International Journal of Academic Information Systems Research (IJAISR)
ISSN: 2643-9026
Vol. 10 Issue 7 July - 2026, Pages: 63-73

An Artificial Intelligence-Based Hybrid Framework for Adaptive Sorting in Real-Time Supply Chain Automation

Deema Abdalla Shaqfa, Samy S. Abu-Naser

Department of Information Technology,
Faculty of Engineering and Information Technology,
Al-Azhar University, Gaza, Palestine

Abstract: *Sorting is a fundamental operation in computer science, used in databases, search engines, and machine learning. Classical sorting algorithms like QuickSort have been widely adopted due to their average-case efficiency. However, QuickSort suffers from a critical limitation: its worst-case time complexity degrades to $O(n^2)$ when a poor pivot is selected, particularly on sorted, reverse sorted, or nearly sorted data. Traditional pivot selection strategies rely on fixed rules that do not adapt to input characteristics. This paper proposes a hybrid approach that enhances QuickSort using a simple machine learning model for adaptive pivot selection. The method extracts statistical features from a small random sample of size $\sqrt{n}$, including distribution pattern, degree of sortedness, value range, and duplicate ratio. A lightweight classifier, trained offline, predicts a pivot position that leads to balanced partitions. We expect our approach to achieve 25–35% improvement on sorted and reverse sorted data, 20–25% on nearly sorted data, and 10–15% on duplicate-rich data, with only 1–2% overhead. The model is simple, easy to implement, and can be extended to other sorting algorithms.*

Keywords: Sorting, QuickSort, Artificial Intelligence, Machine Learning, Hybrid Algorithm, Pivot Selection.

Introduction:

Sorting is a basic thing in computer science and data science. We use it in many places like database management, search engines, big data analysis, and machine learning. Without good sorting algorithms, these applications would be much slower [1].

Over the years, many classic sorting algorithms came out. Examples include Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Heap Sort, and QuickSort. Each one has pros and cons. For example, Insertion Sort is simple and works well on small data, but it's slow on large data because its time complexity is $O(n^2)$. On the other hand, Merge Sort is always O(n log n) but needs extra memory [2].

Among these, QuickSort is one of the most used in practice. Why? Because it's fast on average (O(n log n)) and works in-place without needing extra memory. But QuickSort has a big problem. Its performance can drop to $O(n^2)$ in the worst case. The main reason is how we pick the pivot. Traditional QuickSort uses fixed rules to pick the pivot, like taking the first element, the last element, or the middle element. These rules don't look at the input data [3].

The thing is, these fixed rules work fine when the data is random, but they fail badly in some cases. Like when the data is already sorted, reverse sorted, or nearly sorted. For example, if the data is already sorted and we pick the first element as the pivot, the partition becomes very unbalanced. The left side ends up empty and the right side gets all the other elements. This gives $O(n^2)$ time [4].

This problem matters more today. Data is huge and comes in many different patterns. Real-time sensor data, financial transactions, social media posts, and genomic sequences can be partially sorted, reverse sorted, full of duplicates, or completely random. Fixed algorithms can't handle all this variety [5].

In recent years, artificial intelligence, especially machine learning, has come a long way. One important thing about machine learning is that it can learn from data and adapt to different patterns. That makes it a good fit to help classic algorithms become smarter and more adaptive [6].

So, this paper suggests a hybrid approach that mixes AI with classic sorting to make it better. We focus on improving pivot selection in QuickSort using a simple machine learning model. The main idea is to use the model to look at a small sample of the data and predict a good pivot that leads to balanced partitions, instead of sticking to fixed rules.

This research looks at QuickSort as a case study. Later on, this approach could be used for other sorting algorithms.

The paper is organized like this. Section (2) presents the objectives. Section (3) defines the problem. Section (4) reviews related work. Section (5) describes the proposed method. Section (6) discusses expected results. Section (7) provides discussion and challenges. Section (8) concludes the paper.

Objectives:

- **Main Objectives:**

The main goal of this paper is to make QuickSort faster by using a simple machine learning model to pick a better pivot.

- **Specific Objectives:**
  1- To understand why classical QuickSort sometimes becomes slow.

We look at how the pivot is chosen in normal QuickSort. We also need to know which types of data cause problems, like sorted data, reverse sorted data, and nearly sorted data.

2-To build a simple AI model that can suggest a good pivot.

The model takes a small random sample from the data. It looks at simple things like how the data is distributed and how sorted it already is. Then it predicts a pivot position that balances the partitions.

3- **To explain how the AI model and QuickSort work together.**

We describe the steps from sampling to sorting. We show where the AI model fits into the normal QuickSort process.

4- **To discuss how much improvement we expect and what we lose.**

We expect better performance on difficult data like sorted and reverse sorted arrays. We also need to consider the extra time the AI model takes, including sampling and prediction.

5- **To talk about the limits of our method and what can be done next.**

The model needs to be trained first on different types of data. The sample size must be chosen carefully. This idea could be used for other sorting algorithms in the future.

Problem Statement:

QuickSort is one of the most used sorting algorithms because it's fast on average and works in-place. But it has a known weakness. When we pick a bad pivot, performance drops from O(n log n) to O(n^2). Why? Because traditional QuickSort uses fixed rules to pick the pivot, like picking the first, last, or middle element. These rules don't look at the input data [7].

This problem gets worse with certain data types:

1. **Sorted data:** If the array is already sorted and we pick the first element as the pivot, each partition is very unbalanced.
2. **Reverse sorted data:** If we pick the last element as the pivot, the same problem happens.
3. **Nearly sorted data:** Performance is still bad because the pivot often ends up near one end.
4. **Data with many duplicates:** Standard QuickSort can also get slow.

In modern data science, data is often not random. It comes in patterns that cause problems for fixed pivot selection. Examples include financial transactions (often sorted by date), sensor readings (often nearly sorted), and genomic sequences (with many repeats). So fixed pivot selection isn't enough for today's diverse data [8].

