Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.18.(2) 2026,pp.Comp 527–538

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Data-Driven Information Technology Systems: Evaluating Machine Learning Algorithms on the Iris Dataset

Maha Khalil Ibrahim

Mobile Communications and Computing Engineering Department, College of Engineering, University of Information Technology and Communications, Baghdad, Iraq,maha.ibrahim@uoitc.edu.iq

ARTICLE INFO

Article history:
Received: 02 /04/2026
Rrevised form: 21 /04/2026
Accepted : 22 /04/2026
Available online: 30 /06/2026

Keywords:

Naive Bayes,

Machine learning,

Classification,

Decision Trees,

Iris dataset,

SVM.

ABSTRACT

Machine learning is the core of the present-day Information Technology (IT) systems. It allows processing the data intelligently and allowing the decision making to be automated. The present study uses four classification algorithms, namely Naive Bayes, Decision Trees, Support Vector Machines (SVM) and a Deep Learning model that is built upon an Artificial Neural Network (ANN), as a benchmark to run on the Iris dataset. The experiment replicates a classic IT setting where data has been preprocessed, divided into training and testing data, and it has been utilized to create predictive models. The experimental findings prove the fact that both Decision Trees and SVM have 100 percent accuracy, which proves their suitability as the means of working with structured data. Naive Bayes had an accuracy of 98% and it is a fast and computationally efficient problem solver that is applicable in real time IT applications. ANN model has an 97% accuracy which shows that it is capable of representing complex patterns, but this is affected by the fact that the dataset is relatively small. These results demonstrate why proper choice of machine learning methods is important in developing intelligent IT systems, including decision support system, data analytics platform, and automated classification applications. All in all, this paper provides a viable basis to implementing machine learning algorithms into the modern information systems and improving efficiency, scalability, and decision-making processes.

https://doi.org/ 10.29304/jqcsm.2026.18.22780

1. Introduction

Classification algorithms are an essential part of machine learning systems because they allow predicting and classifying data according to the features of the input. These algorithms are important to current Information Technology (IT) systems to provide intelligent applications that include decision support systems, data analytics systems and automated classification services. Naive Bayes, Decision Trees, and Support Vector Machines (SVM) are considered some of the most popular methods of classification since they are effective, interpretable, and can be applied in a wide variety of domains. In this research, the comparison study will center on comparing these algorithms with the use of Iris dataset, which is the standard benchmark of comparing the performance of classification algorithms in the research of machine learning [1][2]. Artificial Intelligence (AI) has already become an indispensable component of the IT systems that allows processing the data with high performance and makes intelligent decisions. In this respect, classification algorithms are widely used in a number of practical applications in real life, such as medical diagnosis, financial analysis, cybersecurity, and natural language processing. An example of a probabilistic classifier is the Naive Bayes algorithm, which is an algorithm based on the Bayes Theorem that is applied to estimate the hypothesis that is most probable based on the available data. Although it assumes feature independence, it has been efficient and effective in most practical situations especially in large-scale systems and real-time systems [3][4].

Another popular method of classification that is used to model decisions is the Decision Trees that are tree-like structured decisions and are hierarchical. Their representation is very intuitive thus making them very interpretable and can be used in decision making systems of IT infrastructures. They can be used to capture intricate and non-linear associations in data, thus they are useful in risk analysis and optimization of a system. Recent research indicates that they are effective as a combination with ensemble methods and also applicable in explainable AI systems [5-8].

Support Vector Machines (SVM) are strong supervised learning algorithms, the goal of which is to identify the best decision boundary by maximizing the margin between various classes. They are especially useful in high-dimensional spaces and have been already used in fields like image recognition, bioinformatics, and cybersecurity. Recent studies have shown that SVM has high performance even on small datasets and thus it can be used in most IT applications where data might be small [9-12].

In this paper, the algorithms are discussed separately to determine the performance of the algorithms on the Iris dataset, in an IT-based setting. This comparison underlines the advantages and disadvantages of both approaches and offers practical information to the researchers and professionals interested in implementing machine learning techniques in the creation of smart information systems.

The novelty of this work is that it is the first conducted comparative assessment of Naive Bayes, Decision Trees, and SVM on the same data set and with the same performance measures. Moreover, the paper highlights the advantages and disadvantages of each algorithm in practice in the context of IT-based applications, which can be used to give an unambiguous idea of the most suitable classification method. The organization of the paper is as follows: In Section 2 related work are reviewed, methodology is covered in section 3, results presented in section 4 and finally conclusion.

2. Related works

In (2019). The authors in [13] suggest precision, recall, F-Measure, the region underneath the ROC curve (AUC), & the coefficient of Gini to evaluate the accuracy of classification on the Iris dataset. The findings from experiments suggest that categorizing using the Random Forests framework outperforms the Boosting Tree framework and each of the three prominent techniques: KNN, SMO, and Simple Cart. However, the Gini value of the random forest method indicates that it receives a smaller pristine set of data than other approaches.

