function value (e.g., classification error) associated with moving from i to j.

After all ants have constructed their solutions, the pheromone levels are updated to reflect the quality of the solutions found. The pheromone update rule can be expressed as:

$$\tau_{ij}(1-\rho).\tau_{ij}+\sum_{all\ ants}\Delta\tau_{ij}^{\ k} \quad (2)$$

- *Support Vector Machine Hyperparameter Optimization*

Support Vector Machines (SVMs) are effective for classification and regression tasks, but their performance is sensitive to the selection of hyperparameters. Key hyperparameters include the penalty parameter C, the kernel parameter γ (for radial basis function kernels), and other kernel-specific parameters (Gamba, 2024). Optimizing these parameters involves searching for the configuration that maximizes the SVM's performance on validation data. Several optimization techniques Grid Search, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) are commonly employed for this purpose.

SVM aims to find the optimal hyperplane that maximizes the margin between two classes. Given a training dataset (x_i,y_i), where $x_i \in R^n$ and y_i $\in\{-1,1\}$yi the optimization problem for a linear SVM is:

$$\min\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i \quad (3)$$

Subject to:

$$y_i(w.x_i+b)\geq 1-\xi_i, \qquad \xi_i\geq 0 \quad (4)$$

Here, w is the weight vector, b is the bias, C is the penalty parameter controlling the trade-off between margin size and misclassification, and ξ_i are slack variables for soft-margin SVMs. For non-linear problems, a kernel function $K(x_i, x_j)$ maps data into a higher-dimensional space, allowing the SVM to construct non-linear decision boundaries.

➢ *Genetic Algorithm (GA)*

GA mimics natural evolution, using selection, crossover, and mutation to iteratively improve solutions (Othman et al., 2022). For SVM, hyperparameters C and γ are encoded as chromosomes. Each chromosome x = (C, γ) is evaluated using a fitness function, such as cross-validation accuracy:

f(x) = Accuracy (C,γ)

- *The GA Process Involves:*

- ✓ Selection: Choosing top-performing chromosomes based on their fitness scores.
- ✓ Crossover: Combining pairs of chromosomes to produce offspring:

$$x_{offspring}=\alpha x_{parent1}+(1-\alpha)x_{parent2}, \qquad \alpha \epsilon [0,1].$$

- ✓ Mutation: Introducing random changes to maintain diversity:

$$x_{mutated}=x+\delta, \qquad \delta\sim N(0,\sigma^2)$$

GA effectively explores the search space, avoiding local optima while balancing exploration and exploitation[84].

➢ *Particle Swarm Optimization (PSO)*

Particle Swarm Optimization (PSO) is another popular method derived from the collective behavior of swarms (Gad, 2022). Each particle represents a candidate solution in the search space, and particles adjust their positions based on individual and group performance. This algorithm is particularly effective in continuous search spaces, offering faster convergence than GA while maintaining a balance between local and global searches. Studies have demonstrated PSO's superiority in optimizing SVM hyperparameters, particularly in terms of execution speed and computational efficiency (Kalita et al., 2020). PSO is inspired by the social behavior of swarms. Each particle represents a candidate solution x=(C, γ), with position x_t and velocity v_t at iteration t. Particles update their velocities and positions based on personal best (p_{best}) and global best (g_{best}) solutions:

$$v_{t+1}=\omega v_t+c_1r_1(p_{best}-x_t)+c_2r_2(g_{best}-x_t), \quad (5)$$

$$x_{t+1}=x_t+v_{t+1}$$

Here, ω is the inertia weight, c_1 and c_2 are acceleration coefficients, and $r_1,r_2 \sim U(0, 1)$ are random weights. PSO balances exploration and exploitation, converging quickly to optimal solutions. It has demonstrated superior performance in SVM hyperparameter optimization.

➢ *ACO-SVM based Classifiers*

ACO is a probabilistic technique inspired by the foraging behavior of ants, particularly their ability to find the shortest path between their colony and a food source (Choudhary, 2024). Ants deposit pheromones on paths they traverse, and the concentration of pheromones influences the probability of other ants following the same path. Over time, shorter paths accumulate higher pheromone concentrations, guiding the colony toward optimal solutions. In the context of SVM optimization, ACO uses a graph representation of the hyperparameter space and constructs solutions through stochastic exploration (Alsarhan et al., 2023). Classifying unbalanced datasets presents a significant challenge, as traditional Support Vector Machine (SVM) algorithms often struggle to deliver accurate results (Manakkadu & Dutta, 2024).

τ_c be the pheromone level for box constraint C

τ_S be the pheromone level for the kernel scale S

To optimize SVM parameters τ_c and τ_s using ACO, the pheromone update equation for each parameter can be adapted as[91];

$$\tau_c \leftarrow (1-\rho)*\tau_c+\sum_{k=1}^{m}\frac{\alpha}{score_k}*C_k \quad (6)$$

$$\tau_s \leftarrow (1-\rho)*\tau_s+\sum_{k=1}^{m}\frac{\alpha}{score_k}*S_k \quad (7)$$

- τ_c and τ_s are the pheromone levels associated with the SVM parameters C (box constraint) and S (kernel scale).
- ρ be the evaporation rate
- $score_k$ is the performance score (e.g., accuracy, or its negative for minimization) of the kth ant.
- α be the memory influence factor
- C_k and S_k are the specific values of C and S chosen by the *k*th ant.

➢ *Review of Related Studies*

Several studies have demonstrated the successful application of ML algorithms in diagnosing diabetes and related complications.

Poldrack et al. (2020) highlights the effectiveness of machine learning techniques in predicting diabetes using the Pima Indian Diabetes Dataset. The primary aim was to enhance diagnostic accuracy through robust preprocessing and classification methods. Eight machine learning models were employed, including Random Forest, Decision Tree (J48), Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression. Preprocessing methods like feature selection, missing value imputation, normalization, and standardization were applied to improve model performance. Among these models, Random Forest emerged as the most accurate, achieving an accuracy rate of 80.87%. This was significantly higher compared to other models, underscoring its suitability for diabetes prediction. The study also focused on comparing the efficiency of various preprocessing techniques and their impact on diagnostic accuracy. Normalization and feature selection proved crucial in enhancing the model's performance. The authors emphasized the need for preprocessing steps, as raw data often contains noise and missing values, which can lead to inaccurate predictions. By addressing these challenges, the study successfully demonstrated that machine learning could be an effective tool for early diabetes diagnosis. However, the authors also highlighted some limitations, such as the dependency on dataset quality and the need for external validation with diverse datasets.

Thabit et al. (2022) investigated the application of machine learning algorithms for early diabetes detection using the Pima Indian Diabetes Dataset. The study explored the efficacy of various classification algorithms, including Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), and Naïve Bayes (NB). To optimize model performance, preprocessing techniques like data cleaning, feature selection, scaling, and splitting were applied. The Random Forest classifier outperformed other models, achieving an accuracy of up to 100% during testing. This remarkable performance demonstrates the potential of Random Forest in medical diagnostics, particularly for diabetes. The study’s implementation used Python within the ANACONDA environment, leveraging tools like Jupyter Notebook and Spyder for data analysis and model building. The research also highlighted the importance of preprocessing steps in ensuring data quality and improving model performance. Features such as glucose levels, BMI, and age were identified as critical determinants of diabetes risk, making feature selection a pivotal part of the workflow.

