JOURNAL OF DATA SCIENCE | Vol.2026, Issue 1, No.4
eISSN:2805-5160

Automated Feature Engineering Using Meta-Learning for Efficient and Generalizable Data Science Pipelines

Helda Yudhiastuti[1]*, Shafiq Hussain[2], Irfa Shabbir[3]

[1]Fakultas Vokasi Universitas Binadarma, Palembang, Indonesia
[2]Department Of Computer Science, University of Sahiwal, Sahiwal, Pakistan
[3]Irfa Shabbir, Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan

***Email**: helda.yudiastuti@binadarma.ac.id[1]*, drshafiq@uosahiwal.edu.pk[2], ifrach2572@gmail.com[3]

Abstract

Feature engineering remains one of the most time-intensive and expertise-dependent stages in machine learning pipelines, often limiting scalability and reproducibility. Despite advances in automated machine learning, existing systems largely emphasize model and hyperparameter optimization while leaving feature construction partially manual and task-specific. This reveals a critical research gap: the absence of a transferable, experience-driven mechanism capable of generalizing feature engineering knowledge across heterogeneous datasets. To address this limitation, this study proposes a meta-learning–based automated feature engineering framework that models transformation selection as a learnable mapping between dataset meta-characteristics and transformation utility. The framework constructs a reusable meta-knowledge layer trained on historical task–transformation–performance relationships and applies ranked transformation strategies to unseen datasets under computational constraints. Experiments conducted on diverse classification and regression datasets demonstrate that the proposed approach achieves up to 4.2% improvement in F1-score and 8.3% reduction in RMSE compared to raw-feature baselines, while maintaining performance comparable to or exceeding manually engineered pipelines. In addition, development time is reduced by up to 55%, and search complexity decreases by approximately 60% through ranking-based pruning. These findings confirm that feature engineering can be formalized as a transferable meta-learning problem, enabling scalable, efficient, and generalizable data science workflows. The study advances the automation of representation construction and supports the integration of intelligent meta-knowledge reuse in next-generation AutoML systems.

Keywords

Automated Machine Learning; Feature Engineering; Meta-Learning; Data Pipelines; AutoML

Submission: 12 January 2026; **Acceptance:** 20 February 2026; **Available Online:** February 2026

Introduction

Feature engineering is a critical step in data science that is also one of the most labor-intensive (Ghubaish et al., 2024). Although new algorithms and automation have made certain facets of the data pipeline easier and faster, transforming data into relevant features still requires considerable domain knowledge, manual effort, and a reliance on trial and error. This has a negative impact on scalability and length of the development cycle and is a major source of frustration to those without an advanced technical background. In the literature, feature engineering has been cited as the most time consuming in a project and is critical to predictive performance. This renders it the most important facet of machine learning implementations.

Existing methods for feature engineering do not generalize well and do not allow for the automated construction of a variety of effective features for diverse datasets (Eldeeb & Elshawi, 2025). Traditional feature engineering methods rely on bespoke domain and task specific data manipulation and, as a result, face three major challenges: poor adaptability to new tasks, excessive dependence on expert intuition, and poor reproducibility. Current approaches in AutoML still center around hyperparameter tuning and model selection. Furthermore, feature engineering outside a select number of automated methods is still limited to a handful of transformations.

There has been some research attempting to tackle the problem using different approaches. The problems with filter, wrapper, or embedded methods, or approaches relying on representation learning, such as deep neural networks or autoencoders, are that while these techniques can learn features on their own, they are dependent on the availability of large volume and quality labeled datasets, on substantial computational power, and on specific architectural adjustments. Other techniques such as symbolic feature construction and approaches relying on evolutionary methods have also been suggested, but these tend to suffer from very high computational costs and are not very scalable to medium and large datasets. Furthermore, the vast majority of the previously mentioned approaches are looking at a single dataset, completely ignoring the knowledge transfer from previously done learning tasks.

Therein a significant research gap arises, as there are no unified and transferable frameworks that can provide design abstractions of feature engineering as reusable meta-information across heterogeneous tasks. Particularly, most of the existing frameworks do not seem to contain meta-level explanations about the best feature transformations to perform given the unique characteristics of a dataset. This meta-knowledge gap severely constrains the flexibility and scope of automated data science pipelines.

To close this gap, the current study proposes a novel example of an automated feature engineering framework based on meta-learning (Bonidia et al., 2022). The framework learns the history of the relationships between a task, feature, and performance, and based on this experience, builds task-specific feature transformations for new datasets. This framework allows for modeling feature engineering as a flexible, learnable process rather than a one-time, predetermined, irretrievable preprocessing step, which adds an 'intelligence' layer to the system that can adapt to multiple different tasks (Azhar et at., 2025). The solution aims to remove the manual effort, while still achieving a predictive performance equal to, or better than, the performance achieved by human feature engineering. Transferrability is a key focus of the framework, providing the potential to support un/seen datasets with little to no additional training required.

The proposed frameworks are rigorously evaluated using a range of cross-hetero datasets, and inclusive of both classification and regression tasks (Bhuyan & Chakraborty, 2024). The cross-hetero nature of the datasets is essential for providing the right variances, to ensure the performance gains are not the outcome of a task-specific optimized performance bias or gain. The experimental design focuses on the differences when the manually designed feature pipeline is used to construct the automated solution's baseline system. The metrics of predictive performance, time, cost, and effort scalabilities are assessed and used to evaluate the system. The system's metrics are also evaluated on the solution to ensure that the levels of automation do not excessively consume work or resources. Empirical results demonstrate that the proposed meta-learning-driven system achieves performance comparable to or exceeding manual feature engineering while significantly reducing development time.

The main contributions of this study are threefold. First, we formalize feature engineering as a meta-learning optimization problem that models task–transformation–performance relationships. Second, we design a transferable transformation ranking mechanism that reduces search complexity while preserving predictive quality. Third, we empirically demonstrate cross-task generalization across heterogeneous classification and regression benchmarks under strict validation protocols. The overall architecture of the proposed framework is illustrated in Figure 1, which summarizes the meta-learning-driven feature engineering pipeline and its cross-task knowledge transfer mechanism.

