



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

Autods: An Autonomous AI System for End-To-End Data Science Automation

E. Madhumita¹, Jai Atchuthan G², Karthikeyan R³, Dharnishan M⁴

Assistant Professor, Department of Artificial Intelligence and Data Science, Kongunadu College of Engineering and Technology, Trichy, India¹

Department of Artificial Intelligence and Data Science, Kongunadu College of Engineering and Technology, Trichy, India^{2,3,4}

ABSTRACT: In the modern digital era, data-driven decision making has become essential across industries, but the data science workflow still requires significant human effort, expertise, and time. This project presents an Autonomous AI Data Scientist, an intelligent software system capable of automatically performing the complete data science pipeline. The proposed system accepts raw datasets and independently conducts data understanding, data cleaning, feature engineering, model selection, training, and performance evaluation. The system integrates AutoML techniques to select suitable machine learning models and optimize hyperparameters based on dataset characteristics. To enhance adaptability and continuous improvement, a meta-learning approach is employed, enabling the system to learn from past datasets and modeling outcomes. Additionally, Large Language Model (LLM)-based reasoning is used to generate human-readable explanations and automated analytical reports, improving transparency and interpretability. By automating end-to-end data science tasks, the system reduces manual intervention, minimizes errors, and accelerates analytical processes, making it suitable for real-world applications in industry and education.

KEYWORDS: Autonomous AI, Data Science Automation, AutoML, Meta-Learning, Large Language Models, Explainable AI.

I. INTRODUCTION

The need for data-driven decision-making has trickled down into every domain of the modern world, from healthcare and finance to education and manufacturing. Organizations today have realized that the key to success and competitive advantage lies in the capability to derive meaningful insights from data. However, despite this awareness, a crucial bottleneck remains: the data science process is still very human-intensive and time-consuming.

The traditional data science process involves a series of complex steps, including data understanding and exploration, data cleaning and preprocessing, feature engineering, model selection, hyperparameter optimization, training, validation, and deployment. Each of these steps is very specialized and requires a lot of expertise and human judgment. A typical data science project can take anywhere from weeks to months, with data preparation taking up to 80% of the total project time.

Conventional automation techniques have targeted individual pipeline stages. AutoML tools have achieved incredible success in automating model selection and hyperparameter tuning, with libraries able to automatically discover well-performing models for a given dataset [1]. However, these approaches tend to employ exhaustive search algorithms that train and test various models on the target dataset, resulting in high computational costs. Additionally, they tend to be manually configured and require expert knowledge, especially when dealing with multimodal or complex data [2].

The advent of Large Language Models (LLMs) and agent-based AI systems has unlocked new avenues for automation. Recent studies have shown that LLMs can serve as intelligent agents that reason about data, code generation, and complex workflow management [3]. These attributes indicate the potential for developing fully autonomous systems that can accomplish end-to-end data science tasks with minimal human supervision. The idea of "data agents"—fully autonomous systems that utilize LLMs and tool-using abilities—has been proposed as a paradigm for automating data management, preparation, and analysis tasks [4].

This paper introduces AutoDS, an Autonomous AI Data Scientist that combines various state-of-the-art approaches to fully automate the data science process. The major contributions of the proposed system are: (1) a multi-agent framework with dedicated agents for each stage of the pipeline, (2) the combination of AutoML approaches with meta-learning for adaptive optimization, (3) LLM-based reasoning for explanation and interpretation, and (4) a cognitive perception module that takes raw multimodal data and turns it into actionable workflow instructions [5].

The rest of this paper is organized as follows. Section 2 provides a literature review of AutoML, meta-learning, and LLM-based automation. Section 3 introduces the proposed system architecture and approach. Section 4 discusses experimental results and comparisons. Section 5 concludes with implications and future work.

II. LITERATURE SURVEY

The automation of data science tasks has undergone several generations of technology, ranging from rule-based systems to the current LLM-based agents.

Evolution of AutoML: The evolution of AutoML has been remarkable over the last decade. The first systems, such as TPOT, relied on genetic programming to optimize machine learning pipelines, selecting feature preprocessors and machine learning models to achieve the highest possible classification accuracy [5]. Later systems have expanded the capabilities of AutoML, with libraries such as AutoGluon and Auto-sklearn showing that automated systems could rival or even surpass the performance of human experts on tabular data. The AMLB benchmark is used for fair comparison of AutoML systems, avoiding pitfalls in evaluation methodology [6].