The study underscores the transformative potential of machine learning in healthcare, paving the way for more accurate and efficient diagnostic tools for diabetes management.

Almahdawi et al. (2022) evaluated the performance of machine learning models in predicting diabetes using a dataset collected from Iraqi patients. The study tested three classifiers: Random Forest, K-Nearest Neighbor (KNN), and Multilayer Perceptron (MLP). Two experimental setups were employed to assess the impact of feature selection on model performance. In the first experiment, all 12 features of the dataset were used, with Random Forest achieving the highest accuracy of 98.8%. In the second experiment, only five features were selected for training, leading to improved performance for KNN and MLP. However, Random Forest's accuracy slightly decreased to 97.5%, highlighting the importance of comprehensive feature selection. The authors emphasized the significance of feature engineering in enhancing model performance. Attributes like glucose levels, BMI, and family history were identified as key predictors of diabetes. Additionally, the study underscored the need for robust preprocessing techniques, including data cleaning and normalization, to ensure high-quality input for machine learning models.

Ahmed et al. (2022) introduced a novel approach to diabetes prediction using a hybrid model that combines Support Vector Machines (SVM) and Artificial Neural Networks (ANN). This fused model leverages fuzzy logic to classify patients as diabetic or non-diabetic, achieving a prediction accuracy of 94.87%. The study employed a dataset divided into 70% training and 30% testing. Outputs from the SVM and ANN models were fed into a fuzzy logic system, which made the final diagnosis. The hybrid approach was stored on a cloud platform, allowing real-time predictions based on patient data. The research emphasized the importance of integrating multiple machine learning models to enhance diagnostic accuracy. By combining the strengths of SVM and ANN, the fused model demonstrated superior performance compared to standalone models. The use of fuzzy logic further improved decision-making by accounting for uncertainties in the data. However, the authors noted some challenges, such as the computational complexity of the fused model and the need for extensive training data. Mansouri et al. (2024) explored the use of the K-Nearest Neighbors (KNN) algorithm for predicting gestational diabetes mellitus (GDM). The study utilized the Pima Indian Diabetes Dataset and employed various preprocessing steps, including handling missing values, feature scaling, and data splitting. The KNN classifier achieved an accuracy of 76%, demonstrating its potential for early diabetes detection. The study also highlighted the importance of hyperparameter tuning and data preprocessing in improving model performance. Metrics such as recall, precision, and F1-score were used to evaluate the classifier's effectiveness. One of the key contributions of the research was its focus on gestational diabetes, a condition often linked to prediabetic states and future Type 2 diabetes. The findings underscore the need for early detection tools to mitigate long-term health risks. The authors acknowledged some limitations, including the relatively low accuracy of KNN compared to other algorithms and the need for more extensive datasets.

Uddin & Ali (2023) examines the application of machine learning algorithms for predicting diabetes. Utilizing patient data such as glucose levels, BMI, and family history, the authors implemented various models, including Decision Trees, SVMs, and Neural Networks. The dataset encompassed a combination of demographic and clinical factors. The research focused on comparing algorithm performance using metrics like sensitivity, specificity, and AUC-ROC. The Decision Tree and Random Forest models demonstrated the highest levels of accuracy and sensitivity, with predictions exceeding 90%. Moreover, the study emphasized the importance of model optimization, including feature selection and hyperparameter tuning, to improve predictive outcomes. One of the key contributions of this research lies in its emphasis on early detection and prevention strategies. By analyzing patient lifestyle data and clinical measurements, the study showcases how machine learning can provide personalized risk assessments. Such applications could significantly impact preventive healthcare strategies, allowing for timely interventions. Their study reinforces the transformative potential of machine learning in diabetes diagnostics, paving the way for innovative tools to assist in early detection and management.

Agliata et al. (2023) delved into the role of machine learning techniques, particularly artificial neural networks (ANN), in diagnosing Type 2 diabetes. The study utilized datasets such as NHANES and MIMIC-III to develop a binary classifier capable of distinguishing diabetic from non-diabetic patients based on clinical and demographic data. With a primary evaluation metric of accuracy, the ANN-based model achieved a notable performance of 86%, complemented by a robust ROC-AUC score of 0.934. The study highlighted the utility of ML models in identifying nonlinear relationships between various patient attributes and diabetes risk, which traditional methods might overlook. Additionally, the research emphasized the potential of ANN models to support long-term diabetes management by integrating predictive analytics into clinical workflows. The study also explored the importance of balanced datasets in training machine learning models, ensuring unbiased predictions across diverse patient groups.

III. METHODOLOGY

➢ *Selection of Datasets*

The study utilized the PIMA Indian Dataset (PID). The PID was chosen due to its relevance to modern lifestyles characterized by high processed food intake and low physical activity. PID, composed of a total of 768 instances, with 268 diabetic and 500 non-diabetic samples. The dataset features eight key attributes influencing diabetes prediction, namely pregnancies, BMI, insulin level, age, blood pressure, skin thickness, glucose, and Diabetes Pedigree Function. The dataset also contains a label function as shown in Table 1. The dataset was split 80/20 for training and validation.

Table 1 Description of PIMA Indian Dataset Attributes

No.	Selected Attributes	Description of Selected Attributes	Range
1	Pregnancy	Number of times a participant is pregnant	0 – 17
2	Glucose	Plasma glucose concentration 2 hours after an oral glucose tolerance test	0 – 199
3	Diastolic Blood Pressure	Diastolic blood pressure (pressure exerted into arteries between heartbeats) (mm Hg)	0 – 122
4	Skin Thickness	Triceps skinfold thickness (mm), related to collagen content	0 – 99
5	Serum Insulin	2-hour serum insulin (mu U/ml)	0 – 846
6	BMI	Body mass index (weight in kg/(height in m)^2)	0 – 67.1
7	Diabetes Pedigree Function	A metric used in diabetes prognosis	0.078 – 2.42
8	Age	Age of participants	21 – 81
9	Outcome	Diabetes class variable; "Yes" indicates diabetic, "No" indicates non-diabetic	Yes/No

- ➢ *Performance Comparison of Majorly Utilized Machine Learning Classification Techniques*

The performance comparison of majorly utilized Machine Learning Classification Techniques was appraised so that the classification technique with optimal performance would be selected. In this regard, 4 ML algorirthms was considered; Logistic Regression, Support Vector Machine, Naive Bayes and Random Forest. The selection of these classifiers was based on high frequency utilization of these classifiers in research journals and publications. Also, the algorithms for these classifiers were coded using MatLab and the snippet codes are highlighted below.

```
%comparison logistic regression, SVM, NB and random forest

%include recal, accuracy and precision

clear; clc;

% Load the PIMA Indian Diabetes dataset

data = readtable('diabetes.csv'); %

% Split the data into features (X) and target (y)

X = data{:,1:end-1};

y = data{:,end};
```