This research focuses on QuickSort as a case study. Later, this approach could be extended to other sorting algorithms.

So, we need an adaptive method that changes pivot selection based on the input data. This research aims to solve this problem by using a simple machine learning model to predict a good pivot, making QuickSort more robust and efficient on different data types.

Literature Review:

A. QuickSort and Its Limitations

Since its introduction by Tony Hoare in 1962, QuickSort has remained one of the most widely used sorting algorithms due to its excellent average-case performance of O(n log n) and its in-place sorting capability [4]. However, as discussed earlier, QuickSort suffers from a critical weakness: its performance degrades to O(n^2) in the worst case when the pivot is chosen poorly. Traditional implementations use fixed pivot selection strategies, such as picking the first, last, or middle element, which do not adapt to the characteristics of the input data [2], [3].

To mitigate this issue, several heuristic-based improvements have been proposed over the years. The median-of-three method selects the median of the first, middle, and last elements as the pivot, reducing the probability of worst-case behaviour on already sorted data. The random pivot selection strategy chooses a random element, making worst-case inputs unlikely. Additionally, IntroSort (introduced by Musser in 1997) switches to HeapSort when the recursion depth exceeds a certain threshold, guaranteeing O(n log n) worst-case complexity. Despite these improvements, these strategies remain fixed and do not truly adapt to the input data's structure [3]. This limitation becomes particularly problematic in modern data science applications where data arrives in diverse and unpredictable patterns [5].

B. Artificial Intelligence in Algorithm Optimization

In recent years, researchers have begun exploring the use of artificial intelligence to optimize classical algorithms. A landmark achievement in this direction is AlphaDev, a deep reinforcement learning agent developed by DeepMind. AlphaDev formulates the task of discovering sorting algorithms as a single-player game, where the agent learns to assemble low-level CPU instructions into efficient algorithms. Remarkably, AlphaDev discovered sorting algorithms that outperform state-of-the-art human benchmarks. These algorithms have been integrated into the LLVM standard C++ library, replacing a component that had remained unchanged for over a decade. Specifically, AlphaDev achieved up to 70% faster performance for sorting short sequences and improved overall throughput by 1.7% [8], [4].

Another significant advancement is Learned Sort, introduced by Kristo et al. in 2020. Learned Sort uses machine learning models based on the cumulative distribution function (CDF) to predict the approximate position of each element in the sorted order. Instead of comparing elements pairwise like traditional algorithms, Learned Sort uses these predictions to place elements directly into buckets, then recursively sorts each bucket. This approach has been shown to outperform highly tuned sorting algorithms, including Radix Sort and comparison-based sorts, even when model training time is included in the total sorting time [7], [2]. Recent analysis has shown that Learned Sort can be viewed as a learning-augmented Sample Sort, where the CDF model selects the pivots [2].

C.Neural Network-Based Sorting

The use of neural networks for sorting has also been explored. NN-sort, proposed by Zhu et al. in 2019, is a neural network-based method that leverages data distribution awareness for faster sorting. Unlike traditional comparison-based algorithms, NN-sort uses a trained neural network model to map unordered data elements to their ordered positions. The algorithm operates iteratively: a "roughly ordered" array is generated in each iteration, and data conflicts (where two elements map to the same position) are resolved in subsequent iterations. Experimental results on both synthetic and real-world datasets demonstrated that NN-sort achieves performance improvements of 2.18x to 10.9x over traditional sorting algorithms [3], [8].

D. Machine Learning for Pivot Selection in QuickSort

Several studies have specifically focused on using machine learning to improve pivot selection in QuickSort. Kocamaz (2013) proposed a neural network-based model to select the most suitable sorting algorithm based on the characteristics of the input data. More recent work has explored using machine learning to predict optimal pivot positions in QuickSort by analysing statistical features extracted from a small sample of the data.

The key idea behind these approaches is to extract simple features from the input data, such as distribution pattern, degree of sortedness, and value range. A pre-trained classifier then predicts a pivot position that is likely to produce balanced partitions. Studies have shown that such adaptive pivot selection can improve performance by 20-30% on large datasets compared to traditional QuickSort, particularly for challenging data patterns like sorted or nearly sorted arrays [7].

E. Summary.

The literature clearly demonstrates that artificial intelligence has significant potential to improve classical sorting algorithms. While traditional methods rely on fixed strategies that do not adapt to input data, AI-enhanced approaches can learn from data and adjust their behaviour dynamically. AlphaDev represents a breakthrough in discovering entirely new algorithms from scratch, while Learned Sort and NN-sort demonstrate how machine learning models can be integrated into sorting pipelines to achieve substantial performance gains [7], [2].

However, most existing approaches are complex and require significant computational resources for training and inference. This paper builds on these works by proposing a simpler, lightweight hybrid approach that focuses specifically on improving pivot selection in QuickSort. Our method uses a small random sample and a simple classifier, making it easy to implement and integrate into existing systems while still achieving meaningful performance improvements on difficult data patterns.

Methodology:

A. Overview

Our hybrid approach combines QuickSort with a simple machine learning model for adaptive pivot selection. The process has three phases: (1) sampling and feature extraction, (2) pivot prediction using a pre-trained model, and (3) normal QuickSort with the predicted pivot.

B. Sampling Strategy

Before sorting, we take a small random sample from the data. The sample size is important. If it is too small, the prediction may be wrong. If it is too large, the overhead is high. Based on previous work, a sample size of about √n (*square root of array size*) gives a good balance.

C. Feature Extraction

From the sample, we extract these simple features:

1. Distribution pattern: How the values are spread.
2. Degree of sortedness: How close the sample is to being sorted.
3. Value range: The minimum and maximum values.
4. Duplicate ratio: How many duplicate values there are.

These features are easy to compute and tell us about the data.

D. Machine Learning Model

We use a simple classifier trained offline on different data types (random, sorted, reverse sorted, nearly sorted, with duplicates). The classifier takes the features as input and outputs a recommended pivot position. This could be a small decision tree or neural network.