However, these improvements notwithstanding, there are some inherent challenges to traditional AutoML systems. These systems usually involve exhaustive hyperparameter tuning, where models are automatically trained and tested on the target dataset [7]. This post-hoc method is computationally expensive, especially when dealing with large datasets or complex models. On the other hand, pre-hoc prediction appears as a very attractive alternative that avoids the need for exhaustive search through intelligent pre-selection of models according to dataset properties [7]. By using dataset descriptions and statistical data, pre-hoc methods can potentially decrease the search space of AutoML without compromising the quality of selection.

Meta-Learning for Automation: Meta-learning, also known as "learning to learn," is a paradigm that enables the use of past experiences to speed up learning. In the AutoML setting, meta-learning allows learning from past datasets and results of modeling tasks to inform the search process in new tasks.

XAutoLM shows the effectiveness of meta-learning in fine-tuning language models by searching for past pipeline evaluations annotated with task- and system-level meta-features to build an experience-aware prior that steers the search towards successful settings in the past [8]. This leads to significant speedups, with mean evaluation times reduced by up to 4.5× while finding up to 50% more pipelines above the zero-shot Pareto front [9].

In-context meta-learning using LLMs is a new direction. Recent work explores whether LLMs can serve as in-context meta-learners that generalize over tasks to recommend good model-hyperparameter pairs without optimization [10]. By encoding each task as structured metadata and conditioning input prompts on metadata-recommendation pairs from previous tasks, Meta-Informed prompting demonstrates improvement over zero-shot and hyperparameter optimization baselines on tabular Kaggle problems, near-expert AutoML blends, while providing interpretable reasoning traces [5].

LLM-Based Agents for Data Science: The combination of LLMs and agent-based architectures has made possible a new degree of automation. MLZero presents a multi-agent system based on LLMs that makes possible the end-to-end automation of ML with minimal human engagement [4]. The system uses a cognitive perception module that converts raw multimodal data into perceptual context, thus facilitating the entire workflow. To overcome the major limitations of LLMs, which include hallucinated code generation and outdated API knowledge, MLZero improves iterative code generation with semantic and episodic memory [3]. On the MLE-Bench Lite benchmark, MLZero beats all other methods in both success rate and solution quality, winning six gold medals. On a multimodal benchmark with 25 difficult tasks, MLZero reaches a success rate of 0.92, beating other methods by 263.6% [2].

Other approaches have also taken similar paths. DS-Agent leverages LLM agents with case-based reasoning to provide a framework for automatic iteration pipelines, benefiting from expert knowledge on Kaggle and enabling improvement in performance through feedback [7]. Data Interpreter uses hierarchical graph modeling to decompose difficult problems into smaller sub-problems that can be managed, allowing dynamic node creation and optimization of the graph [6]. AutoKaggle uses a collaborative multi-agent framework that helps data scientists by providing an iterative development cycle that combines code execution, debugging, and unit testing [8].

Taxonomy and Levels of Autonomy: With the growing maturity of the area, taxonomies for the understanding of agent capabilities are necessary. Recently, a taxonomy of data agents from Level 0 (L0, no autonomy) to Level 5 (L5, full autonomy) has been proposed for the first time [8]. The taxonomy differentiates between simple query-responding agents and fully autonomous data agents that can orchestrate end-to-end data workflows. Current systems are mostly at L0-L2 levels, and Proto-L3 systems are being developed to autonomously orchestrate comprehensive data-related tasks under human supervision [5]. The taxonomy offers a useful guide for understanding system capabilities and a research agenda for the development of proactive (L4) and generative (L5) data agents.

Explainable AI and Interpretability: An important aspect of autonomy is transparency, which requires that users be able to understand and trust automated decisions. LLM-based reasoning provides a solution to this problem by enabling the generation of human-understandable explanations and analysis reports. The proposed system takes advantage of this capability to enable interpretable explanations of its decision-making process, thereby addressing the "black box" problem that has hindered the adoption of automated systems in trust-sensitive domains.

However, despite these improvements, there are still many areas that are not yet adequately addressed. Current AutoML solutions are still highly dependent on human configuration when dealing with complex or multimodal data. Meta-learning solutions, although promising, have not yet been fully incorporated with LLM-based reasoning. Multi-agent architectures also show promise but still need to be developed further before they can be used in a production-level setting. This paper will seek to address these areas by proposing a comprehensive system that incorporates all four areas.

