Exploratory Landscape Analysis as Meta-Features for Hyperparameter Tuning Recommendation: A Study on SVMs

Estéfane de Lima dos Santos[1], **Guilherme Moreira Lima Furlaneto**[1]
Márcio P. Basgalupp[3], **Jadson Castro Gertudes**[2], **Rafael Gomes Mantovani**[1]

[1]Federal University of Technology – Paraná (UTFPR), Apucarana, Paraná, Brazil

[2]Federal University of Ouro Preto (UFOP), Ouro Preto, Minas Gerais, Brazil

[3]Federal University of São Paulo (UNIFESP), São José dos Campos, São Paulo, Brazil

{estefane, guilhermefurlaneto}@alunos.utfpr.edu.br,
basgalupp@unifesp.br, jadson.castro@ufop.edu.br,
rafaelmantovani@utfpr.edu.br

Abstract. *A fundamental challenge in applying classification algorithms is selecting appropriate hyperparameter settings for a given dataset. Recently, AutoML has emerged as a paradigm for addressing this burden, with some applications based on Metalearning (MtL). By observing which algorithms and settings performed well across datasets, MtL can recommend promising configurations without extensive search. In this work, we investigate whether Exploratory Landscape Analysis (ELA) descriptors are useful as meta-features in a recommender system that predicts whether tuning Support Vector Machine (SVM) hyperparameters is worthwhile. We extracted 50 ELA features from 2D SVM optimization surfaces on* 169 *classification datasets. Experiments considered three meta-datasets: ELA-only, a state-of-the-art baseline (pyMFE), and their combination. These meta-datasets were evaluated using eight different algorithms and preprocessing sub-tasks. Experimental results showed that the combined meta-dataset achieved the best AUC of* 0.825 *when combined with the XGBoost algorithm. The Friedman-Nemenyi test confirmed it as significantly superior to the baseline. A SHAP analysis of the best model showed that* 6 *of the* 10 *most important features are ELA descriptors, confirming that landscape information provides a complementary and discriminative signal for the tune-versus-default decision.*

1. Introduction

Machine Learning (ML) has become pervasive in modern society, influencing domains as diverse as healthcare, finance, education, and industry [Marsland 2009]. Both domain experts and non-specialists increasingly rely on ML-based systems to support or automate decision-making, making the accessibility and reliability of these methods a pressing concern. Different tasks can be modeled, but classification stands out as one of the most widely studied and applied, encompassing problems such as spam detection, medical diagnosis, and fault identification in industrial processes.

A fundamental challenge in applying classification algorithms is selecting and configuring appropriate models for a given dataset. The literature offers a rich landscape of classification algorithms, ranging from Decision Trees (DTs) to kernel-based

methods, such as Support Vector Machines (SVMs) [Cortes and Vapnik 1995], each with distinct inductive biases and sensitivities. Notably, these algorithms are considerably sensitive to the choice of their hyperparameters, where suboptimal hyperparameter values can substantially degrade predictive performance [Mantovani et al. 2024]. Identifying suitable hyperparameter settings traditionally involves trial-and-error, exhaustive grid/random search, or even optimization, which are time-consuming and require considerable expertise.

At the same time, Automated Machine Learning (AutoML) [Hutter et al. 2019] has emerged as a paradigm to address this burden, combining techniques such as algorithm selection, hyperparameter optimization, and neural architecture search within unified frameworks. At the core of many AutoML systems lies Meta-learning (MtL) [Brazdil et al. 2022], which leverages experience from prior learning tasks to inform decisions on new, unseen tasks, effectively allowing the system to "*learn how to learn*". By observing which algorithms and configurations performed well across a portfolio of datasets, MtL systems can recommend promising configurations without exhaustive search, democratizing access to high-quality ML solutions for users of all expertise levels.

However, for MtL to be effective, tasks must be characterized in a way that captures properties relevant to algorithm behavior. This is typically achieved through meta-features: measurable descriptors extracted from a dataset that summarize its statistical, structural, or geometric properties. Traditional meta-features include dataset-level statistics such as the number of instances, feature dimensionality, class imbalance ratio, and measures of feature correlation. While broadly used, these descriptors may not fully capture the complexity of the underlying data distribution as it relates to classifier performance.

