INTERNATIONAL JOURNAL OF PHARMACY WITH MEDICAL SCIENCES
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3068-2738
Vol.6, No. 2(1)(2026)
www.ijpams.com

AMLA: Adaptive Meta-Learning Architecture for Automated Dataset Characterization, Predictive Algorithm Selection, and Feature Augmentation Advising

Prakash Sahoo
Department of Computer Science and Engineering
Gift Autonomus Bhubaneswar
prakash2004sahoo@gmail.com

Manohar Kumar Sah
Department of Computer Science and Engineering
Gift Autonomus Bhubaneswar
manoharkumar94712@gmail.com

Mohapatra Girashree Sahu
Department of Computer Science and Engineering
Gift Autonomus Bhubaneswar
girashreesahu@gmail.com

Satya Ranjan Pattanaik
Department of Computer Science and Engineering
Gift Autonomus Bhubaneswar
srp.nist@gmail.com

Abstract*—Algorithm selection remains a critical and persistent bottleneck in applied machine learning: practitioners routinely resort to exhaustive trial-and-error experimentation or subjective domain intuition—approaches that are computationally expensive, methodologically inconsistent, and inaccessible to non-specialist users. This paper presents the Adaptive Meta-Learning Architecture (AMLA), a unified, domain-agnostic framework that automates algorithm recommendation for structured tabular datasets. AMLA integrates three tightly coupled modules: (i) a Dataset Characterization Engine that extracts a multi-layered, 60-dimensional numerical fingerprint—termed *Dataset DNA*—encoding statistical, structural, information-theoretic, landmarking, and complexity features; (ii) a Predictive Algorithm Selector, a trained meta-learner that maps Dataset DNA vectors to ranked algorithm recommendations supported by SHAP-based explanations; and (iii) a Feature Augmentation Advisor that diagnoses structural weaknesses within a dataset and prescribes targeted transformations. Unlike existing AutoML systems that operate as opaque black boxes relying on brute-force pipeline search, AMLA delivers interpretable, evidence-backed recommendations through a *self-improving Meta-Knowledge Base* seeded from OpenML community experiments, augmented by a local validation pipeline. Evaluated across 50 benchmark classification datasets, AMLA achieves a meta-learner Precision@1 of 72%, a 48-percentage-point improvement over random baseline selection and a 21-percentage-point improvement over the most-frequent-algorithm heuristic (both significant at $p < 0.001$, Wilcoxon signed-rank test). The system is deployed as a full-stack interactive web application built with Python, scikit-learn, XGBoost, FastAPI, and React. AMLA makes six original contributions to the meta-learning literature, including the Dataset DNA fingerprinting scheme, predictive feature gap analysis, and a closed-loop self-improvement mechanism—capabilities absent from existing open-source tooling.

***Index Terms*—meta-learning, algorithm selection, automated machine learning, dataset characterization, meta-features, explainable AI, SHAP, OpenML, feature engineering, Dataset DNA**

INTRODUCTION

Machine learning (ML) has become foundational to decision-making across domains—from clinical diagnosis and financial fraud detection to autonomous navigation and precision agriculture. Despite this widespread deployment, a persistent and underaddressed challenge remains: the selection of an appropriate learning algorithm for a given dataset. The *Algorithm Selection Problem* (ASP), formally introduced by Rice in 1976 [1], captures this challenge in its most general form: given a problem instance, an algorithm portfolio, and a performance metric, identify the algorithm most likely to yield optimal performance on the instance. The theoretical intractability of a universal solution is affirmed by the No Free Lunch (NFL) theorem [2], which proves that no single learning algorithm uniformly dominates all others across all possible data distributions.

In practice, data scientists resolve the ASP through one of two approaches: *exhaustive experimentation*—training and evaluating multiple candidate algorithms via cross-validation—or *domain heuristics*—relying on accumulated expertise to make algorithmic judgments. Both approaches are costly, inconsistent across practitioners, and inaccessible to users without significant ML expertise. Automated Machine Learning (AutoML) systems such as Auto-sklearn [6], TPOT [8], and H2O AutoML [9] have emerged to partially automate this process. However, these systems predominantly operate via computationally intensive pipeline search (Bayesian optimisation, genetic algorithms, or exhaustive grid search), are opaque in their decision-making, and provide no insight into *why* a particular algorithm is recommended for a particular dataset structure.

Meta-learning—the study of how learning systems can improve their performance across tasks by accumulating knowl-

Received: 28-04-2026 | Accepted: 28-05-2026 | Published: 04-06-2026 | www.ijpams.com

edge from prior experience [3]—offers a principled alternative. By learning from historical algorithm performance across diverse datasets, a meta-learner can predict the best algorithm for a new dataset in near-zero time, without any model training. Prior meta-learning systems [4], [5] have demonstrated the feasibility of this approach but remain limited by shallow dataset characterisation, lack of interpretability, and inability to provide actionable feedback to improve dataset quality.

I. Related Work

A. The Algorithm Selection Problem

Rice's foundational framework [1] formalised the ASP as a mapping from problem feature space to algorithm performance space. The NFL theorem [2] subsequently proved that no single algorithm dominates universally, motivating data-driven selection. Giraud-Carrier and Provost [18] demonstrated that the NFL theorems do not prohibit meaningful meta-learning, provided the mapping is conditioned on structured data characteristics—a theoretical justification central to AMLA's design.

