Performance Comparison of Support Vector Machine and Decision Trees for the Classification of Handwritten Digits

Bilikisu Temilade Azeez[1], Stephen Olatunde Olabiyisi[2], Modupe Oluwaseun Alade[3] and Isiaka Akinkunmi Adeyemo[4]

[1,2,4]Department of Computer Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

[3]Department of Cybersecurity, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150600265

Received: 19 July 2026; Accepted: 24 July 2026; Published: 03 August 2026

ABSTRACT

Handwritten digit recognition remains one of the fundamental applications of machine learning and pattern recognition due to its widespread use in banking, postal services, document processing and intelligent information systems. Although deep learning approaches have demonstrated remarkable performance, classical machine learning algorithms such as Support Vector Machine (SVM) and Decision Tree (DT) remain attractive because of their simplicity, computational efficiency and interpretability. This study evaluates and compares the performance of Support Vector Machine and Decision Tree classifiers for handwritten digit recognition using the Modified National Institute of Standards and Technology (MNIST) dataset. The dataset comprises 70,000 grayscale images of handwritten digits (0–9), each with a resolution of 28 × 28 pixels. Image preprocessing involved normalization of pixel values and standard data preparation before model implementation. Both classifiers were implemented using Python and the Scikit-learn library under identical experimental conditions, and model performance was evaluated using accuracy, precision, recall and F1-score. Experimental results showed that the Support Vector Machine outperformed the Decision Tree across all evaluation metrics. The SVM achieved an accuracy of 94.32%, precision of 94.41%, recall of 94.32%, and an F1-score of 94.35%, whereas the Decision Tree recorded 87.36% accuracy, 87.42% precision, 87.36% recall and 87.35% F1-score. The findings indicate that SVM provides superior classification performance for handwritten digit recognition, while Decision Tree offers faster implementation and greater interpretability. The study concludes that SVM is more suitable for applications requiring high recognition accuracy, whereas Decision Tree remains appropriate for applications where computational simplicity and model transparency are prioritized

Keywords: Handwritten Digit Recognition, Support Vector Machine, Decision Tree, MNIST Dataset, Machine Learning, Image Classification.

INTRODUCTION

Handwritten digit recognition has become an important research area in machine learning, pattern recognition and computer vision because of its extensive application in banking, postal automation, document digitization and intelligent information systems. The ability of computers to accurately recognize handwritten numerical characters has significantly improved the efficiency of automatic document processing while reducing human effort and operational errors. Despite advances in optical character recognition technologies, handwritten digits remain challenging because of the wide variations in individual writing styles, stroke thickness, orientation and image quality.

The Modified National Institute of Standards and Technology (MNIST) dataset has become the most widely accepted benchmark for evaluating handwritten digit recognition algorithms. The dataset contains 70,000 grayscale images of handwritten digits distributed across ten numerical classes, providing a standardized platform for comparing machine learning techniques under identical experimental conditions. Its balanced class

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

distribution and public availability have made it one of the most frequently used datasets in machine learning research.

Among the classical machine learning algorithms used for handwritten digit recognition, Support Vector Machine (SVM) and Decision Tree (DT) remain two of the most popular approaches because they employ fundamentally different learning mechanisms. Support Vector Machine performs classification by constructing an optimal hyperplane that maximizes the separation margin between classes, making it highly effective for high-dimensional image classification problems. In contrast, Decision Tree performs recursive partitioning of the feature space into homogeneous regions using simple decision rules, providing an interpretable classification model that is easy to understand and implement.

Although several studies have reported the effectiveness of these algorithms, many existing comparisons rely mainly on classification accuracy without considering other important performance measures such as precision, recall and F1-score. In addition, differences in preprocessing techniques, implementation environments and evaluation procedures often make comparisons across studies inconsistent. Consequently, there remains a need for a standardized comparative evaluation of Support Vector Machine and Decision Tree classifiers using identical datasets, preprocessing methods and evaluation metrics.

