An-Najah University Journal for Research – A
Natural Sciences

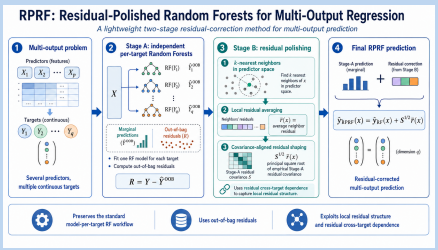
RPRF: Residual-Polished Random Forests for Multi-Output Regression

Abdelrahman Eid[1*], Mohammad Hawawreh[2]

Type: Full Article. Received: 10th May. 2026, Accepted: 5th July. 2026

Accepted Manuscript, In Press

Abstract: Multi-output regression requires prediction of several continuous outcomes from the same predictor set. Independent per-target models are simple to train, but they may leave residual association between targets unused. We propose a residual-polishing extension of per-target Random Forest regression that preserves the standard model-per-target workflow while adding a post-hoc correction based on out-of-bag residuals, local residual averaging, and residual-covariance structure. The method is evaluated using controlled synthetic scenarios and real multi-output datasets. The evaluation combines real-data benchmark comparisons with repeated train/validation/test experiments, including confidence intervals, paired statistical tests, and comparisons with several multi-output baselines. The results show that residual polishing can improve the independent Random Forest baseline when recoverable local and cross-target residual structure remains after marginal fitting.



Keywords: **Keywords:** multi-output regression; multi-target prediction; Random Forests; residual modeling; dependence-aware learning; k-nearest neighbors; stacking

Introduction

Multi-output regression, often termed multi-target regression, studies how to predict several continuous outcomes from the same set of input features [1, 2]. This setting appears across many areas of applied machine learning where outcomes are linked through shared mechanisms and are therefore statistically dependent [1, 3]. When dependence is present, prediction quality is not only about reducing the error of each target. It may also depend on whether useful cross-target structure remains after marginal prediction and can still be exploited to improve overall predictive accuracy [1, 2].

A common baseline is to train one model for each target [1, 2]. This approach is easy to train and validate, and it allows each target model to be tuned separately. Its limitation is that it does not use cross-target information. Even after conditioning on the inputs, residual errors can remain coupled across targets. Ignoring this structure may leave useful predictive information unexploited, especially when residual dependence remains after conditioning on the inputs [1–3].

Many multi-output methods use target dependence more directly [1]. These approaches can improve prediction when targets are related, but they often require additional modeling structures beyond the standard independent per-target workflow. This creates a practical need for methods that preserve the simplicity of per-target prediction while still allowing residual cross-target information to be used after the main marginal models have been fitted.

This paper introduces RPRF, a residual-polishing extension of per-target Random Forest regression for multi-output prediction. The method is designed to use residual structure that remains after the independent target models have been trained. Its contribution is to add a structured post-hoc correction layer while keeping the original model-per-target workflow unchanged. Therefore, the study introduces a lightweight residual-correction layer for independent Random Forest predic-

[1] Department of Mathematics and Physics, Faculty of Science, An-Najah National University, Nablus, Palestine.
[2] Business Intelligence and Data Analysis Graduate Program, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine, m.hawawreh@najah.edu.
* Corresponding author: abed.eid@najah.edu.
ORCID iD: 0000-0003-3512-9763

tion, not as a universally dominant multi-output learner. The full formulation is given in Section 3.

Related work

Multi-output regression methods are commonly grouped into problem transformation and algorithm adaptation [1]. Problem transformation retains a standard single-output learner but changes the representation to share information across targets, while algorithm adaptation modifies learning to predict an output vector directly.

A widely used transformation direction trains one model per target while allowing information from other targets to enter through additional inputs. Stacked single-target learning and regressor chains follow this idea and can improve accuracy when targets are dependent [2]. A central requirement is leakage control, since target-based inputs should be generated out of sample during training so that the learner sees the same type of information it will have at prediction time [2, 4]. Random linear target combinations provide another transformation-based approach by constructing new targets from random linear combinations of the original outputs [5]. Recent extensions have further developed chain-based and ensemble-based designs for multi-target regression, showing that target-dependence modelling remains an active direction [6, 7].