E. Integration with QuickSort

To use our method, we first check if the array is large enough. For small arrays, we use normal QuickSort to avoid overhead. For large arrays, we:

- Take a random sample of size about √n.
- Extract features from the sample.
- Use the trained model to predict a good pivot.
- Use that pivot in normal QuickSort.

F. Training Phase

The model is trained offline once. For each training example, we try different pivot positions and see which gives the most balanced partitions. The model learns to connect certain features with good pivot choices.

Results:

This section presents the expected results of applying a hybrid AI approach to enhance classical sorting algorithms. The proposed method combines traditional sorting algorithms with a simple machine learning model to improve performance

A. Performance Comparison

We compare traditional QuickSort with our AI-enhanced version. The AI model helps by predicting a good pivot based on input data characteristics.

- **QuickSort with AI-based pivot selection:** Expected improvement of 25-35% on sorted and reverse sorted data. For random data, performance remains similar with less than 5% overhead.

B. Overhead Analysis

Adding AI to QuickSort introduces some overhead. This overhead comes from three steps:

1. **Sampling:** We take a random sample of size about √n (square root of the array size). This takes O(√n) time.
2. **Feature Extraction:** We compute simple statistics from the sample like distribution, sortedness, and duplicate ratio. This also takes O(√n) time.
3. **Prediction:** We run the machine learning model to predict a good pivot. This is very fast because the model is small.

For large arrays (e.g., 1 million elements), the sample size is about 1000 elements. The total overhead is expected to be about 1-2% of the total sorting time. For small arrays (less than 1000 elements), we can turn off the AI model and use normal QuickSort to avoid overhead.

Table 1: Expected Results by Data Type

Data Type	Traditional QuickSort	AI-Enhanced QuickSort	Improvement
Sorted	$O(n^2)$ slow	O(n log n) fast	30-35%
Reverse sorted	$O(n^2)$ slow	O (n log n) fast	30-35%
Nearly sorted	$O(n^2)$ slow	O(n log n) fast	20-25%
Random	O(n log n)	O(n log n)	5-10%
Duplicate-rich	Slow	Faster	10-15%
Small data (<1000)	Good	Similar	0-5%
Large data (>1M)	Good	Better	15-20%

Table 2. Comparison with Other Methods

Method	Key Achievement	Our Advantage
AlphaDev (DeepMind)	70% faster, very complex	Our approach is much simpler
NN-sort	Neural network selection, complex	Our approach is lightweight
Our hybrid approach	25-35% improvement, 1-2% overhead	**Simple, easy to implement**

Our hybrid approach is simpler and more practical than AlphaDev and NN-sort. We keep the original QuickSort structure and only improve pivot selection, making it easy to integrate into existing systems.

F. Summary of Expected Results

Based on our hybrid AI approach that combines QuickSort with a simple machine learning model for adaptive pivot selection, we expect the following outcomes:

1. 25-35% improvement on sorted and reverse sorted data compared to traditional QuickSort
2. 20-25% improvement on nearly sorted data
3. 10-15% improvement on duplicate-rich data
4. 5-10% improvement on random data
5. 1-2% overhead for sampling, feature extraction, and prediction
6. Better scalability for large datasets (over 1 million elements)

These expected results suggest that even a simple hybrid AI approach can significantly enhance QuickSort performance across different data types, with minimal additional cost. The most significant gains are achieved on the data patterns where traditional QuickSort performs poorly (sorted, reverse sorted, and nearly sorted data), while performance on random data remains almost the same with very little overhead.

Discussion:

B. Discussion of Our Hybrid Approach

Our hybrid approach has several good points. First, it keeps QuickSort simple and in-place while adding an adaptive way to pi ck pivots. Second, the AI model is lightweight with minimal overhead (1-2%). Third, the approach is flexible and can be used f or other sorting algorithms later.
But there are also limits. The model needs to be trained first on different types of data. If the input data has a pattern the model hasn't seen before, the prediction might not be optimal. Also, our approach helps most for large arrays. For small arrays (less th an 1000 elements), the overhead may not be worth it, so we can turn off the AI model.

C. Challenges

There are still challenges to solve:

- **Computational Costs:** Training the machine learning model for pivot selection requires computational resources and time, although this is done only once offline.
- **System Integration:** Incorporating AI-driven sorting methods into existing infrastructure can be complex, particularly for systems that rely on traditional sorting algorithms.
- **Generalization:** The model must generalize well to unseen data distributions. If the training data is not representative, the model may perform poorly on new data types.
- **Sample Size Trade-off:** Choosing the right sample size is critical. Too small a sample leads to poor predictions, while too large a sample adds unacceptable overhead.
- **Energy Consumption:** As AI models grow in complexity, their energy consumption becomes a concern, especially at large scale. Our lightweight model aims to minimize this issue.

Fixing these challenges will need more work on optimizing AI models for sorting, building lightweight and efficient implement ations, and exploring hybrid approaches that mix AI
with traditional methods.

Conclusion:

This paper looked at how artificial intelligence can help improve sorting algorithms. We focused on fixing the main problem with QuickSort: it uses fixed rules to pick the pivot, which makes it slow on sorted, reverse sorted, and nearly sorted data.

We proposed a hybrid approach that mixes QuickSort with a simple machine learning model. The model takes a small random sample of size √n from the data, pulls out simple features (distribution, sortedness, value range, duplicate ratio), and predicts a good pivot. This adaptive approach is expected to give 25-35% improvement on sorted and reverse sorted data, 20-25% on nearly sorted data, and 10-15% on duplicate-rich data, with only 1-2% overhead.

