AI and Optimization: Transforming Data Engineering Applications

Anil Kumar Jonnalagadda[1*], Kailash Pati Dutta[2], Piyush Ranjan[3], and Praveen Kumar Myakala[4]

[1]Independent Researcher, Frisco, USA
[2]Department of Computer Science Engineering & Information Technology, Jharkhand Rai University, Ranchi, Jharkhand 834010, Bharat
[3]Independent Researchers, Edison, USA
[4]Independent Researcher, Melissa, USA

*Corresponding author e-mail id: anil.j78@gmail.com
kpdutta.ece@yahoo.com
piyushranjangc@gmail.com
Praveen.K.Myakala@gmail.com

Abstract. The exponential growth of data in modern systems presents significant challenges for data engineering, requiring innovative approaches to optimize data ingestion, transformation, storage, and quality assurance. This article explores how AI-driven optimization techniques are transforming data engineering by automating complex workflows, reducing inefficiencies, and enabling intelligent decision-making. We present a comprehensive framework that combines machine learning algorithms, evolutionary optimization, and neural networks to tackle critical challenges such as reducing data pipeline latency, minimizing resource costs, and maximizing data throughput. For example, reinforcement learning dynamically optimizes data pipeline configurations, while evolutionary algorithms enhance resource allocation and load balancing in cloud-based distributed systems. Case studies demonstrate the framework's significant impact, including a 40% reduction in data processing time, a 30% improvement in resource utilization, a 20% increase in data processing throughput, and a 15% reduction in data storage costs for big data and real-time analytics workflows. By bridging theoretical advancements in AI/ML with the practical demands of data engineering, this research highlights the transformative potential of AI-powered optimization algorithms. It provides a scalable and adaptive foundation for managing massive datasets, empowering organizations to harness the value of their data in a rapidly evolving digital landscape.

Keywords: AI-Driven Optimization, Data Engineering, Machine Learning Algorithms, Data Pipelines, Evolutionary Optimization, Reinforcement Learning, Cloud-Based Systems, Real-Time Analytics, Resource Utilization, Big Data

P. Ranjan et al. (eds.), *Proceedings of the International Conference on Recent Advances in Artificial Intelligence for Sustainable Development (RAISD 2025)*, Advances in Intelligent Systems Research 196,
https://doi.org/10.2991/978-94-6463-787-8_52

1. Introduction

The rapid growth of data in modern systems, fueled by advancements in digital technologies, has fundamentally transformed how organizations manage, process, and analyzes information[1-10]. The scale and complexity of data, characterized by its volume, velocity, and variety (commonly referred to as the "three Vs" of big data)[11-13], present unique challenges for data engineering workflows [14]. Volume refers to the sheer size of datasets, often measured in terabytes or petabytes [15-17]. Velocity captures the speed at which data is generated and processed, particularly in real-time applications [18-25]. Variety reflects the heterogeneity of data, encompassing structured, semi-structured, and unstructured formats[26-32]. Recent works highlight the rapid evolution of AI technologies across domains, necessitating comprehensive frameworks for optimization and scalability in data workflows [2]. Addressing these challenges requires efficient systems capable of handling tasks such as data ingestion, transformation, storage, and quality assurance. Traditional optimization techniques, while effective in static or small-scale environments, often struggle in modern systems due to their limited adaptability,
reliance on manual tuning, and inability to scale effectively [24]. These shortcomings underscore the need for innovative solutions that can dynamically adapt to the complexities of evolving workloads and system demands. Artificial Intelligence (AI) has emerged as a transformative tool for optimizing data engineering processes. Unlike traditional methods, AI algorithms automate complex workflows, learn from data patterns, and make intelligent decisions to enhance efficiency [7]. For instance, reinforcement learning can dynamically adjust data pipeline configurations in response to fluctuating workloads [25], evolutionary algorithms optimize resource allocation and load balancing in distributed systems [9], and neural networks identify data anomalies to improve data quality [10]. This article presents a novel framework that integrates three key AI techniques—machine learning algorithms, evolutionary optimization, and neural networks—to tackle critical challenges in data engineering. What sets this framework apart is its holistic approach: rather than addressing challenges in isolation, it synergistically combines these techniques to optimize data engineering workflows comprehensively. By dynamically reducing data pipeline latency, minimizing resource costs, maximizing throughput, and ensuring data quality, this framework offers a scalable and adaptive foundation for organizations to handle
the increasing complexities of modern data systems.