Algorithm-adaptation approaches aim to learn predictors that output all targets together. Tree-based methods are a central example, including structured-output tree ensembles, predictive clustering tree ensembles, and recent multi-output Random Forest applications [3, 8]. This line has been extended with ensemble designs that improve robustness for multi-target prediction, including random output selection and extremely randomized predictive clustering tree ensembles [9, 10]. Rule ensemble methods provide another joint modeling approach and can offer interpretable structure at the rule level [11].

A related viewpoint comes from meta-learning, where a second stage is used to reduce systematic error by combining signals produced by base learners. Stacking is a classic example and it highlights the role of out-of-sample base predictions when learning a second-stage correction [4, 12]. Later work studied when stacking helps and how to limit overfitting in the second stage [13]. The super learner framework follows a similar principle and uses cross-validation to combine a set of candidate learners while targeting prediction risk [14].

The proposed method is also related to residual learning and error-correction strategies, where a second-stage procedure is used to model the error left by an initial predictor. In this view, the first-stage model estimates the main signal, while the second stage attempts to recover systematic residual structure. Recent multi-output work has also used residual correction to combine interpretable first-stage models with Random Forest residual modeling [15]. This work follows the same general motivation of correcting remaining error, but differs in its construction.

The present work is closest to meta-learning approaches in which a second stage uses out-of-sample base-model information to reduce systematic prediction error [4, 12]. However, existing target-as-feature methods, including stacked single-target learning and regressor chains, typically use target values or target predictions as additional inputs to another learner [2]. Structured-output tree ensembles follow a different direction by adapting the learning algorithm itself to produce joint outputs [3]. These approaches show the value of using target dependence, but they also move away from the simple independent model-per-target workflow. This leaves room for a post-hoc residual-correction strategy that preserves separate target models while introducing dependence only after the marginal predictions have been learned.

Method

Problem formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ denote n observed pairs, where $x_i \in \mathbb{R}^p$ is the feature vector and $y_i = (y_{i1}, \ldots, y_{iq})^\top \in \mathbb{R}^q$ is a vector of q continuous targets [1]. The goal is to learn a predictor $\widehat{f} : \mathbb{R}^p \to \mathbb{R}^q$ that yields accurate prediction for each target while making effective use of cross-target dependence when it remains informative after conditioning on the inputs. Throughout, we use the equivalent notation

$$\widehat{f}(x) = \widehat{Y}(x) = (\widehat{y}_1(x), \ldots, \widehat{y}_q(x))^\top \in \mathbb{R}^q.$$

For training samples, $\widehat{y}_{ij}$ denotes the prediction of target j at case i.

We use a training partition of size n_{train} and a validation partition of size N_{val} for selecting the tuning parameter introduced below. We write I_q for the $q \times q$ identity matrix. The neighbor count candidates are collected in a finite set $\mathcal{K} \subset \{1, 2, \ldots\}$, and the selected value is denoted by $k^\star$.

Residual-Polished Random Forests

Residual-Polished Random Forests, denoted RPRF, is a two-stage procedure. The first stage fits one Random Forest regressor per target and produces out-of-bag residuals [16]. Random Forests are used because they are a strong default for tabular regression and often achieve competitive accuracy with limited tuning [16, 17]. The second stage builds a residual correction using local

averaging in feature space and then couples that correction across targets using the empirical residual covariance. This follows the general principle that a correction layer should be learned from out-of-sample style errors rather than from in-sample fits [4,12]. The full procedure, including validation-based selection of k, is detailed below and summarized in Algorithm 1.

Stage A: Per-target Random Forests and out-of-bag residuals

For each target $j = 1, \dots, q$, we train a Random Forest regressor $f_j : \mathbb{R}^p \to \mathbb{R}$ on the training partition [16]. For each training case i and target j, Random Forests provide an out-of-bag prediction $\widehat{y}_{ij}^{\mathsf{OOB}}$ computed by aggregating only trees that did not include case i in their bootstrap sample [16]. We define

$$\widehat{y}_{ij}^{\mathsf{OOB}} = f_j^{\mathsf{RF,OOB}}(x_i), \qquad r_{ij} = y_{ij} - \widehat{y}_{ij}^{\mathsf{OOB}}, \tag{1}$$

where $f_j^{\mathsf{RF,OOB}}$ denotes the out-of-bag prediction rule induced by the fitted forest f_j. Residuals are stacked rowwise as

$$R_{\mathsf{train}} = [r_{ij}] \in \mathbb{R}^{n_{\mathsf{train}} \times q}, \qquad S = \mathrm{cov}(R_{\mathsf{train}}) \in \mathbb{R}^{q \times q}. \tag{2}$$