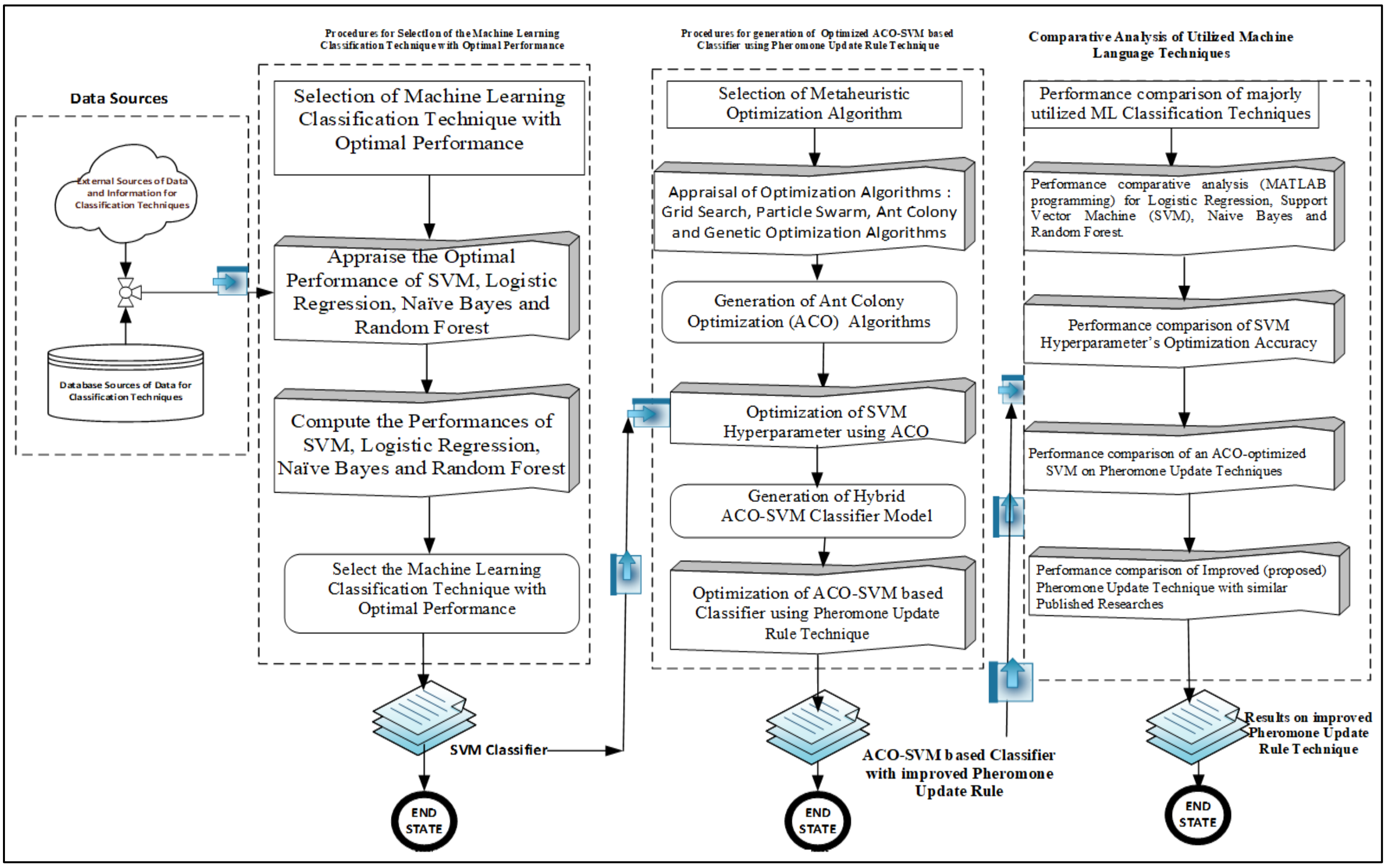
Data Sources
External Sources of Data and Information for Classification Techniques
Database Sources of Data for Classification Techniques
Procedures for Selection of the Machine Learning Classification Technique with Optimal Performance
Selection of Machine Learning Classification Technique with Optimal Performance
Appraise the Optimal Performance of SVM, Logistic Regression, Naïve Bayes and Random Forest
Compute the Performances of SVM, Logistic Regression, Naïve Bayes and Random Forest
Select the Machine Learning Classification Technique with Optimal Performance
SVM Classifier
END STATE
Procedures for generation of Optimized ACO-SVM based Classifier using Pheromone Update Rule Technique
Selection of Metaheuristic Optimization Algorithm
Appraisal of Optimization Algorithms : Grid Search, Particle Swarm, Ant Colony and Genetic Optimization Algorithms
Generation of Ant Colony Optimization (ACO) Algorithms
Optimization of SVM Hyperparameter using ACO
Generation of Hybrid ACO-SVM Classifier Model
Optimization of ACO-SVM based Classifier using Pheromone Update Rule Technique
ACO-SVM based Classifier with improved Pheromone Update Rule
END STATE
Comparative Analysis of Utilized Machine Language Techniques
Performance comparison of majorly utilized ML Classification Techniques
Performance comparative analysis (MATLAB programming) for Logistic Regression, Support Vector Machine (SVM), Naive Bayes and Random Forest.
Performance comparison of SVM Hyperparameter's Optimization Accuracy
Performance comparison of an ACO-optimized SVM on Pheromone Update Techniques
Performance comparison of Improved (proposed) Pheromone Update Technique with similar Published Researches
Results on improved Pheromone Update Rule Technique
END STATE

Fig 2 The Conceptual Diagram for the Research Study Methodologies

➢ *Selection of Metaheuristic Optimization Algorithm*

Metaheuristic algorithms are computationally intelligent algorithms used for solving sophisticated optimization problems (Almufti et al., 2023). Metaheuristic algorithms mimic specific animal behaviors or evolutionary processes to explore a solution space through randomized search methods, ultimately identifying the optimal solution through this approach (Darvishpoor et al., 2023). Moreover, the Metaheuristic Optimization Algorithms proposed for metaheuristic optimization algorithm are;

- Grid Search Optimization Algorithms
- Particle Swarm Optimization (PSO) Algorithm
- Ant Colony Optimization (ACO) algorithm and
- Genetic (Optimization) Algorithms (GA)

The snippet codes utilized to select optimization algorithm with optimum performance are;

```
% Display the best SVM parameters, accuracy, recall, and precision

disp(['Best BoxConstraint (C): ', num2str(bestC)]);

disp(['Best KernelScale: ', num2str(bestKernelScale)]);
disp(['Best accuracy: ', num2str(bestAccuracy)]);

disp(['Best recall: ', num2str(bestRecall)]);

disp(['Best precision: ', num2str(bestPrecision)]);

% Classification and evaluation function

function [accuracy, recall, precision] = classifyAndEvaluate(data, target, C, KernelScale)

% Split data into training and testing sets

cv = cvpartition(size(data, 1), 'HoldOut', 0.3);

trainData = data(cv.training, :);

trainTarget = target(cv.training);

testData = data(cv.test, :);

testTarget = target(cv.test);
```

The outcomes of the optimization algorithm performance analysis show that Ant Colony Optimization (ACO) algorithm has the optimum performance (details of results are in next section), thus, ACO is utilized to form a hybrid classifier with SVM.