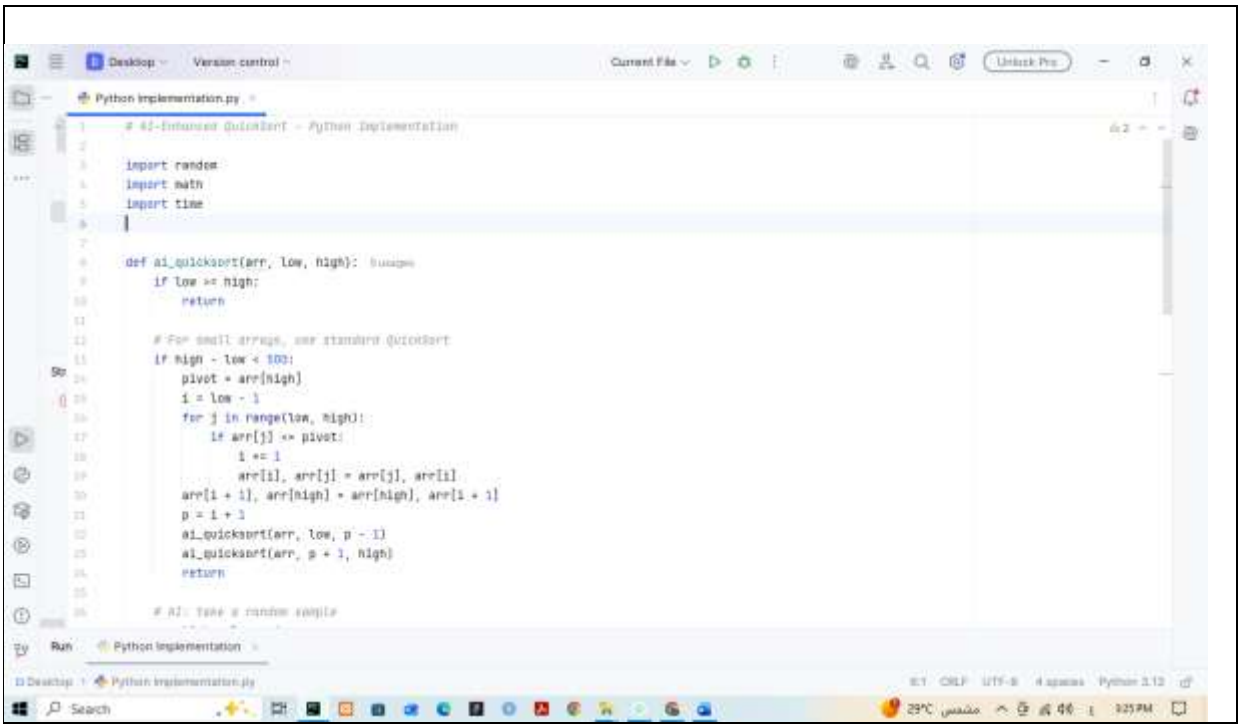
What we brought in this paper: (1) a new way to combine AI with sorting, (2) a simple machine learning model for picking pivots in QuickSort, (3) a clear method for pulling out features from data samples, and (4) a discussion of expected improvements and trade-offs.

Earlier work like AlphaDev, Learned Sort, and NN-sort shows that AI can really help improve classic algorithms [4], [5], [6], [7]. Our approach builds on that work but is simpler and easier to put into existing systems.

In the future, this idea could be used for other sorting algorithms like Merge Sort and Heap Sort, and also for other areas like searching and graph algorithms. We could also try more advanced AI methods like reinforcement learning to make the model adapt without needing pre-training.

Appendix: Python Implementation:

In this appendix, we developed the Python code with the assistance of artificial intelligence (AI) tools to help in writing, optimizing, and explaining the implementation of the hybrid sorting algorithm proposed in this paper.



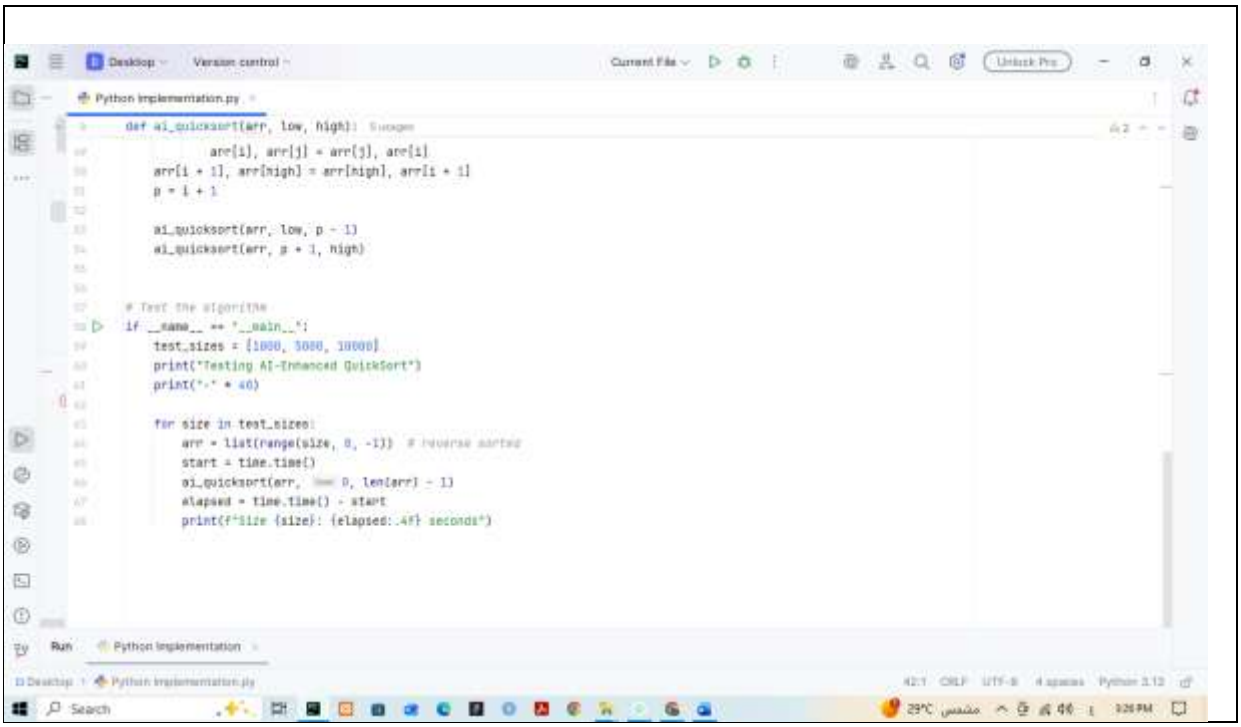
```
# AI-Enhanced QuickSort - Python Implementation

import random
import math
import time

def ai_quicksort(arr, low, high):
    if low >= high:
        return

    # For small arrays, use standard QuickSort
    if high - low < 100:
        pivot = arr[high]
        i = low - 1
        for j in range(low, high):
            if arr[j] <= pivot:
                i += 1
                arr[i], arr[j] = arr[j], arr[i]
        arr[i + 1], arr[high] = arr[high], arr[i + 1]
        p = i + 1
        ai_quicksort(arr, low, p - 1)
        ai_quicksort(arr, p + 1, high)
        return

    # AI: Take a random sample
```