We use the standard convention where rows correspond to samples and columns correspond to targets, so S is the empirical covariance across the q target residual coordinates. The matrix R_{train} collects residual row vectors $r_i \in \mathbb{R}^{1 \times q}$ (the ith row of R_{train}), and we write $r_i^\top \in \mathbb{R}^q$ for the corresponding column vector. Out-of-bag residuals are leakage safe because each training case is predicted by trees that did not use it during fitting [16]. This makes R_{train} and its covariance S reflect out-of-sample style errors, which is the relevant signal for learning a correction layer [4,12].

Remark 1. Fix a target index j and a training case index i. In a Random Forest, the out-of-bag prediction $\widehat{y}_{ij}^{\mathsf{OOB}}$ aggregates only those trees whose bootstrap samples do not contain case i. Consequently, the residual $r_{ij} = y_{ij} - \widehat{y}_{ij}^{\mathsf{OOB}}$ in Eq. (1) is not a resubstitution residual.

Proof. By definition, the set of trees contributing to $\widehat{y}_{ij}^{\mathsf{OOB}}$ consists only of trees trained on bootstrap samples that exclude case i [16]. Therefore, y_{ij} is not used in fitting any tree that contributes to $\widehat{y}_{ij}^{\mathsf{OOB}}$, and r_{ij} is computed from a prediction obtained without using the response of case i. □

Stage B: Local residual prototype and covariance-aware correction

For a query $x \in \mathbb{R}^p$, let $\mathcal{N}_k(x) \subset \{1, \dots, n_{\mathsf{train}}\}$ denote the indices of its k nearest training points under Euclidean distance. Local averaging in feature space is a standard nonparametric idea behind nearest neighbor regression and related local methods [18, 19]. The uniform local mean residual is

$$\overline{r}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} r_i^\top \in \mathbb{R}^q, \tag{3}$$

where r_i is the ith row of R_{train}. We treat $r_i^\top$ and $\overline{r}(x)$ as column vectors in $\mathbb{R}^q$.

This local averaging step can be interpreted through a residual-decomposition view of the problem. After Stage A, the prediction error can be written as a residual vector field over the feature space. If the marginal Random Forest models capture the dominant signal but leave a non-negligible conditional residual mean, then nearby training residuals provide information about the remaining local error structure around a query point x. In this setting, the k-nearest-neighbor average acts as a simple estimator of a structured local residual mean. The polishing step is therefore most useful when two conditions hold simultaneously: the residuals remain locally structured in feature space, and cross-target dependence is still present after marginal fitting. Under these conditions, Stage B is intended to recover part of the remaining systematic error rather than to replace the main predictive signal already captured by Stage A.

The benefit of the polishing step is expected to depend on the operating regime. It is most useful when Stage A leaves non-negligible structured residual error, nearby points in feature space have similar residual vectors, and cross-target dependence remains informative after marginal fitting. Its benefit may be more limited when the remaining residuals are close to noise, when Euclidean neighborhoods do not capture similarity in residual behavior, or when the empirical residual covariance is weak or unstable. These regimes help clarify that Stage B is intended as a targeted dependence-aware correction rather than as a uniformly dominant replacement for the Stage-A predictor.

To couple the correction across targets, we use a matrix square root of S. Let $S = U \Lambda U^\top$ be an eigendecomposition with $U \in \mathbb{R}^{q \times q}$ orthonormal and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_q)$ with $\lambda_\ell \geq 0$. We define

$$S^{1/2} = U \, \Lambda^{1/2} U^\top, \tag{4}$$

where $\Lambda^{1/2} = \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_q})$. The matrix square root is a standard matrix function and provides a principled way to transfer covariance structure through a linear transform [20]. Since empirical covariance estimation can be unstable in finite samples, especially when some eigenvalues are very small, we floor eigenvalues below a small constant $\delta > 0$ before forming $\Lambda^{1/2}$ [21]. Concretely, we replace λ_ℓ by $\max(\lambda_\ell, \delta)$ before forming $\Lambda^{1/2}$.