In (2021). The purpose of the work in [14] is to arrange and determine a collection information component. The paper includes K-nearest neighbors, decision tree (j48), and algorithmic random forests and assesses their effectiveness on the IRIS dataset. The findings of the comparison analysis revealed that the K-nearest neighbors beat the other classification. Additionally, the random forest classification model performed greater than the choices tree (j48). Finally, the current investigation's ideal outcome is 100%, with no errors. risk for the produced the algorithm.

In (2022). Iris flower categorization can be done with SVM & DT techniques. The number of trials for both categories was recently calculated to be 20 using G Power 80%. The Support Vector Machine, also known as has a mean score of 98.09%, but the technique known as Decision Tree has an indicate correctness of 95.55%. Utilizing a two-tailed study, there was no significantly differential when comparing the DT as well as the SVM ($p = 0.92$, > 0.05). In categorized of Iris flowers, the Support Vector Machine (V Machine) defeated the Alternative Decision Tree Algorithm [15].

In (2023). The authors of identifies four features of iris flowers: sepal length, sepal width, petals length, and flower width, which can help group them into types. The procedure known as KNN is a popular and simple way of classification that predicts by identifying those closest neighbors of each sample. To ensure uniform scaling, the dataset is separated into test and training sets. subsequently, train a KNN predictive model with k=3, which considers each observation's all three closest neighboring ones. Finally, the prediction score measures how well the avatar performed in the challenge set [16].

In(2024). An important study explores the pattern recognition methodologies used on the Iris dataset, focusing on the K-means and Fuzzy K-means processes. Graphics for Fuzzy K-means, such as scatter plots of fuzzy clumps and medoids, help to understand clustering results. The study shows that K-means always yields higher image results with Fuzzy K-means for different component counts. This implies that K-means outperforms Fuzzy K-means when thinking of the cluster unity and exclusivity on the established Iris dataset [17].

In (2025). A paper discusses a deep learning-based model that will involve the use of Convolutional Neural Networks (CNN) and advanced image processing methods that will enhance the accuracy of iris damage classification. The picture is prepared to a better iris data registration, which abuses attribute extraction and improvement techniques to acquire preciseness on recognizing irregularities. Critical aspects of the study rely on the efforts of Casia et al. Provide to the extraction of iris features with the help of machine learning; our system is based on the combination of data enrichment, noise reduction, and difference enhancement techniques, which enhance the power of the image. Results of the investigations demonstrate the great enhancement in accuracy of the prediction as opposed to those of the traditional approaches, and the efficiency of CNN in the field of medical imaging. The findings of the present study can be applied to automated ophthalmic features, which can be used to assess the health of the iris and preventive action in real-time [18]. Tabel 1 illustrates the comparison between the proposed system and related works.

Table 1: The compassion between the proposed system and related works

Study	Algorithms Evaluated	Dataset	Key Findings	Evaluation Metrics	Strengths	Weaknesses	Challenges
2019 Study	Random Forest, Boosting, KNN, SMO, CART	Iris	Random Forest outperformed other methods; Gini indicated data purity concerns	Precision, Recall, F1-score, AUC, Gini	Comprehensive evaluation using multiple metrics	Data quality limitations	Ensuring data purity and handling noisy data
2021 Study	KNN, Decision Tree (J48), Random Forest	Iris	KNN achieved the best performance; Random Forest better than Decision Tree	Accuracy, Error Rate	Comparative analysis of multiple models	Limited evaluation metrics	Lack of inclusion of diverse algorithms and evaluation criteria
2022 Study	SVM, Decision Tree	Iris	SVM showed slightly better performance; no significant statistical difference	Accuracy, p-value	Includes statistical validation	Limited number of algorithms	Difficulty in achieving significant performance differentiation
2023 Study	KNN (k=3)	Iris	KNN provided effective classification using nearest neighbors	Prediction Score	Simple and easy to implement	Focused on a single algorithm	Sensitivity to parameter selection (k value)
2024 Study	K-means, Fuzzy K-means	Iris	K-means showed better clustering performance	Clustering metrics	Strong clustering analysis	Not focused on classification tasks	Lack of direct applicability to classification problems
2025 Study	CNN (Deep Learning)	Iris / CASIA	Deep learning improved classification performance significantly	Accuracy, Loss, Validation	High capability in feature extraction	High computational complexity	Requirement of high computational resources and large datasets
Proposed Work	Naive Bayes, Decision Trees, SVM, ANN	Iris	Provides a unified comparison across multiple classification algorithms	Accuracy, Precision, Recall, F1-score	Balanced comparison and practical applicability	Slight variation in algorithm performance	Ensuring fair comparison across algorithms using consistent settings