A complementary family of meta-features originates from Exploratory Landscape Analysis (ELA) [Mersmann et al. 2011], a collection of techniques originally developed for continuous black-box optimization to characterize the fitness landscape of optimization problems. They quantify properties such as modality, ruggedness, and separability of an objective function's landscape. When adapted for classification settings, these descriptors can capture aspects of the data distribution that conventional meta-features do not encode. Thus, in this work, we investigate the potential of ELA-derived meta-features in a MtL recommender system, specifically one tasked with predicting whether hyperparameter tuning of SVMs is likely to yield meaningful gains over their default hyperparameter settings.

The remainder of this paper is organized as follows. Section 2 presents some of the necessary concepts about MtL and ELA, describing related work. The experimental methodology is presented in Section 3. The results are discussed in Section 4, while the conclusions and final considerations of the study are presented in Section 5.

2. Background

2.1. Meta-learning

Meta-learning (MtL) emerged as a distinct area of study within ML, motivated by the observation that learning systems can improve their efficiency through accumulated experience. It investigates how the learning mechanism interacts with the context in which

it is applied, and how this interaction can be exploited to make the learning process itself more effective [Vanschoren 2019]. More precisely, MtL can be defined as the study of principled methods that exploit *meta-knowledge* to obtain efficient models and solutions by adapting ML and data mining processes [Brazdil et al. 2022]. The meta-knowledge in question typically comprises information derived from previous learning tasks, such as task descriptions (features), pipelines, or resulting models.

The conceptual foundation of MtL can be traced back to the *algorithm selection problem* formalized by [Rice 1976]. In this formulation, given a problem $x \in \mathcal{P}$ drawn from a problem space $\mathcal{P}$, and an algorithm $a \in \mathcal{A}$ from a set of candidate algorithms $\mathcal{A}$ applicable to problems in $\mathcal{P}$, the goal is to determine a selection mapping $S : \mathcal{P} \rightarrow \mathcal{A}$. Letting $p(a, x)$ denote a performance measure of algorithm a applied to problem x, the objective is to find a mapping that associates each problem with the algorithm that maximizes performance:

$$\|p(S(x), x)\| \geq \|p(a, x)\| \quad \forall a \in \mathcal{A}. \tag{1}$$

A central insight of Rice's framework is the role of *problem features*, where rather than operating on the raw problem x, the mapping is defined over a feature space $\mathcal{F}$ that characterizes each problem through measurable descriptors. This idea is the direct precursor of modern *meta-features*, the dataset descriptors that allow learning experience to be transferred across tasks. In a typical MtL workflow, a *meta-dataset* is assembled by extracting meta-features from a collection of base-level datasets and recording the performance attained by one or more algorithms on each of them. A meta-model is then trained on this meta-dataset to map meta-features to the target of interest, so that recommendations can be issued for new, unseen datasets without exhaustive search [Vanschoren 2019].

In this work, the MtL target is a binary decision: given a dataset, predict whether tuning the hyperparameters of SVMs is likely to yield meaningful gains over their default configurations. The quality of any such recommendation depends critically on how well the chosen meta-features capture the properties of the data relevant to classifier behavior [Mantovani et al. 2019], which motivates the search for richer dataset characterizations, as discussed next.

2.2. Exploratory Landscape Analysis

Exploratory Landscape Analysis (ELA) is a mean to numerically characterize the *fitness landscape* of an optimization problem [Mersmann et al. 2011]. The underlying motivation is closely related to the algorithm selection problem discussed above, since the performance of an optimization algorithm is strongly tied to the structure of the landscape it must traverse, so characterizing that structure provides a principled basis for understanding and predicting algorithm behavior.

In their original formulation, the authors defined a set of numerical measures organized into several groups of cheaply computable *low-level* features, covering aspects such as the distribution of objective values, level-set properties, meta-model fit, local search behavior, curvature, and convexity. These descriptors together approximate higher-level, more intuitive properties of the landscape, such as multimodality, global structure, and separability [Mersmann et al. 2011] when the objective function is unknown.

ELA has grown into a broad family of landscape descriptors and has become a standard tool for automated algorithm selection and configuration in the optimization community. Comprehensive surveys of the field document how landscape analysis has expanded well beyond the original feature set, discussing the variety of measures now available, the sampling strategies used to compute them, and their applications across optimization problems [Malan 2021]. While these descriptors were conceived to characterize optimization landscapes, the properties they quantify are general geometric notions. When a classification dataset is reframed so that its instances and target values define a surface amenable to landscape analysis, ELA features can in principle describe aspects of the data distribution that conventional, statistically oriented meta-features do not capture. This possibility is precisely what our work sets out to investigate.