The use of $S^{1/2}$ is interpreted in this work as a practical covariance-aligned residual-shaping step, not as a claim of theoretical optimality. The identity correction would use local residual smoothing without cross-target covariance structure, while direct multiplication by S may produce a more aggressive scaling of the correction. The principal square root keeps the eigen-directions of the empirical residual covariance but applies square-root scaling to their eigenvalues. In this sense, $S^{1/2}$ is used as a moderate covariance-aligned operator that introduces cross-target coupling without treating the covariance matrix itself as a direct weight matrix. Other operators, including scalar, diagonal, correlation-based, shrinkage, or target-standardized corrections, are possible alternatives and are left for future work.

The covariance-aware polishing vector and the final multi-output prediction are

$$r^{\text{polish}}(x) = S^{1/2}\,\overline{r}(x), \qquad \widehat{Y}^{\text{RPRF}}(x) = \widehat{Y}^{\text{RF}}(x) + r^{\text{polish}}(x), \tag{5}$$

where $\widehat{Y}^{\text{RF}}(x) = \big(f_1(x), \ldots, f_q(x)\big)^\top$ stacks the per-target Random Forest predictions (with f_j as defined in Stage A). The product $S^{1/2}\overline{r}(x)$ is well defined because $S^{1/2} \in \mathbb{R}^{q\times q}$ and $\overline{r}(x) \in \mathbb{R}^q$. Multiplication by $S^{1/2}$ injects the empirical residual-covariance geometry into the local correction before it is added to the marginal Random Forest prediction.

Lemma 1. *Let $z \in \mathbb{R}^q$ be a random column vector with finite second moments, and let $A \in \mathbb{R}^{q\times q}$ be deterministic. Then*

$$\operatorname{cov}(Az) = A \operatorname{cov}(z)\, A^\top.$$

In particular, if $\operatorname{cov}(z) = I_q$, then $\operatorname{cov}(S^{1/2}z) = S$.

Proof. Let $\mu = \mathbb{E}[z]$. By linearity, $\mathbb{E}[Az] = A\mu$, hence $Az - \mathbb{E}[Az] = A(z-\mu)$. Therefore,

$$\begin{aligned}\operatorname{cov}(Az) &= \mathbb{E}\big[(Az - \mathbb{E}[Az])(Az - \mathbb{E}[Az])^\top\big] \\ &= A\,\mathbb{E}\big[(z-\mu)(z-\mu)^\top\big]\,A^\top \\ &= A \operatorname{cov}(z)\, A^\top.\end{aligned}$$

For the special case, set $A = S^{1/2}$ and $\operatorname{cov}(z) = I_q$, giving $\operatorname{cov}(S^{1/2}z) = S^{1/2} I_q (S^{1/2})^\top = S$, since $S^{1/2}$ is symmetric in Eq. (4). □

Lemma 1 clarifies the covariance-shaping effect of applying a fixed linear operator to a residual vector. In RPRF, this result motivates the use of $(S^{1/2})$ as a practical covariance-aligned transformation derived from the empirical Stage-A residual covariance. The method does not require the local residual prototype $(\overline{r}(x))$ to have identity covariance, and $(S^{1/2})$ is therefore used as a structured residual-shaping operator rather than as an optimal correction rule.

Proposition 1. *If the empirical residual covariance S in Eq.* (2) *is diagonal, then $S^{1/2}$ is also diagonal and the polishing step in Eq.* (5) *does not mix targets. In this case, each target receives a local residual correction scaled only by the square root of its own marginal residual variance.*

Proof. If S is diagonal with nonnegative diagonal entries, then an eigendecomposition is given by $U = I_q$ and $\Lambda = S$. Hence, $S^{1/2} = U\Lambda^{1/2}U^\top = \Lambda^{1/2}$ is diagonal. Therefore, $S^{1/2}\overline{r}(x)$ scales each coordinate of $\overline{r}(x)$ and does not form linear combinations across targets. □

According to Proposition 1, the cross-target mixing in RPRF is introduced only when the empirical Stage-A residual covariance contains off-diagonal structure.

Model selection and evaluation metrics

The method introduces a single hyperparameter k, the number of neighbors in $\mathcal{N}_k(x)$. The value of k is selected on the validation split by minimizing macro-averaged RMSE. A monotone grid over k is evaluated, with candidate values satisfying $1 \le k < n_{\text{train}}$, and the value with the lowest validation MacroRMSE is selected.