➢ *Optimization of SVM Hyperparameter using ACO*

Given that τ_c be the pheromone level for box constraint C and τ_S be the pheromone level for the kernel scale S, thus, to optimize SVM parameters τ_c and τ_s using Ant Colony Optimization (ACO), the pheromone update equation for each parameter can be adapted as;

$$\tau_c \leftarrow (1-\rho)*\tau_c + \sum_{k=1}^{m} \frac{\alpha}{score_k} * C_k \qquad (8)$$

$$\tau_s \leftarrow (1-\rho)*\tau_s + \sum_{k=1}^{m} \frac{\alpha}{score_k} * S_k \qquad (9)$$

- τ_c and τ_s are the pheromone levels associated with the SVM parameters C (box constraint) and S (kernel scale).
- ρ be the evaporation rate
- $score_k$ is the performance score (e.g., accuracy, or its negative for minimization) of the kth ant.
- α be the memory influence factor
- C_k and S_k are the specific values of C and S chosen by the *k*th ant.

Meanwhile, accuracy was used for the performance score ($score_k$) when performance comparison with an existing pheromone update technique (illustrated with equation 8) and the proposed improved pheromone update technique iterated in equation 9. Thus, the snippet codes utilized are shown below.

```
clear;

% Read the PIMA dataset

data = readtable('diabetes.csv');

% Display the first few rows of the dataset

disp(head(data));

% Number of ants

numAnts = 250;%200

% Number of iterations

numIterations = 150;%100
```

➢ *Generation of an Improved Pheromone Update Rule Technique*

The proposed Hybrid Adaptive Pheromone Update technique would combine the following techniques:

- Hybrid Pheromone Update: The hybrid technique will Combines adaptive pheromone scaling with a memory-based approach to account for historical best solutions.
- Dynamic Exploration-Exploitation Balance: This will adjust exploration versus exploitation dynamically based on solution convergence.
- Pheromone Influence Factor: This uses a factor to prioritize promising regions in the parameter space.

The above methodologies were employed on PIMA data and the combination of pheromone variables that produced optima performance was utilized to generated algorithmic model illustrated in equation 10 and 11. The snippet codes utilized are:

% Determine the best accuracy, recall, and precision of this iteration

[maxAccuracy, maxIndex] = max(accuracies);

maxRecall = recalls(maxIndex);

maxPrecision = precisions(maxIndex);

% Memory update

*Memory_C = Memory_C + bestC * bestAccuracyMemory;*

*Memory_Scale = Memory_Scale + bestKernelScale * bestAccuracyMemory;*

% Dynamic exploration-exploitation balance

beta = 1 - (std(accuracies) / mean(accuracies));

- ➢ *Algorithmic Outline of proposed Hybrid Adaptive Pheromone Update Technique*

The algorithmic outline for the pheromone update equation proposed for the technique to be adopted is as state in equation 3.1.

$$\tau_c=(1-\rho)\tau_c+ \alpha\times Memory_C+ \beta\times\Delta_{\tau C} \quad (10)$$

$$\tau_s=(1-\rho)\tau_S+ \alpha\times Memory_s+ \beta\times\Delta_{\tau S} \quad (11)$$

Where

τ_c is the pheromone level for box constraint C

τ_S is the pheromone level for the kernel scale.

ρ is the evaporation rate.

Moreso, the changes $\Delta_{\tau C}$ and $\Delta_{\tau S}$ in pheromone levels for C and Kernel Scale are defined as illustrated in equation 10 & 11.

$$\Delta_{\tau C} = \left(\frac{accuracy_{ant}}{accuracy_{best}}\right)\times C_{ant}\times\frac{accuracy_{best}}{100} \quad (12)$$

$$\Delta_{\tau S}=\left(\frac{accuracy_{ant}}{accuracy_{best}}\right)\times Scale_{ant}\times\frac{accuracy_{best}}{100} \quad (13)$$

Where

$\Delta_{\tau C}$ and $\Delta_{\tau S}$ are the changes in pheromone levels for C and Kernel Scale

$Accuracy_{best}$ is the best accuracy found so far

$Accuracy_{ant}$ is the accuracy of the current ant

C_{ant} and $Scale_{ant}$ be the parameters chosen by the current ant

α be the memory influence factor

β be the exploration-exploitation balance factor

In addition, the memory-based influence and the dynamic exploration-exploitation balance are defined as stated in equations 14 and 16.

- *Memory-Based Influence:*

This will incorporate historical best pheromone levels,

Where:

$$Memory_C=Memory_C+ BestMemory_C\times bestAccuracyMemory \quad (14)$$

$$Memory_{Scale}=Memory_{Scale}+ BestMemory_{Scale}\times bestAccuracyMemory \quad (15)$$

- *Dynamic Exploration-Exploitation Balance*

Adjust the balance factor β based on convergence using the equation below

$$\beta=1-\frac{std(accuracies)}{mean(accuracies)} \quad (16)$$

The snippet codes for the aforementioned procedures are below. Full programming code is in appendix I;

% Update pheromone for SVM parameters with hybrid approach

*pheromoneC = (1 - initialEvaporationRate) * pheromoneC + alpha * Memory_C + beta * ((accuracies(maxIndex) / maxAccuracy) * Cs(maxIndex) * (maxAccuracy / 100))*

*pheromoneScale = (1 - initialEvaporationRate) * pheromoneScale + alpha * Memory_Scale + beta * ((accuracies(maxIndex) / maxAccuracy) * Scales(maxIndex) * (maxAccuracy / 100))*

- ➢ *Algorithmic Outline for the Improved ACO Pheromone Update Technique*

The parameters, BoxConstraint and KernelScale of the Support Vector Machine (SVM) classifier was optimized using an Ant Colony Optimization (ACO) approach. The proposed hybrid adaptive pheromone update technique will be used as the ACO pheromone update technique. Below is the step-by-step algorithmic outline of the process:

- *Step 1: Initialization*

✓ *Dataset Loading*

D=readDataset(‘PIMA_DIABETICS.csv’) (17)

Here D is the dataset read from a.csv file

- ✓ *Normalization*

$$X=normalize\left(D\left[:,1:n\text{-}1\right]\right) \tag{18}$$

Where X is the normalized feature matrix,

$$y=D\left[:,1:n\right]) \tag{19}$$

Where y is the target vector.

- ✓ *ACO Parameters:* set ACO parameters (numerical values given below will be utilized in this work)

$$numAnts=200$$

$$numIterations=100$$

$$initialEvaporationRate\ (\rho_i)=0.3$$

$$\alpha=0.5$$

$$bestAccuracyMemory=0.1$$

$$memoryFactor=0.05$$

- ✓ *Pheromone Initialization:*

$$\tau_c=1, \qquad \tau_S=1$$

Where τ_c and τ_S are the pheromone levels for the SVM parameters BoxConstraint (C) and KernelScale.

- ✓ *Memory Initialization:*

$$Memory_C=0, \qquad Memory_{Scale}=0$$

- ✓ *Best Parameter Initialization:*

$$bestC=1,\ bestKernelScale=1,\ bestAccuracy=0$$

- *Step 2: Iterative Optimization Loop*

  For each iteration (iteration = 1 to numIterations):

- ✓ *Ant Parameter Selection*

$$C_i=\tau_C{*}rand(), \tag{20}$$

$$KernelScale_i=\tau_S{*}rand()$$

Where C_i and $KernelScale_i$ are the SVM parameters selected by the *i-th* ant.