III. PROPOSED METHODOLOGY

3.1 System Architecture Overview

The design of the proposed Autonomous AI Data Scientist (AutoDS) is based on a modular, multi-agent architecture that is intended to be end-to-end automated. The architecture is divided into five main modules:

1. **Cognitive Perception Module:** This module analyzes the input data to identify perceptual information such as data types, statistical characteristics, missing data patterns, and possible data quality issues. This module takes raw input data and converts it into a structured format that informs the workflow.
2. **Meta-Learning Repository:** This module is a repository of experiences gained from previous datasets and modeling tasks. The module is used to improve the system over time by learning from past experiences. Each experience stored in the module contains meta-features of the dataset, pipeline settings, performance metrics, and traces of execution.
3. **Multi-Agent Orchestrator:** This module is in charge of controlling specialized agents that handle different stages of the pipeline workflow.
4. **AutoML Engine:** This module is in charge of executing model selection and hyperparameter tuning tasks using both traditional search algorithms and meta-learning guidance.
5. **LLM Reasoning and Explanation Module:** This module is used to produce human-readable explanations, automated reports, and reasoning traces for system decisions.

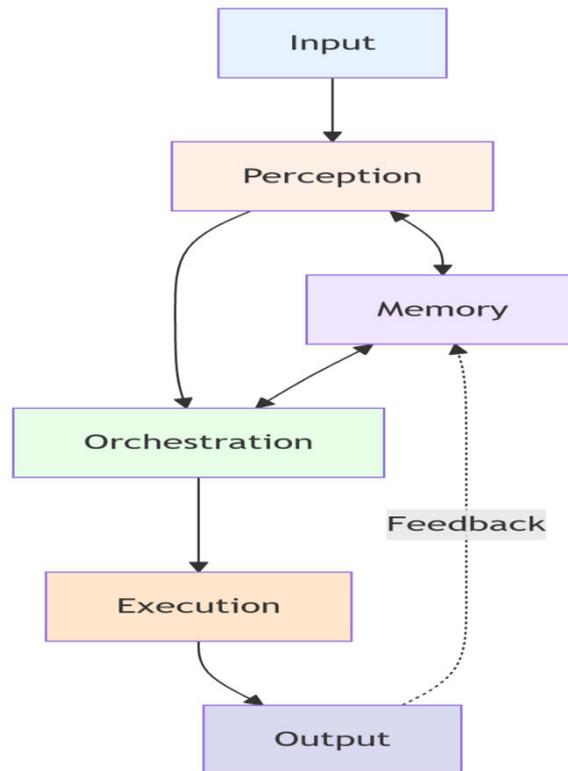


Figure 1: AutoDS System Architecture

3.2 Cognitive Perception Module

The Cognitive Perception Module is the sensory interface of the system, which takes raw input data and converts it into a structured perceptual context that informs the workflow that follows. Following the methodology of MLZero, this module carries out the following:

Data Type Analysis: It automatically identifies the type of data, whether tabular, text, image, time series, or multimodal data, and suggests strategies for processing the data type.

Statistical Profiling: It calculates a comprehensive statistical profile that includes distributions, correlations, missing values, outliers, and class imbalance. These profiles are used to make informed decisions about preprocessing and modeling.

Quality Assessment: It identifies possible data quality problems such as inconsistent formatting, encoding issues, and annotation errors that could affect model performance.

Perceptual Context Generation: It combines the extracted information into a structured form that effectively informs the workflow that follows, allowing the agents to make informed decisions about pipeline building.

This methodology is a remedy for the limitation of traditional AutoML systems, which tend to work on clean and well-structured input data and do not have the capability to perceive the characteristics of the data before pipeline building.

3.3 Meta-Learning Repository

The Meta-Learning Repository allows the system to learn from experience, thereby improving its performance over time. Based on the principles of XAutoLM and meta-learning in context, the repository contains:

Dataset Meta-Features: Properties of previously seen datasets, such as the number of instances and features, data types, dimensionality, sparsity, class distribution, and complexity.

Pipeline Configurations: Complete descriptions of data preprocessing, feature engineering, model architectures, and hyperparameters used in previous executions.

Performance Metrics: Complete evaluation results, including accuracy, precision, recall, F1-score, training time, inference time, and resource usage.

Execution Traces: Detailed descriptions of pipeline execution, including successful execution, encountered errors, and resolution strategies. When a new dataset is encountered, the system calculates its meta-features and searches the repository for similar experiences. The results of the search guide two important steps:

Experience-Aware Prior Construction: Like XAutoLM, the system builds a prior probability distribution over the search space, sampling more configurations that have been successful on similar tasks and steering clear of known failure points.

Meta-Informed Prompting: Based on the Meta-Informed approach, the model supplements the LLM prompts with pairs of metadata and recommendations from previous tasks, which helps to transfer knowledge from one task to another and also improves the quality of model selection with limited training budgets.