For an evaluation set of size N, the target-specific MAE and RMSE are defined as

$$\text{MAE}_j = \frac{1}{N}\sum_{i=1}^{N} |y_{ij} - \widehat{y}_{ij}|, \qquad \text{RMSE}_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{ij} - \widehat{y}_{ij})^2}. \tag{6}$$

The reported MAE and RMSE values are macro-averaged across targets:

$$\text{MacroMAE} = \frac{1}{q}\sum_{j=1}^{q}\text{MAE}_j, \qquad \text{MacroRMSE} = \frac{1}{q}\sum_{j=1}^{q}\text{RMSE}_j. \tag{7}$$

We also report the coefficient of determination. For target j, let $\bar{y}_j = \frac{1}{N}\sum_{i=1}^{N} y_{ij}$. Then

$$R_j^2 = 1 - \frac{\sum_{i=1}^{N}(y_{ij} - \widehat{y}_{ij})^2}{\sum_{i=1}^{N}(y_{ij} - \bar{y}_j)^2}. \tag{8}$$

When a single R^2 value is reported, it is macro-averaged across targets.

Algorithm 1 gives a step-by-step summary of the two-stage RPRF pipeline. Stage A fits the independent per-target Random Forest models and extracts out-of-bag residuals. Stage B uses nearest-neighbor residual averaging and covariance-aligned residual shaping before adding the correction to the Random Forest prediction.

The computational cost follows directly from the two-stage structure. Stage A has the same training cost as fitting q Random Forest models [16]. Stage B adds

a nearest neighbor lookup and a $q \times q$ linear transform. With brute-force search, the distance computation is $O(n_{\text{train}}p)$ per query, forming $\overline{r}(x)$ costs $O(kq)$, and applying $S^{1/2}$ costs $O(q^2)$.

Algorithm 1: Two-stage Residual-Polished Random Forests (RPRF) procedure

Require: Training split $(X_{\text{train}}, Y_{\text{train}})$, validation split $(X_{\text{val}}, Y_{\text{val}})$, test inputs X_{test}, candidate set $\mathcal{K}$

Ensure: Predicted $\widehat{Y}^{\text{RPRF}}(X_{\text{test}})$

Fit per target Random Forests $f_1, \ldots, f_q$ on $(X_{\text{train}}, Y_{\text{train}})$.
Obtain out-of-bag predictions $\widehat{y}_{ij}^{\text{OOB}}$ and residuals $r_{ij} = y_{ij} - \widehat{y}_{ij}^{\text{OOB}}$.
Form $R_{\text{train}} = [r_{ij}]$ and $S = \text{cov}(R_{\text{train}})$.
Compute $S^{1/2} = U\Lambda^{1/2}U^\top$ with eigenvalue flooring.
for $k \in \mathcal{K}$ **do**
  for $x \in X_{\text{val}}$ **do**
    Find $\mathcal{N}_k(x)$ in X_{train} using Euclidean distance.
    Compute $\overline{r}(x) = \frac{1}{k}\sum_{i\in\mathcal{N}_k(x)} r_i^\top$.
    Form $\widehat{Y}^{\text{RF}}(x) = (f_1(x), \ldots, f_q(x))^\top$.
    Compute $\widehat{Y}^{\text{RPRF}}(x) = \widehat{Y}^{\text{RF}}(x) + S^{1/2}\overline{r}(x)$.
  end for
  Compute $\text{MacroRMSE}(k)$ on $(X_{\text{val}}, Y_{\text{val}})$ using Eq. (7).
end for
Select $k^\star = \arg\min_{k\in\mathcal{K}} \text{MacroRMSE}(k)$.
for $x \in X_{\text{test}}$ **do**
  Find $\mathcal{N}_{k^\star}(x)$ in X_{train}.
  Compute $\overline{r}(x) = \frac{1}{k^\star}\sum_{i\in\mathcal{N}_{k^\star}(x)} r_i^\top$.
  Form $\widehat{Y}^{\text{RF}}(x) = (f_1(x), \ldots, f_q(x))^\top$.
  Compute $\widehat{Y}^{\text{RPRF}}(x) = \widehat{Y}^{\text{RF}}(x) + S^{1/2}\overline{r}(x)$.
end for

Computing $S^{1/2}$ by eigen-decomposition is performed once per training split and costs $O(q^3)$ [20]. Consequently, training time is dominated by fitting the q forests, and prediction time per query is the sum of (i) computing per-target Random Forest predictions, (ii) a brute-force kNN query, and (iii) applying the $q \times q$ transform. Under brute-force neighbor search, this yields approximately

$$O(C_{\text{RF}}(q, T) + n_{\text{train}}p + kq + q^2), \tag{9}$$

where $C_{\text{RF}}(q, T)$ denotes the cost of evaluating q forests with T trees each.