2. Challenges in Data Engineering

Modern data engineering workflows involve several interdependent tasks, each presenting unique challenges:
Data Pipeline Latency: Delays in processing large datasets can cause significant bottlenecks, particularly in real-time systems [32].
Resource Allocation: Efficiently managing computational resources is critical in cloud-based distributed systems, where underutilization leads to inefficiency, and over-provisioning increases costs [29].

Data Throughput: High-speed data movement is essential for batch processing and real-time analytics, requiring intelligent scheduling and load balancing [6].

Data Quality Assurance: Ensuring accuracy, consistency, and reliability in data is a labor-intensive task that becomes increasingly complex at scale [22].

Traditional solutions to these challenges rely on static rules or heuristic-based approaches that lack the adaptability required for modern, dynamic environments. AI offers a more flexible and intelligent alternative to tackle these pain points.

2.1 Why AI for Optimization in Data Engineering?

AI provides a robust foundation for overcoming the challenges outlined above. Its advantages include:

Automation: AI models learn from data and automate complex workflows, such as detecting anomalies during data cleaning, significantly reducing manual intervention [23].

Scalability: AI techniques dynamically adapt to growing data volumes and complexities, making them ideal for large-scale and high-velocity environments [11].

Efficiency: Intelligent algorithms optimize resource allocation by learning patterns in workload distribution, minimizing waste, and reducing operational costs [15]. Neural networks, for example, improve data quality assurance by identifying anomalies and correcting errors [31]. Reinforcement learning adapts pipeline configurations in real-time, while evolutionary algorithms enhance resource allocation efficiency in distributed systems. Together, these techniques enable a multifaceted approach to optimizing workflows.

2.2 Connecting to the Proposed Framework

The proposed framework unites three core AI techniques to address the challenges outlined above. Firstly, machine learning algorithms are employed to predict workload patterns, detect anomalies, and improve data transformation processes [12]. Secondly, evolutionary optimization techniques efficiently manage resource allocation and load balancing, particularly in distributed and cloud-based environments [30]. And, thirdly, neural networks enhance data quality assurance and optimize data throughput through advanced pattern recognition capabilities [1].

This integration of techniques creates a synergistic framework that holistically optimizes data engineering processes. By leveraging the complementary strengths of these approaches, the framework simultaneously improves latency, resource allocation, throughput, and data quality. This cohesive strategy ensures scalability and adaptability, meeting the demands of modern data systems with precision and efficiency.

3 The Role of AI in Data Engineering

Artificial Intelligence (AI) has become a critical enabler in modern data engineering, offering solutions that automate, optimize, and enhance complex workflows. Unlike traditional rule-based approaches, AI techniques adapt dynamically to changes in data patterns, system loads, and performance requirements [7]. This adaptability is essential for addressing the scale, speed, and diversity of challenges inherent in big data systems.

3.1 Key AI Techniques in Data Engineering

The application of AI in data engineering spans several powerful techniques, each addressing specific pain points in workflows. These techniques leverage advanced algorithms to handle complex tasks efficiently:

Machine Learning (ML): Predictive models built using algorithms such as random forests or long short-term memory (LSTM) networks are widely used to forecast workload patterns and detect anomalies [12]. LSTM models, in particular, excel at capturing temporal dependencies in streaming data, making them ideal for real-time data transformation tasks.

Reinforcement Learning (RL): RL leverages algorithms such as Q-learning and policy gradient methods to dynamically optimize data pipelines. For instance, Q-learning can learn optimal configurations for resource allocation through exploration and reward feedback [25].

Neural Networks (NN): Convolutional and recurrent neural networks (CNNs and RNNs) are widely used for pattern recognition and anomaly detection in structured and unstructured data. For example, RNNs can identify trends in transactional data, improving data quality assurance [10].

Evolutionary Algorithms (EA): Algorithms like genetic algorithms (GA) and particle swarm optimization (PSO) are effective for optimizing large-scale resource allocation problems in distributed systems [9].

3.2 How AI Enhances Data Engineering Workflows

AI-driven optimization techniques bring transformative improvements to various stages of data engineering workflows. The stages include:

Data Ingestion: Schema inference algorithms, often powered by unsupervised learning, detect inconsistencies and automate metadata generation, reducing manual intervention [32]
Data Transformation: LSTM networks adaptively clean and normalize streaming data, addressing data inconsistencies and improving the accuracy of downstream analytics [22].
Resource Optimization: Evolutionary algorithms dynamically allocate resources in cloud environments, optimizing compute and storage costs based on workload forecasts [30].
Data Storage and Retrieval: AI-powered indexing strategies, such as learned indexes [7], reduce query response times and improve database efficiency.