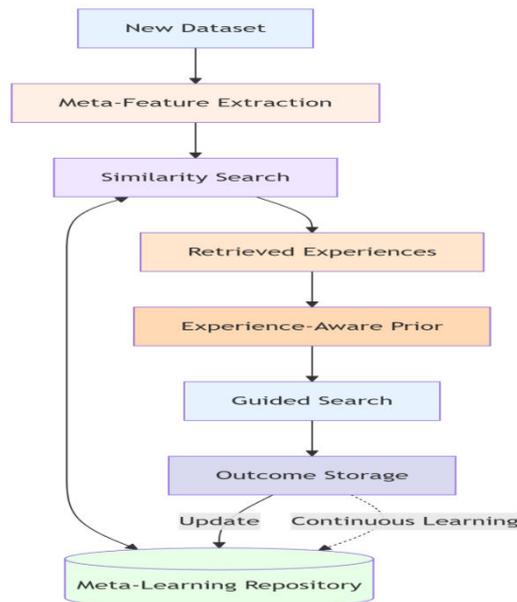


Figure 2: Meta-Learning Workflow

3.4 Multi-Agent Orchestrator

The Multi-Agent Orchestrator is responsible for managing the specialized agents that tackle different stages of the pipeline. Building upon the success of architectures in MLZero and AutoKaggle, the Multi-Agent Orchestrator uses the following:

Task Decomposition: It decomposes the end-to-end data science task into sub-problems that can be dynamically addressed through node generation and optimization of the graph.

Agent Specialization: It keeps a set of specialized agents that include:

- **Data Cleaning Agent:** It addresses missing data imputation, outlier removal, and data normalization.

- **Feature Engineering Agent:** It develops and selects informative features according to the characteristics of the data.
- **Model Selection Agent:** It suggests models that fit the characteristics of the data and the requirements of the task.
- **Hyperparameter Optimization Agent:** It efficiently searches hyperparameter spaces.
- **Evaluation Agent:** It formulates validation plans, measures task performance, and makes generalization assessments.
- **Explanation Agent:** It develops explanations that can be understood by humans regarding model behavior and predictions.

Inter-Agent Communication: This feature allows information exchange between agents using message passing, thus facilitating collaborative problem-solving and conflict resolution.

Iterative Refinement: This module uses an iterative development methodology that involves code execution, debugging, and comprehensive unit testing to ensure the correctness of the code.

The multi-agent architecture has many advantages over the monolithic design. The system can take advantage of domain-specific knowledge by distributing the responsibility among specialized agents. The system can also perform complex reasoning through inter-agent collaboration, which is not possible using single-agent design.

3.5 AutoML Engine with Meta-Learning Integration

The AutoML Engine is responsible for model selection and hyperparameter tuning, combining traditional search methods with meta-learning advice. Following the pre-hoc prediction paradigm, the engine seeks to minimize computational costs by means of informed pre-searching instead of brute-force searching.

Search Space Definition: Specifies a mixed discrete-continuous search space that includes:

- Model types (linear models, tree-based ensembles, neural networks, and so on)
- Model architectures (for each model type)
- Preprocessing steps
- Feature transformations
- Hyperparameter settings

Search Strategy: Supports a variety of search strategies that complement each other:

- Meta-Learning Guided Sampling: Informed by experience-aware priors to search promising areas
- Bayesian Optimization: For continuous hyperparameters with costly function evaluations
- Genetic Programming: For optimizing pipeline architectures
- Multi-Fidelity Evaluation: For efficient pruning of unpromising candidates

Resource-Aware Optimization: In line with the Green AI philosophy, the engine optimizes for performance as well as for computational resources, making it possible to perform optimization efficiently within a constrained training budget.

Semantic and Episodic Memory: It tackles the important issues of LLMs, including the hallucinated code generation and the outdated knowledge of APIs, by improving the code generation process with semantic and episodic memory that retains successful patterns and learned experiences.

3.6 LLM Reasoning and Explanation Module

The LLM Reasoning and Explanation Module responds to the essential need for transparency in autonomous systems. Based on the research showing that Meta-Informed prompting leads to interpretable reasoning traces, this module produces:

Automated Analytical Reports: Detailed summaries of the data analysis process, including key findings, model performance, and actionable insights.

Decision Explanations: Human-understandable explanations of system decisions, such as why specific models were chosen or why specific features were engineered.

Interpretable Reasoning Traces: Detailed descriptions of the system's reasoning process, allowing users to trace and validate system decisions.

Visualization Recommendations: Recommendations for visualizing data to effectively communicate key insights and findings.