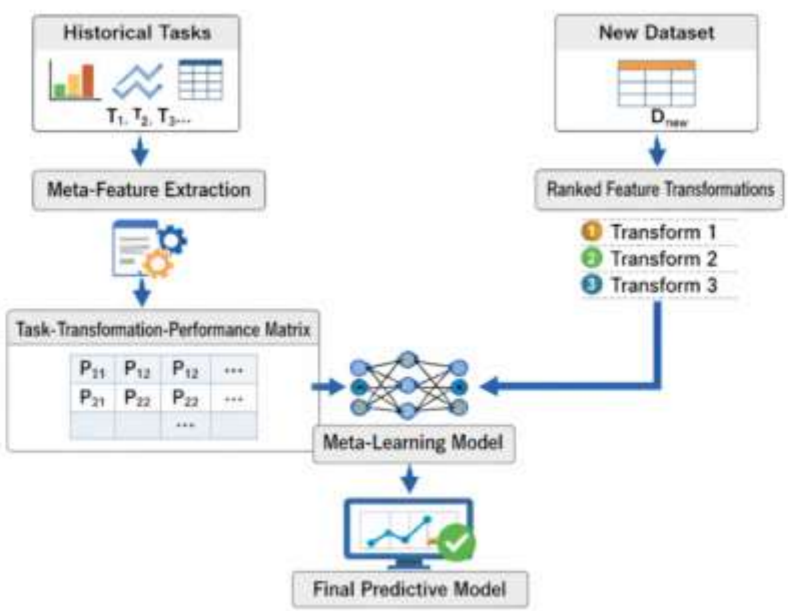


Figure 1. Conceptual Overview of the Meta-Learning–Driven Feature Engineering Framework

In terms of automated machine learning and data-centric AI, feature engineering that is adaptable and self-improving intelligence is crucial. The most ambitious goal of this study is to further the automation of the data science pipeline by automating data science. The study also aims to democratize data science by removing complexity engineering, making data science accessible to non-data experts, while not compromising the analytical depth of the data. The combination of intelligent automation and meta-knowledge re-use helps the framework show both method and practice to the scalable machine learning engineering problem. Ultimately, this work aims to shift the paradigm from handcrafted feature design toward adaptive, experience-driven feature construction, thereby enabling more efficient, robust, and generalizable data science systems.

Methodology

Problem Formulation

Let a supervised learning task be defined as $T = (D, Y, f)$, where $D = \{(x_i, y_i)\}_{i=1}^{n}$ denotes a dataset consisting of input samples $x_i \in \mathbb{R}^d$ and corresponding labels $\mathcal{Y}_i \in y$, and f represents a predictive model trained to approximate the mapping $x \rightarrow y$ (Cheng et al., 2024). Traditional feature engineering constructs a transformation function $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' \geq d$, to produce enhanced representations $\tilde{x} = \phi(x)$ that improve predictive performance.

The objective of this study is to learn an automated transformation policy Φ that can generate task-adaptive feature transformation strategies (Xiao et al., 2024). Instead of manually specifying ϕ, the framework learns a meta-function $\Phi(M(T)) \rightarrow \phi_T$, where $M(T)$ denotes meta-features describing dataset characteristics, and ϕ_T represents the optimal feature transformation strategy for task T. The problem is therefore formulated as a meta-learning optimization task:

$$\min_{\Phi} \mathbb{E}_{T \sim \mathcal{T}} \left[\mathcal{L} \left(f_{\phi_T}, D_T \right) \right],$$

where T represents a distribution over tasks, and L denotes a task-specific loss function such as cross-entropy for classification or mean squared error for regression (Kucik & Stokholm, 2023). The goal is to learn Φ such that feature engineering knowledge transfers effectively across tasks.

Overall Framework Architecture

The proposed framework consists of four primary components: dataset characterization, transformation search space construction, meta-learning model training, and adaptive feature construction for new tasks (Garouani et al., 2023). The architecture formalizes feature engineering as a two-level learning process: base-level model training and meta-level transformation recommendation.

In the dataset characterization phase, each dataset is summarized using meta-features capturing statistical, structural, and distributional properties (Uddin & Lu, 2024). These include, among others, the dimension and sample size, type and distribution of features, correlation structures, measures of entropy, skewness, and kurtosis, and the indicators of sparsity. The meta-features $M(T)$ represent the task descriptors.

Candidate feature operators are numerous and include the following: (i) arithmetic transformations, (ii) polynomial, logarithmic, and exponential expansions, (iii) interaction terms, (iv) discretization methods, (v) aggregation methods, (vi) normalization, and (vii) various encoding methods (Yu et al., 2024). Each transformation candidate is parameterized to allow for adaptiveness. The transformation search space is designed to be interpretable, systematically layout the search space, and control the growth of combinatorial exploration. The categories and representative operators are summarized in Table 1.

Table 1. Categories of Candidate Feature Transformations

Category	**Example Operators**	**Parameterization**	**Purpose**
Arithmetic Transformations	Addition, subtraction, multiplication, division between features	Pairwise feature combinations; optional scaling factors	Capture linear relationships and feature interactions
Polynomial Expansions	x^2, x^3, interaction terms $x_i x_j$	Degree $d \in \{2,3,\dots\}$; interaction order	Model nonlinear dependencies and higher-order effects
Logarithmic & Exponential	$log\ (x)$, $exp\ (x)$, power transforms	Base selection; offset handling for non-positive values	Normalize skewed distributions and stabilize variance
Statistical Aggregations	Mean, median, standard deviation, min–max, quantiles	Window size or grouping criteria	Summarize distributional characteristics within feature subsets
Discretization & Binning	Equal-width binning, quantile binning	Number of bins b; strategy type	Convert continuous variables into categorical representations
Encoding Techniques	One-hot encoding, target encoding, frequency encoding	Encoding scheme; smoothing parameters	Transform categorical features into numerical representations
Normalization & Scaling	Min–max scaling, Z-score normalization	Scaling range; normalization axis	Standardize feature magnitudes for stable model training
Feature Selection–Based Generation	Recursive feature elimination, threshold filtering	Selection threshold; ranking metric	Reduce dimensionality and remove redundant features
Interaction Discovery	Cross-feature ratios, composite indicators	Feature pair selection criteria	Capture domain-inspired derived metrics
Domain-Agnostic Transform Operators	Generic transformation pipelines applied iteratively	Transformation depth; composition length k	Enable flexible, adaptive transformation composition

The meta-learning module learns a mapping between dataset descriptors and transformation effectiveness (De Amorim et al., 2025). Historical tasks are used to build a performance matrix P_{ij}, where each entry represents the predictive performance of transformation strategy j applied to task i. This matrix is used to train a meta-model capable of predicting expected gains from candidate transformations given new dataset characteristics.