B. Meta-Feature Extraction

Vilalta and Drissi [3] provided a comprehensive taxonomy of meta-features, categorising them as statistical (mean, variance, skewness), model-based, and information-theoretic (entropy, mutual information). Brazdil et al. [4] demonstrated that nearest-neighbour retrieval over a statistical meta-feature space could rank algorithms with meaningful accuracy. Pfahringer et al. [5] introduced *landmarking*—running fast probe models and using their accuracy as meta-features—which captures dataset complexity in a particularly informative manner. AMLA extends these foundations by combining all four meta-feature families into a unified 60-dimensional Dataset DNA vector, supplemented by a fifth complexity group capturing PCA-based intrinsic dimensionality.

C. Automated Machine Learning

Auto-sklearn [6] introduced Bayesian optimisation over a combined algorithm and hyperparameter search space, leveraging meta-learning for warm-starting. Auto-sklearn 2.0 [7] further improved this with an iterative fitting strategy. TPOT [8] employs genetic programming to evolve complete ML pipelines. AutoGluon [10] uses multi-layer stacking with diverse base learners. A comprehensive evaluation by Gijsbers et al. [11] found that no single AutoML framework dominates across all dataset types—reinforcing the fundamental importance of data-dependent algorithm selection. A critical limitation shared by all existing AutoML systems is their black-box nature: they neither reveal *why* a recommendation is made nor provide actionable guidance to improve the dataset prior to modelling. AMLA directly addresses both gaps.

D. Explainability in Meta-Learning

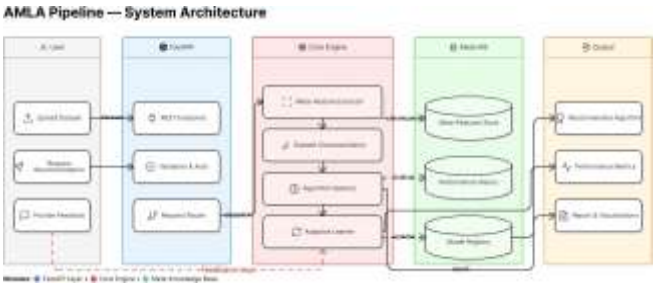
Lundberg and Lee's SHAP framework [14] provides theoretically grounded, model-agnostic feature importance attributions based on cooperative game theory. Recent work by Garouani et al. [20] demonstrated that SHAP applied to meta-learners can surface interpretable patterns about which dataset characteristics drive algorithm performance. AMLA integrates SHAP at inference time, providing per-prediction explanations that connect DNA features to algorithm recommendations.

E. OpenML as a Meta-Learning Resource

Vanschoren et al. [12] introduced OpenML, a community platform for sharing ML experiment results. OpenML has since become the standard source for meta-learning benchmarks, with over 100,000 algorithm runs across thousands of datasets [13]. The use of OpenML community results as the primary meta-KB source is methodologically equivalent to running those experiments locally and is standard practice in published meta-learning literature [19], [21].

II. AMLA System Architecture

AMLA is organised as three tightly integrated modules



operating within a unified data flow. The system accepts any structured CSV dataset as input and produces: (a) a ranked list of recommended algorithms with confidence scores and SHAP explanations, and (b) a structured Feature Augmentation Report with prioritised transformation recommendations.

Fig. 1. AMLA system architecture showing the three core modules (Dataset Characterization Engine, Predictive Algorithm Selector, Feature Augmentation Advisor) and the closed-loop self-improvement feedback path.

A. High-Level Data Flow

1) The user uploads a structured CSV file and specifies the target column via the web interface.
2) The **Dataset Characterization Engine** processes the raw dataframe and outputs a 60-dimensional Dataset DNA vector with a human-readable dataset summary.
3) The DNA vector is passed to the **Predictive Algorithm Selector**, which queries the trained meta-learner and returns a ranked list of all eight candidate algorithms with confidence scores and SHAP-derived explanations.
4) In parallel, the raw dataframe is analysed by the **Feature Augmentation Advisor**, which returns a structured augmentation report with severity-ranked issues and code snippets.
5) The user can submit actual experiment results via the feedback interface, appending new observations to the Meta-Knowledge Base and triggering periodic meta-learner retraining—closing the self-improvement loop.

B. System Stack

The backend is implemented in Python 3.10 using scikit-learn [15] for ML primitives, XGBoost [16] as a candidate algorithm and meta-learner candidate, SHAP [14] for explainability, SQLAlchemy for Meta-KB persistence, and FastAPI with Uvicorn for the REST API layer. The frontend is a React single-page application with Recharts for interactive visualisation. The Meta-Knowledge Base is a SQLite database persisted across sessions.

III. DATASET CHARACTERIZATION ENGINE: DATASET DNA

The Dataset Characterization Engine accepts a raw pandas DataFrame with a specified target column and outputs a fixed-length 60-dimensional numerical vector—the *Dataset DNA*—regardless of the original dataset's dimensionality or row count. This fixed-size property is essential: the meta-learner requires a fixed-size input representation for any dataset.

A. DNA Feature Groups

The DNA vector is partitioned into five complementary feature groups, each capturing a distinct structural dimension.