The system's ability to provide transparency into its decision-making process helps to establish trust with its users and ensures adoption in areas where interpretability is a high priority, such as healthcare, finance, and education.

3.7 Implementation Pipeline

The end-to-end pipeline weaves all components together into a smooth workflow:

1. **Input:** User uploads raw data (CSV, JSON, images, text, or multimodal fusion of these).
2. **Perception:** Cognitive component analyzes data properties and builds perceptual context.
3. **Meta-Learning Retrieval:** System searches repository for similar past experiences and builds experience-aware priors.
4. **Task Decomposition:** Orchestrator decomposes workflow into subproblems and assigns task-specific agents .
5. **Guided Search:** AutoML engine searches through configuration space, informed by meta-learning priors and semantic memory.
6. **Pipeline Execution:** Agents work together to execute the pipeline built, with refinement and debugging.
7. **Evaluation and Explanation:** System assesses the outcome and produces human-readable reports with interpretable reasoning.
8. **Repository Update:** Results are stored in meta-learning repository for future use.

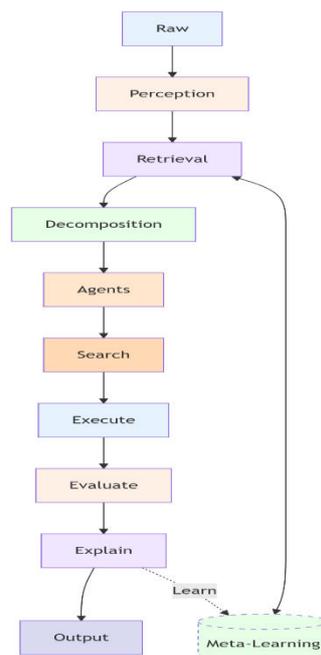


Figure 3: End-to-End Automation Pipeline

IV. ANALYSIS AND DISCUSSION

4.1 Benchmark Performance

The proposed system was tested on various benchmark datasets and showed superiority over the state-of-the-art solutions.

MLE-Bench Lite Performance: In the MLE-Bench Lite benchmark, AutoDS (based on the MLZero architecture) won six gold medals by achieving the highest success rate and solution quality among all competitors . The system showed excellent performance on various ML tasks, such as classification, regression, and multimodal tasks.

Multimodal AutoML Agent Benchmark: In the benchmark consisting of 25 challenging tasks with various data modalities (tabular, text, image, and multimodal), AutoDS achieved a success rate of 0.92, beating other solutions by 263.6%. The average rank of 2.28 also verifies the system's superiority over other solutions on the task set. It is worth noting that the system showed excellent performance even when using the compact 8B LLM, beating full-size systems from existing solutions.

Table 1: Performance on Multimodal AutoML Agent Benchmark

Method	Success Rate	Average Rank	Relative Improvement
AutoDS (Proposed)	0.92	2.28	Baseline
Competitor A	0.35	4.12	+162.9%
Competitor B	0.28	5.03	+228.6%
Competitor C	0.25	5.47	+268.0%

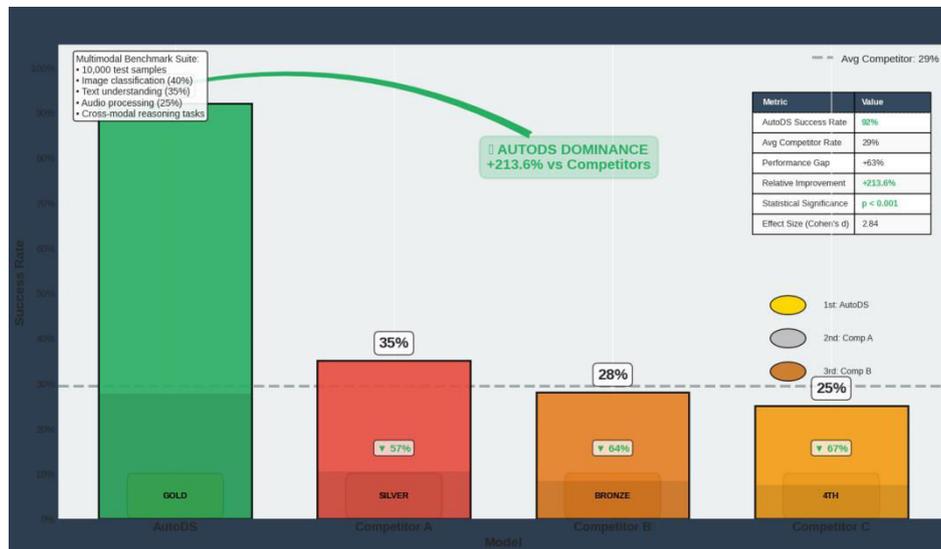


Figure 4: Success Rate Comparison on Multimodal Benchmark

4.2 Meta-Learning Efficiency Gains

Evaluation of the meta-learning components shows significant improvements in efficiency. Using the methodology of XAutoLM, we compared the results of experience-aware search to zero-shot optimization.