This study therefore evaluates and compares the performance of Support Vector Machine and Decision Tree classifiers for handwritten digit recognition using the MNIST dataset. Specifically, the study preprocesses the dataset, implements both classifiers using Python and Scikit-learn, and compares their performances using accuracy, precision, recall and F1-score. The outcome provides useful guidance for selecting appropriate machine learning algorithms for handwritten digit recognition and related image classification applications.

Related Work

Several researchers have applied machine learning algorithms to handwritten digit recognition using benchmark datasets such as the Modified National Institute of Standards and Technology (MNIST) dataset. Among the various classification techniques, Support Vector Machine (SVM) and Decision Tree (DT) have received considerable attention because of their effectiveness, computational efficiency and ease of implementation. This section reviews selected empirical studies relevant to the present research.

LeCun, Bottou, Bengio and Haffner (1998) introduced the MNIST dataset as a standardized benchmark for evaluating handwritten digit recognition algorithms. The dataset contains 70,000 grayscale images of handwritten digits distributed across ten classes and has become the most widely adopted benchmark for comparing machine learning models. Their work established a common experimental platform that enables researchers to evaluate different classifiers under identical conditions.

Cortes and Vapnik (1995) proposed the Support Vector Machine as a supervised learning algorithm based on statistical learning theory. The algorithm constructs an optimal separating hyperplane that maximizes the margin between different classes, thereby improving generalization performance. Due to its robustness in high-dimensional feature spaces, SVM has been successfully applied to handwriting recognition, image classification and text categorization.

Decoste and Schölkopf (2002) investigated the application of Support Vector Machines to handwritten digit recognition and reported that SVM achieved excellent classification performance on the MNIST dataset. Their findings demonstrated that appropriate feature scaling and kernel selection significantly improve recognition accuracy. The study established SVM as one of the most reliable classical machine learning algorithms for handwritten digit classification.

Breiman, Friedman, Olshen and Stone (1984) introduced the Classification and Regression Tree (CART) algorithm, which forms the theoretical foundation of modern Decision Tree classifiers. Decision Trees classify data by recursively partitioning the feature space into increasingly homogeneous subsets using decision rules derived from the training data. The algorithm is computationally efficient, highly interpretable and requires minimal data preprocessing.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Kotsiantis (2013) reviewed Decision Tree learning algorithms and observed that although Decision Trees are simple to implement and easy to interpret, they generally produce lower classification accuracy than Support Vector Machines when applied to image recognition problems. The study also emphasized that appropriate pruning strategies are necessary to reduce overfitting and improve generalization.

Patel and Thakore (2013) compared several supervised machine learning algorithms for image classification and reported that Support Vector Machine consistently outperformed Decision Tree in terms of classification accuracy. However, Decision Tree exhibited lower computational complexity and faster model construction, making it suitable for applications requiring rapid decision making.

More recently, Voloshchenko (2021) compared Logistic Regression, Decision Tree, Random Forest, k-Nearest Neighbour and Support Vector Machine for handwritten digit recognition. The study reported that Support Vector Machine achieved one of the highest classification accuracies among the evaluated classical machine learning algorithms, while Decision Tree produced comparatively lower predictive performance despite its computational simplicity.

The findings of Olatunji *et al*. (2025) align with broader research trends in chaos-enhanced optimization algorithms for image and handwriting recognition tasks. Also, Okunlola *et al* (2026) integrated chaotic maps into the Particle Swarm Optimisation (PSO) algorithm to enhance population diversity, prevent premature convergence and improve segmentation quality and Moronkeji *et al*. (2026) evaluated three chaos-enhanced PSO (CE-PSO) variants, Logistic+PSO, Sinusoidal+PSO and Gaussian+PSO, for image segmentation using multilevel thresholding on the Berkeley Segmentation Dataset (BSDS500).

The reviewed studies consistently indicate that Support Vector Machine generally provides superior predictive accuracy, whereas Decision Tree offers advantages in interpretability and computational efficiency. However, many previous studies primarily focused on classification accuracy without simultaneously evaluating precision, recall and F1-score under identical preprocessing conditions. Furthermore, differences in implementation procedures and evaluation protocols make direct comparison of reported results difficult.