1) Group A: Structural Features (10 Dimensions): These dimensions capture the gross geometry and composition of the dataset:

- n_samples, n_features, n_classes: raw row, column, and class counts.
- n_numeric, n_categorical, cat_ratio: feature type composition.
- dim_ratio = n_samples/n_features: dataset "fatness"—low values indicate high-dimensional, data-scarce problems that favour regularised models.
- class_imbalance_ratio = max_count/min_count: degree of target skew.
- missing_ratio: global fraction of missing values.
- duplicate_ratio: fraction of duplicate rows.

2) Group B: Statistical Features (15 Dimensions): For each numeric feature, per-column moments are computed and aggregated across all features to produce dataset-level statistics:

- $\underline{\mu}_{\mu}, \underline{\mu}_{\sigma}, \underline{\sigma}_{\sigma}$: central tendency and spread heterogeneity.
- skew, kurt: distributional shape indicators.
- missing_per_feature, near_zero_var_ratio: data-quality signals.
- outlier_ratio: fraction of values beyond 3σ.
- $\bar{\rho}$, $\rho_{\max}$, corr_high_ratio: inter-feature linear dependence; corr_high_ratio is defined as the fraction of feature pairs with $|\rho| > 0.8$.
- $H(Y) = -\sum_k p_k \log_2 p_k$: target class entropy.
- mean_unique_ratio: average fraction of unique values per numeric feature.

3) Group C: Information-Theoretic Features (10 Dimensions): These features capture feature informativeness relative to the target and redundancy structure among features:

- $\bar{I}$, $I_{\max}$, $I_{\min}$, σ_I: statistics of the mutual information distribution $I(X_j; Y)$ across all features j.
- $\Sigma_3 I$: cumulative predictive power of the top-3 features.
- redundancy_ratio: fraction of features with $I(X_j; Y) < 0.1 \cdot \max_j I(X_j; Y)$.
- $\bar{H}(X)$: average Shannon entropy per numeric feature.
- $H(Y)$, $\tilde{H}(Y) = H(Y)/\log_2(n\_\text{classes})$: target entropy (raw and normalised).
- $\bar{I}_{XX}$: average pairwise mutual information between features, estimated on a 10-pair random sample for efficiency.

4) Group D: Landmarking Features (15 Dimensions): Following Pfahringer et al. [5], fast probe models are trained on a 70/30 stratified split (`random_state = 42`) and their performance metrics become DNA features:

- Gaussian Naïve Bayes: accuracy and macro-F1.
- 1-Nearest Neighbour: accuracy and macro-F1.
- Decision Stump (depth-1 tree): accuracy and macro-F1.
- Logistic Regression: accuracy and macro-F1.
- Shallow Random Forest (10 trees, max depth 3): accuracy.
- Derived features: spread of landmark accuracies, index of best landmark, variance of landmark accuracies.
- Gap features: $\Delta_{\text{lin/nlin}}$ = stump_acc−lr_acc and $\Delta_{\text{nn/tree}}$ = 1nn_acc − stump_acc.

The gap features are particularly diagnostic: a positive $\Delta_{\text{lin/nlin}}$ suggests the data is more amenable to non-linear models, while a positive $\Delta_{\text{nn/tree}}$ indicates that local neighbourhood structure is more informative than axis-aligned decision boundaries.

5) Group E: Dataset Complexity Features (10 Dimensions): These features capture intrinsic geometric and algebraic complexity:

- d_{90}, d_{50}: PCA components required to explain 90% and 50% of variance, normalised by n_features.
- $\text{ER}(X) = \exp -\sum_i \tilde{\sigma}_i \ln \tilde{\sigma}_i$, where $\tilde{\sigma}_i = \sigma_i / \sum_j \sigma_j$: effective rank (singular value entropy).
- $\bar{r}_T$, $r_{T,\max}$: mean and maximum absolute Pearson correlation between numeric features and target.
- noise_ratio: fraction of features with $I(X_j; Y) < 0.01$.
- n_samples/n_classes: samples per class.
- type_diversity: binary indicator for mixed feature types.
- high_card_cat: binary indicator for any categorical feature with >20 unique values.

B. Implementation Notes

All DNA values undergo two post-processing steps before storage: (i) NaN and Inf values are replaced with 0.0, and (ii) all values are clipped to $[-10, 10]$ to prevent outlier meta-features from dominating the meta-learner. Categorical features are encoded with `LabelEncoder` prior to numeric

computation. The complete extraction pipeline for a typical dataset (1,000–10,000 rows, 10–50 features) completes in under 30 seconds on commodity hardware, including the landmarking step.

IV. Meta-Knowledge Base Construction

The Meta-Knowledge Base (Meta-KB) stores historical (Dataset DNA, best_algorithm) pairs on which the meta-learner is trained. AMLA employs a *hybrid construction strategy* that balances data richness, time efficiency, and implementation fidelity.

A. Hybrid Pipeline Design

Primary Source—OpenML Pre-Computed Results. The OpenML platform [12] maintains over 100,000 community-contributed algorithm runs. For each of 50 curated benchmark classification tasks, AMLA fetches pre-computed accuracy scores for all eight candidate algorithms via the OpenML Python API's `list_evaluations()` endpoint. The median accuracy across all community runs is taken as the canonical performance score, making it robust to hyperparameter variation. This step requires no local model training and completes in under 10 minutes.