2.3. Related Work

The use of MtL to support algorithm selection and hyperparameter recommendation has a substantial track record, particularly for classifiers that are highly sensitive to their configuration. An early example combined MtL with search techniques to recommend SVM hyperparameter values, using MtL to suggest promising configurations as starting points for a subsequent search [Gomes et al. 2012]. Rather than recommending specific values, a related line of work handled the problem as a binary decision of whether hyperparameter tuning is worthwhile at all: the *tune-vs-default* question. This was first explored for SVMs [Ridd and Giraud-Carrier 2014] and later consolidated into a full MtL recommender system that predicts, for a given dataset, when tuning improves SVM classifiers [Mantovani et al. 2019].

Other studies have targeted different aspects of the configuration problem. The hyperparameter *initialization* of classification algorithms has been addressed via MtL, which employs dynamic time warping and revisits PCA-based meta-features to characterize datasets [Horváth et al. 2023]. Hyperparameter recommendation has also been studied for specific modern learners, such as selecting XGB configurations from similar datasets identified through their meta-features [Lima Marinho et al. 2024].

Underpinning all of these systems is the question of how datasets are characterized. A systematic treatment of the descriptors used for this purpose organizes meta-features into well-established groups and provides a unifying framework for their definition and extraction [Rivolli et al. 2022]. What stands out across this body of work is that dataset characterization relies almost exclusively on traditional families of meta-features: simple, statistical, information-theoretic, model-based, and landmarking, among others. To the best of our knowledge, ELA-derived descriptors, despite their established value in characterizing optimization landscapes, have not been widely employed as meta-features in MtL recommender systems for classification.

3. Experimental Methodology

An overview of the experimental flow, including sub-steps, is shown in Figure 1. The following subsections will explain all the sub-tasks in detail.

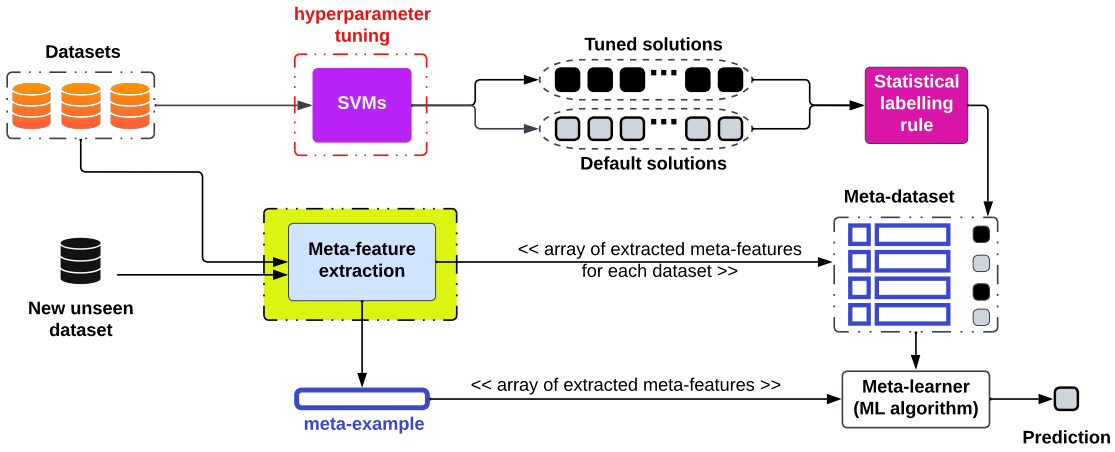


Figure 1. Experimental methodology adopted to predict when to tune SVMs via Meta-learning. Adapted from [Mantovani et al. 2019].

3.1. Meta-Knowledge: Datasets and Tuning results

We explored the meta-knowledge provided by [Mantovani et al. 2019]. The authors curated a collection of 169 classification datasets from OpenML[1]. These datasets were selected following some criteria: a) the number of features does not exceed $1,500$; b) number of instances between 100 and $50,000$; c) must not be a modified version of the original classification problem; d) must not be an adaptation of a regression dataset; e) all the classes must have at least 10 examples, enabling the use of stratified 10-fold CV.

We also processed the datasets, removing constant and identifier features; converting logical features into values $\in \{0, 1\}$; missing values were imputed by the median for numerical features, and a new category for categorical ones; categorical features were also converted into the 1-N encoding; and finally, all features were normalized with $\mu = 0$ and $\sigma = 1$.