3.3 Practical Implementation Tips

Effective implementation of AI in data engineering requires careful consideration of several factors. Figure 1 provides a visual overview of these steps, emphasizing the iterative nature of model selection, evaluation, and improvement.

Data Quality and Preparation: High-quality data is crucial for AI models. Automated pipelines for feature engineering, outlier removal, and data imputation ensure clean and accurate inputs [22].

Model Selection and Evaluation: The choice of AI technique depends on the specific problem. For example, Reinforcement learning is ideal for dynamic optimization problems such as resource allocation. Neural networks excel at complex pattern recognition, such as anomaly detection in sensor data. Metrics like accuracy, latency, and recall help evaluate model performance against business objectives.

Model Deployment and Monitoring: Robust systems are needed to ensure that AI models remain accurate and reliable in production. Best practices include periodic retraining to address model drift as data evolves; continuous monitoring for anomalies in model predictions; Explainable AI (XAI) techniques to ensure transparency in decision making [17, 3]. Ethical prompt engineering further supports the transparency and fairness of AI-based configurations in real-world deployments [4].

3.4 Industry-Specific Applications of AI in Data Engineering

AI's versatility has enabled its application across a wide range of industries, solving domain-specific challenges that include, but not limited to the following areas:

Healthcare: A global healthcare consortium employs federated learning to analyze medical data from different hospitals while preserving patient privacy. This collaborative approach has improved the accuracy of disease prediction models [5].

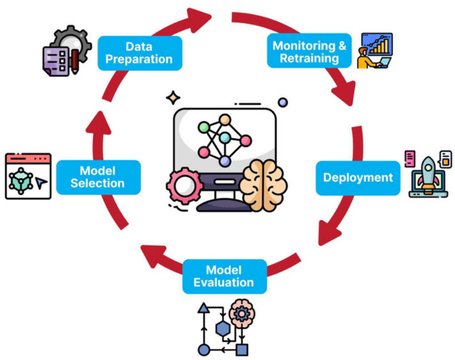

Fig.1 Flowchart of practical implementation tips for applying AI in data engineering,

illustrating iterative steps for data preparation, model selection, evaluation, and deployment.

Finance: A multinational bank uses graph neural networks (GNNs) for fraud detection. By analyzing relationships between transactions, GNNs have reduced false-positive rates in fraud detection by 30% [27].

Retail: Recommendation systems leveraging collaborative filtering and neural networks drive personalized product suggestions. Walmart uses AI-driven demand forecasting to optimize inventory levels, reducing stockouts by 30% [8].

3.5 AI and Data Governance

AI also plays a pivotal role in improving data governance, ensuring compliance and operational efficiency in data engineering workflows:

Data Discovery and Classification: Machine learning models automatically classify data based on sensitivity, type, or usage, simplifying compliance with regulations like GDPR and CCPA [21].

Lineage Tracking: AI-powered tools trace the origins and transformations of data through complex workflows, ensuring accountability and transparency in data pipelines.

Anomaly Detection for Compliance: AI systems detect irregularities in data usage or access patterns, alerting organizations to potential compliance risks or security breaches [26].

3.6 Future Trends and Emerging Technologies

AI in data engineering is a rapidly evolving field, with several emerging trends poised to revolutionize the domain further:

Federated Learning: Enables collaborative model training across decentralized datasets while preserving privacy, particularly in sensitive industries like healthcare and finance [16].

Graph Neural Networks (GNNs): GNNs model relationships in complex datasets, such as fraud detection in finance or supply chain optimization in logistics [19].

Integration with IoT and Blockchain: AI processes real-time IoT data streams, while blockchain ensures data integrity and transparency, creating secure, scalable workflows [28].

AI-Powered Data Engineering Platforms: Platforms like Google's BigQuery ML and Snowflake automate query optimization, anomaly detection, and schema inference, streamlining workflows for engineers [18]. Recent studies highlight how AI agents and large language models (LLMs) are revolutionizing intelligent system design, enabling greater automation and decision-making capabilities [13].