This study addresses these limitations by implementing Support Vector Machine and Decision Tree classifiers using the same preprocessing pipeline, identical experimental environment and standardized evaluation metrics. The comparative analysis based on accuracy, precision, recall and F1-score provides a more comprehensive assessment of classifier performance for handwritten digit recognition using the MNIST dataset.

METHODOLOGY

The study adopted a comparative experimental design to evaluate the performance of Support Vector Machine (SVM) and Decision Tree (DT) classifiers for handwritten digit recognition. The methodology consisted of five sequential phases: dataset acquisition, data preprocessing, model implementation, model evaluation and performance comparison. Both classifiers were implemented under identical experimental conditions to ensure that any observed differences in performance resulted solely from the characteristics of the algorithms.

The methodology framework is summarized as follows:

Phase 1: MNIST Dataset Acquisition

↓

Phase 2: Data Preprocessing (Normalization, Feature Standardization and Train-Test Split)

↓

Phase 3: Model Implementation (Support Vector Machine and Decision Tree)

↓

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Phase 4: Model Evaluation (Accuracy, Precision, Recall and F1-Score)

↓

Phase 5: Performance Comparison and Analysis

Dataset Acquisition

The Modified National Institute of Standards and Technology (MNIST) dataset was used for this study. The dataset contains **70,000 grayscale images** of handwritten digits ranging from 0 to 9. Each image has a resolution of **28 × 28 pixels**, resulting in **784 numerical features** after flattening.

As shown in Table 3.1, the dataset consists of:

Table 3.1: MNIST Dataset Distribution

Dataset	Number of Images
Training Set	60,000
Testing Set	10,000
Total	70,000

The MNIST dataset was selected because it is publicly available, balanced across all digit classes, computationally efficient and widely accepted as the standard benchmark for handwritten digit recognition.

Data Preprocessing

Prior to model implementation, the dataset underwent preprocessing to improve learning performance and ensure consistency between the classifiers. The preprocessing steps included:

i. **Normalization:** Pixel intensity values were scaled from the range **0–255** to **0–1** by dividing each pixel value by 255.

ii. **Feature Standardization:** Standardization was applied to improve the performance of the Support Vector Machine by ensuring that features contributed equally during classification.

iii. **Train-Test Split:** The standard MNIST partition of **60,000 training images** and **10,000 testing images** was maintained for reproducibility and fair comparison.

Model Implementation

Two supervised machine learning algorithms were implemented using Python and the Scikit-learn library.

Support Vector Machine

The Support Vector Machine classifier employed a **linear kernel** with a regularization parameter (**C = 1.0**) using the One-vs-Rest multiclass classification strategy. The model classified handwritten digits by identifying the optimal separating hyperplane that maximized the margin between classes.

Decision Tree

The Decision Tree classifier was implemented using the CART algorithm with the **Gini impurity** criterion. To

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

minimize overfitting, the model was configured with a maximum tree depth of **30**, a minimum of **5 samples per split**, and **2 samples per leaf node**.

Experimental Environment

Both classifiers were implemented using **Python** within the **Scikit-learn** machine learning framework. Tables 3.2 and 3.3 present the experimental environment and hardware specifications for the implementation of the models.

Table 3.2: Experimental Environment

Component	**Specification**
Programming Language	Python
Machine Learning Library	Scikit-learn
Dataset	MNIST
Operating System	Windows
Development Environment	Visual Studio Code
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score

Both models were trained and evaluated under identical software and hardware conditions to ensure consistency throughout the experiments.

Table 3.3: Hardware Specifications

Component	**Specification**
Processor	Intel Core i7-10750H @ 2.60GHz
RAM	16 GB DDR4
Storage	512 GB SSD
GPU	NVIDIA GeForce GTX 1650 (optional for acceleration)

Performance Evaluation

The performance of the classifiers was evaluated using four standard classification metrics.

Accuracy measures the proportion of correctly classified handwritten digits among all predictions.

Precision measures the proportion of correctly predicted samples relative to all samples predicted for a particular class.

Recall measures the ability of the classifier to correctly identify all instances belonging to each digit class.