- ✓ *SVM Classification and Evaluation*

$$Accuracy_i=classifyAndEvaluate\ (X,y,C_i,KernelScale_i) \tag{21}$$

- ✓ *Best Accuracy Determination:*

$$maxAccuracy=\max\left(Accuracy_i\right), \quad maxIndex=argmax(Accuracy_i)$$

The best accuracy and corresponding index are determined from the current iteration.

- ✓ *Memory Update:*

$$Memory_C=Memory_C+bestC{*}bestAccuracyMemory \tag{22}$$

$$Memory_S=Memory_S+bestKernelScale{*}bestAccuracyMemory \tag{23}$$

- ✓ *Dynamic Exploration-Exploitation Balance (β):*

$$\beta=1\text{-}\left(\frac{\sigma\left(Accuracy_i\right)}{\mu(Accuracy_i)}\right) \tag{24}$$

Where $\sigma()$ and $\mu()$ represent the standard deviation and mean of the accuracies, respectively.

- ✓ *Pheromone Update:*

$$\tau_C=\left(1\text{-}\rho_i\right)\tau_C+\alpha Memory_C+\beta\left(\frac{Accuracy_{maxIndex}}{maxAccuracy}\times C_{maxIndex}\times\frac{maxAccuracy}{100}\right)3.13(a)$$

$$\tau_S=(1\text{-}\rho_i)\tau_S+\alpha Memory_S+\beta\left(\frac{Accuracy_{maxIndex}}{maxAccuracy}\times KernelScale_{maxIndex}\times\frac{maxAccuracy}{100}\right) \tag{25}$$

- ✓ *Update Best Parameters:*

if maxAccuracy>bestAccuracy, then:

$$bestC=C_{maxIndex,}bestKernelScale=KernelScale_{maxIndex},\ bestAccuracy=maxAccuracy$$

- *Step 3: Display Results*

  After all iterations, display the best SVM parameters and their corresponding accuracy:

Best BoxConstraint (C):bestC

Best KernelScale:bestKernelScale

Best accuracy:bestAccuracy

The snippet codes for the aforementioned procedural outline are shown below.

```
% Classification and evaluation function

function [accuracy, recall, precision] = classify And Evaluate(data, target, C, KernelScale)

% Split data into training and testing sets

cv = cvpartition(size(data, 1), 'HoldOut', 0.3);
```

trainData = data(cv.training, :);

trainTarget = target(cv.training);

➢ *Implementation of MatLab Codes for the Algorithmic Procedures*

The coding of the algorithmic procedures stated in aforementioned equations would be generated using MatLab R2024b version. This version of the MatLab is selected due to the following major updates, transitions and features:

- 5G Toolbox: This will explore 6G waveform generation; the Wireless Waveform Analyzer app will also perform signal quality assessments of the acquired 5G waveforms.
- DSP HDL Toolbox: Uses the interactive DSP HDL IP Designer app to customize, configure DSP algorithms, and generate HDL code and verification components.
- Simulink Control Design: Design and implement nonlinear and data-driven control techniques such as sliding mode and iterative learning control.
- System Composer: Edit sub-setted views; describe system behaviour with activity and sequence diagrams.
- Embedded Coder: As of R2024b, the SoC Blockset Support Package for Infineon® AURIX™ Microcontrollers has been merged into the Embedded Coder Support Package for Infineon® AURIX™ TCx4 Microcontrollers.

IV. RESULT AND ANALYSIS

➢ *Results on Performance Comparison of Majorly Utilized ML Classification Techniques*

The Machine Language (ML) classification techniques majorly reviewed and programmed for optimum performance were Logistic Regression, Support Vector Machine (SVM), Naive Bayes and Random Forest.

Table 2 Comparative Results of ML Classification Techniques

Classifier	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	78.261	65.455	53.731
Support Vector Machine	79.13	69.388	50.746
Naive Bayes	74.348	56.01	55.224
Random Forest	74.348	56.01	55.224

The performance of major machine learning classification techniques Logistic Regression, Support Vector Machine (SVM), Naive Bayes, and Random Forest was evaluated based on accuracy, precision, and recall metrics. Logistic Regression, a robust statistical method for binary classification, demonstrated simplicity and interpretability, achieving an accuracy of 78.26%, precision of 65.46%, and recall of 53.73%. Its strength lies in clearly identifying influential features and providing probabilistic outputs.

Support Vector Machine (SVM) showed competitive performance with an accuracy of 79.13%, precision of 69.39%, and recall of 50.75%. SVM is effective for both classification and regression, particularly in complex, non-linearly separable datasets, using kernels such as radial basis function (RBF) to map data into higher dimensions. The choice of kernel and parameter tuning significantly impacts its performance. Naive Bayes, known for its computational efficiency and scalability, assumes feature independence, simplifying calculations. It achieved an accuracy of 74.35%, precision of 56.01%, and recall of 55.22%. Despite its naïve assumption, it performs well for certain data types, especially text-related tasks. Random Forest, an ensemble method combining multiple decision trees, offers high accuracy and robustness against overfitting. It reported accuracy, precision, and recall metrics matching those of Naive Bayes (74.35%, 56.01%, and 55.22%, respectively), benefiting from majority voting and handling missing data effectively. Overall, SVM was selected as the preferred classifier for this study due to its superior overall accuracy and precision in handling the PIMA Indian diabetes dataset.

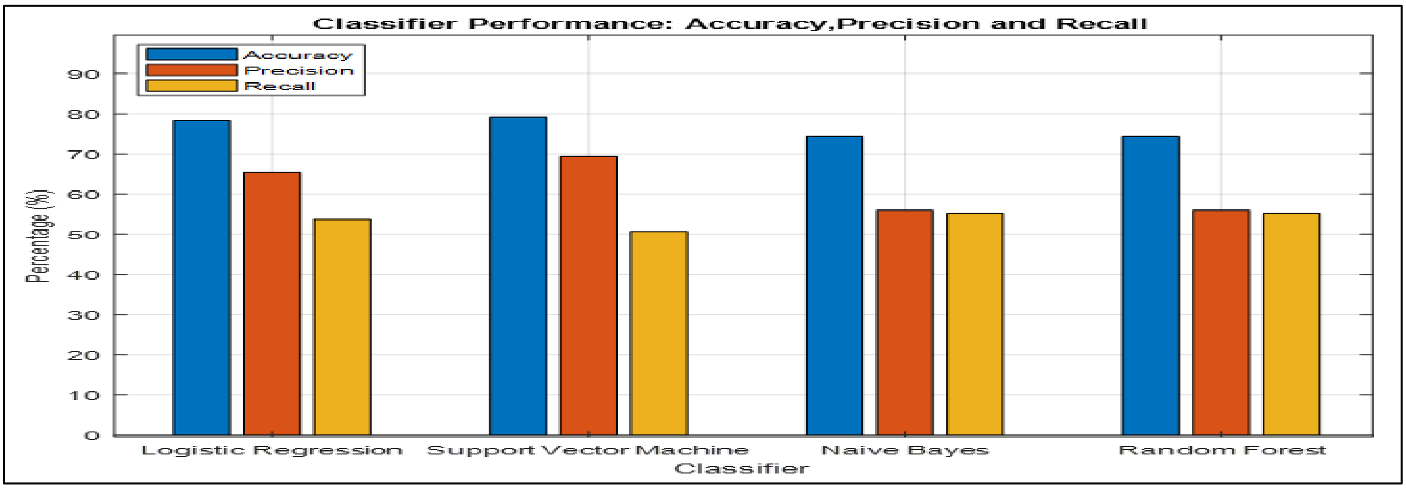

Fig 3 Comparative Analysis of ML Classifiers

- ➢ *Performance Comparison of SVM Hyperparameter's Optimization Accuracy*

The performance comparison of the accuracy of SVM hyperparameters optimization using grid search, Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) techniques shows that ACO optimized SVM shows the optimal accuracy and precision of 83.0435 % and 80.9524%. The results on other SVM optimized classifier are shown in Table 3 and illustrated in Figure 4.