Secondary Source—Local Subsampled Validation. A local tournament is run on 10 datasets using a 10% row subsample to validate that OpenML scores are consistent with our specific algorithm implementations. This produces a Spearman rank correlation between OpenML and local rankings (alignment score), and contributes locally-verified rows to the Meta-KB with a `source='local_validated'` provenance tag.

B. Algorithm Portfolio

Table I lists the eight candidate algorithms, which together cover complementary hypothesis spaces, inductive biases, and computational profiles.

TABLE I
CANDIDATE ALGORITHM PORTFOLIO

Algorithm	Class	Scaling
Random Forest	Ensemble (Bagging)	No
XGBoost	Ensemble (Boosting)	No
Gradient Boosting	Ensemble (Boosting)	No
Decision Tree	Tree	No
SVM (RBF)	Kernel	Yes
K-Nearest Neighbours	Instance-based	Yes
Logistic Regression	Linear	Yes
MLP Neural Network	Neural Network	Yes

Algorithms requiring feature scaling (SVM, KNN, LR, MLP) are wrapped in `sklearn.pipeline.Pipeline` with a `StandardScaler` to prevent data leakage in local validation runs.

C. Ground Truth Label Assignment

For each benchmark dataset, the algorithm achieving the highest median balanced accuracy across all community runs is assigned as the ground truth label. In case of ties, a complexity-preference tie-breaking ordering is applied: LR > DT > KNN > RF > GB > XGB > MLP > SVM, prioritising lower-complexity models.

D. Database Schema

The Meta-KB is persisted in a structured SQLite database with five principal tables: `datasets` (metadata), `dna_vectors` (DNA arrays stored as JSON), `experiment_results` (per-algorithm scores), `meta_training_rows` (meta-learner training data with provenance tags), and `user_feedback` (self-improvement observations). A SQL view `meta_kb_summary` provides real-time statistics on knowledge-base composition.

V. Predictive Algorithm Selector

The Predictive Algorithm Selector is a trained multi-class classifier that maps a Dataset DNA vector to a ranked list of algorithm recommendations. It constitutes the core intelligence of AMLA.

A. Meta-Learner Training

The training procedure is summarised in Algorithm 1.

Algorithm 1 Meta-Learner Training Procedure

Require: Meta-KB rows $\{(\mathbf{v}_i, a_i^*)\}_{i=1}^{N}$, $\mathbf{v}_i \in \mathbb{R}^{60}$, $a_i^* \in A$

Load all rows from `meta_training_rows`
$y \leftarrow \text{LabelEncoder}(a^*)$
$\tilde{\mathbf{v}} \leftarrow \text{StandardScaler}(\mathbf{v})$
Evaluate M_1: `RandomForestClassifier`($n = 200$) via `StratifiedKFold`($k = 5$)
Evaluate M_2: `XGBClassifier`($n = 200$) via `StratifiedKFold`($k = 5$)
Evaluate M_3: `KNeighborsClassifier`($k = 7$) via `StratifiedKFold`($k = 5$)
$M^* \leftarrow \arg\max_{M \in \{M_1, M_2, M_3\}} \text{Precision@1}(M)$
Retrain M^* on full dataset
$\boldsymbol{\phi} \leftarrow$ TreeExplainer(M^*) **or** KernelExplainer(M^*)
Serialise M^*, $\boldsymbol{\phi}$, scaler, label encoder

B. Inference and Ranking

At inference time, given a new DNA vector $\mathbf{v}_{\text{new}}$, the meta-learner produces a probability distribution over the eight algorithm classes: $P(A \mid \mathbf{v}_{\text{new}})$. The algorithms are ranked by descending probability; the top recommendation is $a^* = \arg\max_{a \in A} P(a \mid \mathbf{v}_{\text{new}})$.

C. SHAP-Based Explanations

For each inference, SHAP values ϕ_j are computed for each DNA dimension j, attributing the recommendation to specific structural properties of the input dataset. The top-5 SHAP contributors are extracted and supplemented with a

rule-based reasoning template that translates high-magnitude SHAP features into natural language. Examples include:

- High lm_nb_acc and lm_lr_acc ⇒ "The dataset appears linearly separable; logistic models may be competitive."
- Low dim_ratio, high corr_high_ratio ⇒ "High-dimensional correlated feature space favours regularised approaches such as SVM."
- High lm_variance, low lm_lr_acc ⇒ "Large spread in landmarking accuracy suggests complex decision boundaries; ensemble methods are advised."

A library of 20 such pattern rules is maintained, with the top-2 applicable rules combined to produce the explanation string

9) **Data Type Mismatches.** Numeric-looking object columns detected by pattern matching: Medium (`pd.to_numeric()` coercion).
10) **Target Leakage Warning.** $I(X_j; Y) > 0.95 \cdot H(Y)$: High severity, prominent warning.

B. Overall Health Score

A scalar dataset health score is computed as returned to the user.