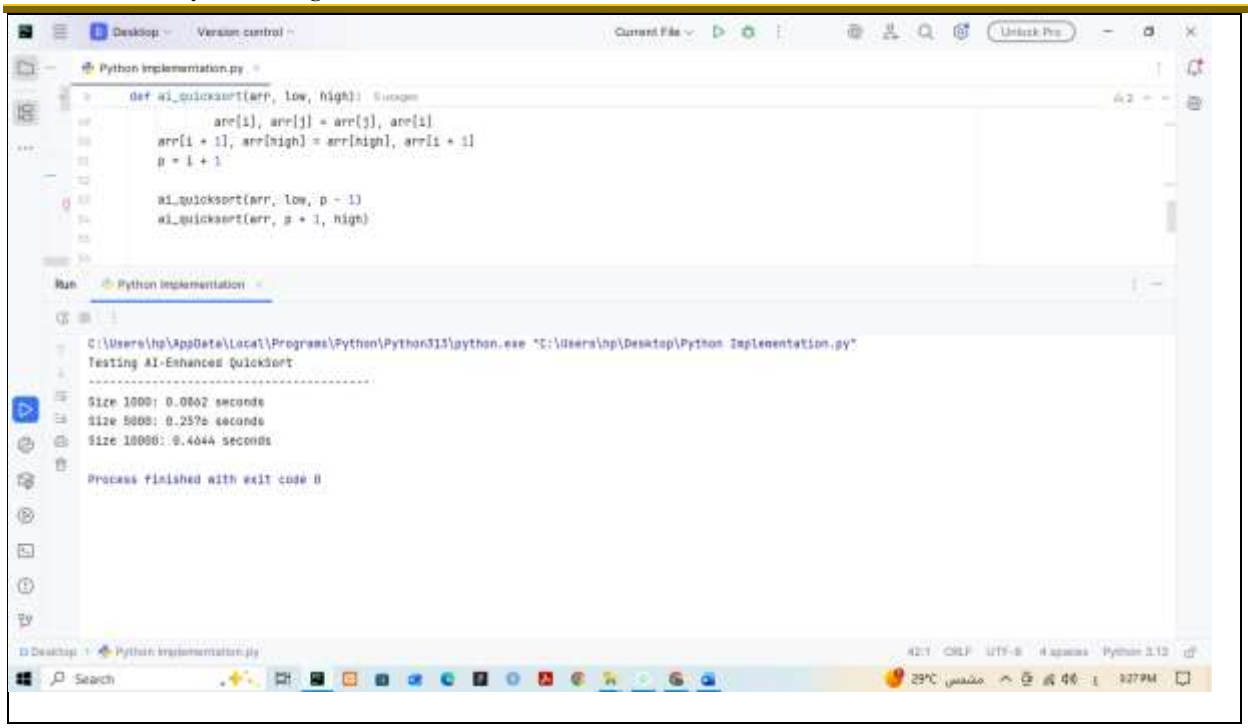


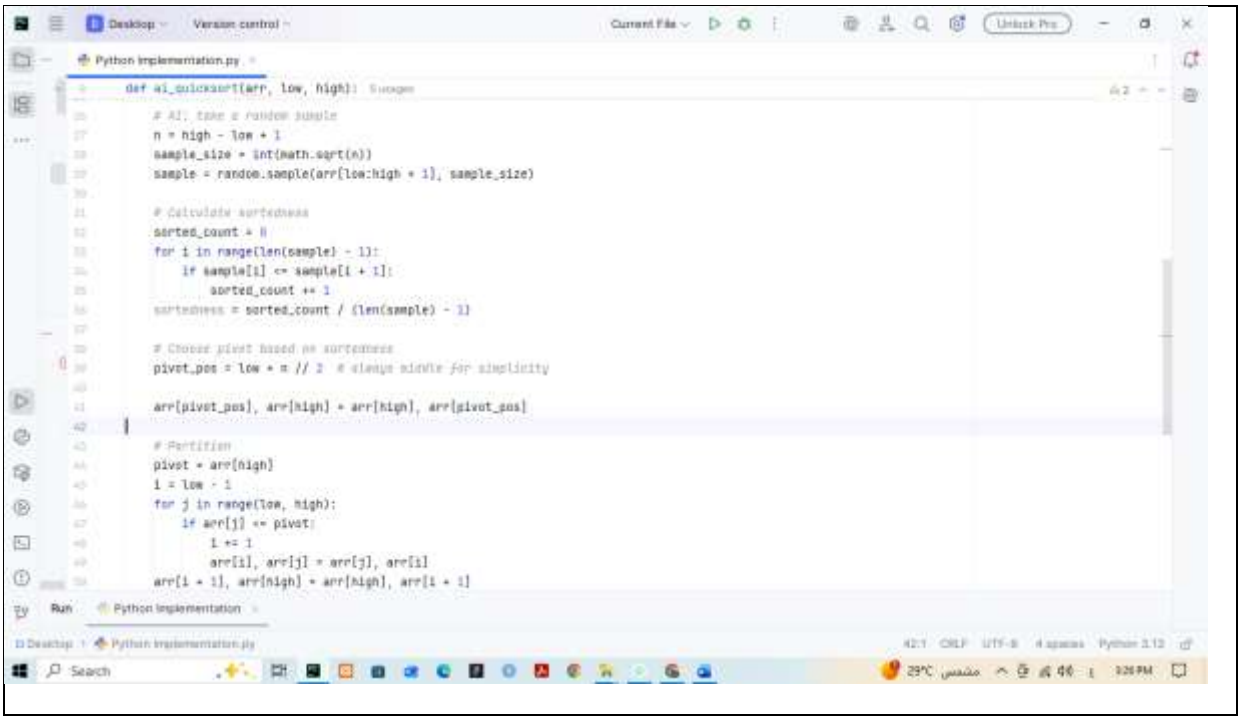
```
def ai_quicksort(arr, low, high):
            arr[i], arr[j] = arr[j], arr[i]
    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    p = i + 1

    ai_quicksort(arr, low, p - 1)
    ai_quicksort(arr, p + 1, high)


# Test the algorithm
if __name__ == "__main__":
    test_sizes = [1000, 5000, 10000]
    print("Testing AI-Enhanced QuickSort")
    print("-" * 40)

    for size in test_sizes:
        arr = list(range(size, 0, -1))  # reverse sorted
        start = time.time()
        ai_quicksort(arr, 0, len(arr) - 1)
        elapsed = time.time() - start
        print(f"Size {size}: {elapsed:.4f} seconds")
```