Table 3 Performance of SVM Optimized Classifiers

Classifier	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
Grid search optimized SVM	80.208	78.713	60.241
GA optimized SVM	81.510	62.687	80.000
PSO optimized SVM	78.385	52.985	78.022
ACO optimized SVM	83.0435	80.9524	65.3846

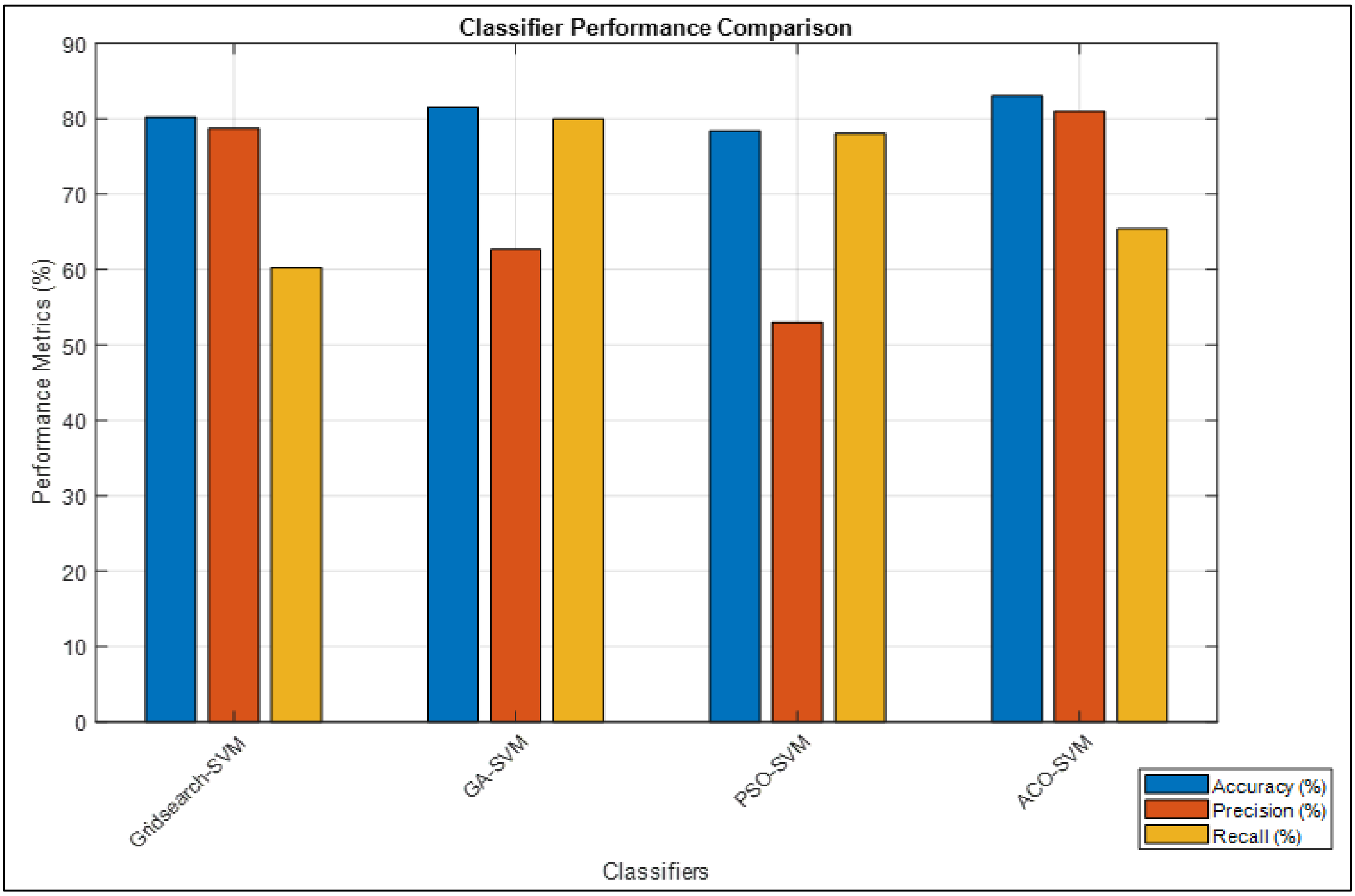

Fig 4 SVM Optimized Classifiers Performance

- ➢ *Performance Comparison of an ACO-optimized SVM on Pheromone Update Techniques*

The research study generated an ACO-optimized SVM for the improved pheromone update technique. The algorithm for the newly generated pheromone update technique is depicted in equation 25 & 26.

$$\tau_c=(1\text{-}\rho)\tau_c+\ \alpha\times Memory_C+\ \beta\times\Delta_{\tau C} \qquad (26)$$

$$\tau_s=(1\text{-}\rho)\tau_S+\ \alpha\times Memory_s+\ \beta\times\Delta_{\tau S} \qquad (27)$$

Where

τ_c is the pheromone level for box constraint C

τ_S is the pheromone level for the kernel scale.

ρ is the evaporation rate.

The comparative analysis of the ACO-optimized SVM with a Default Pheromone Update Technique (DPUT) and ACO-optimized SVM with an Improved (proposed) Pheromone Update Technique (IPUT) shows that IPUT reflected higher performance characteristics. The Accuracy, Precision and Recall are 83.0435 % 80.9524 % and 65.3846 % respectively. This is comparatively illustrated in Figure 5.