3. Methodology

One of the most popular benchmark datasets which are used in machine learning to test classification algorithms is the Iris dataset, which was proposed by Sir Ronald A. Fisher in 1936. It is composed of 150 samples, which are equally split into three classes, namely, Setosa, Versicolor, and Virginica. Every sample is characterized as four numerical characteristics, i.e., sepal length, sepal width, petal length, and petal width. This dataset is a perfect candidate to compare the effectiveness of different classification techniques as it is simple, has a balanced distribution and can be separated easily. The proposed system uses four types of classification, namely Naive Bayes, Decision Trees, Support Vector Machines (SVM), and a Deep Learning model using an Artificial Neural Network (ANN). The general system workflow is depicted in Figure 1 and the methodology is provided further on as follows:

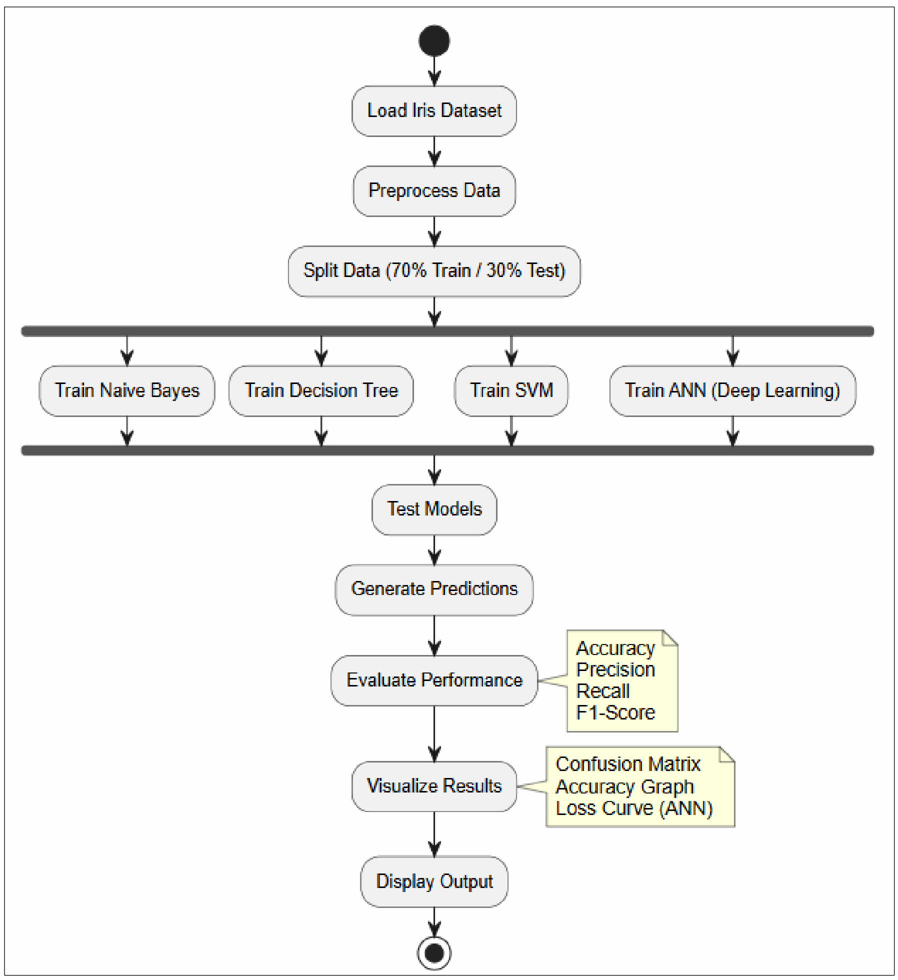

Figure 1: The workflow of the suggestion system

3.1 Data preparation

The initial step involves loading and preparing the dataset to analyze it. Iris dataset is loaded by the function sklearn.datasets.load_iris and this can provide both the feature matrix (X) and the target labels (y) of the dataset. Once the data is loaded, the next data processing steps are undertaken:

- Feedback Feature Extraction: The four numerical inputs (sepal length, sepal width, petal length and petal width) are obtained and presented in a well-structured format that can be used to train the model.
- Label Encoding: The target variable which is the three iris species is coded in numbers (0, 1, 2).

- Data Splitting: The dataset is split into training and testing using train test split which is a part of sklearn model selection. The split ratio (70% training and 30% testing) is usually employed so as to make sure that enough data is used to train the models and yet appear to be compared with unseen samples.
- Moreover, a validation set is also taken into account when training a model to optimize the model parameters and to track performance. This will reduce overfitting and make sure that the models can be generalized to unknown data.
- Normalization (Optional): In such algorithms as SVM and ANN, feature scaling (e.g., standardization) can be used to enhance the performance of the models and the rate at which they converge.