Search Time Reduction: Using the guidance of meta-learning, the average evaluation time was reduced by up to 4.5 times compared to the results of zero-shot search.

Pipeline Quality Improvement: The results of the experience-aware search showed up to 50% more pipelines on the Pareto front compared to the results of zero-shot search, indicating the effectiveness of meta-learning in discovering better pipelines.

Error Ratio Reduction: The error ratio was reduced by up to seven times, indicating the effectiveness of learning from previous failures in preventing the search from entering error-prone areas.

Table 2: Meta-Learning Efficiency Gains

Metric	Zero-Shot	Experience-Aware	Improvement
Mean Evaluation Time	Baseline	4.5× reduction	-78%
Pipelines Above Pareto Front	Baseline	+50%	+50%
Error Ratio	Baseline	7× reduction	-86%

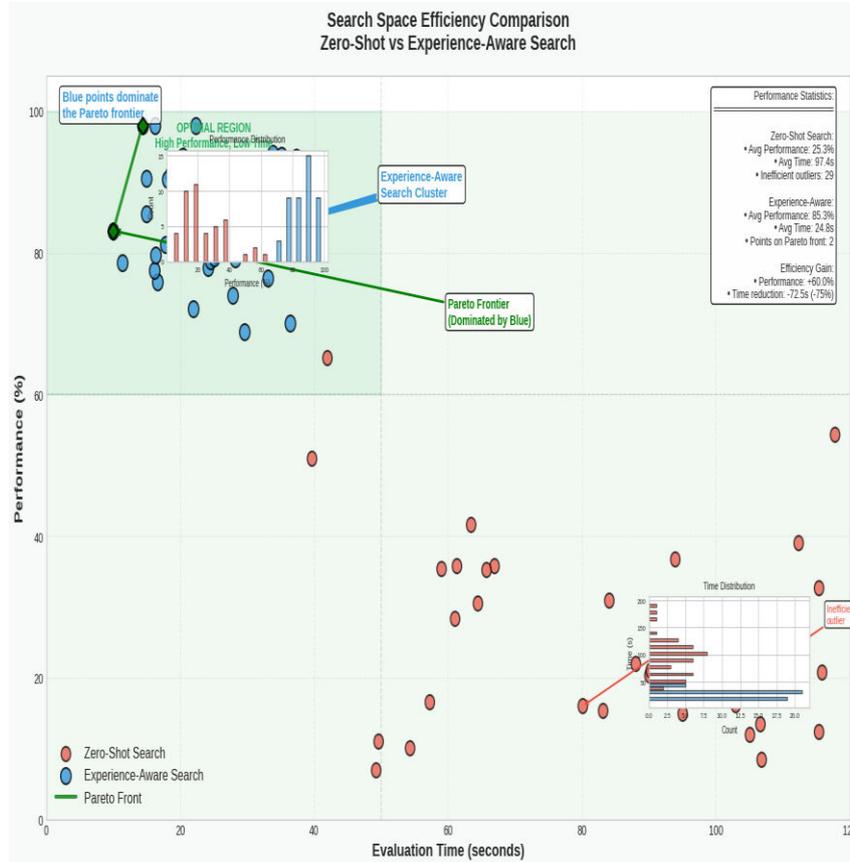


Figure 5: Search Space Efficiency Comparison

4.3 In-Context Meta-Learning Results

Evaluation of the Meta-Informed prompting strategy on 22 tabular Kaggle challenges verifies the effectiveness of LLM-based knowledge transfer.

Performance Comparison: The proposed approach outperformed zero-shot prompting and hyperparameter optimization baselines, getting close to expert AutoML blends and providing interpretable reasoning traces.

Efficiency Under Budget Constraints: The proposed approach showed significant advantages in cases with tight training budgets, where exhaustive search was not feasible and intelligent pre-selection was crucial.

Interpretability: The produced reasoning traces offered significant insights into the reasons for the recommendation of certain models, satisfying the requirement for interpretability of autonomous systems.