Experimental Evaluation

We evaluate RPRF on two real multi-output datasets, together with controlled synthetic scenarios as described in Section 4. All experiments used a 60%/15%/25% train/validation/test split. Repeated-split inference was used for the Energy benchmark and the synthetic experiments, as detailed below. In each repeated experiment, the training split was used to fit the models, the validation split was used to select (k), and the test split was kept untouched until final evaluation. Results from repeated experiments are summarized using the mean and 95% confidence interval of the test-set macro metrics.

Paired tests across repeated splits were used to compare methods. The primary comparison was RPRF versus independent per-target RF, because RPRF is proposed as a residual-polishing extension of that baseline. For error metrics such as RMSE and MAE, a positive paired difference defined as comparator error minus RPRF error indicates lower error for RPRF.

Performance is reported using the macro-averaged MAE, RMSE, and R^2 definitions given in Section 3. As presented in Table 1, the evaluation includes independent per-target RF, XGBoost MIMO, and three RF-based multi-output baselines. The additional baselines are Regressor Chains, stacked single-target learning, and random linear target combinations. These methods were included to compare RPRF with established approaches that use target dependence in multi-output regression [2, 4, 5].

Table (1): Baselines included in the evaluation.

Family	Method	Approach
Independent	Per-target RF	One RF model is fitted independently for each target.
Boosting	XGBOOST MIMO	A boosted-tree baseline fitted on a stacked target representation with one-hot target indicators [2, 22].
Proposed	RPRF	A residual-polished extension of per-target RF using k-nearest-neighbor residual smoothing and $S^{1/2}$.
Problem transformation	RC-RF	Regressor Chains with RF as the base learner [2].
Stacking-based	SST-RF	Stacked single-target learning with RF as the base learner [2, 4].
Target transformation	RLC-RF	Random linear target combinations with RF as the base learner [5].

Datasets

Real datasets

We evaluate RPRF on two real multi-output regression datasets: an environmental VOC prediction dataset and a building-energy benchmark. Both datasets contain strongly associated targets. In the VOCs dataset, the four selected targets showed strong positive pairwise associations, with Pearson correlations ranging from 0.754 to 0.881. In the Energy dataset, the two targets showed a very strong positive correlation (Pearson $= 0.976$).

The VOCs dataset [23] contains $n = 180$ observations, $p = 34$ predictor variables, and $q = 4$ continuous targets. This dataset is also closely related to a recent domain-specific multi-output VOC prediction study, where co-emitted VOCs and demographic variables were used to jointly predict paint-related VOC concentrations, illustrating the practical relevance of cross-target dependence in environmental exposure modeling [24]. The target variables were acetonitrile, n-butyl acetate, Toluene, and m/p-Xylene. Predictor variables were standardized before model fitting because Stage B defines neighborhoods by Euclidean distance. For each

predictor, the mean and standard deviation were estimated from the training split only, and the same scaling parameters were applied to the validation and test splits.

Additionally, the Energy Efficiency dataset (ENB2012) contains $n = 768$ samples with $p = 8$ input features and $q = 2$ continuous targets [25, 26]. We apply the same train-split-preprocessing policy: any scaling is fit on the training split and then applied unchanged to validation and test.

The two real datasets provide different empirical settings: a small four-output environmental dataset (VOCs) and a two-output building-energy benchmark (Energy). The synthetic study below is used to examine controlled changes in sample size, input dimension, output dimension, and residual correlation.

The two real datasets were used differently in the analyses as they differ in size and output structure. The larger Energy benchmark $(n = 768,\ q = 2)$ was used for repeated-split inference, whereas the smaller VOCs dataset $(n = 180,\ q = 4)$ was retained as a descriptive domain-specific benchmark. Repeated random partitions for VOCs would yield small test subsets and less stable inferential summaries.

Synthetic study

To complement the real-data evaluation, synthetic data were generated to vary the sample size n, feature dimension p, output dimension q, and cross-target dependence strength $\rho \in [0, 1)$.