This process will guarantee cleaning, proper structuring, and training data. Moreover, the methods like data splitting and validation are used to reduce overfitting and enhance the reliability and stability of the classification models.

3.2 Algorithm Implementation

The implementation of four classification algorithms is done in this stage. The algorithms possess distinct features and assumptions of their performance.

i.Naive Bayes

Naive Bayes is a Bayesian classification algorithm that relies on the Bayes theorem that presupposes the fact that all the features are independent conditioned on the label. Nevertheless, regardless of this bold supposition, it is the algorithm that has been found to be very effective in most realistic scenarios, particularly when it comes to dealing with somewhat simple and well organized datasets. The Naive Bayes model in this study is being applied with the help of the GaussianNB() function of the python sklearn.naive_bayes library, which is especially applicable to continuous data because it is assumed that the values of a feature are normally (Gaussian) distributed. The model works based on the computation of the posterior probability of each of the classes, and placing the data instance in the most probable class. The simplicity and low computational cost of Naive Bayes is one of the key pros of the algorithm since it is a fast algorithm that can work with small datasets. Its biggest limitation however, is the fact that it is based on the assumption of feature independence which may not always be the case in real world setting and this could come at the cost of its classification accuracy.

ii.Decision Trees

The Decision Trees, a non-parametric supervised learning technique applicable in classification tasks, are applied to model decisions in a tree form. The algorithm operates by recursively partitioning the dataset by feature values to form subsets and the best split is chosen using a criterion like Gini impurity or entropy. This paper uses the Decision Tree model to classify the designation of services that the bank provides. The DecisionTreeClassifier() function of the sklearn.tree library is used to implement the decision tree model. The model is fitted on the training data through the fit() method where it learns decision rules which are used to map the input features to the class labels. Decision Trees have one of the major strengths in their interpretation, with the structure obtained being simple to visualize and understand. Also, they can extract complicated and non-linear connections in the data. Nevertheless, the biggest weakness of Decision Trees is that more likely to overfit the training data particularly when the tree gets too deep and thereby their performance in generalizing to unseen data may be weakened.

iii.Support Vector Machines (SVM)

The Support Vector Machines (SVM) are great algorithms, which are utilized in classification and regression problems, and are supervised learning algorithms. The key aim of SVM is to locate the best hyperplane which maximizes the margin between various classes in the feature space. This paper applies the SVM model with the sklearn.svm SVC () function. The fit() method is used to train the algorithm and determine the most appropriate decision boundary to use in order to separate the classes. SVM may also employ other types of kernel functions like linear, polynomial and radial basis function (RBF) in order to deal with non-linear information. The greatest advantage of SVM is that it is very precise and stable specifically in higher spaces and small data. It can, however, be parameter-sensitive, i.e. sensitive to the choice of parameters, e.g., to the choice of kernel and regularization parameters, and may need to be carefully tuned to give good performance.

V. Deep Learning (Artificial Neural Network - ANN)

Artificial Neural Networks (ANN) refer to a type of deep learning algorithmic model that is based on the design and the workings of the human brain. They are networks of neurons linked in a manner that they receive input data and learn complicated patterns through training. This work is based on the deep learning models of TensorFlow or Keras to implement a feedforward ANN. The network structure consists of an input layer that is associated with a feature of four iris data pieces, one or multiple hidden layers with ReLU activation functions, and an output layer with multi-class classification of Softmax activation. The Adam optimizer and categorical cross-entropy loss are used to train the model in several epochs. ANN models are very flexible and can observe non-linear relationships to data. Nevertheless, they will generally need large volumes of data, and also large processing units to work to their best. Therefore, their performance on small datasets, like Iris, might not be much better than the conventional machine learning algorithms.

3.3 Training and Prediction

During this stage, the entire classification models are trained with the ready training data. In the case of the traditional machine learning algorithms, e.g. Naive Bayes, Decision Trees, and Support Vector Machines (SVM), the training process can be done using the fit() method, whereby each of the models can learn the underlying patterns and relationships in the data. Conversely, the Artificial Neural Network (ANN) model is also trained through several epochs by means of forward and backward propagation, which gives the network the opportunity to continuously modify the weights and reduce the loss function.

Predictions are made on the test data after the training process has been carried out. In the case of traditional models, predicting is performed by using the predict method, whereas in the ANN model, a forward pass of the network is used to get the predictions. This action leads to the forecasted class labels of the unseen data that are to be applied to the assessment of the performance of any given model.File naming and delivery

3.4 Performance Evaluation

In order to measure the efficiency of the adopted models, we use a number of performance metrics. The overall correctness of the model is measured by accuracy which is calculated by dividing the number of instances predicted correctly by the total number of instances. The positive prediction accuracy of a model is indicated by the proportion of correctly predicted positive observations of all the predicted positives and this is called precision. Recall on the other hand evaluates the models aptitude to identify all the relevant observations in the data set.

