### Analysis of Results and Insights

#### Implemented SVM :

* **Accuracy:** 86%
* **Performance Evaluation:**
  + The manually implemented Support Vector Machine (SVM) serves as an educational tool, demonstrating the fundamental mechanics of SVMs.
  + It employs a simpler quadratic programming approach, which, while conceptually insightful, lacks the robustness and optimization of advanced libraries.
  + Its lower performance and scalability limitations make it unsuitable for real-world applications or competitive environments.

#### Built-in SVM (Optimized Parameters):

* **Accuracy:** 98%
* **Key Factors for Success:**
  + The model achieves near-perfect performance due to the use of optimized hyperparameters: C=10C=10C=10, gamma=’scale’\text{gamma}=\text{'scale'}gamma=’scale’, and an RBF kernel.
  + This configuration allows the model to balance bias and variance effectively, adapting to the dataset's complexity.
  + However, **SVM demonstrates longer training times compared to Neural Networks** on this dataset, especially as the data size scales up. This is attributed to the computational complexity of solving the quadratic optimization problem inherent in SVMs.
  + Despite its efficiency in terms of implementation and resource usage for inference, the time taken during training can be a bottleneck in practical scenarios.

#### Random Forest (Optimized Parameters):

* **Accuracy:** 97%
* **Advantages and Trade-offs:**
  + The Random Forest algorithm provides robust and consistent performance, excelling in scenarios with mixed feature types.
  + It is naturally resistant to overfitting due to its ensemble nature but slightly underperforms compared to SVMs in numerical datasets like MNIST.
  + Unlike SVMs, Random Forest does not require feature scaling, making it a convenient choice for broader datasets with minimal preprocessing.

#### Neural Network (Optimized Parameters):

* **Accuracy:** 98%
* **Performance Characteristics:**
  + Matches the built-in SVM in terms of accuracy, with high precision, recall, and F1-scores.
  + While Neural Networks are computationally intensive, they **outperform SVM in training speed** for this dataset, due to efficient backpropagation algorithms and parallelization capabilities.
  + Neural Networks require significant computational resources and hyperparameter tuning but prove advantageous in capturing complex patterns efficiently over time.

### Conclusion and Recommendations

#### Best Models:

* Both the **Built-in SVM** and **Neural Network** stand out as the top-performing models, achieving 98% accuracy on the MNIST dataset.

#### Recommendation:

* Considering the dataset's purely numerical nature and balancing computational efficiency, accuracy, and training time:
  + **Neural Networks** are advantageous for faster training, particularly for large-scale datasets or iterative experimentation.
  + However, **Built-in SVM** is preferred for straightforward implementation, slightly lower computational requirements for inference, and comparable accuracy. **Its slower training time should be factored into decisions when scalability and time efficiency are critical.**