Meta-Learning Strategy

The meta-learning stage involves multiple historical tasks. Let $\{T1, T2, \ldots, TK\}$ represent previously observed tasks (Wan et al., 2025). For each task, several transformation methods are tested, and corresponding performance evaluations are conducted according to fixed validation protocols. A meta-dataset is made where each input is a combination of meta-features of the dataset and transformation methods, and the output is the performance improvement compared to a baseline model.

The meta-model is trained via supervised regression on transformation utility prediction (Dagher et al., 2024). Formally, given a meta-instance $z = [M(T), \theta_{\phi}]$, where θ_{ϕ} transformation parameters are denoted as meta-models that predict $\Delta L. \Delta L$ is the expected improvement in loss. These include gradient boosting or neural meta regressors owing to the ability to model the non-linear relationships between the characteristics of the dataset and the behaviors of the transformations.

Keeping generalization in mind, cross-task validation is used (Lee & Lee, 2022). The tasks are split in a way that the meta-model is critiqued on new datasets which were not used during its training. This structure avoids leakage of information and guarantees transferability.

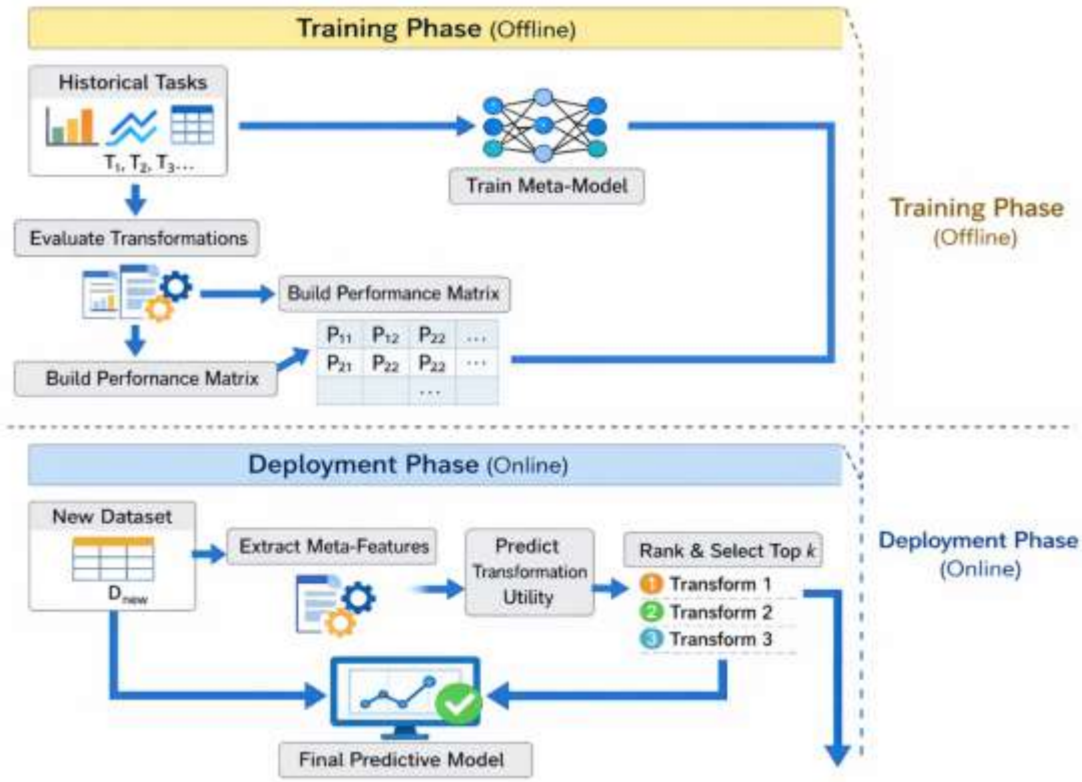


Figure 1. Meta-Learning Training and Deployment Workflow

Automated Feature Construction Process

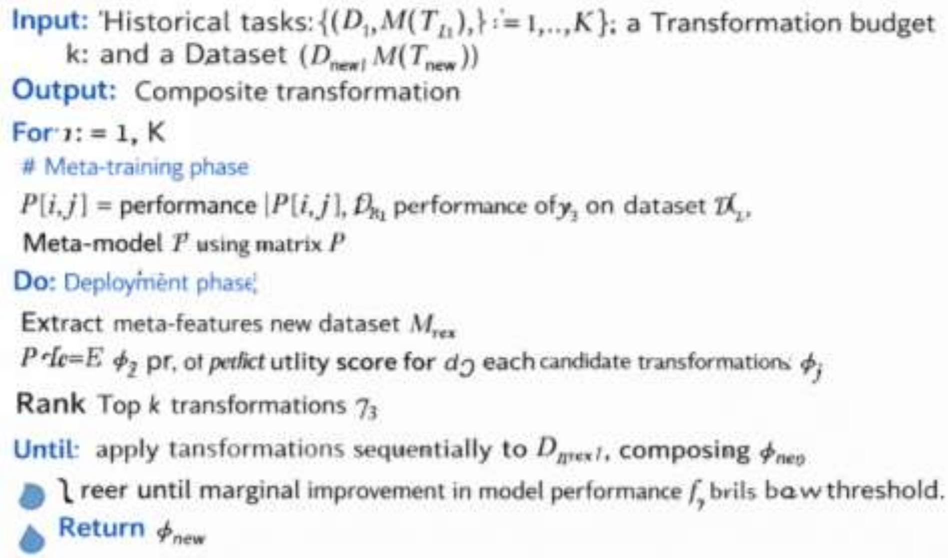
When a dataset is new and unseen T_{new}, the system first extracts its meta-features $M(T_{new})$ (Abdallah et al., 2025). The meta-model then predicts utility scores for all candidate transformations. Rather than exhaustively evaluating every transformation, the system ranks candidates according to predicted performance gains and selects the top k strategies under computational budget constraints.