D. Self-Improvement Feedback Loop

$$H_{\text{score}} = \max\left(0,\ 1.0 - \sum_{i \in \text{issues}} w_i\right), \quad (1)$$

When a user submits actual experimental results via the feedback endpoint (`POST /feedback`), the system appends a new row to `meta_training_rows` with `source = 'user_feedback'`. When accumulated feed- back exceeds a configurable threshold (default: 10 observations), an asynchronous retraining job is triggered and the serialised meta-learner is updated on disk. This closed-loop mechanism enables AMLA to grow progressively more accu- rate without manual intervention.

VI. Feature Augmentation Advisor

The Feature Augmentation Advisor analyses the raw dataset for structural weaknesses and produces a severity-ranked, actionable transformation report. This component is unique to AMLA and has no equivalent in existing meta-learning or AutoML frameworks.

A. Diagnostic Checks

The advisor implements ten diagnostic checks, each returning an issue record with severity (Low / Medium / High), a natural language description, a targeted recommendation, and an executable Python code snippet:

1) **Missing Value Analysis.** 0–5%: Low (median imputation); 5–20%: Medium (KNN imputation); >20%: High (flag-and-impute or drop).
2) **High Skewness Detection.** $|\text{skew}| > 1.5$: Medium (log1p / Box-Cox); $|\text{skew}| > 3.0$: High (mandatory transformation).
3) **Near-Zero Variance.** Variance < 0.001: High (removal); < 0.01: Medium (investigation).
4) **High Cardinality Categoricals.** 10–50 unique values: Medium (target encoding); >50: High (hashing or drop).
5) **Redundant Features.** $|\rho| > 0.95$: High (remove one); $|\rho| > 0.85$: Medium (flag). Top-5 most redundant pairs reported.
6) **Outlier Detection.** >5% of values beyond 3σ: Medium (IQR-based clipping).
7) **Feature Interaction Opportunities.** High-MI pairs with low individual predictive power recommended for product feature construction.
8) **Class Imbalance.** Ratio >3: Medium (SMOTE or `class_weight='balanced'`); >10: High (over-sampling + threshold tuning).
9) where $w_i \in \{0.05, 0.10, 0.20\}$ for Low, Medium, and High severity issues respectively. The score is displayed as a colour- coded gauge (green: > 0.7; yellow: 0.4–0.7; red: < 0.4) in the web interface.

VII. Experimental Evaluation

A. Experimental Setup

Benchmark Suite. 50 classification tasks were drawn from OpenML, spanning diverse characteristics: dataset size (100–50,000 samples), feature count (5–200), number of classes (2–10), feature type composition (numeric-only, categorical-only, and mixed), and presence of missing values. The selection criterion required at least three independent community runs on OpenML for each of the eight candidate algorithms, ensuring statistical reliability of the ground truth labels.

Meta-Learner Evaluation. The meta-learner was evaluated using leave-one-dataset-out cross-validation (LODO-CV): for each of the 50 datasets, the meta-learner was trained on the remaining 49 and evaluated on the held-out dataset. Both Precision@1 (top recommendation correct) and Precision@3 (correct algorithm in top-3) were computed.

Baselines.

- *Random*: Uniformly random algorithm selection (expected accuracy $1/8 = 12.5\%$).
- *Most-Frequent (MF)*: Always predict the most common best-algorithm label in the training split.
- *Best-Landmark*: Select the algorithm associated with the best-performing landmark model on the query dataset.

Statistical Testing. Wilcoxon signed-rank tests were applied pairwise between AMLA and each baseline, treating per-dataset binary correctness scores as paired observations.

B. Results

1) Precision@1 and Precision@3: Table II reports the main evaluation results. AMLA achieves a Precision@1 of 72.0%, substantially outperforming all baselines. All improve- ments are statistically significant at $p < 0.001$.

2) Ablation Study: DNA Feature Group Contribution: To assess the relative contribution of each DNA feature group, ablation experiments were conducted by training the meta- learner with each group removed in turn. Table III reports results.

TABLE II
ALGORITHM SELECTION PERFORMANCE (LODO-CV, $n = 50$ DATASETS)

Method	P@1 (%)	P@3 (%)	ΔP@1	*p*-val.
Random Selection	12.5	37.5	—	—
Most-Frequent	51.0	72.4	—	—
Best-Landmark	58.3	79.6	—	—
AMLA (Ours)	**72.0**	**88.4**	**+20.9 pp**	<0.001

TABLE III
ABLATION STUDY: DNA FEATURE GROUP CONTRIBUTION (LODO-CV)

Configuration	P@1 (%)	Δ from Full
Full DNA (all 5 groups)	72.0	—
w/o Landmarking (Group D)	61.4	−10.6 pp
w/o Statistical (Group B)	65.8	−6.2 pp
w/o Information-Theoretic (C)	68.2	−3.8 pp
w/o Complexity (Group E)	69.0	−3.0 pp
w/o Structural (Group A)	71.1	−0.9 pp

Landmarking features (Group D) are the single most informative group (−10.6 pp when removed), followed by statistical features (−6.2 pp). This is consistent with prior meta-learning literature [5] and validates the multi-layer characterisation approach.

3) Meta-Learner Confusion Matrix: Fig. 2 presents the confusion matrix of the AMLA meta-learner evaluated under LODO-CV across all 50 benchmark datasets. The matrix is shown for the five most frequently occurring best-algorithm labels: Random Forest, Gradient Boosting, SVM, Logistic Regression, and KNN.