F1-Score represents the harmonic mean of precision and recall and provides a balanced measure of classification performance.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

These evaluation metrics were selected because they provide a comprehensive assessment of classifier effectiveness and enable objective comparison between Support Vector Machine and Decision Tree models.

RESULTS AND DISCUSSION

Experimental Results

The performance of the Support Vector Machine (SVM) and Decision Tree (DT) classifiers was evaluated using the MNIST handwritten digit dataset under identical experimental conditions. Both models were assessed using four standard classification metrics: accuracy, precision, recall and F1-score. These metrics provide a comprehensive evaluation of each classifier's predictive performance.

Table 4.1: Performance Comparison of Support Vector Machine and Decision Tree

Performance Metric	**Support Vector Machine**	**Decision Tree**
Accuracy (%)	94.32	87.36
Precision (%)	94.41	87.42
Recall (%)	94.32	87.36
F1-Score (%)	94.35	87.35

The results demonstrate that the Support Vector Machine consistently outperformed the Decision Tree across all evaluation metrics. The SVM achieved an accuracy of **94.32%**, representing an improvement of **6.96 percentage points** over the Decision Tree, which achieved **87.36%** accuracy. Similar improvements were observed for precision, recall and F1-score, confirming the superior classification capability of the SVM for handwritten digit recognition.

DISCUSSION OF RESULTS

The superior performance of the Support Vector Machine can be attributed to its ability to construct an optimal separating hyperplane with maximum margin between different digit classes. Since each handwritten digit image consists of 784 numerical features, the high-dimensional nature of the MNIST dataset favours the SVM, which is specifically designed to perform well in high-dimensional feature spaces. The linear kernel adopted in this study effectively separated the handwritten digit classes while maintaining good generalization on unseen test samples.

The Decision Tree classifier also produced satisfactory classification performance, achieving an accuracy of **87.36%**. Its relatively lower performance is mainly due to its tendency to generate overly specific decision boundaries, making it more susceptible to overfitting when handling complex image data. Nevertheless, the Decision Tree offers significant advantages in terms of model interpretability and computational simplicity because the classification process can be represented using straightforward decision rules.

The precision values obtained indicate that the Support Vector Machine generated fewer false-positive classifications than the Decision Tree. Likewise, the higher recall achieved by the SVM demonstrates its greater ability to correctly identify handwritten digit samples across all classes. Consequently, the higher F1-score recorded by the SVM confirms that it maintains a better balance between precision and recall.

Confusion Matrix Analysis

The confusion matrices further revealed that most handwritten digits were correctly classified by both models, with the majority of observations concentrated along the principal diagonal. However, a small number of misclassifications occurred between visually similar digits such as **4 and 9**, as well as **7 and 1**. These misclassification patterns are consistent with previous studies on handwritten digit recognition and arise from similarities in handwriting styles rather than deficiencies in the classifiers themselves.

Compared with the Decision Tree, the Support Vector Machine exhibited fewer off-diagonal classification errors, indicating stronger discrimination between visually similar digit classes.

Comparison with Previous Studies

The findings of this study agree with previous research reporting that Support Vector Machines generally outperform Decision Trees in handwritten digit recognition tasks. The superior classification accuracy achieved by the SVM supports earlier findings by Cortes and Vapnik (1995), Decoste and Schölkopf (2002), and Patel and Thakore (2013), who demonstrated that margin-based classifiers possess stronger generalization capability for high-dimensional image classification problems.

Although the Decision Tree achieved lower predictive performance, its simple structure, ease of interpretation and relatively low computational requirements make it a practical choice for applications where transparency and rapid implementation are more important than achieving maximum classification accuracy.

Overall, the experimental results demonstrate that Support Vector Machine provides the most reliable performance for handwritten digit classification using the MNIST dataset. The higher accuracy, precision, recall and F1-score obtained by the SVM indicate its suitability for applications requiring high recognition accuracy, while the Decision Tree remains useful for systems where model simplicity and interpretability are primary considerations.