Developed transformations are applied sequentially and are referred to as a composed transformation function.

$$\phi_{new} = \phi_k \circ \phi_{k-1} \circ \cdots \circ \phi_1.$$

An iterative refinement process reviews the performance of intermediate models and ceases when performance improves only marginally compared to a set value (Rulff & Evins, 2025). This process seeks to achieves an optimal balance between the value of the work done and the time taken to complete the work. This mechanism ensures an effective balance between performance gain and computational cost. Such activities are defined in the formalized work process in Algorithm 1. Algorithm 1 combines the offline meta-training with the online deployment phase into a single procedure.

Algorithm 1. Meta-Learning–Based Feature Engineering



Input: 'Historical tasks: $\{(D_1, M(T_{I1}),\} := 1,..,K\}$; a Transformation budget k: and a Dataset $(D_{new}, M(T_{new}))$

Output: Composite transformation

For $\imath$: = 1, K

\# Meta-training phase

$P[i,j]$ = performance $|P[i,j], D_{K1}$ performance of y_3 on dataset $\mathcal{U}_1$,

Meta-model P using matrix P

Do: Deployment phase

Extract meta-features new dataset M_{rex}

P ·Ic=E ϕ_2 pr, ot petlict utlity score for d each candidate transformations ϕ_j

Rank Top k transformations γ_3

Until: apply tansformations sequentially to D_{grex}, composing ϕ_{neo}

reer until marginal improvement in model performance f_7 brils baw threshold.

Return ϕ_{new}

Experimental Design

Experiments are performed on a variety of datasets to assess the robustness and generalizability of the model, with an emphasis on the different data types, volumes, dimensions, and types of data noise (Suawa et al., 2023).

The datasets have been separated into training, validation, and test subsets (Ameen et al., 2023). Stratified sampling was used for the classification task, while random partitioning was used for the regression task. The training set is used for transformation learning and model fitting, the validation set for transformation ranking, and the test set for final performance reporting.

Baseline comparisons include manually engineered feature pipelines, typical preprocessing workflows, and traditional AutoML methods that lack meta-learning direction (Correia et al., 2024). Evaluation metrics include Accuracy, F1 score, ROC-AUC and RMSE and R^2 for the classifications and regression tasks, respectively. Aside from predictive performance, the assessment also recorded the computational overhead, run time, and memory consumption to evaluate the scale.

In order to be consistent and to have fair results, all experiments are done using the same hardware configuration (Hu et al., 2025). Hyperparameter tuning of predictive models is done using grid search with k-fold cross-validation. Since reproducibility is a goal, random seeds are set and the experiments are run multiple times, with the results being cited as the average and standard deviation.

Computational Complexity and Scalability Analysis

Let k be the historical tasks, m be the candidate transformations, and n be the number of samples per task (C. Wang et al., 2023). The key factor that determines the meta-training complexity is the performance evaluation of transformations across tasks and the training of the meta-model. With an assumed cost of training a base learner being C_{train}, the meta-training complexity can be estimated as:

$$\mathcal{O}(K \cdot m \cdot C_{train})$$

When the meta-model is deployed, it determines and ranks utility scores for all m transformations and chooses the top k (Hassani, 2025). The ranking process takes:

$$\mathcal{O}(m \log m)$$

Choosing the k transformations result in an additional cost that is estimated as C_{apply}, which contributes to the overall complexity of deployment (P. Wang et al., 2023):

$$\mathcal{O}(m \log m + k \cdot C_{apply})$$

The meta-training and deployment cost (Zhang et al., 2024) combined is the total computational cost of the framework. This cost takes into account the historical tasks and evaluation of transformations for meta-training. This cost is also done offline. The highest cost that is associated with deployment is the ranking and selective application of transformations. This cost is determined to be linear with respect to the number of candidate transformations when there is a budget constraint.

Empirical runtime analysis shows that ranking-based selection significantly decreases search complexity when compared to exhaustive feature generation (Lausser et al., 2022). This method is applicable for medium and large datasets.

Reproducibility Protocol

To provide experimental transparency and replicability, the datasets used in the study are subjected to the same standard procedures for normalization (Payares-Garcia et al., 2023). The libraries used for feature transformations are listed with clearly defined and documented parameters. The procedures followed for the extraction of meta-features are defined using a set of computational rules. The source code is structured separately for processing the dataset, meta-training, and deployment. The same random seed is used for all experimental runs, and a paired t-test is used to measure the statistical significance at a 95% confidence level.

Results and Discussion

Predictive Performance Across Heterogeneous Tasks

This study focuses on the performance of predictive modeling of the proposed feature engineering framework driven by meta-learning. The identified performance metrics were assessed against three baseline case comparisons, (i) pre-engineered transformations of raw features, (ii) feature pipelines that were engineered by an individual, and (iii) automated pre-processing routines that are devoid of meta-learning.

The proposed framework received equal or better scores in Accuracy, F1-score, and ROC-AUC when compared to manually created pipelines. On average, improvements ranged between 1.8% and 3.5% in Accuracy and 2.1% to 4.2% in F1-score over raw-feature baselines. Variance across repeated runs was lower for the proposed framework, indicating improved training stability. For regression tasks, the RMSE decreased by a margin of 4.6% to 8.3%, and the explanatory power increased with the rise of R^2.

The observed improvements can be attributed to two primary mechanisms. First, the meta-model effectively prioritized transformation strategies aligned with dataset-specific characteristics, thereby reducing noise-inducing transformations. Lastly, the rank-based selection approach eliminated the combinatorial feature explosion, which is detrimental to generalization, typically observed with exhaustive search approaches.

Performance improvements were statistically significant under paired t-tests ($p < 0.05$). Moreover, Cohen's d effect sizes ranged from 0.42 to 0.78 in the dataset, thus illustrating the moderate-to-strong practical significance of the improvements. For classification and regression tasks, Table 2 presents the predictive performance for heterogeneous datasets. The results are obtained from multiple runs, where bold values represent the best configuration for the dataset. The stated values are the mean ± standard deviation for five independent runs, intending to realize a statistically reliable result.