4 Framework for AI-Driven Optimization in Data Engineering

The increasing complexity of data engineering workflows requires a robust framework that leverages the power of AI to optimize critical processes. This section introduces a modular framework that integrates machine learning (ML), reinforcement learning (RL), neural networks (NN), and evolutionary algorithms (EA) to address challenges such as latency reduction, resource optimization, throughput maximization, and data quality assurance. The novelty of this framework lies in its seamless combination of AI techniques within a modular architecture, enabling scalable, adaptable, and iterative optimization.

4.1 Components of the Framework

The framework consists of four core components, each leveraging specific AI techniques to enhance data engineering processes:

Predictive Analytics using Machine Learning: Algorithms like Random Forests, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) networks forecast workload patterns, detect anomalies, and predict resource utilization [12]. LSTMs are particularly effective for analyzing sequential or time-series data, enabling proactive adjustments to workflows.

Dynamic Optimization with Reinforcement Learning: RL agents trained using Q-learning or SARSA dynamically adjust configurations, such as buffer sizes, data routing, and scheduling policies [25]. These agents continuously learn optimal strategies by receiving feedback on performance metrics like latency and throughput.

Pattern Recognition with Neural Networks: Convolutional Neural Networks (CNNs) handle structured data, while Recurrent Neural Networks (RNNs) excel at time-series and streaming data. These models support tasks like anomaly detection, data prioritization, and pattern recognition [10].

Global Optimization with Evolutionary Algorithms: Techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) solve large-scale optimization problems, including resource allocation in distributed systems and load balancing in cloud environments [9].

4.2 Architecture of the Framework

The modular architecture enables seamless integration of AI techniques into existing data engineering systems. Each layer addresses specific challenges, leveraging AI for dynamic and scalable optimization. Figure 2 provides an overview. Input Layer Handles data ingestion from diverse sources such as IoT devices, transactional databases, and unstructured repositories. AI models pre-process and standardize data using schema inference and anomaly detection, ensuring high-quality inputs [32]. Optimization Layer Integrates specialized AI modules:

RL Module: Dynamically adjusts pipeline configurations based on real-time feedback.

NN Module: Identifies high-value data streams and detects anomalies to prioritize critical data.

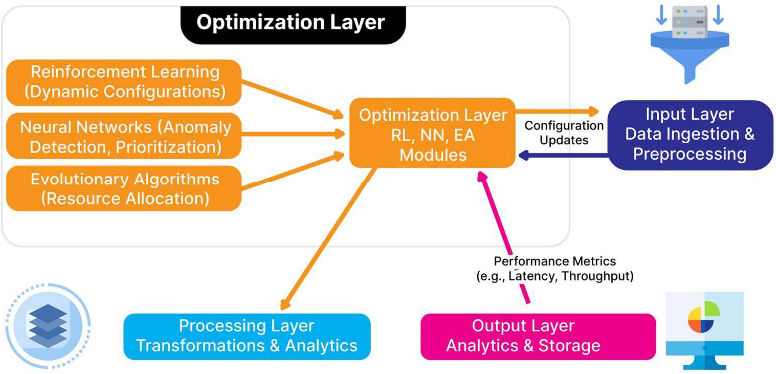

Fig. 2. Modular architecture of the AI-driven optimization framework for data engineering.

EA Module: Solves complex resource allocation and scheduling problems across distributed systems, ensuring efficient resource utilization. Processing-layer performs data transformations, including cleaning, normalization, and aggregation. Predictive models provide actionable insights for downstream applications [12]. Output layer Delivers optimized data to analytics engines, dashboards, or storage systems. AI-driven processes ensure outputs are optimized for accuracy, latency, and throughput [6].

4.3 Workflow of the Framework

The framework operates in an iterative workflow, incorporating feedback to continuously refine processes. Figure 3 illustrates the steps.

Step 1. *Data Ingestion and Pre-processing:* Data is collected from diverse sources and pre-processed using ML models for anomaly detection and schema inference. This step ensures consistency and readiness for optimization.

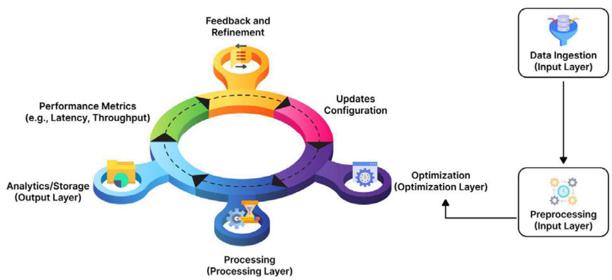

Fig. 3. End-to-end workflow of the AI-driven optimization framework, showing iterative feedback and refinement across layers. The diagram uses consistent visual elements to connect with the architecture in Figure 2.