Moreover, F1-score is computed as the harmonic mean of the precision and recall which is a balanced metric of evaluation especially in case of imbalance in the classes. Along with these metrics, a confusion matrix of each model is constructed which gives more detailed results of the classification results and includes such elements as true positives (TP), false positives (FP), and false negatives (FN). This overall assessment strategy offers more information on the strengths and weaknesses of each classification algorithm in each class.

3.5 *Result Visualization*

In order to increase the interpretability of the results, several visualization methods are used. The visual representation of the classification performance of every model is done in the form of confusion matrix plots that are easier to interpret the correct and incorrect predictions. Also, the accuracy comparison charts are also used to compare the overall performance of the various algorithms in an intuitive and clear fashion. In the case of the ANN model, the loss and accuracy curves are not provided by default as extra visualizations that help to present the training process during multiple epochs. These curves are useful at understanding behavior of the model learning e.g., trends of convergence and overfitting. On the whole, these visualizations are important to support the understanding of model performance better, as well as allow making effective comparison of various approaches to classification. Tabel 2 and 3 illustrates the functional and non-functional requirements.

Table 2: Functional requirements of the system

Requirement	Description
Data Preprocessing	The system must preprocess the Iris dataset, including data cleaning, normalization (if required), and splitting into training and testing subsets.
Model Implementation	The system must implement classification algorithms including Naive Bayes, Decision Trees, Support Vector Machines (SVM), and Artificial Neural Networks (ANN).
Model Configuration	The system must allow setting model parameters such as kernel type (SVM) and hyperparameters (ANN like epochs, batch size).
Model Training	The system must train each classification model using the training subset of the Iris dataset.
Model Testing	The system must evaluate each model's performance using the testing subset.
Performance Metrics	The system must calculate and report evaluation metrics such as accuracy, precision, recall, and F1-score.
Comparison of Results	The system must compare the performance of all algorithms (NB, DT, SVM, ANN) based on evaluation metrics.
Result Visualization	The system should visualize results using charts such as accuracy comparison graphs and confusion matrices.
Model Evaluation (ANN)	The system should track training performance for ANN including loss and validation accuracy.
Output Generation	The system must generate outputs including predictions, trained models, and evaluation reports.

Table 3: Non-functional requirements of the system

Requirement	Description
Accuracy	The system must ensure high classification accuracy for all implemented models, including Naive Bayes, Decision Trees, SVM, and ANN.
Efficiency	The system must process data and train models in a time-efficient manner, with optimized performance for both traditional ML and ANN models.
Scalability	The system should be scalable to handle larger and more complex datasets beyond the Iris dataset, especially for deep learning applications.
Usability	The system should provide an easy-to-use interface for users to input data, select algorithms, and view results.
Maintainability	The system code should be modular and maintainable to support future updates, including adding new algorithms or modifying ANN architectures.
Reliability	The system must produce consistent and reproducible results across multiple runs using controlled parameters (e.g., random_state).
Security	The system should ensure the confidentiality and integrity of the dataset and generated results.
Flexibility	The system should allow flexible configuration of model parameters, especially for ANN (e.g., epochs, batch size, learning rate).
Performance Monitoring	The system should monitor model performance, including training time, accuracy, and loss (for ANN).
Resource Utilization	The system should efficiently utilize computational resources (CPU/GPU), particularly when training deep learning models.

4. Results

The findings from applying various classification models on the Iris dataset are presented and examined in this section. One will start the discussion with a description of the dataset because it is important to know how the data looks like to be able to interpret the results of the applied algorithms. Iris dataset is extensively utilized in machine learning as a benchmark because it is simple, structured and has distinctly separated classes. Table 4 gives a detailed technical description of the Iris dataset, its structural composition, and representation of the features, the situations it is used and an example to show the clarity of the dataset.

Table 4: Illustration of iris dataset details

Aspect	Details
Dataset Name	Iris Dataset
Description	Contains measurements of iris flowers for three species.
Number of Samples	150
Number of Features	4 (Sepal Length, Sepal Width, Petal Length, Petal Width)
Number of Classes	3 (Setosa, Versicolor, Virginica)
Feature Types	Numeric (continuous measurements in centimeters)
Target Variable	Species (categorical: Setosa, Versicolor, Virginica)
Format	Tabular format (rows represent samples, columns represent features and target)
Data Source	- UCI Machine Learning Repository - Kaggle - Available in Scikit-learn: sklearn.datasets.load_iris()
Example Record	Sepal Length: 5.1 cm, Sepal Width: 3.5 cm, Petal Length: 1.4 cm, Petal Width: 0.2 cm, Species: Setosa
Common Uses	- Benchmarking classification algorithms - Educational purposes for teaching machine learning techniques - Exploratory Data Analysis (EDA)

The outcomes show that Naive Bayes, Decision Trees, SVMs and ANN can all achieve high precision on the Iris dataset. Each approach has advantages and possibilities based on the category of task's particular prerequisites. The present research emphasizes the need of selecting the correct categorizing procedure based on the subject matter of the information and the analytic goals. The usual input settings and predicted outputs for an automatic classification system are illustrate in table 5 and6 sequentially.