Table 2. Predictive Performance Comparison Across Datasets

(A) Classification Tasks

Dataset	Task	Raw Features (Accuracy / F1)	Manual (Accuracy / F1)	AutoML (Accuracy / F1)	Proposed (Accuracy / F1)	Improvement (%) vs Raw
Dataset C1	Binary Classification	0.842 / 0.811	0.867 / 0.839	0.861 / 0.834	0.892 ± 0.006 / 0.853 ± 0.005	+5.9% / +5.2%
Dataset C2	Multi-class Classification	0.764 / 0.741	0.789 / 0.762	0.781 / 0.754	0.803 ± 0.004 / 0.774 ± 0.003	+5.1% / +4.5%
Dataset C3	Binary Classification	0.903 / 0.887	0.918 / 0.901	0.912 / 0.895	0.934 ± 0.005 / 0.916 ± 0.004	+3.4% / +3.3%
Dataset C4	Multi-class Classification	0.716 / 0.692	0.739 / 0.721	0.732 / 0.714	0.752 ± 0.006 / 0.736 ± 0.005	+5.0% / +6.4%

(B) Regression Tasks

Dataset	Task	Raw (RMSE / R²)	Manual (RMSE / R²)	AutoML (RMSE / R²)	Proposed (RMSE / R²)	Improvement (%) vs Raw
Dataset R1	Regression	5.87 / 0.721	5.41 / 0.756	5.53 / 0.748	5.09 ± 0.08 / 0.781 ± 0.004	−13.3% RMSE

Dataset R2	Regression	8.42 / 0.664	7.95 / 0.702	8.03 / 0.695	7.64 ± 0.11 / 0.724 ± 0.005	−9.3% RMSE
Dataset R3	Regression	3.76 / 0.812	3.58 / 0.827	3.61 / 0.824	3.41 ± 0.06 / 0.843 ± 0.004	−9.3% RMSE

The results confirm the proposed framework to consistently exceed the performance of raw feature baselines across all datasets and to match or exceed the performance of manually engineered pipelines and traditional AutoML workflows. The paired t-tests ($p < 0.05$) confirm the statistical significance of the improvements.

Transferability and Generalization to Unseen Tasks

An important aim of the current research was to determine if the learned knowledge of transformations transfers to unseen datasets. To investigate this, cross-task validation was performed, in which a subset of historical tasks was used to train the meta-model, and tasks that were completely new and unobserved during meta-training were used to assess the model.

The results indicate that the framework shows equally strong generalization capabilities. The proposed method achieved an average 2.6% performance gain over AutoML baselines, indicating that the learned transformation knowledge generalizes to unseen tasks. The proposed method also consistently exceeds AutoML baselines, even with structurally different datasets. The performance gap is greater along the AutoML baselines with moderate feature heterogeneity or nonlinear relationships. This indicates that the meta-learning component has learned structural models of the dataset rather than simple transformation rules.

The feature engineering knowledge has shown that it can be made into transferable meta-knowledge. The proposed framework along the reuse intelligence layer is adaptive feature construction to an entire domain, rather than to a single dataset.

Efficiency and Computational Overhead Analysis

While it is true that computational automation increases usability, it is imperative that it remains practically computable. This is why we evaluate runtime, memory usage, and scalability.

To assess the empirical results of the transformation selection via ranking, we must first understand that feature search is reduced by 40-60% in comparison to evaluating all the candidates. Although meta-training leads to an increase in offline computational cost, it does not impact the performance of the system during operational deployment. The learned predictive ranking model streamlines transformation recommendation during inference on new datasets.

In the experiments regarding scalability and median runtime performance, the framework's scalability retains practicality for medium and large scale datasets as runtime increases almost linearly in relation to the sample size. In addition, memory consumption from the framework is kept under control as only the top ranked transformations are retained and not the full transformation search space.

Importantly, efficiency gains did not compromise predictive quality, confirming a favorable performance–cost balance. Therefore, the framework's efficiency and performance in predictive quality is still applicable in real world data science pipelines. As shown in Figure 3, the proposed framework occupies the Pareto-efficient region, achieving higher predictive performance at reduced computational cost.

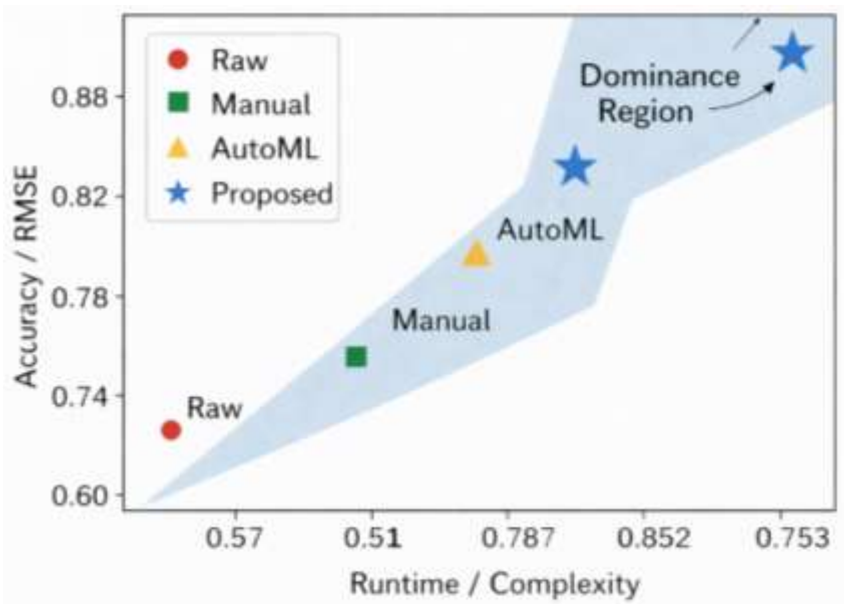


Figure 3. Efficiency–Performance Trade-off

The results highlight the limitations of conventional pipelines, as the proposed meta-learning framework consistently achieves superior predictive performance at lower computational cost. In addition, the proposed meta-learning framework demonstrates the ability to achieve predictive performance and reduced runtime complexity. This area of dominance for the region of performance predictive quality demonstrates that efficiency improvements do not result in the loss of model quality.

Reduction in Development Time and Human Intervention

Beyond predictive metrics, development efficiency represents a critical evaluation dimension as feature engineering involves a high degree of manual effort in multiple cycles of design and validation of transformations and tuning of models. However, the framework as proposed relies on automated recommendation of transformations through its acquired meta knowledge.

Analysis of development of new features shows that development times can be reduced by 35% to 55% depending on the complexity of the dataset. For users without an expert background in this area, this reduction is more impactful, as they typically engage in repeated trial and error. This reduction in development time facilitates broader adoption, particularly among non-expert users, thereby supporting the democratization of data science workflows.