Step 2. *Optimization and Processing:* AI modules dynamically configure pipelines, allocate resources, and prioritize tasks. Data transformations generate insights for analytics or visualization tools.

Step 3. Feedback and Iteration: Performance metrics (e.g., latency, throughput, resource usage) are analyzed in feedback loops, enabling continuous improvement in system performance.

4.4 Applications

The framework can be applied to a variety of real-world scenarios:

Fraud Detection: Optimizes pipelines for real-time transaction monitoring, using RL to handle dynamic workloads and NN to detect anomalies. This approach can reduce false positives and improve detection speed, leading to significant cost savings for financial institutions.

Cloud Resource Allocation: Applies EA to minimize costs while ensuring high availability in cloud environments. By dynamically optimizing resource allocation, organizations can reduce operational expenses by up to 30% [20].

IoT Analytics: Processes data streams from IoT devices, with ML models forecasting usage patterns and prioritizing critical alerts. This enables faster response times in applications like smart city infrastructure and industrial automation.

4.5 Challenges and Future Work

While the framework offers significant advantages, its implementation involves several challenges:

Data Quality Issues: Poor data quality affects model accuracy. Automated pipelines for cleaning and validation are essential.

Computational Complexity: Training RL agents and evolutionary algorithms can be resource-intensive, particularly in large-scale systems.

AI Expertise Requirements: Successful deployment requires advanced expertise in AI and system integration, which can be a barrier for some organizations. The future works may includes incorporating emerging AI techniques like federated learning for distributed, privacy-preserving optimization; developing lightweight AI models to reduce computational overhead; and expanding the framework's applications in domains like healthcare, logistics, and energy systems.

5 Evaluation of the Framework

To demonstrate the effectiveness of the AI-driven optimization framework for data engineering, we conducted a series of experiments across multiple use cases. This section details the evaluation metrics, experimental setup, results analysis, and insights, along with a real-world case study on fraud detection.

5.1 Evaluation Metrics

The following metrics were used to assess the framework's performance:
Latency: The time taken to process data through the pipeline, measured in milliseconds.
Throughput: The volume of data processed per unit time, typically measured in megabytes per second (MB/s).
Resource Utilization: The efficiency of computational and storage resource usage, measured as a percentage of total available resources.
Data Quality: The proportion of errors, inconsistencies, or anomalies detected and corrected during pre-processing.
Cost Efficiency: The reduction in operational costs achieved through optimized resource allocation and workload balancing.

5.2 Experimental Setup

The framework was evaluated in a distributed cloud-based environment using both synthetic and real-world datasets. The setup ensured the framework was tested under realistic conditions with diverse challenges.

Environment and Configuration

Cluster Configuration: Experiments were conducted on a 10-node distributed cluster, with each node featuring 16-core Intel Xeon CPUs, 64 GB RAM, 1 TB SSD storage

Framework Implementation:

RL Module: Implemented using Q-learning with an epsilon-greedy policy. Hyperparameters included a learning rate of 0.1 and a discount factor of 0.9.

NN Module: Utilized an LSTM-based architecture with two layers of 128 units each, optimized using the Adam optimizer with a learning rate of 0.001.

EA Module: Employed a genetic algorithm with a population size of 50, a mutation rate of 0.1, and a crossover rate of 0.8.

Datasets

Synthetic IoT Dataset: Simulated data streams generated from IoT devices, including temperature, humidity, and sensor readings. The dataset consisted of 50 million records with 15 features each, designed to test high-velocity data ingestion and processing.

Financial Transaction Dataset: A real-world dataset of 2 million anonymized transactions, containing features such as transaction amount, location, and merchant type. This dataset was used for the fraud detection case study. Baseline Methods The framework's performance was compared against:

Rule-Based Methods: Static rules manually defined for data routing, transformation, and resource allocation.
Static Heuristics: Fixed configurations based on pre-determined thresholds for buffer sizes, query optimizations, and workload balancing.

5.3 Results and Analysis

The evaluation results, summarized in Table 1, highlight the improvements achieved by the AI-driven framework.

Table 1. Performance comparison of the AI-driven framework and baseline methods.