For each dataset, two SVM hyperparameters were adjusted: the cost constant (C) and the width of the RBF kernel (γ), assessed using the Balanced Accuracy (BAC) evaluation measure. Table 1 presents the 2D continuous hyperspace explored by the tuning. Each dataset had SVMs tuned 30 times using a Random Search (RS) [Bergstra and Bengio 2012]. The tuning was performed using a nested-CV resampling technique: an inner CV with 3 folds and an outer loop with 10. Each single tuning execution consumed a budget of 300 evaluations. So, tuning a single dataset evaluated a total of $10 \times 300 \times 30 = 30,000$ hyperparameter settings, and this optimization path was later used to extract meta-features. The datasets and their final optimization tuning paths compose the "meta-knowledge" explored in the experiments.

3.2. ELA Meta-feature Extraction

Figure 2 highlights the meta-feature extraction presented in Figure 1. For each dataset, we collected the optimization paths for $30,000$ hyperparameter settings and extracted ELA meta-features from the corresponding 2D continuous landscapes. Each tuning job can be modeled as a function: it receives a dataset and an algorithm (SVM), and returns the most

[1]https://www.openml.org/

Table 1. SVM hyperspace that generated optimization paths provided by meta-knowledge [Mantovani et al. 2019].

Symbol	Hyperparameter	Type	Range/Options	Scale	Default	Tuned
k	kernel	categorical	{RBF}	-	RBF	x
C	cost	real	$[2^{-15}, 2^{15}]$	log	1	✓
γ	width of the kernel	real	$[2^{-15}, 2^{15}]$	log	$1/N$	✓

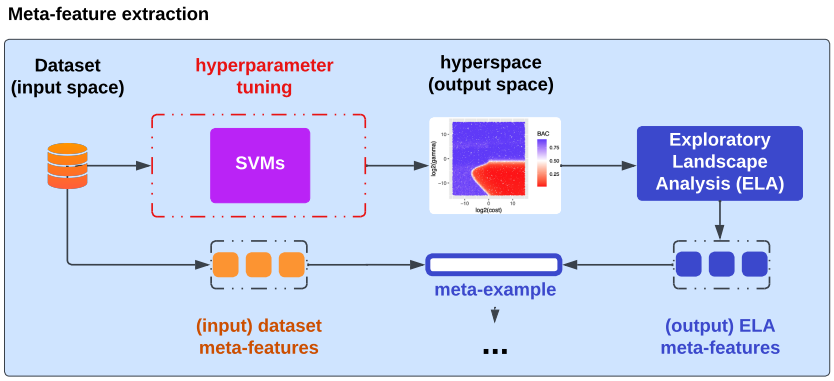


Figure 2. Meta-feature extraction task. For each dataset, meta-features are extracted from the input (datasets) and output (hyperparameters) spaces.

suitable hyperparameter setting, after evaluating a set of candidate solutions. The dataset lies in the "*input space*", while the set of candidate solutions and their corresponding BAC values are considered the "*output space*". Most related studies exclusively extract meta-features from datasets, and the optimization surfaces remain uninvestigated.

3.3. Meta-datasets and learning tasks

We created a binary classification MtL task to predict whether SVMs should be tuned using the optimization results from meta-knowledge and ELA meta-features. The meta-target values were defined by the "meta-label rule" presented in [Mantovani et al. 2019]. The Wilcoxon paired test compares the solutions obtained by the RS technique and the default hyperparameter settings. A meta-example receives the label "tuning" if the optimized performance is higher and statistically significant (i.e., tuning provides better results); otherwise, it receives the label "default", indicating that default hyperparameters are sufficient. We executed three meta-tasks: (i) our ELA meta-dataset, with all the 50 meta-features; (ii) the original meta-dataset (baseline) with a collection of 80 graph and network-based meta-features implemented in the pyMFE Python package [Alcobaça et al. 2020]; and (iii) a combined meta-dataset, with the union of all available meta-features (ELA + pyMFE). All meta-tasks have the same number of examples (169) and class distribution (122 tuning $\times$ 47 defaults), differing in the set and number of meta-features: i) 50; ii) 80, and iii) 130, respectively.

3.4. Meta-learning Experiments

Prior to each MtL experiment, we run two preprocessing steps: i) removing constant meta-features (zero variance); and ii) pairs of meta-features with absolute Pearson correlation exceeding a threshold τ are filtered, retaining one representative from each correlated

Table 2. ELA feature groups extracted from 2D SVM continuous hyperspace.