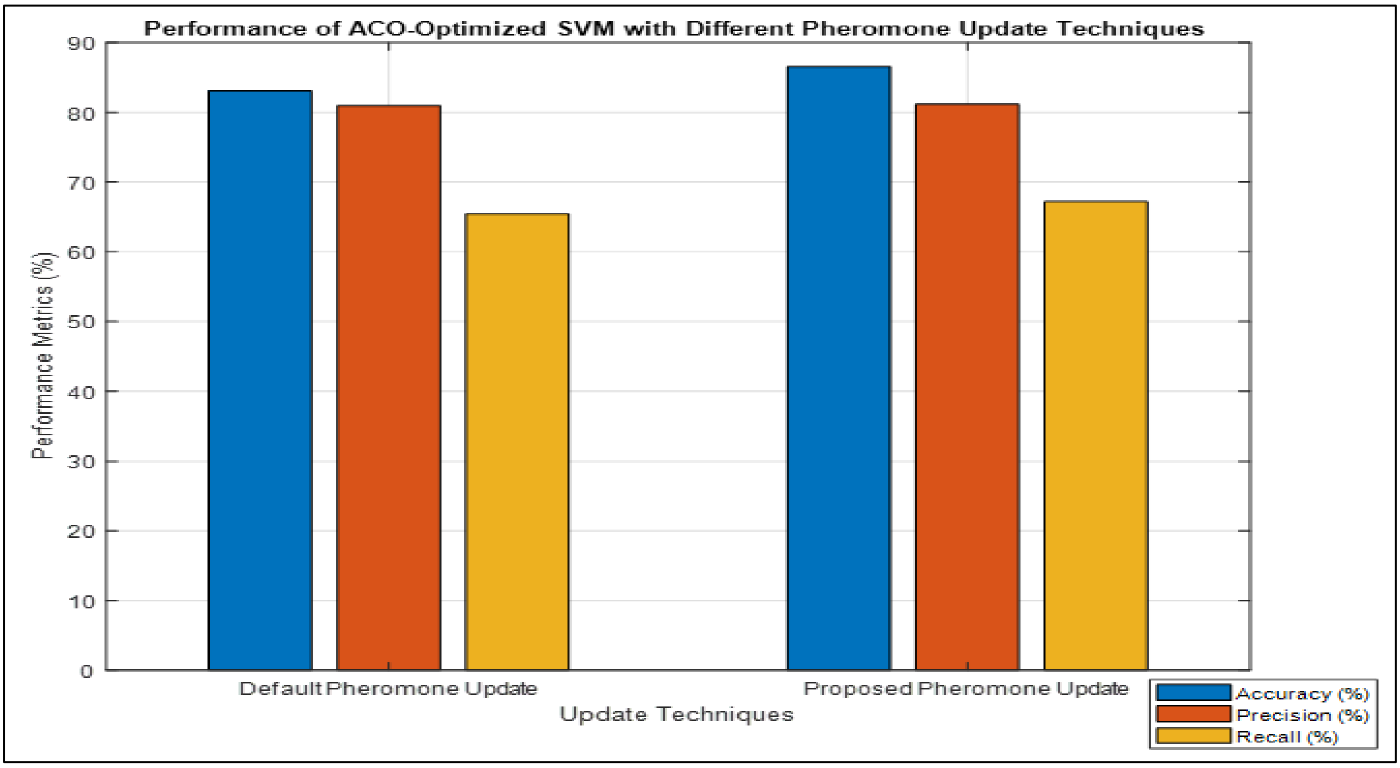

Fig 5 Performance Comparison of an ACO-optimized SVM on Default Pheromone Update Technique and Improved (Proposed) Pheromone Update Techniques

- *Performance Comparison of IPUT with Similar Published Researches*

The comparative analysis of the ACO-optimized SVM with an Improved (proposed) Pheromone Update Technique (IPUT) generated was carried out with respect to similar published research works. The ACO optimized SVM with the improved (proposed) pheromone update technique has Accuracy, Precision and Recall of 86.520 %, 81.130 % and 67.187 %, respectively. All the analysis were run using the MATLAB code as illustrated in in Appendices A1 and A2.

Table 4 Performance Comparison with some Similar Published Works (PIMA dataset)

SVM Classifier	Accuracy (%)	Precision	Recall (%)
Reza, *et. al*., 2023	85.5	87	83.4
Anggoro & Permatasari, 2023. (polynomial kernel)	80	65	74
Shrestha, *et. al.*, 2023	85.64	-	-
ACO optimized SVM with the proposed pheromone update technique	86.52	81.13	67.187

V. CONCLUSION AND RECOMMENDATION

This study successfully enhanced the performance of Support Vector Machine (SVM) classifiers for diabetes diagnosis using an improved pheromone update technique within the Ant Colony Optimization (ACO) framework. By initializing pheromone trails and incorporating memory, the algorithm enabled ants to select optimal SVM parameters, balancing exploration and exploitation while leveraging historical accuracy. This hybrid approach effectively converged to the best parameter combination including KernelScale resulting in superior classification accuracy. The proposed improved pheromone update technique (IPUT) outperformed both the default pheromone update and similar published methods, achieving notable gains in accuracy, precision, and recall on the PIMA Indian Diabetes dataset. These results confirm the viability and effectiveness of metaheuristic optimization to enhance machine learning classifiers in medical diagnosis tasks.

- *Recommendations*

Building on these findings, future research should explore the application of the improved ACO-optimized SVM approach to other medical classification problems and diverse datasets to validate its generalizability. Additionally, investigating other metaheuristic algorithms such as Particle Swarm Optimization and Genetic Algorithms in combination with advanced hyperparameter tuning could further enhance classifier performance. Research could also focus on integrating feature selection techniques to reduce dimensionality and improve interpretability.

REFERENCES

[1]. Abdulghani, B. A., & Abdulghani, M. A. (2024). A comprehensive review of ant colony optimization in swarm intelligence for complex problem solving. *Acadlore Transactions on Machine Learning*, *3*(4), 214-224.

[2]. Addissouky, T. A., Ali, M. M., El Sayed, I. E. T., & Wang, Y. (2024). Type 1 diabetes mellitus: retrospect and prospect. *Bulletin of the National Research Centre*, *48*(1), 42.

[3]. Agliata, A., Giordano, D., Bardozzo, F., Bottiglieri, S., Facchiano, A., & Tagliaferri, R. (2023). Machine learning as a support for the diagnosis of type 2 diabetes. *International Journal of Molecular Sciences*, *24*(7), 6775.

[4]. Ahmed, U., Issa, G. F., Khan, M. A., Aftab, S., Khan, M. F., Said, R. A., ... & Ahmad, M. (2022). Prediction of diabetes empowered with fused machine learning. *Ieee Access*, *10*, 8529-8538.

[5]. Ahmed, Z., Mohamed, K., Zeeshan, S., & Dong, X. (2020). Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. *Database*, *2020*, baaa010.

[6]. Aiad, B. A. E., Zarif, K. B., Gadallah, Z. M., & Abd EL-kareem, H. (2021, August). Support vector machine kernel functions comparison. In *The International Undergraduate Research Conference* (Vol. 5, No. 5, pp. 84-88). The Military Technical College.

[7]. Aisha, H., Nashiya, F., Shyma, Z., Farooqui, S., & Zulekha, S. (2024). Salivary Glucose as a Potential Biomarker for Monitoring Blood Glucose Levels in Type 2 Diabetes Mellitus: Current Insights and Future Prospects. *Indian Journal of Pharmacy Practice*, *17*(2).

[8]. Akinyelu, A. A., Ezugwu, A. E., & Adewumi, A. O. (2020). Ant colony optimization edge selection for support vector machine speed optimization. *Neural Computing and Applications*, *32*(15), 11385-11417.

[9]. Alghlayini, S., Al-Betar, M. A., & Atef, M. (2025). Enhancing non-invasive blood glucose prediction from photoplethysmography signals via heart rate variability-based features selection using metaheuristic algorithms. *Algorithms*, *18*(2), 95.

[10]. Almahdawi, A., Naama, Z. S., & Al-Taie, A. (2022, December). Diabetes prediction using machine learning. In *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)* (pp. 186-190). IEEE.

[11]. Almufti, S. M., Shaban, A. A., Ali, Z. A., Ali, R. I., & Fuente, J. D. (2023). Overview of metaheuristic algorithms. *Polaris Global Journal of Scholarly Research and Trends*, *2*(2), 10-32.