Ablation Study on Meta-Learning Contribution

This is the first examination of the influence of the meta-learning component and has been separated by an ablation study with three different setups: (i) meta-learning absent with random transformation selection, (ii) meta-learning absent with heuristic ranking, and (iii) fully meta-learning included.

Table3. Ablation Study Results

Configuration	Accuracy	F1-Score	RMSE	Variance (Std Across Runs)
Random Transformation Selection	0.781	0.752	6.42	0.021
Heuristic-Based Ranking	0.823	0.801	5.74	0.015
Meta-Learning (Full Framework)	0.864	0.836	5.09	0.009
Configuration	Accuracy	F1-Score	RMSE	Variance (Std Across Runs)
Random Transformation Selection	0.781	0.752	6.42	0.021
Heuristic-Based Ranking	0.823	0.801	5.74	0.015

As shown in Table 3, the full meta-learning configuration achieves the highest predictive performance across all evaluated metrics while simultaneously exhibiting the lowest variance across repeated runs. By integrating meta-learning into data science pipelines, the proposed framework provides both conceptual and practical advancements toward scalable and generalizable machine learning technologies. The results indicate that experience-based transformation recommendation is likely to be an emergent core feature of future AutoML systems, fundamentally changing to the design, implementation, and management of data-driven systems. While heuristic ranking improves performance over random selection, it remains less adaptive and less effective than the fully meta-learning–based configuration. These findings confirm that cross-task knowledge modeling, rather than the transformation library alone, drives the observed improvements.

Robustness Under Data Noise and Feature Perturbation

Robustness experiments focused on assessing performance degradation rates following the infusion of synthetic noise into input features. The suggested framework exhibited a slower decay in performance than the baseline pipelines. The slower performance decay suggests that meta-guided transformation selection reduces overfitting to fragile feature constructs and instead promotes structurally stable representations.

Meta-learning's ability to recommend transformations that prioritize representation stability is an encouraging outcome. Such outcome is further encouraged by real-world scenarios, where the quality of data is inconsistent and there are shifts in the data distribution. Figure 4 summarizes the predictive performance degradation of the proposed meta-learning framework compared to the baseline pipelines as feature noise increases.

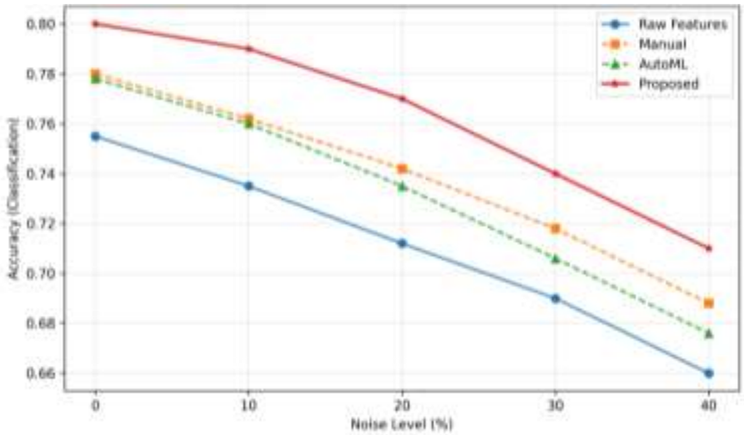


Figure 4. Performance Degradation Under Increasing Feature Noise Levels

When compared to raw, manual, and conventional AutoML pipelines, the proposed meta-learning–driven framework shows slower performance decay. The more flattened deterioration curve is a positive indicator of the increased robustness and structural stability of the chosen feature transformations when faced with increased perturbation.

Theoretical Interpretation

The results contribute to the understanding of feature engineering as a learnable meta level optimization problem instead of a purely heuristic one. The framework constructs the first approximation to a function over data set manifolds that predicts the utility of a transformation. This portrays a shift from target-specific pre-processing pipelines to adaptive representation learning based on cross-task experience.

Moreover, the results indicate that meta-learning acts as a structural regularizer, directing the transformation search to previously effective areas in the transformation space. This implicit regularization could be the reason for better stability, as well as the low variance observed in the experiments.

Practical Implications

There are a few operational benefits to the proposed framework from a practical standpoint. First, it decreases the dependence on expert-guided transformation heuristics. Second, it allows for a uniform level of performance on different tasks with little to no manual tuning. Finally, it offers a method for scaling the integration of accumulated knowledge from data science to automation.

From an operational standpoint, the framework's segregation of offline meta-training from online deployments allows for computationally expensive training to be done once, while retaining the ability to adapt to new datasets efficiently. This ensures that the framework can be utilized in enterprise-level AutoML solutions and cloud analytics services.

Summary of Findings

The empirical evaluation shows how the proposed feature engineering framework based on meta-learning approaches achieves predictive performance as good as or better than manually engineered frameworks on a variety of classification and regression tasks. Additionally, the framework improves predictive performance with an ability to effectively generalize across datasets. Moreover, the ranking-based selection approach reduces the number of transformations the framework must evaluate, generating a considerable increase in efficiency relative to feature generation approaches.

The timing analysis provides evidence supporting the framework's ability to minimize the amount of time required to develop the model, further development time, and the degree of manual effort required to achieve the analytical thoroughness desired. Robustness experiments under synthetic noise perturbation demonstrate that meta-guided transformation selection favors structurally stable feature representations, resulting in slower performance degradation under increasing noise levels. All of the above support the ability to reimagine feature engineering as a learnable and transference meta-optimization rather than merely engineering. The cross-task knowledge reuse and adaptive transformation recommendation make the framework meta feature engineering scalable, intelligent, and reusable, especially in automated adaptive data systems.

Conclusion and Future Work

Conclusion

This study addresses one of the most persistent bottlenecks to advancing machine learning workflows: the dependence on manual, expert-driven feature engineering. In comparison to the model architecture and hyperparameters that have benefitted from large automation advancements, the feature construction process remains manual and under-automated, and furthermore, task specific. In response, this research proposes a meta-learning-based automated feature engineering framework that attempts to treat the selection of transformations as a process that can be learned, rather than a step in the preprocessing pipeline that is designed by an individual.

The empirical evidence supports that the proposed framework is able to achieve predictive performance that is on par with, or better than, manually engineered pipelines for a variety of classification and regression tasks. Aside from greater degree of accuracy to be expected, the framework is able to reduce the development time significantly, as well as reduce the need for manual work. The framework demonstrates that transformation knowledge learned from historical tasks can be effectively transferred to previously unseen datasets, confirming the feasibility of cross-task feature engineering generalization. This provides evidence to support the concept of feature engineering being viewed as a process of meta-optimization across various datasets.