#	Category /Acronym	Description	Cont.
1	Meta-Model (`ela_meta`)	Fits linear and quadratic surrogate models to the landscape and reports adjusted R^2, intercept, and coefficient statistics. High linear R^2 indicates a globally smooth and predictable landscape.	5
2	Dispersion (`ela_disp`)	Compares mean pairwise Euclidean distances of good configurations (top 10%) against the remaining ones. Ratio < 1 indicates that good solutions cluster together in the search space.	4
3	Information Content (`ela_ic`)	Shannon entropy over consecutive symbol pairs along the nearest-neighbor (NN) walk, ordered by Euclidean distance. High entropy indicates a rugged landscape; low entropy indicates a smooth or plateau-like structure.	5
4	Cell Mapping (`ela_cm`)	Partitions the 2D continuous space into a uniform 5×5 grid and summarizes fitness statistics per cell. High standard deviation of cell means indicates a multimodal landscape.	4
5	Landscape Correlation (`ela_lc`)	Pearson correlation between fitness and Euclidean distance to the sample centroid, and spatial autocorrelation via the nearest neighbor. Also includes distributional statistics of fitness values (standard deviation, skewness, kurtosis).	5
6	Nearest Better Clustering (`nbc`)	For each configuration, compares the Euclidean distance to its nearest neighbor against the distance to its nearest *better* neighbor. Ratio ≈ 1 indicates a funnel structure; high ratio suggests multimodality or deception.	6
7	Fitness-Distance Corr. (`fdc`)	Pearson correlation between fitness and Euclidean distance to the best observed configuration. $r < -0.15$: big-valley structure (easy); $r > 0.15$: deceptive landscape.	5
8	Gradient Homogeneity (`gh`)	Local gradients estimated by weighted least squares (WLS) over k nearest Euclidean neighbors. Directional consistency is measured via pairwise cosine similarity of gradient vectors. $\overline{gh} \to 1$: clean funnel; $\overline{gh} \to 0$: chaotic gradients.	6
9	GP Length Scale (`ls`)	Length scales from an anisotropic Matérn-5/2 GP kernel fitted to the landscape. Large scale: smooth dimension; small scale: rugged dimension. The ratio between dimensions captures anisotropy.	6
10	Sample Statistics (`sample`)	Basic statistics of the sampled configurations: total sample size, and minimum, maximum, and mean fitness value.	4
	Total		**50**

group. We investigated four values of τ: $\{0.80, 0.85, 0.90, 0.95\}$. This procedure is applied within each experimental combination to avoid data leakage across the evaluation loop.

Also, eight supervised learning algorithms were evaluated as meta-learners: i) Support Vector Machine (SVM) with linear kernel (SVM-Lin); ii) Support Vector Machine (SVM) with RBF kernel (SVM-RBF); iii) Random Forest (RF); iv) K-Nearest Neighbors (kNN); v) Decision Tree (DT); vi) Naïve Bayes (NB); vii) Logistic Regression (LR); and viii) eXtreme Gradient Boosting (XGB). All algorithms are wrapped in a pipeline that applies standard score normalization ($\mu = 0$, $\sigma = 1$) before model fitting, ensuring that scale-sensitive learners are not affected by the differing ranges of meta-features. All the algorithms were implemented in Python and used their default hyperparameters.