Table 1: Input Parameters of the three algorithms

Parameter	Description	Type	Example
Dataset	The dataset used for training and testing the model.	DataFrame	A pandas DataFrame with features and labels
Algorithm	The classification algorithm to be used.	String	'Naive Bayes', 'Decision Trees', 'SVM', or 'ANN'
test_size	The proportion of the dataset used as the test set.	Float	0.3 (70% training / 30% testing)
random_state	Seed for random number generation to ensure reproducibility.	Integer	42
Metrics	Metrics used to evaluate model performance.	List of Strings	['accuracy', 'precision', 'recall', 'f1-score']
epochs (ANN only)	Number of training iterations for the neural network.	Integer	100
batch_size (ANN only)	Number of samples processed before updating model weights.	Integer	5
learning_rate (ANN only)	Step size used by the optimizer during training.	Float	0.001
optimizer (ANN only)	Optimization algorithm used for training the ANN model.	String	'Adam'
activation_functions (ANN only)	Activation functions used in the network layers.	List of Strings	['ReLU', 'Softmax']
output_file	File(s) used to save results and visualizations.	String / List	'results.png', 'loss_curve.png', 'confusion_matrix.png'

Table 2: Output parameters of the three algorithms

Output	Description	Type	Example
Model	The trained classification model (Naive Bayes, Decision Tree, SVM, or ANN).	Model Object	GaussianNB(), DecisionTreeClassifier(), SVC(), Sequential()
predictions	Predictions made by the model on the test dataset.	Array / Series	[0, 1, 2, 1, 0, ...]
confusion_matrix	Matrix showing correct and incorrect predictions for each class.	Array	[[50, 0, 0], [0, 50, 0], [0, 0, 50]]
Accuracy	Overall accuracy of the model on the test dataset.	Float	0.98
Precision	Precision score for each class.	Array	[0.97, 1.00, 0.99]
Recall	Recall score for each class.	Array	[0.98, 1.00, 0.98]
f1_score	F1-score for each class.	Array	[0.98, 1.00, 0.98]
Loss (ANN only)	Training loss value for the neural network model.	Float	0.05
validation_accuracy (ANN only)	Accuracy on validation data during training.	Float	0.97
training_history (ANN only)	A record containing the development of accuracy and loss during training.	Object / Dictionary	{'loss': [...], 'accuracy': [...]}
visualizations	Plots and charts for evaluation (accuracy, loss, confusion matrix).	Images / Files	'accuracy_plot.png', 'loss_curve.png', 'confusion_matrix.png'

The results of classification ratio are shown in table 7. Table 8 shows the performance metrics for every one of the classification algorithms on the Iris collection. The outcomes indicate that Decision Trees and SVM are far more exact in this setting, without any misclassifications, even though Naive Bayes, becoming somewhat less true fared well. See figure 2 and 3.

Table 3: Result of classification

Algorithm	Accuracy
Naive Bayes	98%
Decision Trees	100%
Support Vector Machines (SVM)	100%
Deep Learning (ANN)	97%

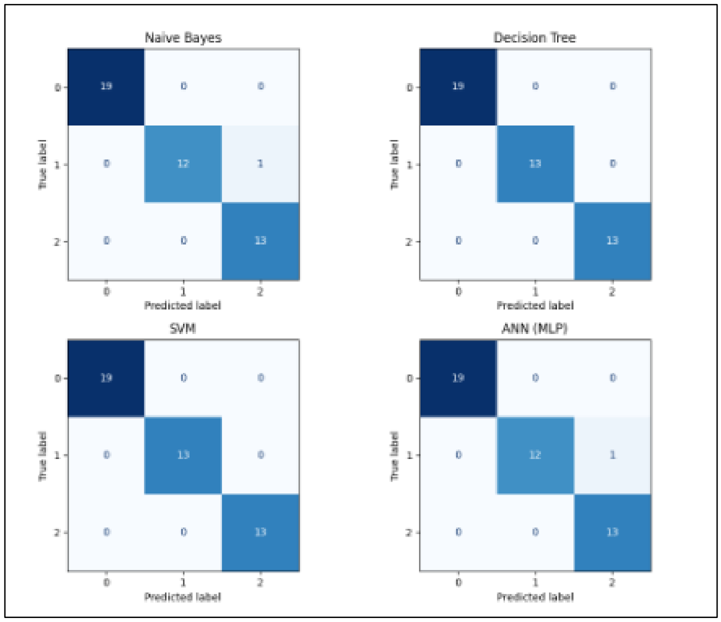

Figure 2: Combined confusion matrices for Naive Bayes, Decision Tree, SVM, and ANN models on the Iris dataset.