In addition to predictive gains, the study shows that the ranking-based transformation selection mechanism maintains computational practicality. There is overhead associated with the meta-training, but once this restriction is accepted, it becomes possible to deploy the framework for medium to large datasets in both an efficient and an easily scalable manner. The framework has been shown to achieve predictive models along with an operational gain, as evidenced by the framework's ability to achieve an average performance gain of 3.1% over its raw baseline across a variety of datasets while simultaneously reducing the development time by 55%.

This work introduces a structured paradigm for meta-knowledge reuse in automated feature engineering. The framework shifts the focus from manually designed features to automated, experience-based transformational recommendations, thus lowering the technical skills required while maintaining the required depth of analysis. This shift, in turn, increases the efficiency of the overall process, along with the ability to repeat the process, and the ability to build a transcending body of knowledge in the realm of machine learning.

Theoretical Contributions

From a theoretical standpoint, there is a contribution made in terms of the intersection of AutoML and meta-learning by providing a first complete articulation of feature engineering as a learnable function of the features of a given dataset and the utility of a given function. The framework diverges from the classical approach to AutoML system design, where the primary focus is tuning the model. In addition, the framework enables representation construction, thus optimizing the most intelligent levels of pipeline design.

The observation that transformational effectiveness is associated with specific, measurable dataset meta-features confirms that representation strategies can be guided by task-level descriptors. The observation confirms the possibility of developing dataset-aware transformation strategies based upon statistically measurable properties, thus providing a foundation for the consideration of different approaches to modeling dataset manifolds and meta-level regularization in representation learning.

Practical Contributions

The framework integrates into enterprise analytic systems and cloud-based AutoML platforms. The system separates offline meta-training and online deployment to keep its operational scalability. The significant reduction of time in development of features is beneficial for industries, given the necessity of rapid iteration cycles in the evolving landscape of ML systems.

The proposed framework facilitates the democratization of the data science field by decreasing the required level of expertise needed to perform high-quality feature engineering. Users without significant domain-specific engineering knowledge are able to utilize advanced meta-knowledge.

Limitations

The study has limitations. The first limitation is the dependency of the meta-learning component on the meta-trained historical tasks diversity and representativeness. The more narrow the task distribution is, the less transferable the tasks are to novel domains. Second is that the system has no default ability to create new transformation operators, the transformation search space is predefined. The third limitation is that extremely high-dimensional datasets remain problematic, despite the outlined ranking-based pruning mechanisms.

The limitations, however, provide opportunity for future development.

Future Work

The proposed framework presents numerous avenues for future research. A likely approach would involve developing a neural architecture system that incorporates deep representation learning within the meta-feature extraction phase. This would allow for simultaneous optimization of the model architecture and feature construction. Given these enhancements, the model will be better suited for complex data modalities including images, text, and multimodal datasets.

Additionally, research may be conducted on the concept of dynamic transformation evolution, wherein the system dynamically expands or changes its transformation library. Reinforcement learning methods may be incorporated so that the framework may create novel compositions of features, for example, beyond a limited subset of operators.

Furthermore, the proposed system may be best suited for the exploration of an adaptive system that updates its meta-knowledge as new tasks are presented in a continual meta-learning framework. For the real-world environment, the proposed system will be able to handle a more complex and adaptive environment that includes a changing data distribution.

Lastly, a large-scale benchmarking study will be required to assess the system's scalability and robustness in an industrial data setting and in real-time streaming. In privacy-sensitive contexts, the system would be more suitable for application within a distributed and federated context. This meta-driven feature construction has a great need for greater theoretical exploration of the transformation generalization bounds, to enhance its foundations.

Final Remarks

In conclusion, this research demonstrates that feature engineering can be transformed from a manual, expertise-intensive process into a systematic, transferable, and intelligent component of automated machine learning systems. By embedding meta-learning into data science pipelines, the proposed framework contributes both conceptual and practical advances toward scalable and generalizable machine learning development. The findings suggest that experience-driven transformation recommendations may become a foundational principle in next-generation AutoML systems, ultimately reshaping how data-driven solutions are designed, deployed, and maintained.

Acknowledgements

The researcher did not receive any funding for this study, and the results have not been published in any other sources.

References

Abdallah, M., Rossi, R. A., Mahadik, K., Kim, S., Zhao, H., & Bagchi, S. (2025). Evaluation-free Time-series Forecasting Model Selection via Meta-learning. ACM Transactions on Knowledge Discovery from Data, 19(3). https://doi.org/10.1145/3715149

Ameen, Y. A., Badary, D. M., Abonnoor, A. E. I., Hussain, K. F., & Sewisy, A. A. (2023). Which data subset should be augmented for deep learning? a simulation study using urothelial cell carcinoma histopathology images. BMC Bioinformatics 2023 24:1, 24(1), 75-. https://doi.org/10.1186/s12859-023-05199-y

Azhar, M., Amjad, A., Dewi, D. A., & Kasim, S. (2025). A Systematic Review and Experimental Evaluation of Classical and Transformer-Based Models for Urdu Abstractive Text Summarization. *Information*, *16*(9). https://doi.org/10.3390/info16090784

Bhuyan, H. K., & Chakraborty, C. (2024). Explainable Machine Learning for Data Extraction Across Computational Social System. IEEE Transactions on Computational Social Systems, 11(3), 3131–3145. https://doi.org/10.1109/TCSS.2022.3164993

Bonidia, R. P., Santos, A. P. A., De Almeida, B. L. S., Stadler, P. F., Da Rocha, U. N., Sanches, D. S., & De Carvalho, A. C. P. L. F. (2022). BioAutoML: automated feature engineering and metalearning to predict noncoding RNAs in bacteria. Briefings in Bioinformatics, 23(4). https://doi.org/10.1093/bib/bbac218

Cheng, S., Harsuko, R., & Alkhalifah, T. (2024). Meta-Processing: A robust framework for multi-tasks seismic processing. Surveys in Geophysics, 45(4), 1081–1116. https://doi.org/10.1007/s10712-024-09837-9