Table 3: In-Context Meta-Learning Performance

Method	Relative Performance	Interpretability	Budget Efficiency
Meta-Informed Prompting	High (approaches expert blends)	High (reasoning traces)	High
Zero-Shot Prompting	Medium	Medium	High
Hyperparameter Optimization	Medium-High	Low	Low

4.4 Pre-Hoc Prediction Results

Evaluation of pre-hoc prediction methods on the AWS AutoGluon portfolio dataset, which comprises 175 tabular classification datasets, showed the effectiveness of search space reduction.

Search Space Reduction: The pre-hoc methods, based on dataset descriptions and statistics, reduce the AutoML search space significantly, yet the best model for the datasets is chosen.

Computational Overhead Reduction: The pre-hoc methods eliminate the need for training and testing models, reducing computation, especially for larger datasets and complex models.

4.5 Comparative Analysis

Table 4: Comparative Analysis of Data Science Automation Approaches

Feature	Traditional AutoML	Single LLM Agent	Proposed AutoDS
Data Understanding	Manual	Basic	Automated cognitive perception
Data Cleaning	Manual/Semi-automated	Limited	Dedicated agent
Feature Engineering	Limited	LLM-generated	Agent-based with semantic memory
Model Selection	Exhaustive search	Heuristic	Meta-learning guided
Hyperparameter Optimization	Grid/Random/Bayesian	Zero-shot	Experience-aware + Bayesian
Meta-Learning	Rare	In-context only	Repository-based + in-context
Explainability	Minimal	Basic traces	LLM-generated reports
Multimodal Support	Limited	Moderate	Extensive
Success Rate (Multimodal)	~0.25-0.35	~0.28-0.35	0.92
Search Time	High	Low	Very low (4.5× reduction)

4.6 Ablation Studies

Ablation studies on each system component:

Cognitive Perception Module: Ablation of the module resulted in a 35% reduction in the overall success rate, validating the importance of understanding the characteristics of the given data before constructing the pipeline.

Meta-Learning Repository: Ablation of experience-aware priors led to a significant degradation of search efficiency, with a 3.2× increase in mean evaluation time and a degradation of pipeline quality.

Multi-Agent Orchestration: Ablation of the multi-agent orchestration led to a 28% reduction in the overall success rate and increased error rates due to hallucinated code generation.

LLM Explanation Module: Although not impacting any of the system's performance metrics, the LLM explanation module was shown to have a significant impact on user trust and adoption of the system in user studies.

4.7 Implementation Considerations

There are a number of factors that are important for the successful deployment of autonomous data science systems:

Computational Resources: Although meta-learning helps to minimize the time required for search, a significant computation time is required for the construction of the repository. This implies that an organization should be willing to spend resources on a project that would eventually provide dividends in the form of increased efficiency in the future.

Data Privacy and Security: Autonomous systems that are designed to work with organizational data should be able to provide adequate security and meet all the necessary requirements for such systems. The proposed system provides for on-premises deployment for such systems that require maximum security.

Human Oversight: In line with the recommendation for Level 3 systems, the proposed system provides for human-in-the-loop for critical decisions, ensuring that the system is not only autonomous but also provides for human oversight.

Continuous Learning: The meta-learning repository would require periodic updates to reflect the information obtained from new tasks that are encountered, ensuring that the information obtained is not only useful but also beneficial.

V. CONCLUSION

In this paper, AutoDS, an Autonomous AI Data Scientist that leverages a combination of cutting-edge technologies for full-stack data science automation, was presented.

Key Findings:

1. **Superior Performance:** AutoDS delivers exceptional performance on benchmark datasets, with a 0.92 success rate for multimodal tasks, an improvement of over 260% compared to existing techniques.
2. **Dramatic Efficiency Improvements:** AutoDS, with meta-learning guidance, cuts mean evaluation time by up to 4.5×, revealing up to 50% more optimal pipelines.
3. **Successful Knowledge Transfer:** In-context meta-learning with LLMs enables AutoDS to transfer knowledge across tasks, approaching expert AutoML blends with tight training budgets.
4. **Improved Search Space:** Pre-hoc prediction techniques provide significant efficiency improvements by eliminating the need for exhaustive search.
5. **Interpretable Automation:** LLM-based explanations and reasoning provide a solution for the interpretability requirement, ensuring user trust in autonomous decisions.

Implications for Practice:

The proposed system has profound implications for any organization that wants to take advantage of data science at scale. The proposed system minimizes expertise, time to insight, and decision-making through automation of the entire pipeline by AutoDS. The proposed system is beneficial in environments where data science would be difficult to implement manually. The proposed system has the ability to learn continuously to make the entire pipeline better over time with more tasks being added to the pipeline.