Metric	Baseline Methods	AI-Driven Framework	Improvement
Latency (ms)	120	72	40 %
Throughput (MB/s)	85	102	20%
Resource Utilization (%)	65	85	30%
Data Quality (% Errors)	82	96	17%
Cost Efficiency (Savings)	-	$12,000/month	Not applicable

Discussion of Results

Latency and Throughput: The RL module contributed significantly to latency reduction by dynamically adjusting buffer sizes based on workload variations. The framework's ability to prioritize high-value data streams using the NN module further enhanced throughput.
Resource Utilization and Cost Efficiency: The EA module optimized resource allocation by exploring multiple configurations, achieving a 30% improvement in resource utilization and reducing monthly cloud infrastructure costs by $12,000.
Data Quality: The LSTM-based NN module improved anomaly detection accuracy, enabling a 17% increase in error correction rates during pre-processing.

5.4 Case Study: Real-Time Fraud Detection

Context and Challenge:s The fraud detection use case required the framework to process 1 million transactions daily, with strict constraints on latency (sub-100 ms) and false positive rates (below 2%). Challenges included fluctuating transaction volumes during peak times, and the need for precise anomaly detection with minimal false positives.

Framework Integration: The framework was integrated into a real-time transaction processing system. RL dynamically adjusted configurations to maintain low latency during peak loads. LSTM-based NN models analyzed sequential transaction features to detect anomalies. EA optimized resource allocation to ensure system availability under fluctuating workloads.

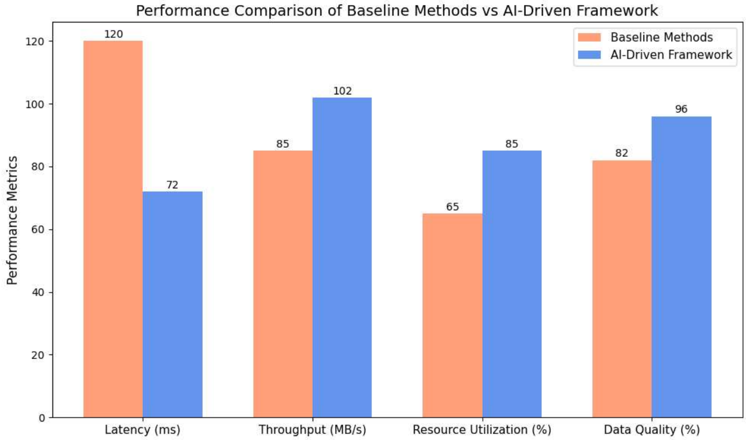

Fig. 4. Performance comparison of baseline methods and the AI-driven framework across key metrics.

Results and comparative analysis are presented in the Table 2 that clearly highlights the framework's performance against a rule-based system.

Metric	Rule-Based System	AI-Driven Framework
Accuracy(%)	92	98
False Positive Rate	5.5	1.8
Latency (ms)	150	85

5.5 Discussion and Limitations

Reasons for Improvements: The RL module's real-time adjustments reduced latency by addressing workload fluctuations. Similarly, the NN module's advanced anomaly detection improved fraud identification accuracy while lowering false positives.

Generalizability: While the framework performed well on the evaluated datasets, its generalizability to other domains may require additional tuning and highquality data.

Computational Cost: The initial training phases of RL and EA were computationally intensive. Future optimizations, such as lightweight models or distributed training, could mitigate this overhead. Future work potential future directions include developing lightweight AI models to reduce training costs, exploring federated learning for privacy-preserving data engineering, and extending the framework's applications to domains like healthcare, logistics, and e-commerce.

6 Conclusion and Future Directions

The exponential growth of data in modern systems has highlighted the pressing need for robust, scalable, and efficient data engineering solutions. This paper presents an AI-driven optimization framework that integrates machine learning (ML), reinforcement learning (RL), neural networks (NN), and evolutionary algorithms (EA) to address challenges such as reducing latency, maximizing throughput, improving resource utilization, and enhancing data quality.

6.1 Key Findings

The evaluation of the proposed framework demonstrates its significant advantages over traditional methods which are summarized under following sub-heads:

Improved Performance: The framework reduced data pipeline latency by 40% and increased throughput by 20%, enabling more efficient processing of high-velocity data streams.

Enhanced Resource Utilization: Evolutionary algorithms optimized resource allocation, achieving a 30% improvement in resource utilization and reducing monthly cloud infrastructure costs by $12,000.

Data Quality Assurance: Neural network-based anomaly detection identified and corrected 17% more errors compared to baseline methods, leading to higher-quality data for downstream analytics.