CONCLUSION

This study presented a comparative evaluation of Support Vector Machine (SVM) and Decision Tree (DT) classifiers for handwritten digit recognition using the Modified National Institute of Standards and Technology (MNIST) dataset. Both classifiers were implemented under identical experimental conditions to ensure a fair comparison, and their performances were assessed using four standard evaluation metrics: accuracy, precision, recall and F1-score.

The experimental results showed that the Support Vector Machine consistently outperformed the Decision Tree across all evaluation metrics. Specifically, the SVM achieved an accuracy of **94.32%**, precision of **94.41%**, recall of **94.32%** and an F1-score of **94.35%**, whereas the Decision Tree recorded an accuracy of **87.36%**, precision of **87.42%**, recall of **87.36%** and an F1-score of **87.35%**. The superior performance of the SVM demonstrates its effectiveness in handling the high-dimensional feature space of handwritten digit images and its strong ability to generalize to unseen data.

Although the Decision Tree produced lower classification performance, it remains an effective classifier because of its simplicity, ease of interpretation and relatively low computational complexity. These characteristics make it suitable for applications where model transparency and rapid implementation are more important than maximizing prediction accuracy.

Overall, the findings of this study establish that Support Vector Machine is the more suitable algorithm for handwritten digit recognition on the MNIST dataset when high classification accuracy is required. The study also provides a practical comparison of two widely used supervised machine learning algorithms and contributes to the growing body of knowledge on handwritten digit recognition using classical machine learning techniques.

Future studies may extend this work by evaluating additional machine learning and deep learning models, including Random Forest, k-Nearest Neighbour, Convolutional Neural Networks (CNNs) and Vision Transformers. Further investigations may also consider larger handwritten character datasets, advanced feature

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

extraction techniques and hybrid learning approaches to improve recognition performance and computational efficiency.

REFERENCES

1. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
2. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Wadsworth International Group.
3. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.
4. Decoste, D., & Schölkopf, B. (2002). Training invariant support vector machines. Machine Learning, 46(1–3), 161–190.
5. Demšar, J. (2006). Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research, 7, 1–30.
6. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification (2nd ed.). John Wiley & Sons.
7. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow (3rd ed.). O'Reilly Media.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
9. Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
10. Kotsiantis, S. B. (2013). Decision trees: A recent overview. Artificial Intelligence Review, 39(4), 261–283.
11. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.
12. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
13. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.
14. Moronkeji, I. O., Olabiyisi, S. O., & Adedeji, O. T. (2026). A comparative study of selected chaos maps to improve particle swarm optimisation in image segmentation tasks. EIRA Journal of Multidisciplinary Research and Development, 2(4). 17-26.
15. Murphy, K. P. (2022). Probabilistic Machine Learning: An Introduction. MIT Press.
16. Okunlola, S. O., Baale, A. A., & Olabiyisi, S. O. (2026). Performance Evaluation of Selected Chaos-Enhanced Particle Swarm Optimisation for Image Segmentation. *Engineering and Technology Journal, 11*(7), 10904–10911. https://doi.org/10.47191/etj/v11i07.13
17. Olatunji, Babatunde Lekan, Olabiyisi, Stephen Olatunde, Oyeleye, Christopher Akinwale and Omotade, Adedotun Lawrence, 2025. "An Enhanced Chicken Swarm Optimization Algorithm Using Gaussian and Tent Chaotic Map Functions," International Journal of Research and Innovation in Applied Science, International Journal of Research and Innovation in Applied Science (IJRIAS), vol. 10(7), pages 653-664.
18. Patel, H. R., & Thakore, D. G. (2013). A survey on feature extraction techniques for image classification. International Journal of Computer Applications, 70(9), 1–7.
19. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
20. Python Software Foundation. (2024). Python Language Reference. https://www.python.org
21. Scikit-learn Developers. (2024). Scikit-learn User Guide. https://scikit-learn.org
22. Vapnik, V. N. (1998). Statistical Learning Theory. John Wiley & Sons.
23. Zhang, C., & Ma, Y. (2012). Ensemble Machine Learning: Methods and Applications. Springer.