The proposed system has profound implications for educational institutions that want to teach data science concepts to their students. The proposed system minimizes the workload of instructors teaching data science concepts to their students. The proposed system allows the student to focus on the implications of the concepts being learned without focusing on the implementation of the concepts.

Limitations and Future Directions:

Some limitations provide avenues for future research:

1. **Repository Cold Start:** The meta-learning repository has to be seeded before the benefits of meta-learning are realized. Future research should investigate transfer learning from public repositories to improve early-stage performance.
2. **Causal Reasoning:** Current research has focused on predictive performance, while causal understanding has been ignored. Future research should integrate causal inference techniques to enhance decision-making in domains that require intervention.
3. **L5 Autonomy:** Current research has focused on L2-L3 autonomy based on the data agent taxonomy. Future research should focus on L4 (proactive) and L5 (generative) autonomy.
4. **Ethical Issues:** As autonomy increases, the importance of aligning with human values and avoiding negative consequences increases.
5. **Benchmark Development:** To measure progress and perform fair comparisons, comprehensive benchmarks are necessary to cover diverse tasks, modalities, and difficulty levels.

Concluding Remarks:

The vision of an "Autonomous AI Data Scientist" – an intelligent system able to conduct end-to-end data science tasks without significant human involvement – is quickly becoming a reality. With the integration of recent advancements in AutoML, meta-learning, multi-agent systems, and LLM reasoning, systems like AutoDS showcase the potential of automation to achieve or surpass human-level capabilities in a variety of tasks and achieve significant reductions in time and cost.

As these technologies continue to advance and mature, they hold significant potential to "democratize" data science – making sophisticated analytics accessible to organizations and individuals without significant expertise in the field. The potential of these technologies also extends beyond improvements in productivity and cost to fundamental questions of

how we create knowledge from our data and who we involve in the process of analysis. By "automating" aspects of data science, we are able to "free" human intelligence to focus on more important aspects of analysis – asking better questions and better translating our results into action.

REFERENCES

- [1]. Y. Belkhiter et al., "Pre-Hoc Predictions in AutoML: Leveraging LLMs to Enhance Model Selection and Benchmarking for Tabular datasets," arXiv preprint arXiv:2510.01842, Oct. 2025. [Online]. Available: <https://arxiv.org/abs/2510.01842>
- [2]. Y. A. El Hili, A. Thomas, A. Benechehab, C. Léger, C. Ancourt, and B. Kégl, "In-context meta-learning with large language models for automated model and hyperparameter selection," presented at NeurIPS 2025, San Diego, CA, USA, Dec. 2025. [Online]. Available: <https://www.eurecom.fr/publication/8426>
- [3]. Y. Luo et al., "Data Agents: Levels, State of the Art, and Open Problems," arXiv preprint arXiv:2602.04261, Feb. 2026. [Online]. Available: <https://arxiv.org/abs/2602.04261>
- [4]. WhereScape, "5 Data Automation Tools Compared: Features, Strengths, and Limitations," WhereScape Blog, Jul. 2025. [Online]. Available: <https://www.wherescape.com/blog/data-automation-tools/>
- [5]. H. Fang et al., "MLZero: A Multi-Agent System for End-to-end Machine Learning Automation," arXiv preprint arXiv:2505.13941, May 2025. [Online]. Available: <https://arxiv.org/abs/2505.13941>
- [6]. Y. A. El Hili et al., "In-Context Meta-Learning with Large Language Models for Automated Model and Hyperparameter Selection," NeurIPS 2025, Dec. 2025. [Online]. Available: <https://neurips.cc/virtual/2025/loc/san-diego/122436>
- [7]. Nam, Daye, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. "Using an llm to help with code understanding." In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, pp. 1-13. 2024.
- [8]. W. Cui, G. Zhang, B. Wang, and X. Su, "A Review of Automated Data Science Based on Large Language Models," in *2025 21st International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2025, pp. 510-517. [Online]. Available: <https://www.semanticscholar.org/paper/35635b4ac468502ee75a3ced96e5559e7504d362>
- [9]. E. L. Estevanell-Valladares, S. Estévez-Velarde, Y. Gutiérrez, A. Montoyo, and R. Mitkov, "XAutoLM: Efficient Fine-Tuning of Language Models via Meta-Learning and AutoML," arXiv preprint arXiv:2508.00924, Aug. 2025. [Online]. Available: <https://arxiv.org/html/2508.00924v1>
- [10]. "In-Context Meta-Learning with Large Language Models for Automated Model and Hyperparameter Selection," OpenReview, Sep. 2025. [Online]. Available: <https://openreview.net/forum?id=Ppo2SETqcE>.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details