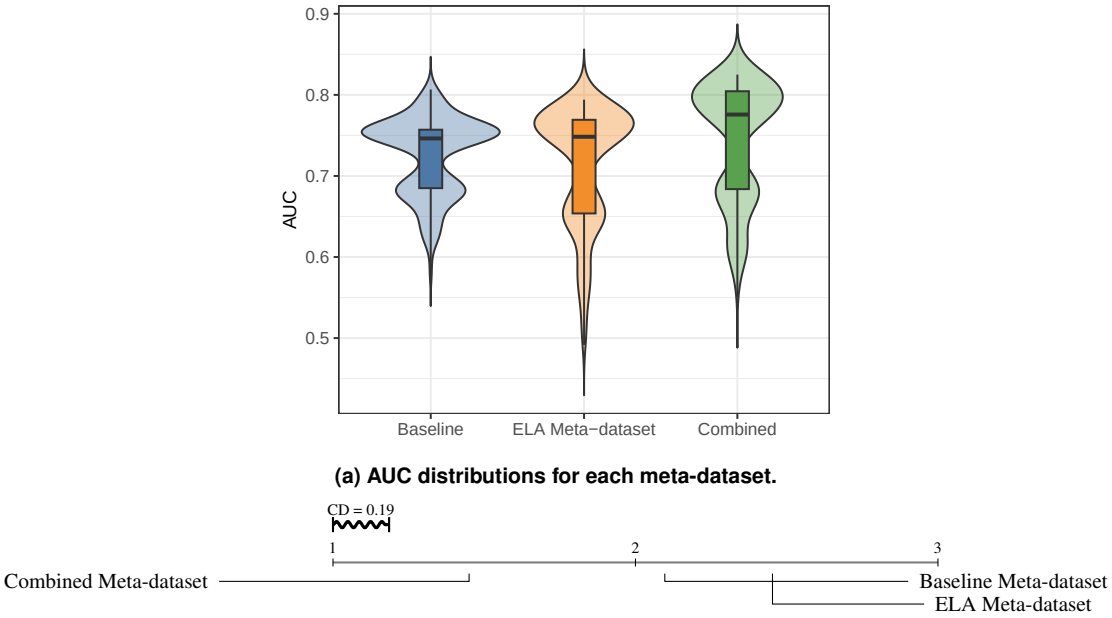
Model performance is estimated via Leave-One-Out Cross-Validation (LOOCV). In each iteration, one of the 169 examples serves as the sole test instance in a separate fold, and the remaining train the meta-models. Class probability values are collected across all folds and used to compute three performance metrics: weighted F_1-score, BAC, and Area Under the ROC Curve (AUC). The entire experiment is repeated across 10 random seeds to account for the stochasticity of probabilistic algorithms, and the results are aggregated by averaging metrics. The full experiment comprises: 3 meta-datasets $\times 4$ correlation thresholds $\times 8$ meta-learners $\times 10$ seeds = 960 LOOCV jobs. All the codes and meta-datasets used and generated in this study are publicly available[2].

4. Results

The overall results obtained by all meta-datasets are depicted in Figure 3a. We aggregated results generated with different algorithms, seeds, and correlation thresholds. The original meta-dataset (baseline) has AUC values mostly in the range $\{0.7, 0.75\}$, with a median near 0.72. The ELA meta-dataset exhibits similar behavior but with a broader distribution. Some algorithms benefit from these meta-features, but others, such as DT and NB, perform worse, leading to greater dispersion. The most interesting result is the superior performance of the combined dataset: a higher median (≈ 0.78) and the highest overall AUC (0.825). This indicates that combining network and graph with ELA meta-features generally yields a more consistently discriminative meta-dataset, not just for some algorithms.

We applied the Friedman test with $\alpha = 0.05$ to assess statistical significance, treating all combinations of algorithms and correlation thresholds as experimental blocks. The Nemenyi post hoc test was subsequently applied to identify which pairs of meta-feature sets differed significantly. The results are presented as a Critical Difference (CD) diagram, in which meta-feature sets are ranked from best to worst by their average rank and connected by a horizontal bar when no statistically significant difference exists between them. Figure 3b shows that all pairwise comparisons are statistically significant: the Combined meta-dataset achieves the best average rank (1.45), followed by the baseline (2.10) and ELA Meta-datasets (2.45). Notably, the ELA meta-dataset alone performs significantly worse than the baseline, indicating that these meta-features only provide a statistically meaningful gain when combined with the original network-based features.

[2] `https://github.com/ic2d/ela_tuning`



(a) AUC distributions for each meta-dataset.

(b) CD diagram of the Friedman-Nemenyi test with $\alpha = 0.05$.

Figure 3. AUC distribution and statistical comparison across meta-datasets, aggregated over 10 executions and different correlation thresholds (τ).

4.1. Algorithms' Performance

Figure 4 shows the individual performance of each algorithm for each meta-dataset and correlation coefficient values. The baseline meta-dataset, composed solely of network and graph-based features, achieves its best result with SVM-RBF at $\tau = 0.80$ (AUC $= 0.789$). Incorporating just ELA meta-features yields a modest improvement, with XGB reaching AUC $= 0.794$ at $\tau = 0.95$, a small improvement over the baseline. The best results are observed for the Combined meta-dataset using XGB with $\tau = 0.85$, which achieves an AUC of 0.825 and surpasses both meta-datasets.