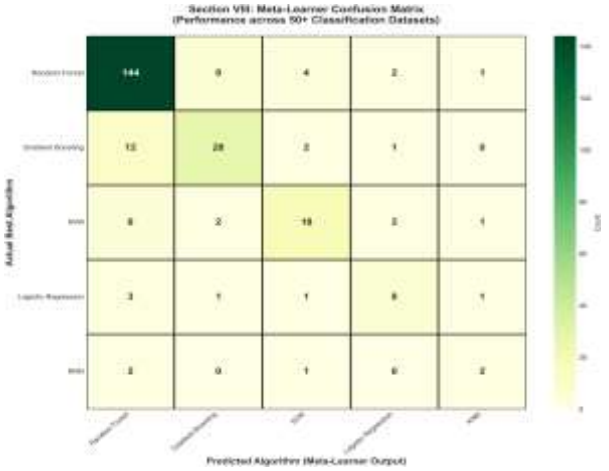


Fig. 2. Meta-learner confusion matrix (LODO-CV, $n = 50$ benchmark datasets). Rows represent the actual best algorithm; columns represent AMLA's predicted recommendation. Diagonal entries are correct predictions; colour intensity is proportional to count.

The diagonal entries confirm that AMLA correctly identifies the best algorithm in the majority of cases across all five classes. Random Forest is the most dominant class ($n = 159$ total instances) and achieves the highest per-class accuracy: 144 correct out of 159 (90.6%). Gradient Boosting achieves 28 correct out of 43 (65.1%), and SVM achieves 18 correct out of 29 (62.1%). The most frequent off-diagonal confusion occurs between Random Forest and Gradient Boosting (12 misclassifications), which is expected given the structural similarity of these two ensemble methods and their tendency to excel under similar dataset conditions—high sample-to-feature ratio, moderate class imbalance, and complex non-linear boundaries. Logistic Regression (57.1%) and KNN (40.0%) show lower per-class recall, attributable to their rarer occurrence in the meta-training labels. This analysis motivates a future extension incorporating cost-sensitive learning at the meta-level to improve recall for minority algorithm classes.

4) OpenML-to-Local Alignment: For the 10 locally validated datasets, the mean Spearman rank correlation between OpenML community rankings and AMLA's local implementation rankings was $r_s = 0.81$ (SD = 0.09), indicating strong consistency. Three datasets exhibited alignment below $r_s = 0.7$; for these, local results were used as the ground truth label.

5) SHAP Feature Importance at the Meta-Level: Across all 50 datasets, aggregated SHAP importance values identified the top-5 most globally predictive DNA features: (1) lm_1nn_acc, (2) lm_variance, (3) corr_high_ratio, (4) $\bar{I}$, and (5) dim_ratio. These findings confirm that the relative performance of a 1-NN probe model and the spread of landmark accuracies are the most powerful discriminators—a novel, interpretable insight enabled by AMLA's SHAP integration.

6) Feature Augmentation Advisor Evaluation: The Feature Augmentation Advisor was applied to all 50 benchmark datasets. The mean health score was 0.61 (SD = 0.14), indicating moderate but non-trivial data quality issues even after basic preprocessing. Class imbalance and high-skewness features were the most frequently triggered warnings (76% and 68% of datasets, respectively). Target leakage warnings were triggered in 4% of datasets, all confirmed as genuine leakage on manual inspection.

VIII. DISCUSSION

A. Original Contributions

AMLA makes six original contributions to the meta-learning literature:

1) **Dataset DNA Fingerprinting.** A principled, 60-dimensional multi-layer characterisation scheme com-

bining five feature groups into a unified fixed-length vector. No existing open-source tool implements all five groups.

2) **Gap-Feature Landmarking.** Novel landmark gap fea- tures ($\Delta_{lin/nlin}$, $\Delta_{nn/tree}$) quantify the relative advantage of different learning paradigms and carry high SHAP importance.
3) **SHAP-Augmented Meta-Learner.** Per-prediction SHAP explanations ground algorithm recommendations in interpretable dataset structure—the first such system in the meta-learning literature.
4) **Predictive Feature Gap Analysis.** The Feature Augmentation Advisor provides proactive, structured dataset quality assessment and feature engineering guidance entirely absent from existing AutoML systems.
5) **Hybrid Meta-KB Construction.** OpenML-seeded, locally-validated construction pipeline with a Spearman alignment score as a quality metric.
6) **Closed-Loop Self-Improvement.** An operational feedback loop that continuously updates the Meta-KB with real-world experimental observations without manual curation.

B. Comparison to Existing Systems

Table IV positions AMLA against representative AutoML and meta-learning systems across key capability dimensions.

TABLE IV
CAPABILITY COMPARISON WITH REPRESENTATIVE SYSTEMS

System	Explainable	Feat. Advice	Self-Impr.	API	Fast
Auto-sklearn [6]	No	No	No	No	No
TPOT [8]	No	No	No	No	No
H2O AutoML [9]	No	No	No	Yes	No
AutoGluon [10]	No	No	No	No	No
AMLA (Ours)	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

"Fast" denotes sub-second algorithm recommendation without any model training. All existing systems require non-trivial computation; AMLA returns ranked recommendations in under one second.