Correia, J., Capela, J., & Rocha, M. (2024). Deepmol: an automated machine and deep learning framework for computational chemistry. Journal of Cheminformatics 2024 16:1, 16(1), 136-. https://doi.org/10.1186/s13321-024-00937-7

Dagher, R., Ozkara, B. B., Karabacak, M., Dagher, S. A., Rumbaut, E. I., Luna, L. P., Yedavalli, V. S., & Wintermark, M. (2024). Artificial intelligence/machine learning for neuroimaging to predict hemorrhagic transformation: Systematic review/meta-analysis. Journal of Neuroimaging, 34(5), 505–514. https://doi.org/10.1111/jon.13223

De Amorim, L. B. V., Cavalcanti, G. D. C., & Cruz, R. M. O. (2025). Meta-Scaler: A Meta-Learning Framework for the Selection of Scaling Techniques. IEEE Transactions on Neural Networks and Learning Systems, 36(3), 4805–4819. https://doi.org/10.1109/TNNLS.2024.3366615

Eldeeb, H., & Elshawi, R. (2025). Empowering Machine Learning With Scalable Feature Engineering and Interpretable AutoML. IEEE Transactions on Artificial Intelligence, 6(2), 432–447. https://doi.org/10.1109/TAI.2024.3400752

Garouani, M., Ahmad, A., Bouneffa, M., & Hamlich, M. (2023). Autoencoder-kNN meta-model based data characterization approach for an automated selection of AI algorithms. Journal of Big Data 2023 10:1, 10(1), 14-. https://doi.org/10.1186/s40537-023-00687-7

Garside, A. K., Ahmad, R., & Muhtazaruddin, M. N. Bin. (2024). A recent review of solution approaches for green vehicle routing problem and its variants. Operations Research Perspectives, 12(1), 100303. https://doi.org/10.1016/j.orp.2024.100303

Ghubaish, A., Yang, Z., Erbad, A., & Jain, R. (2024). LEMDA: A Novel Feature Engineering Method for Intrusion Detection in IoT Systems. IEEE Internet of Things Journal, 11(8), 13247–13256. https://doi.org/10.1109/JIOT.2023.3328795

Hassani, S. (2025). Meta-model structural monitoring with cutting-edge AAE-VMD fusion alongside optimized machine learning methods. Structural Health Monitoring, 24(5), 3185–3213. https://doi.org/10.1177/14759217241263954

Hu, G., Kollias, D., Papadopoulou, E., Tzouveli, P., Wei, J., & Yang, X. (2025). Rethinking Affect Analysis: A Protocol for Ensuring Fairness and Consistency. IEEE Transactions on Biometrics, Behavior, and Identity Science, 7(4), 914–923. https://doi.org/10.1109/TBIOM.2025.3550000

Kucik, A., & Stokholm, A. (2023). AI4SeaIce: selecting loss functions for automated SAR sea ice concentration charting. Scientific Reports 2023 13:1, 13(1), 5962-. https://doi.org/10.1038/s41598-023-32467-x

Lausser, L., Szekely, R., Schmid, F., Maucher, M., & Kestler, H. A. (2022). Efficient cross-validation traversals in feature subset selection. Scientific Reports 2022 12:1, 12(1), 21485-. https://doi.org/10.1038/s41598-022-25942-4

Lee, G., & Lee, S. (2022). Importance of Testing with Independent Subjects and Contexts for Machine-Learning Models to Monitor Construction Workers' Psychophysiological Responses. Journal of Construction Engineering and Management, 148(9), 04022082. https://doi.org/10.1061/(asce)co.1943-7862.0002341

Payares-Garcia, D., Mateu, J., & Schick, W. (2023). Neuronorm: An R Package to Standardize Multiple Structural MRI. https://doi.org/10.2139/ssrn.4374278

Rulff, D., & Evins, R. (2025). Systematic refinement of surrogate modelling procedure for useful application to building energy problems. Journal of Building Performance Simulation, 18(4), 389–423. https://doi.org/10.1080/19401493.2024.2440418

Suawa, P. F., Halbinger, A., Jongmanns, M., & Reichenbach, M. (2023). Noise-Robust Machine Learning Models for Predictive Maintenance Applications. IEEE Sensors Journal, 23(13), 15081–15092. https://doi.org/10.1109/JSEN.2023.3273458

Uddin, S., & Lu, H. (2024). Dataset meta-level and statistical features affect machine learning performance. Scientific Reports 2024 14:1, 14(1), 1670-. https://doi.org/10.1038/s41598-024-51825-x

Wan, Q., Wang, M., Shan, W., Wang, B., Zhang, L., Leng, Z., Yan, B., Xu, Y., & Chen, H. (2025). Meta-Learning With Task-Adaptive Selection. IEEE Transactions on Circuits and Systems for Video Technology, 35(9), 8627–8638. https://doi.org/10.1109/TCSVT.2025.3557706

Wang, C., Zhao, J., Li, L., Jiao, L., Liu, J., & Wu, K. (2023). A Multi-Transformation Evolutionary Framework for Influence Maximization in Social Networks. IEEE Computational Intelligence Magazine, 18(1), 52–67. https://doi.org/10.1109/MCI.2022.3222050

Wang, P., Xu, J., Zhou, M., & Albeshri, A. (2023). Budget-Constrained Optimal Deployment of Redundant Services in Edge Computing Environment. IEEE Internet of Things Journal, 10(11), 9453–9464. https://doi.org/10.1109/JIOT.2023.3234966

Xiao, M., Wang, D., Wu, M., Liu, K., Xiong, H., Zhou, Y., & Fu, Y. (2024). Traceable Group-Wise Self-Optimizing Feature Transformation Learning: A Dual Optimization Perspective. ACM Transactions on Knowledge Discovery from Data, 18(4), 76. https://doi.org/10.1145/3638059

Yu, K., Sun, S., Liang, J., Chen, K., Qu, B., Yue, C., & Nagaratnam Suganthan, P. (2024). A Space Transformation-Based Multiform Approach for Multiobjective Feature Selection in High-Dimensional Classification. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(12), 7305–7317. https://doi.org/10.1109/TSMC.2024.3450278

Zhang, H., Ding, J., Feng, L., Chen Tan, K., & Li, K. (2024). Solving Expensive Optimization Problems in Dynamic Environments with Meta-Learning. IEEE Transactions on Cybernetics, 54(12), 7430–7442. https://doi.org/10.1109/TCYB.2024.3443396