For each configuration, features were sampled independently as

$$x_i \sim \mathcal{N}(0, I_p), \qquad i = 1, \ldots, n, \tag{10}$$

and targets were generated from a shared linear signal plus correlated noise:

$$y_i = B^\top x_i + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \Sigma_\rho), \tag{11}$$

where $B \in \mathbb{R}^{p \times q}$ controls the signal contribution and $\Sigma_\rho \in \mathbb{R}^{q \times q}$ has unit marginal variances and constant off-diagonal correlation:

$$(\Sigma_\rho)_{jj} = 1, \quad (\Sigma_\rho)_{j\ell} = \rho \ \ (j \neq \ell). \tag{12}$$

For each scenario, the coefficient matrix B was generated once, with entries sampled independently from a uniform distribution, and then held fixed across all evaluated methods.

We used a one-factor-at-a-time sweep design: one of $\{n, p, q, \rho\}$ was varied while the remaining factors were held fixed. This design isolates changes in performance due to sample size, input dimension, output dimension, and dependence strength under the same data-generation mechanism (Tables 6–13).

The original synthetic scenarios provide a controlled test of residual-correction behaviour under a shared signal and constant-correlation Gaussian noise. Additional scenarios were included with nonlinear signal, heteroscedastic errors, heavy-tailed errors, block covariance, and null residual correlation, as summarized in Table 2. These settings test whether residual polishing remains useful when the residual structure is more complex than the original compound-symmetry design.

Table (2): Additional synthetic scenarios used to extend the original compound-symmetry design.

Scenario	Setting	Purpose
S13	Nonlinear signal	Adds smooth nonlinear transformations of selected predictors to the shared signal.
S14	Heteroscedastic errors	Allows the error variance to change with the feature values.
S15	Heavy-tailed errors	Uses non-Gaussian heavy-tailed noise to assess robustness to outlying errors.
S16	Block covariance	Replaces the compound-symmetry covariance with a block covariance structure among targets.
S17	Null residual correlation	Examines performance when cross-target residual dependence is weak or absent.

The additional scenarios were constructed as follows. S13 introduced a nonlinear signal component by augmenting the shared linear signal with smooth nonlinear transformations of selected predictors. S14 used heteroscedastic errors whose variance changed with the feature values. S15 used heavy-tailed errors to assess robustness to non-Gaussian noise. S16 replaced the compound-symmetry covariance with a block covariance structure among targets. S17 used a null residual-correlation setting to examine performance when cross-target residual dependence is weak or absent. These scenarios preserve the same train/validation/test evaluation protocol and are intended as controlled stress tests rather than exhaustive coverage of all possible multi-output data-generating mechanisms.

Results

Real-data results

We report macro-aggregated mean absolute error (MAE) and root mean squared error (RMSE) across targets, together with macro-averaged R^2. Lower MAE and RMSE indicate better point accuracy, while higher R^2 indicates better explained variance. Table 3 reports the VOCs comparison, and Table 4 reports the repeated-split Energy comparison with 95% confidence intervals.

In the VOCs analysis, XGBOOST MIMO gave the lowest macro-MAE and macro-RMSE and the highest macro-averaged R^2. The residual-polishing step produced a small reduction in error relative to independent RF. Thus, the VOCs experiment supports RPRF as an

improvement over the independent RF baseline, but not as the

Table (3): VOCs results ($n = 180$, $p = 34$, $q = 4$).

Model	MAE	RMSE	R^2
Per-target RF	1.2328	1.7517	0.8902
RPRF	1.2171	1.7456	0.8899
XGBOOST MIMO	1.0835	1.5801	0.9080
RC-RF	1.2407	1.7543	0.8904
SST-RF	1.2424	1.7888	0.8870
RLC-RF	1.2275	1.7350	0.8857

best-performing model in this dataset.

Table (4): Energy repeated-split results (mean [95% CI]; $n = 768$, $p = 8$, $q = 2$).