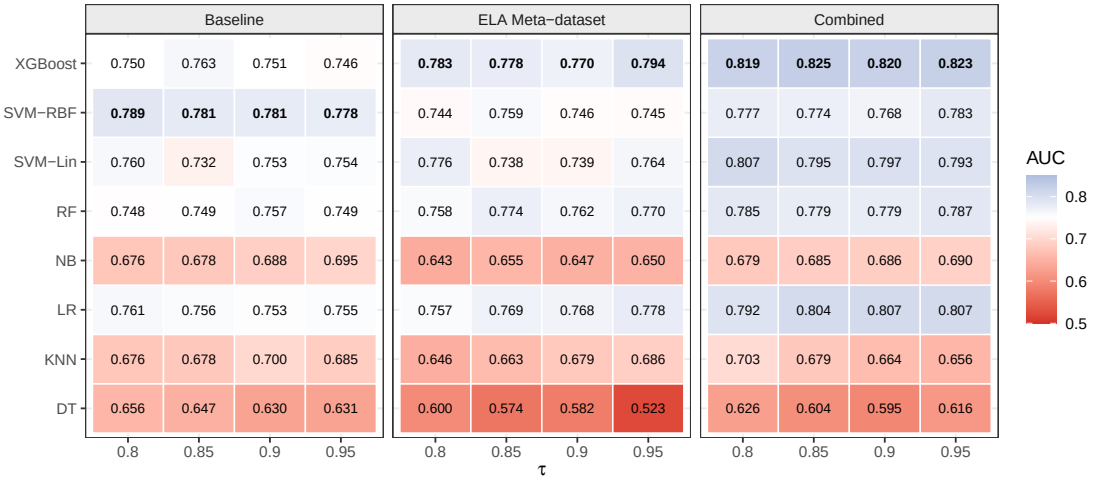


Figure 4. Mean AUC values of each meta-learner across all experiments. Bold values indicate the best-performing algorithm for each meta-dataset.

The heatmap also shows heterogeneous behavior across algorithms when incor-

porating ELA meta-features. XGB and LR benefit the most, achieving the best results on the Combined meta-dataset, with AUC values of 0.825 and 0.807, respectively. RF also improves consistently across the meta-feature sets. In contrast, SVMs-RBF, despite being the best algorithm on the Baseline, shows a slight degradation when adding ELA features, never surpassing its original performance. DT remains the weakest algorithm regardless of the feature set, while kNN and NB show only marginal and inconsistent gains. The correlation threshold τ has a limited overall impact on predictive performance, with variations rarely exceeding 0.01 AUC within the same algorithm and meta-dataset configuration.

4.2. SHAP interpretability for the best meta-model

To support the interpretability of the XGB results on the Combined meta-dataset, we applied the SHapley Additive exPlanations (SHAP) [Lundberg et al. 2018] technique, a eXplainable Artificial Intelligence (XAI) framework that quantifies the contribution of each meta-feature to individual model predictions. Figure 5 presents two complementary views: the beeswarm plot (left) displays the SHAP value of each instance for the top-10 ranked meta-features. In the beeswarm plot, each point represents one meta-instance, where the horizontal position encodes the contribution of that meta-feature to the model output for that specific instance. The relationship with the target classes is therefore indirect: blue points shifted to the right (high feature value, positive SHAP) indicate that the model was pushed towards the *Tuning* class, while red points shifted to the left (low feature value, negative SHAP) indicate a push towards *Defaults*. When blue points consistently appear on the right and red points on the left, as observed for `ela_lc.y_skewness`, `gh.norm.sd`, and `landmarking.stump_sd`, the feature exhibits a monotonic positive relationship with the *Tuning* class, meaning that higher feature values systematically increase the model's tendency to predict that a dataset benefits from SVM hyperparameter tuning.



Figure 5. SHAP analysis best meta-model overall (XGB trained on the Combined meta-dataset ($\tau = 0.85$)). (Left) Beeswarm plot showing the impact of each feature on model predictions. (Right) Mean |SHAP| values for the top-10 most influential features.

The Figure 5(right) summarizes the mean absolute SHAP value for each meta-feature, indicating their overall importance. The most important feature is

`landmarking.stump_sd`, a landmarking meta-feature from pyMFE that captures the variability of a simple decision stump. It is followed by two ELA features: `ela_lc.y_skewness` and `gh.norm.sd`, which characterize the shape and gradient structure of the hyperparameter landscape. The predominance of ELA meta-features (6 of top-10) confirms that landscape information provides a complementary and discriminative signal even when combined with a rich set of network-based meta-features.

5. Conclusions

This study investigated whether ELA descriptors, extracted from 2D SVM continuous optimization landscapes, can be used as meta-features for a MtL recommender system for the tune-versus-default task. We extracted 50 ELA features and compared the generated meta-dataset against a state-of-the-art baseline (pyMFE) and the combination of the two. The results showed that ELA features used in isolation are not competitive, performing significantly worse than the traditional baseline. However, when both sets are combined, they yield a consistently more discriminative meta-dataset, achieving the best overall performance (AUC = 0.825 with XGB as the meta-learner).