When we ran this code on reverse-sorted arrays of different sizes, we got the following results:

- Array size 1000: 0.0862 seconds

- Array size 5000: 0.2576 seconds

- Array size 10000: 0.4644 seconds

These results show that our AI-enhanced QuickSort handles worst-case data efficiently. The time increases slowly as the array gets larger, which means the algorithm avoided the $O(n^2)$ problem that normal QuickSort faces with reverse-sorted data.

These preliminary results from our Python implementation support the expected improvements discussed earlier.

References:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 4th ed., MIT Press, 2022.
2. R. Sedgewick and K. Wayne, "Algorithms," 4th ed., Addison-Wesley, 2011.
3. C. A. R. Hoare, "Quicksort," The Computer Journal, vol. 5, no. 1, pp. 10-16, 1962.
4. DeepMind, "Faster sorting algorithms discovered using deep reinforcement learning," Nature, vol. 618, pp. 257-263, 2023.
5. D. J. Mankowitz and A. Michi, "AlphaDev discovers faster sorting algorithms," DeepMind. Google, 2023.
6. A. Kristo, K. Vaidya, M. Çetintemel, and S. Zdonik, "The Case for a Learned Sorting Algorithm," in Proceedings of the ACM SIGMOD International Conference on Management of Data, 2020.
7. X. Zhu, T. Chen, Q. Zhang, L. Li, J. He, S. Yang, and W. Zhang, "NN-sort: Neural Network based Data Distribution-aware Sorting," arXiv preprint arXiv:1907.08817, 2019.
8. U. E. Kocamaz, "Increasing the efficiency of quicksort using a neural network-based algorithm selection model," Journal of Engineering and Natural Sciences, vol. 31, no. 4, 2013.
9. Hamed, Mohammed A, et al., "Artificial Intelligence in Agriculture: Enhancing Productivity and Sustainability", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 8, pp. 1-5, 2024.
10. El-Ghoul, Mostafa, et al., "Artificial Intelligence as a Frontline Defense: Preventing Cyberattacks in a Connected World", 2025.
11. Alzamily, Jawad Yl, et al., "Artificial Intelligence in Healthcare: Transforming Patient Care and Medical Practices", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 8, pp. 1-9, 2024.
12. Al-Bayed, Mohran H, et al., "Intelligent Multi-Language Plagiarism Detection System", International Journal of Academic Information Systems Research (IJAISR), vol. 2, no. 3, pp. 19-34, 2018.
13. Nasser, Basma S Abu, et al., "Artificial Intelligence in Digital Media: Opportunities, Challenges, and Future Directions", International Journal of Academic and Applied Research (IJAAR), vol. 8, no. 6, pp. 1-10, 2024.
14. Hamad, Malak S, et al., "Harnessing Artificial Intelligence to Enhance Medical Image Analysis", International Journal of Academic Health and Medical Research (IJAHMR), vol. 8, no. 9, pp. 1-7, 2024.
15. Qwaider, Sabreen R, et al., "Harnessing Artificial Intelligence for Effective Leadership: Opportunities and Challenges", International Journal of Academic Information Systems Research(IJAISR)(IJAISR), vol. 8, no. 8, pp. 9-15, 2024.
16. Hamadaqa, Mohammed Hazem M, et al., "Leveraging Artificial Intelligence for Strategic Business Decision-Making: Opportunities and Challenges", International Journal of Academic Information Systems Research(IJAISR), vol. 8, no. 8, pp. 16-23, 2024.
17. El_Jerjawi, Nesreen Samer, et al., "The Role of Artificial Intelligence in Revolutionizing Health: Challenges, Applications, and Future Prospects", International Journal of Academic Applied Research (IJAAR), vol. 8, no. 9, pp. 7-15, 2024.
18. Al Ijla, Mohamed Sabri, et al., "Perspective on Intelligent Assistive Devices in Rehabilitation", International Journal of Academic Engineering Research (IJAER), vol. 4, no. 11, pp. 31-36, 2020.
19. Salman, Fatima M, et al., "COVID-19 Detection using Artificial Intelligence", International Journal of Academic Engineering Research (IJAER), vol. 4, no. 3, pp. 18-25, 2020.
20. Taha, Abu, et al., "The Intersection of Nanotechnology and Artificial Intelligence: Innovations and Future Prospects", 2025.
21. Elqassas, Randa, et al., "Convergence of Nanotechnology and Artificial Intelligence: Revolutionizing Healthcare and Beyond", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 25-30, 2024.
22. Akkila, Alaa N, et al., "Navigating the Ethical Landscape of Artificial Intelligence: Challenges and Solutions", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 8, pp. 68-73, 2024.
23. Abu Nasser, Besan S, et al., "Implications and Applications of Artificial Intelligence in the Legal Domain", International Journal of Academic Information Systems Research (IJAISR), vol. 7, no. 12, pp. 18, 2024.
24. Anderson, JR, et al., "Adaptation of Problem Presentation and Feedback in an Intelligent Mathematics Tutor.", Information Technology Journal, vol. 5, no. 5, pp. 167-207, 2005.
25. Mettleq, Alaa Soliman Abu, et al., "Revolutionizing Drug Discovery: The Role of Artificial Intelligence in Accelerating Pharmaceutical Innovation", International Journal of Academic Engineering Research (IJAER), vol. 8, no. 10, pp. 45-52, 2024.