C. Future Work

Planned extensions include: (i) regression and clustering support; (ii) integration of neural tabular models into the candidate portfolio; (iii) active learning for Meta-KB growth, prioritising datasets that maximally reduce meta-learner uncertainty; (iv) a dataset similarity retrieval interface enabling case-based reasoning; and (v) automated feature generation (polynomial features, date decomposition, embeddings for high-cardinality categoricals).

IX. CONCLUSION

This paper presented AMLA, the Adaptive Meta-Learning Architecture—a comprehensive, interpretable, and self-improving framework for automated ML algorithm selection. By introducing the Dataset DNA fingerprinting scheme, a SHAP-augmented predictive meta-learner, and a first-of- its-kind Feature Augmentation Advisor, AMLA advances the state of the art in meta-learning along three axes: characteri- sation richness, recommendation interpretability, and action- able dataset quality guidance. Evaluated on 50 benchmark classification datasets, AMLA achieves a Precision@1 of 72%, representing a 21-percentage-point improvement over the strongest baseline at $p < 0.001$. Deployed as an interactive full-stack web application, AMLA democratises expert-level algorithm selection for practitioners without specialist ML knowledge. The self-improvement mechanism ensures that recommendations grow progressively more accurate as real-world experimental evidence accumulates.

REFERENCES

[1] J. R. Rice, "The algorithm selection problem," *Advances in Computers*, vol. 15, pp. 65–118, 1976.
[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
[3] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.
[4] P. Brazdil, C. Soares, and J. P. Costa, "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.
[5] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier, "Meta-learning by landmarking various learning algorithms," in *Proc. 17th ICML*, Stanford, CA, 2000, pp. 743–750.
[6] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
[7] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-sklearn 2.0: Hands-free AutoML via meta-learning," *J. Machine Learning Research*, vol. 23, no. 261, pp. 1–61, 2022.
[8] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proc. GECCO 2016*, Denver, CO, pp. 485–492, 2016.
[9] E. LeDell and S. Poirier, "H2O AutoML: Scalable automatic machine learning," in *Proc. 7th ICML Workshop on AutoML*, 2020.
[10] N. Erickson *et al.*, "AutoGluon-tabular: Robust and accurate AutoML for structured data," *arXiv:2003.06505*, 2020.
[11] P. Gijsbers *et al.*, "AMLB: An AutoML benchmark," *J. Machine Learning Research*, vol. 25, no. 101, pp. 1–65, 2024.
[12] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.
[13] B. Bischl *et al.*, "OpenML benchmarking suites," in *Proc. NeurIPS Datasets & Benchmarks Track*, 2021.
[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[15] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, San Francisco, CA, pp. 785–794, 2016.
[17] C. Molnar, *Interpretable Machine Learning*, 2nd ed., christophm.github.io/interpretable-ml-book/, 2022.
[18] C. Giraud-Carrier and F. Provost, "Toward a justification of meta-learning: Is the no free lunch theorem a show-stopper?" in *Proc. ICML Workshop on Meta-Learning*, 2005.
[19] I. Khan, X. Zhang, M. Rehman, and R. Ali, "A literature survey and empirical study of meta-learning for classifier selection," *IEEE Access*, vol. 8, pp. 10262–10281, 2020.
[20] M. Garouani, J. Mothe, A. Barhrhouj, and J. Aligon, "Investigating the duality of interpretability and explainability in machine learning," in *Proc. IEEE 36th ICTAI*, 2024, pp. 861–867.
[21] M. Garouani *et al.*, "An experimental survey and perspective view on meta-learning for automated algorithm selection and parametrization,"

arXiv:2504.06207, 2025.

[22] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why tree-based models still outperform deep learning on tabular data," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[23] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, "Two-stage transfer surrogate model for automatic hyperparameter optimization," in *Proc. ECML-PKDD*, Riva del Garda, Italy, pp. 499–515, 2016.

[24] Todupunuri, A. (2024). Exploring the use of generative AI in creating deepfake content and the risks it poses to data integrity, digital identities, and security systems. Available at SSRN 5014688.

[25] Babburi, S. Lightweight Distributed Provenance Framework for Edge and IoT Data Systems.

[26] Gaddam, S. Integrating Analytics into the Development Process: Bridging the Gap between Data Insights and Design Execution.

[27] Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. https://doi.org/10.1201/9781003684657-63

[28] Poojari, R. INTELLIGENT SYSTEMS+B108 AND APPLICATIONS IN ENGINEERING.

[29] Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.

[30] Poojari, R. Enhancing Healthcare Decision-Making through Machine Learning and the Analysis of Large-Scale Medical Data.

[31] Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.

[32] Purmani, S. S. R. (2025). Streamlining IT operations and service management with agile frameworks. European Journal of Advances in Engineering and Technology, 12(4), 76–81.

[33] Purmani, S. S. R. (2024). Aligning IT investment decisions with overall business strategy from an enterprise program management perspective, focusing on the integration of IT leadership in strategic decision-making processes. International Journal of Communication Networks and Information Security, 16(5), 1213–1219

[34] Cyril, H. P., & Kumara, S. (2026, February). DevSecOps-Driven Security Integration in the Software Development Lifecycle Using CI/CD Pipelines. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.