[12]. Alsarhan, A., Alauthman, M., Alshdaifat, E. A., Al-Ghuwairi, A. R., & Al-Dubai, A. (2023). Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. *Journal of Ambient Intelligence and Humanized Computing*, *14*(5), 6113-6122.

[13]. Al-Shourbaji, I., Helian, N., Sun, Y., Alshathri, S., & Abd Elaziz, M. (2022). Boosting ant colony optimization with reptile search algorithm for churn prediction. *Mathematics*, *10*(7), 1031.

[14]. Anggoro, D. A., & Permatasari, D. (2023). Performance comparison of the kernels of support vector machine algorithm for diabetes mellitus classification. *International Journal of Advanced Computer Science and Applications*, *14*(1).

[15]. Choudhary, K. (2024). Ant colony optimization-based models for agriculture price forecasting: Innovations, case studies, and future prospects. In *Optimization Algorithms-Classics and Recent Advances*. IntechOpen.

[16]. Darvishpoor, S., Darvishpour, A., Escarcega, M., & Hassanalian, M. (2023). Nature-inspired algorithms from oceans to space: A comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones. *Drones*, *7*(7), 427.

[17]. Dórea, J. V. F., Borges, W. R., & Ferracioli, P. R. B. (2024). The main diseases related to type 2 diabetes mellitus: a scoping review. *Scientia. Technology, Science and Society*, *1*(2), 17-27.

[18]. El Mrabet, M. A., El Makkaoui, K., & Faize, A. (2021, December). Supervised machine learning: A survey. In *2021 4th International conference on advanced communication technologies and networking (CommNet)* (pp. 1-10). IEEE.

[19]. Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of computational methods in engineering*, *29*(5).

[20]. Gamba, J. (2024). Basic Approaches in Object Detection and Classification by Deep Learning. In *Deep Learning Models: A Practical Approach for Hands-On Professionals* (pp. 1-45). Singapore: Springer Nature Singapore.

[21]. Josephine, P. K., Prakash, V. S., & Divya, K. S. (2021). Supervised learning algorithms: A comparison. *Kristu Jayanti Journal of Computational Sciences (KJCS)*, 01-12.

[22]. Kalita, D. J., Singh, V. P., & Kumar, V. (2020). SVM hyper-parameters optimization using multi-PSO for intrusion detection. In *Social Networking and Computational Intelligence: Proceedings of SCI-2018* (pp. 227-241). Singapore: Springer Singapore.

[23]. Khaltaev, N., & Axelrod, S. (2021). Global trends in diabetes-related mortality with regard to lifestyle modifications, risk factors, and affordable management: a preliminary analysis. *Chronic diseases and translational medicine*, *7*(3), 182-189.

[24]. Kumari, V. A., & Chitra, R. (2013). Classification of diabetes disease using support vector machine. *International Journal of Engineering Research and Applications*, *3*(2), 1797-1801.

[25]. Manakkadu, S., & Dutta, S. (2024). Ant Colony Optimization based Support Vector Machine for Improved Classification of Unbalanced Datasets. *Procedia Computer Science*, *237*, 586-593.

[26]. Manakkadu, S., & Dutta, S. (2024). Ant Colony Optimization based Support Vector Machine for Improved Classification of Unbalanced Datasets. *Procedia Computer Science*, *237*, 586-593.

[27]. Mansouri, S., Boulares, S., & Chabchoub, S. (2024). Machine learning for early diabetes detection and diagnosis. *J Wirel Mob Netw Ubiquitous Comput Dependable Appl*, *15*(1), 216-30.

[28]. Mehrotra, D. (2019). *Basics of artificial intelligence & machine learning*. Notion Press.

[29]. Morales, E. F., & Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In *Biosignal processing and classification using computational learning and intelligence* (pp. 111-129). Academic Press.

[30]. Pirouz, B., & Pirouz, B. (2023). Multi-Objective Models for Sparse Optimization in Linear Support Vector Machine Classification. *Mathematics*, *11*(17), 3721.

[31]. Pociot, F., & Lernmark, Å. (2016). Genetic risk factors for type 1 diabetes. *The Lancet*, *387*(10035), 2331-2339.

[32]. Poldrack, R. A., Huckins, G., & Varoquaux, G. (2020). Establishment of best practices for evidence for prediction: a review. *JAMA psychiatry*, *77*(5), 534-540.

[33]. Reza, M. S., Hafsha, U., Amin, R., Yasmin, R., & Ruhi, S. (2023). Improving SVM performance for type II diabetes prediction with an improved non-linear kernel: Insights from the PIMA dataset. *Computer Methods and Programs in Biomedicine Update*, *4*, 100118.

[34]. Rongali, S., & Yalavarthi, R. (2015, September). Parameter optimization of support vector machine by improved ant colony optimization. In *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 1* (pp. 671-678). New Delhi: Springer India.

[35]. Saravanan, K., Prakash, R. B., Balakrishnan, C., Kumar, G. V. P., Subramanian, R. S., & Anita, M. (2023, December). Support vector machines: Unveiling the power and versatility of svms in modern machine learning. In *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 680-687). IEEE.

[36]. Shrestha, B., Alsadoon, A., Prasad, P. W. C., Al-Naymat, G., Al-Dala'in, T., Rashid, T. A., & Alsadoon, O. H. (2022). Enhancing the prediction of type 2 diabetes mellitus using sparse balanced SVM. *Multimedia Tools and Applications*, *81*(27), 38945-38969.

[37]. Singh, E., & Pillay, N. (2023, July). A study of ant-based pheromone spaces for generation perturbative hyper-heuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 84-92).

[38]. Taylor, R. (2024). Understanding the cause of type 2 diabetes. *The Lancet Diabetes & Endocrinology*, *12*(9), 664-673.

[39]. Thabit, Q. Q., Fahad, T. O., & Dawood, A. I. (2022, September). Detecting diabetes using machine learning algorithms. In *2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)* (pp. 131-136). IEEE.

[40]. Thanmaikora. (2023). introduction of machine learning. Accessed online August 20, 2025 at https://medium.com/@thanmaikora/introduction-of-machine-learning-168cc163b6b8

[41]. Tsai, C. A., & Chang, Y. J. (2023). Efficient selection of Gaussian Kernel SVM parameters for imbalanced data. *Genes*, *14*(3), 583.

[42]. Uddin, A. M., & Ali, A. (2023, December). Diabetes prediction with machine learning techniques. In *2023 Global Conference on Information Technologies and Communications (GCITC)* (pp. 1-6). IEEE.

[43]. Xia, M., Liu, K., Feng, J., Zheng, Z., & Xie, X. (2021). Prevalence and risk factors of type 2 diabetes and prediabetes among 53,288 middle-aged and elderly adults in China: A cross-sectional study. *Diabetes, Metabolic Syndrome and Obesity*, 1975-1985.

[44]. Zhou, X., Gui, W., Heidari, A. A., Cai, Z., Liang, G., & Chen, H. (2023). Random following ant colony optimization: Continuous and binary variants for global optimization and feature selection. *Applied Soft Computing*, *144*, 110513.

[45]. Zhu, B., & Shi, Y. (2023). MKL-$ L_ {0/1} $-SVM. *arXiv preprint arXiv:2308.12016*.