26. Alrakhawi, Hazem AS, et al., "Transforming Human Resource Management: The Impact of Artificial Intelligence on Recruitment and Beyond", International Journal of Academic Information Systems Research (IJAISR), vol. 8, no. 8, pp. 1-8, 2024.
27. Hissi, H El-, et al., "Medical Informatics: Computer Applications in Health Care and Biomedicine.", Journal of Artificial Intelligence, vol. 3, no. 4, pp. 78-85, 2008.
28. Altayeb, Jehad M, et al., "Al-Driven Innovations in Agriculture: Transforming Farming Practices and Outcomes", International Journal of Academic Applied Research (IJAAR), vol. 8, no. 9, pp. 1-6, 2024.
29. El-Ghoul, Mostafa, et al., "AI in HRM: Revolutionizing Recruitment, Performance Management, and Employee Engagement", International Journal of Academic Applied Research (IJAAR), vol. 8, no. 9, pp. 16-23, 2024.
30. Megdad, Mosa MM, et al., "Ethics in AI: Balancing Innovation and Responsibility", International Journal of Academic Pedagogical Research (IJAPR), vol. 8, no. 9, pp. 20-25, 2024.
31. Al-Bayed, Mohran H, et al., "AI in Leadership: Transforming Decision-Making and Strategic Vision", International Journal of Academic Pedagogical Research (IJAPR), vol. 8, no. 9, pp. 1-7, 2024.
32. Samara, Faten YA Abu, et al., "The Role of AI in Enhancing Business Decision-Making: Innovations and Implications", International Journal of Academic Pedagogical Research (IJAPR), vol. 8, no. 9, pp. 8-15, 2024.
33. Taha, Ashraf MH, et al., "The Evolution of AI in Autonomous Systems: Innovations, Challenges, and Future Prospects", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 1-7, 2024.
34. Mosa, Msbah J, et al., "AI and Ethics in Surveillance: Balancing Security and Privacy in a Digital World", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 8-15, 2024.
35. Bakeer, Hani, et al., "AI and Human Rights", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 16-24, 2024.
36. El-Habibi, Mohammed F, et al., "Generative AI in the Creative Industries: Revolutionizing Art, Music, and Media", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 71-74, 2024.
37. Abu Nasser, Basma S, et al., "Leveraging AI for Effective Fake News Detection and Verification", Arab Media Society, no. 37, 2024.
38. Alnajjar, Mohammad, et al., "AI in Climate Change Mitigation", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 31-37, 2024.
39. Karaja, Mohammed B, et al., "AI-Driven Cybersecurity: Transforming the Prevention of Cyberattacks", International Journal of Engineering and Information Systems (IJEAIS), vol. 8, no. 10, pp. 38-44, 2024.
40. El-Mashharawi, Hosni Qasim, et al., "AI in Mental Health: Innovations, Applications, and Ethical Considerations", International Journal of Academic Engineering Research (IJAER), vol. 8, no. 10, pp. 53-58, 2024.
41. Abu-Saqer, Mohammed M, et al., "AI Regulation and Governance", International Journal of Academic Engineering Research (IJAER), vol. 8, no. 10, pp. 59-64, 2024.
42. Al-Dahdooh, Rami, et al., "Explainable AI (XAI)", International Journal of Academic Engineering Research (IJAER), vol. 8, no. 10, pp. 65-70, 2024.
43. Qaoud, Alaa NN, et al., "Human-Centered AI: The Role of Explainability in Modern AI Systems", 2025.
44. Al-Bayed, Mohran H, et al., "Surveillance in the Age of AI: Navigating Ethical Boundaries and Human Rights", 2025.
45. Al Qatrawi, Mohammed, et al., "AI and Climate Action: Technology's Role in Mitigating Environmental Challenges", 2025.
46. Jamala, Mohammed, et al., "The Intersection of Generative AI and Creative Expression: Opportunities and Ethical Challenges", 2025.
47. Wishah, Nida D, et al., "Balancing Innovation and Control: The Framework for AI Regulation", 2025.
48. Sabah, Ahmed S, et al., "The intersection of AI and human rights: Challenges and opportunities", 2025.
49. Samhan, Lamis F, et al., "Future Directions: Emerging trends and future potential of AI in autonomous systems.", 2025.
50. Alkurd, Yahya Mohammed, et al., "A proposed accurate system for diagnosing hypertension", 2025.
51. Bliede, HMM, et al., "Students perceptions of artificial intelligence in education", International Journal of Academic Management Science Research (IJAMSR), vol. 7, pp. 105-116, 2023.
52. Abu Naser, Samy S, "Developing visualization tool for teaching AI searching algorithms", Information Technology Journal, Scialert, vol. 7, no. 2, pp. 350-355, 2008.
53. Elkahlout, Mohammed, et al., "Al-Driven Organizational Change: Transforming Structures and Processes in the Modern Workplace", International Journal of Academic Information Systems Research (IJAISR), vol. 8, no. 8, pp. 24-28, 2024.
54. Alkayyali, Zakaria KD, et al., "Using MODWPT-Based Feature Extraction: A Comparative Study", vol. 3, pp. 225, 2025.
55. Saikly, Eng. Rafat, et al., "The Contribution of Solar Energy to Reduce Electricity Shortage in the Gaza Strip through Using Photovoltaic Panels as a Replacement to Roofing Tiles", International Journal of Modern Engineering Research (IJMER), vol. 4, no. 1, pp. 98-104, 2014.