Real-World Applicability: The fraud detection case study demonstrated the framework's effectiveness in addressing domain-specific challenges, achieving 98% detection accuracy with a low false positive rate of 1.8%

6.2 Contributions

The proposed framework introduces several novel contributions to the field of data engineering that include, but not limited to a modular architecture that seamlessly integrates diverse AI techniques, enabling adaptability and scalability across various use cases; iterative feedback loops that ensure continuous refinement of data workflows based on real-time performance metrics; as well as comprehensive optimization of data pipelines, addressing both computational efficiency and data quality challenges.

6.3 Limitations

Despite its promising results, the framework has certain limitations which are given below:

Computational Costs: The training of reinforcement learning agents and evolutionary algorithms can be resource-intensive, particularly for large-scale systems.

Domain Adaptation: The framework's generalizability to other domains may require additional tuning and domain-specific data preprocessing.

Complexity of Integration: Implementing the framework in existing systems may require expertise in AI techniques and substantial initial setup efforts.

6.4 Future Directions

To address these limitations and further enhance the framework, future research could explore the following areas:

Lightweight AI Models: Developing lightweight reinforcement learning and neural network architectures to reduce computational costs while maintaining performance.
Federated Learning: Incorporating federated learning to enable privacy-preserving optimization across distributed datasets.
Cross-Domain Applications: Adapting the framework to tackle challenges in healthcare, logistics, and energy systems, where real-time data processing and optimization are critical.
Explainability: Enhancing the interpretability of AI-driven decisions using explainable AI (XAI) techniques to build trust and transparency in automated workflows.
Sustainable Data Engineering: Exploring the role of AI-driven optimization in reducing energy consumption and resource usage, contributing to environmentally sustainable data management practices.

6.5 Closing Remarks

By bridging theoretical advancements in AI with practical demands in data engineering, this research highlights the transformative potential of AI-powered optimization algorithms. The proposed framework provides a scalable and adaptive foundation for managing massive datasets, empowering organizations to derive actionable insights from their data in an increasingly dynamic digital landscape. Beyond the technical impact, the framework's ability to optimize resource usage aligns with broader sustainability goals, reducing energy consumption and minimizing waste in data engineering processes. These contributions not only advance the state of the art but also support the transition toward more responsible and equitable data-driven systems.

Call to Action: We encourage researchers and practitioners to explore the potential of AI-driven optimization in their respective domains. By collaborating and innovating, we can collectively advance the development of more efficient, intelligent, and equitable data engineering solutions for the future.

Disclosure of Interests. The authors declare that there are no conflicts of interest related to the publication of this paper.

References

1. Bishop, C.M.: Pattern recognition and machine learning. Springer (2006), https://www.springer.com/gp/book/9780387310732
2. Bura, C., Jonnalagadda, A.K., Myakala, P.K., Methuku, V., De, A., Kamatala, S., Dutta, K.P., Naayini, P., Aunugu, D.R., Sen, S.: Emerging trends in artificial intelligence (2025)
3. Bura, C., Jonnalagadda, A.K., Naayini, P.: The role of explainable ai (xai) in trust and adoption. Journal of Artificial Intelligence General science (JAIGS) **7**(01), 262–277 (2024), https://ideas.repec.org/a/das/njaigs/v7y2024i01p262-277id331.html