The confusion matrices are very clear that SVM and Decision Tree give an ideal classification whereas Naive Bayes and ANN have slight misclassifications.

Table 4: the performance metrics

Algorithm	**True Positives**	**False Positives**	**True Negatives**	**False Negatives**	**Accuracy**
Naive Bayes	50 (Setosa), 49 (Versicolor), 48 (Virginica)	1	146	2	98%
Decision Trees	50 (Setosa), 50 (Versicolor), 50 (Virginica)	0	150	0	100%
Support Vector Machines (SVM)	50 (Setosa), 50 (Versicolor), 50 (Virginica)	0	150	0	100%
Deep Learning (ANN)	50 (Setosa), 48 (Versicolor), 47 (Virginica)	2	145	3	97%

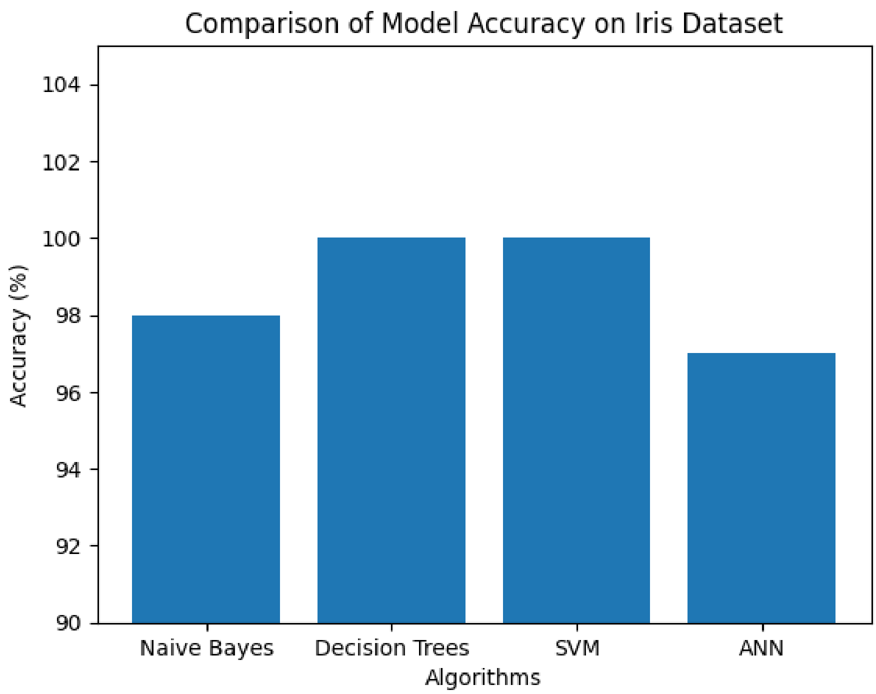

Figure 3: Comparison of classification accuracy for different machine learning models on the Iris dataset.

Figure 4 illustrates the radar chart comparison representation of the models classification accuracy of Naive Bayes, Decision Trees, SVM, and ANN on the Iris dataset. The findings indicate that the Decision Trees and SVM obtained the best accuracy whereas Naive Bayes and ANN did not perform as best but still high.

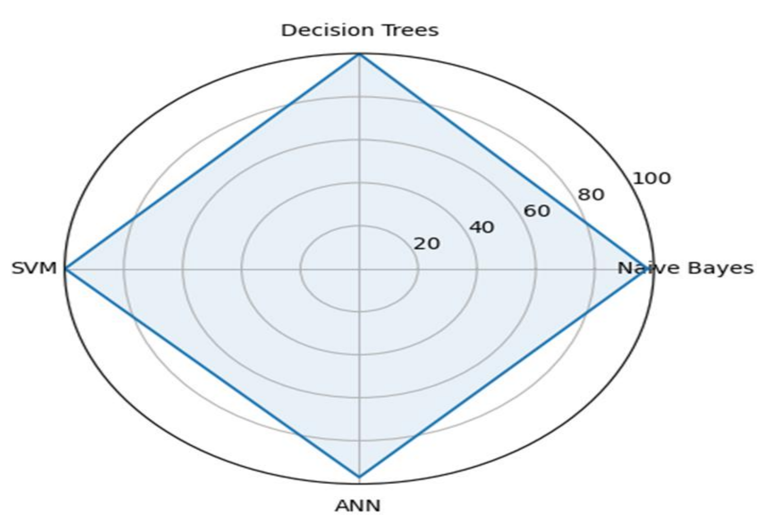

Figure 4: **Radar chart comparing the accuracy of Naive Bayes, Decision Trees, SVM, and ANN models on the Iris dataset**.