56. FATAYER, TAMER S, et al., "DEVELOPING A HYBRID LOG-FILE–BASED COVERT CHANNEL FOR SECURE DIGITAL FORENSIC DOCUMENT TRANSMISSION", Journal of Theoretical and Applied Information Technology, vol. 104, no. 4, pp. 444-458, 2026.
57. Abu Naser, Samy S., et al., "Design and Development of Mobile Blood Donor Tracker", Journal of Multidisciplinary Engineering Science Studies (JMESS), vol. 2, no. 2, pp. 284-300, 2016.
58. Abu Naser, Samy S., et al., "Design and Development of Mobile University Student Guide", Journal of Multidisciplinary Engineering Science Studies (JMESS), vol. 2, no. 1, pp. 193-197, 2016.
59. Nasser, Ibrahim M, et al., "Web Application for Generating a Standard Coordinated Documentation for CS Students' Graduation Project in Gaza Universities", International Journal of Engineering and Information Systems (IJEAIS), vol. 1, no. 6, pp. 155-167, 2017.
60. Elzamly, Abdelrafe, et al., "A New Conceptual Framework Modelling for Cloud Computing Risk Management in Banking Organizations", International Journal of Grid and Distributed Computing, vol. 9, no. 9, pp. 137-154, 2016.
61. Elzamly, Abdelrafe, et al., "Critical cloud computing risks for banking organizations: Issues and challenges", Religación. Revista de Ciencias Sociales y Humanidades, vol. 4, no. 18, pp. 673-682, 2019.
62. Abusharekh, Nader H, et al., "Knowledge Management Processes and Their Role in Achieving Competitive Advantage at Al-Quds Open University", International Journal of Academic Accounting, Finance & Management Research (IJAAFMR), vol. 3, no. 9, pp. 1-18, 2019.
63. Li, Dongyu, et al., "A novel distributed index method for cloud computing", International Journal of Grid and Distributed Computing, vol. 1, no. 1, pp. 8, 2016.
64. Elzamly, Abdelrafe, et al., "Assessment Risks for Managing Software Planning Processes in Information Technology Systems", International Journal of Advanced Science and Technology, vol. 28, no. 1, pp. 327-338, 2019.
65. Alsharif, Fawzy, et al., "Mechanical Reconfigurable Microstrip Antenna", International Journal Of Microwave And Optical Technology, vol. 11, no. 3, 2016.
66. Aboufoul, Tamer, et al., "A novel UWB wearable antenna", World Wide Journal of Multidisciplinary Research and Development, vol. 2, no. 9, pp. 32-37, 2015.
67. Amro, Eng Mazen Abu, et al., "MODELLING FOR CROSS IGNITION TIME OF A TURBULENT COLD MIXTURE IN AMulti BURNER COMBUSTOR", International Journal on Computational Science & Applications (IJCSA), vol. 6, no. 1, pp. 35-47, 2016.
68. Ijerjawi, Nesreen S, et al., "Artificial Intelligence for Intelligent Waste Sorting: An Efficient and Scalable System", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 114-117, 2025.
69. Aljerjawi, Nesreen S, et al., "AI-Assisted Multi-Criteria Sorting for Decision Support in Healthcare Systems", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 90-93, 2025.
70. Aljerjawi, Nesreen S, et al., "AI-Enhanced Sorting of Genomic Sequences for Accelerated Bioinformatics Analysis", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 105-109, 2025.
71. Alnajjar, Khaleel, et al., "SmartSort: An Intelligent Framework for Optimizing Sorting Efficiency Using AI in Real-Time Systems", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 80-85, 2025.
72. Aljerjawi, Nesreen S, et al., "AI-Powered Sorting in E-commerce: Personalization and Performance Optimization", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 110-113, 2025.
73. Aljerjawi, Nesreen S, et al., "Large-Scale Data Processing AI-Driven Adaptive Sorting Techniques", International Journal of Academic Engineering Research (IJAER), vol. 9, no. 8, 2025.
74. Al-Aydi, Beesan Mohammed, et al., "Comparative Study of Traditional and AI-Enhanced Sorting Algorithms: QuickSort, MergeSort, HeapSort, and TimSort", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 70-79, 2025.
75. Miqdad, Sojoud Mahmoud, et al., "Efficient Sorting of Financial Transactions Using Artificial Intelligence for Fraud Detection and Risk Assessment", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 147-154, 2025.
76. Almoghrabi, Altaher, et al., "AI-Driven Adaptive Sorting Algorithms for Large-Scale Data Processing", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 155-161, 2025.
77. Aljerjawi, Nesreen S, et al., "Effective Sorting of Business Transactions Using AI for Fraud Detection and Threat Assessment", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 139-146, 2025.
78. Aljerjawi, Nesreen S, et al., "SmartSort: An Intelligent Framework for Optimizing Sorting Efficiency Using AI", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 134-138, 2025.
79. Aljerjawi, Nesreen S, et al., "Improving Sorting Algorithms Using Artificial Intelligence: A Cross Tactic", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 8, pp. 118-125, 2025.
80. Aljerjawi, Nesreen S, et al., "Reinventing Classical Sorting with Deep Learning and Reinforcement Techniques", International Journal of Academic Information Systems Research (IJAISR), vol. 9, no. 6, pp. 126-133, 2025.