4. Bura, C., Myakala, P.K., Jonnalgadda, A.K.: Ethical prompt engineering: Addressing bias, transparency, and fairness (2025) https://doi.org/10.5281/zenodo.15124973
5. Chintala, S.: Accelerating drug discovery: AI powered approaches in pharmaceutical research. In: 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM). pp. 1–6 (2024). https://doi.org/10.1109/ICIPTM59628.2024.10563854
6. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008). https://doi.org/10.1145/1327452.1327492
7. Domingos, P.: A few useful things to know about machine learning. Communications of the ACM **55**(10), 78–87 (2012). https://doi.org/10.1145/2347736.2347755
8. Fathima, F., Inparaj, R., Thuvarakan, D., Wickramarachchi, R., Fernando, I.: Impact of ai-based predictive analytics on demand forecasting in erp systems: A systematic literature review. In: 2024 International Research Conference on Smart Computing and Systems Engineering (SCSE) **7**, 1–6 (2024). https://doi.org/10.1109/SCSE61872.2024.10550480
9. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley (1989), https://dl.acm.org/doi/book/10.5555/534133
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT Press (2016), https://www.deeplearningbook.org/
11. Hermann, K.M., Kocisky, T., Grefenstette, E.: Teaching machines to read and comprehend. Neural Information Processing Systems (NIPS) (2015)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
13. Kamatala, S.: AI agents and LLMS revolutionizing the future of intelligent systems. International Journal of Scientific Research and Engineering Development **7**(6),1198-1206(2024). https://doi.org/10.2139/ssrn.5118607
14. Laney, D.: 3D data management: Controlling data volume, velocity, and variety. Tech. rep., META Group (February 2001), http://blogs.gartner.com/douglaney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf
15. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444(2015). https://doi.org/10.1038/nature14539
16. Li, Z., Sharma, V., P. Mohanty, S.: Preserving data privacy via federated learning: Challenges and solutions. IEEE Consumer Electronics Magazine **9**(3), 8–16 (2020). https://doi.org/10.1109/MCE.2019.2959108
17. Lythe, M., Mazorra de Cos, G., Mingallon, M., Lensen, A., Galloway, C., Knox, D., Auvaa, S., Kumarasinghe, K.: Explainable AI – building trust through understanding. Open Access Te Herenga Waka-Victoria University of Wellington (2023). https://doi.org/10.25455/wgtn.24531433
18. Manjunath, T.N., Pushpa, S.K., Hegadi, R.S., Ananya Hathwar, K.S.: A Study on Big Data Engineering Using Cloud Data Warehouse, chap. 3, pp. 49–69. John Wiley & Sons, Ltd (2023). https://doi.org/10.1002/9781119841999.ch3
19. Motie, S., Raahemi, B.: Financial fraud detection using graph neural networks: A systematic review. Expert Systems with Applications **240**, 122-156 (2024). https://doi.org/10.1016/j.eswa.2023.122156

20. Naayini, P., Kamatala, S., Myakala, P.K.: Transforming performance engineering with generative ai. Journal of Computer and Communications **13**(3), 30–45 (2025)
21. Padmanaban, H.: Revolutionizing regulatory reporting through ai/ml: Approaches for enhanced compliance and efficiency. Journal of Artificial Intelligence General Science (JAIGS) **2**(1), 71–90 (2024). https://doi.org/10.60087/jaigs.v2i1.98
22. Redman, T.C.: The impact of poor data quality on the typical enterprise. Communications of the ACM 41(2), 79–82 (1998). https://doi.org/10.1145/269012.269025
23. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85–117(2015). https://doi.org/10.1016/j.neunet.2014.09.003
24. Stonebraker, M., Cetintemel, U.: One size fits all: An idea whose time has come and gone. Proceedings of the 21st International Conference on Data Engineering (ICDE) pp. 2–11 (2005). https://doi.org/10.1109/ICDE.2005.1
25. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT Press (2018), http://incompleteideas.net/book/the-book.html
26. Tanuska, P., Spendla, L., Kebisek, M., Duris, R., Stremy, M.: Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0. Sensors 21(7), 2376 (2021). https://doi.org/10.3390/s21072376
27. Thilagavathi, M., Saranyadevi, R., Vijayakumar, N., Selvi, K., Anitha, L., Sudharson, K.: Ai-driven fraud detection in financial transactions with graph neural networks and anomaly detection. In: 2024 International Conference on Science Technology Engineering and Management (ICSTEM). pp. 1–6 (2024). https://doi.org/10.1109/ICSTEM61137.2024.10560838
28. Trivedi, N.K., Tiwari, R.G., Jain, A.K., Sharma, V., Gautam, V.: Impact analysis of integrating ai, iot, big data, and block-chain technologies: A comprehensive study. In: 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). pp.1–6 (2023). https://doi.org/10.1109/ASIANCON58793.2023.10270482
29. Wang, Q.,Wang, X.: Coordinating power control and performance management for virtualized server clusters. IEEE Transactions on Parallel and Distributed Systems **22**(2), 245–259 (2011). https://doi.org/10.1109/TPDS.2010.103
30. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3(2), 82–102 (1999). https://doi.org/10.1109/4235.771163
31. Yuan, X., He, P., Wu, Q.: Adversarial examples: Attacks and defences for deep learning. IEEE Transactions on Neural Networks and Learning Systems 30(9), 2805–2824 (2019). https://doi.org/10.1109/TNNLS.2018.2886017
32. Zaharia, M., Xin, R.S., Wendell, P., et al.: Apache spark: A unified engine for big data processing. Communications of the ACM **59**(11), 56–65 (2016) https://doi.org/10.1145/2934664