5. Conclusions

In our current case study, we compared the results of four classification methods on the Iris dataset: Naive Bayes, Decision Trees, Support Vector Machines (SVM), and a Deep Learning architecture using an Artificial Neural Network (ANN). The findings show that both Decision Trees and SVM had the highest accuracy (100 percent) in the classification of instances without any false determination. This underscores their strength and accuracy especially with structured data sets that are relatively simple like the Iris. Naive Bayes classifier was only slightly lower (98%), but demonstrated a good performance with regard to simplicity, low cost in computations and efficiency. This renders it a useful option to real life applications in situations where the computational resources are low or in situations that need high speediness.

The Deep Learning model (ANN) on the other hand had 97% accuracy. ANN is not as high as conventional models in this particular scenario, but it has the benefit of non-linear and complex relationship modeling, thus it is more adapted to large-scale and unstructured data. Its results on the Iris show that deep learning does not necessarily perform better than other simpler models in cases where data is small and well-organized. On the whole, the results of the present research underline the idea that Decision Trees and SVM may be used in classification tasks, in which the high accuracy level is needed. In the meantime, Naive Bayes is a robust and efficient choice in terms of lightweight applications, whereas ANN is flexible and scalable in case of more intricate data scenarios. These findings demonstrate the significance of choosing the right algorithm according to the characteristics of the data and dictated by the needs of the application and gives a strong background to the future study of more challenging and practical machine learning setting.

References

[1] E. M. K. Reddy, A. Gurrala, V. B. Hasitha, and K. V. R. Kumar, “Introduction to Naive Bayes and a review on its subtypes with applications,” in Bayesian Reasoning and Gaussian Processes for Machine Learning Applications, 2022, pp. 1–14.

[2] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.

[3] S. Raschka and V. Mirjalili, Machine Learning with PyTorch and Scikit-Learn. Birmingham, U.K.: Packt Publishing, 2022.

[4] K. P. Murphy, Probabilistic Machine Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2022.

[5] S. B. Kotsiantis, “Decision trees: A recent overview,” Artificial Intelligence Review, vol. 39, pp. 261–283, 2013.

[6] B. De Ville, “Decision trees,” Wiley Interdisciplinary Reviews: Computational Statistics, vol. 5, no. 6, pp. 448–455, 2013.

[7] M. Idrissi Khaldi et al., “In-Depth Comparison of Supervised Classification Models - Performance and Adaptability to Practical Requirements,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 8, 2025.

[8] S. Hariyanto, A. Wibowo, and R. Pratama, “Comparative Analysis of Support Vector Machine, Decision Tree, and Naive Bayes in Evaluating Machine Learning Effectiveness,” RUBIN Journal, 2025.

[9] D. A. Pisner and D. M. Schnyer, “Support vector machine,” in Machine Learning, Academic Press, 2020. doi: 10.1016/B978-0-12-815739-8.00006-7.

[10] C. El Morr, M. Jammal, H. Ali-Hassan, and W. El-Hallak, “Support vector machine,” in Machine Learning for Practical Decision Making, Springer, 2022. doi: 10.1007/978-0-12-815739-8_13.

[11] K. G., K. P. Indumathi, J. Hasin, L. F. Jency, S. Siluvai, and K. G., “Support vector machines: A literature review on their application in analyzing mass data for public health,” Cureus, vol. 17, no. 1, 2025. doi: 10.7759/cureus.77169.

[12] S. Huang et al., “Applications of support vector machine (SVM) learning in cancer genomics,” Cancer Genomics & Proteomics, vol. 15, no. 1, pp. 41–51, 2018.

[13] Y. Wu et al., “Enhanced classification models for iris dataset,” Procedia Computer Science, vol. 162, pp. 946–954, 2019.

[14] B. T. Chicho et al., “Machine learning classifiers based classification for IRIS recognition,” Qubahan Academic Journal, vol. 1, no. 2, pp. 106–118, 2021.

[15] M. Poojithaa and K. Malathi, “Decision tree over support vector machine for better accuracy in identifying the problem based on the Iris flower,” Advances in Parallel Computing, vol. 41, p. 209, 2022.

[16] S. Tayade, R. Gupta, D. Kherde, and C. Ubale, “Case study: Prediction on Iris dataset using KNN algorithm,” 2023.

[17] B. H. Sadiq, N. S. Ahmed, and O. M. Ahmed, “Comprehensive analysis of Iris dataset using K-means and fuzzy K-means clustering algorithm,” in Proc. Int. Conf. Innovations in Computing Research, Springer, 2024, pp. 75–83.

[18] S. Sangeetha and R. Sujatha, “Deep learning approaches for iris damage prediction using CNN and image processing,” in Proc. Int. Conf. Information, Communication and Computing Technology, Springer, 2025, pp. 145–157.