The SHAP analysis reinforced this finding, highlighting that 6 of the top-10 most important meta-features in the best model overall were ELA descriptors. It indicates that landscape information encodes a complementary signal not captured by conventional meta-features. As future work, we intend to extend this investigation to other classification algorithms, explore additional landscape-based descriptors, and assess the generalizability of these findings across larger, more diverse datasets.

A limitation of this study is the cost of obtaining the ELA descriptors, which are computed from dense SVM surfaces ($30,000$ evaluations per dataset). Since our main objective was to establish whether landscape information signals the tune-versus-default decision—a question best tested by a high-fidelity characterization —the results should be read as an upper bound on the predictive value of ELA descriptors, not as a deployable recommender. Future work will compute the same descriptors from small fixed-size designs, measuring how accuracy degrades as the sampling budget shrinks and yielding an explicit cost–benefit curve against the price of tuning.

Use of artificial intelligence: During the writing of this document, we used the *Claude Opus 4.8 LLM* for grammatical correction and text organization. We also take full responsibility for all content, including checking for plagiarism and correcting all text.

Acknowledgments

This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), grant 406417/2022-9, and the Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), grant APQ-02036-24.

References

Alcobaça, E., Siqueira, F., Rivolli, A., Garcia, L. P. F., Oliva, J. T., and de Carvalho, A. C. P. L. F. (2020). Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 21(111):1–5.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305.

Brazdil, P., van Rijn, J. N., Soares, C., and Vanschoren, J. (2022). *Metalearning: Applications to Automated Machine Learning and Data Mining*. Cognitive Technologies. Springer, Cham, 2 edition. Open Access.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

Gomes, T. A., Prudêncio, R. B., Soares, C., Rossi, A. L., and Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, 75(1):3–13.

Horváth, T., Mantovani, R. G., and de Carvalho, A. C. (2023). Hyper-parameter initialization of classification algorithms using dynamic time warping: A perspective on pca meta-features. *Applied Soft Computing*, 134:109969.

Hutter, F., Kotthoff, L., and Vanschoren, J., editors (2019). *Automated Machine Learning - Methods, Systems, Challenges*. Springer.

Lima Marinho, T., do Nascimento, D. C., and Pimentel, B. A. (2024). Optimization on selecting xgboost hyperparameters using meta-learning. *Expert Systems*, 41(9):e13611.

Lundberg, S. M., Erion, G. G., and Lee, S. (2018). Consistent individualized feature attribution for tree ensembles. *CoRR*, abs/1802.03888.

Malan, K. M. (2021). A survey of advances in landscape analysis for optimisation. *Algorithms*, 14(2).

Mantovani, R. G., Horváth, T., Rossi, A. L. D., Cerri, R., Barbon Junior, S., Vanschoren, J., and Carvalho, A. C. P. L. F. d. (2024). Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms. *Data Min. Knowl. Discov.*, 38(3):1364–1416.

Mantovani, R. G., Rossi, A. L., Alcobaça, E., Vanschoren, J., and de Carvalho, A. C. (2019). A meta-learning recommender system for hyperparameter tuning: predicting when tuning improves svm classifiers. *Information Sciences*, 501:193–221.

Marsland, S. (2009). *Machine Learning - An Algorithmic Perspective.* Chapman and Hall / CRC machine learning and pattern recognition series. CRC Press.

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., and Rudolph, G. (2011). Exploratory landscape analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, page 829–836, New York, NY, USA. Association for Computing Machinery.

Rice, J. R. (1976). The algorithm selection problem. volume 15 of *Advances in Computers*, pages 65–118. Elsevier.

Ridd, P. and Giraud-Carrier, C. (2014). Using metalearning to predict when parameter optimization is likely to improve classification accuracy. In Vanschoren, J., Brazdil, P., Soares, C., and Kotthoff, L., editors, *Meta-learning and Algorithm Selection Workshop at ECAI 2014*, pages 18–23.

Rivolli, A., Garcia, L. P., Soares, C., Vanschoren, J., and de Carvalho, A. C. (2022). Meta-features for meta-learning. *Knowledge-Based Systems*, 240:108101.

Vanschoren, J. (2019). *Meta-Learning*, pages 35–61. Springer International Publishing.