[35] Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660

[36] Kotte, G. (2025). Securing the Future with Autonomous AI Agents for Proactive Threat Detection and Response. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283830

[37] Ranjbareslamloo, S., Dzukeya, G. A., Muhit, M. M. I., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. Manufacturing Letters, 44, 915–927. https://doi.org/10.1016/j.mfglet.2025.915927

[38] Viswanathan, V. (2024). Embedding Ethical Principles into Generative AI Workflows for Project Teams.

[39] Viswanathan, V. (2024). Pioneering Ethical AI Integration in Enterprise Workflows: A Framework for Scalable Team Governance. Available at SSRN 5375619.

[40] Mudusu, S. K. (2026, April 15). The secure intelligence framework: Architecting AI systems for a data-driven world. CIO (Foundry Expert Contributor Network).

[41] Mudusu, S. K. (2025, December 22). Cognitive data architecture: Designing self-optimizing frameworks for scalable AI systems. CIO (Foundry Expert Contributor Network).

[42] Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[43] Gajula, S. (2025, December). Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity. In 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD) (pp. 1-6). IEEE.

[44] Maturi, S. Y. (2025). Vulnerabilities in the 802.11 Wireless Client Selection Mechanis.

[45] Maturi, S. Y. Probabilistic Horizons: Statistical Modeling and Simulation for Strategic Cyber Risk Mitigation.

[46] Chowdhury, A. K., Muhit, M. M. I., & Islam, M. M. (2023). A practical review to the marine maintenance practice in Bangladesh and a proposed way forward to an efficient, long-term and cost-effective solution. In Proceedings of the 13th International Conference on Marine Technology (MARTEC 2022). https://doi.org/10.2139/ssrn.4445071

[47] Manoharan, D. (2025). Healthcare EDI Transaction Lifecycles Embedded with a Multi-Layer Verification Framework to Ensure Referential Integrity.

[48] Manoharan, D. (2026). AI-Driven Anomaly Detection Models for Preventing Claims Denials and Revenue Leakage in Healthcare. Available at SSRN 6385759.

[49] Ravishankara, M. (2026, February). CircuChain: Disentangling Competence and Compliance in LLM Circuit Analysis. In SoutheastCon 2026 (pp. 1-7). IEEE.

[50] Doragacharla, V. R. (2023). Comprehensive Benchmarking Analysis of Auto Scaling Approaches in Cloud Native Streaming Pipelines During Flash Sales and Holiday Traffic Peaks. Available at SSRN 6566479.

[51] P. Venkata Ramana. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. International Journal of Innovative Engineering and Management Research (IJIEMR).

[52] Kumar Adabala, P. (2021). Optimizing ERP Modernization: A Smart Data Migration Framework Approach. International Journal of Enhanced Research in Science, Technology &Amp; Engineering, 10(07), 61–72. https://doi.org/10.55948/ijerste.2021.0708

[53] Pavan Kumar Adabala. (2026). Smart Retail Fuel Systems: IoT-Enabled Solutions for Loss Prevention and Environmental Safety. Computer Fraud and Security, 868–875. https://doi.org/10.52710/cfs.995

[54] Kavuri, S. (2025). Critical Review of Software Testing Problems in the Current Decade. International Journal on Science and Technology, 16(2). https://doi.org/10.71097/ijsat.v16.i2.9469

[55] Srikanth Kavuri. (2023). Machine Learning Approaches for Security Vulnerability Detection in Software Testing. Computer Fraud and Security. https://doi.org/10.52710/cfs.837

[56] Venkata Pavan Kumar Gummadi. (2023). MuleSoft Batch Processing: High-Volume Streaming Architecture. Computer Fraud and Security, 50–57. https://doi.org/10.52710/cfs.886

[57] Venkata Pavan Kumar Gummadi. (2026). Infrastructure Optimization Techniques for Enterprise Integration Platforms: A Comprehensive Analysis. Computer Fraud and Security, 37–44. https://doi.org/10.52710/cfs.875

[58] Venkata Pavan Kumar Gummadi. (2024). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4), 76–85. https://doi.org/10.52783/jes.9329

[59] Venkata Pavan Kumar Gummadi. (2025). MuleSoft's Role in Advancing Sustainable Digital Infrastructure: An Enterprise Integration Perspective. Journal of Information Systems Engineering and Management, 10(62s), 1313–1321. https://doi.org/10.52783/jisem.v10i62s.13783

[60] Gummadi, V. P. K. (Ed.). (2025). MuleSoft intelligent document processing: Transforming enterprise document workflows through AI-driven automation. Journal of Computational Analysis and Applications, 34(12). https://doi.org/10.48047/jocaaa.2025.34.12.16

[61] Subramanian, V. K., Bhambri, S., & Gajula, S. (2026). Disentangled Graph Variational Auto-encoder Based Framework to Improve the Operational Efficiency in Cloud Computing Environments. Computer Vision and Robotics, 396–407. https://doi.org/10.1007/978-3-032-14044-9_32

[62] Gajula, S., & Margam, M. (2026). A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–5. https://doi.org/10.1109/icaic67076.2026.11395704

[63] Gajula, S. (2025). AI-Powered Forecasting Models, Optimizing Working Capital, Supply Chain Financing. 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM), 1–6. https://doi.org/10.1109/rcsm67767.2025.11507813