Model	MAE	RMSE	R^2
Per-target RF	0.8726 [0.8562, 0.8890]	1.2726 [1.2430, 1.3021]	0.9794 [0.9785, 0.9804]
RPRF	0.8389 [0.8220, 0.8558]	1.2322 [1.2026, 1.2618]	0.9808 [0.9799, 0.9817]
XGBOOST MIMO	0.4200 [0.3943, 0.4458]	0.6754 [0.6329, 0.7178]	0.9940 [0.9931, 0.9948]
RC-RF	0.8649 [0.8472, 0.8826]	1.2853 [1.2558, 1.3148]	0.9789 [0.9778, 0.9799]
SST-RF	0.7314 [0.7088, 0.7540]	1.1674 [1.1314, 1.2034]	0.9805 [0.9795, 0.9816]
RLC-RF	0.8820 [0.8650, 0.8991]	1.2919 [1.2621, 1.3217]	0.9787 [0.9777, 0.9797]

In the Energy analysis, XGBOOST MIMO gave the lowest macro-MAE and macro-RMSE and the highest macro-averaged R^2. RPRF improved independent RF, reducing macro-RMSE from 1.2726 to 1.2322. However, XGBOOST MIMO and SST-RF achieved lower RMSE. Therefore, the Energy results show that residual polishing added value to the independent RF workflow, while stronger benchmark models remained competitive or superior.

Statistical significance analysis

To assess whether the observed differences were stable across random partitions, paired tests were computed across repeated train/validation/test splits. The primary analysis compared RPRF with independent per-target RF, because RPRF is built as a residual-polishing extension of that baseline. Statistical significance was interpreted together with the direction of the mean difference; a small p-value indicates a stable difference across splits, while the sign of the difference determines whether it favors RPRF or the comparator. For RMSE, the paired difference was defined as comparator RMSE minus RPRF RMSE, so positive values indicate lower error for RPRF.

For the real-data repeated-split analysis, the primary paired comparison is reported for the Energy benchmark.

Table (5): Primary paired RMSE comparison between RPRF and independent RF for the Energy dataset across repeated splits. Positive differences indicate lower RMSE for RPRF.

Dataset	**RPRF RMSE**	**RF RMSE**	Difference	95% CI	**paired t-test p**
Energy	1.2322	1.2726	0.0404	[0.0353, 0.0454]	< 0.001

Synthetic results

To complement the real-data comparisons with controlled dependence settings, we report synthetic results using the same macro-aggregated RMSE summaries. Table 6 reports mean macro-RMSE across the 12 simulated scenarios. Comparisons should be read within each scenario row because n, p, q, and ρ vary across scenarios.

Across the 12 simulated scenarios, RPRF produced lower macro-RMSE than independent RF, RC-RF, and RLC-RF in every scenario. It also produced lower macro-RMSE than SST-RF in 11 scenarios; the exception was S11, where SST-RF obtained a lower error. Relative to XGBOOST MIMO, RPRF gave lower macro-RMSE in 9 scenarios, while XGBOOST MIMO gave lower error in S02, S03, and S11. The largest reductions for RPRF occurred in the higher-dimensional scenarios S05 and S12, where the errors of independent RF, RC-RF, SST-RF, and RLC-RF increased substantially.

Table (6): Synthetic scenarios: mean macro-RMSE across repeated splits.

Scenario	ρ	n	p	q	k	RPRF	RF	XGBOOST MIMO	RC-RF	SST-RF	RLC-RF
S01	0.4	1200	12	3	70	1.4460	2.2303	1.5609	2.2673	1.6907	2.1821
S02	0.4	5000	12	3	109	1.2957	1.9798	1.2489	1.9986	1.5067	1.9515
S03	0.4	15000	12	3	94	1.1974	1.7642	1.1121	1.7880	1.4006	1.8147
S04	0.4	1200	5	3	70	1.1446	1.1897	1.1560	1.1966	1.1723	1.2608
S05	0.4	1200	50	3	200	2.7996	6.3060	5.3325	6.3175	4.7519	6.2151
S06	0.4	1200	12	2	48	1.3598	1.9362	1.4197	1.9499	1.5507	1.9823
S07	0.4	1200	12	5	102	1.4433	2.4419	1.5560	2.4833	1.8431	2.3702
S08	0.4	1200	12	7	113	1.4137	2.1835	1.4374	2.2389	1.7310	2.0595
S09	0.1	1200	12	3	48	1.4136	2.1305	1.5191	2.1503	1.6851	2.0913
S10	0.7	1200	12	3	91	1.4182	2.2746	1.5771	2.3064	1.7381	2.3304
S11	0.7	15000	50	7	200	4.7417	5.8834	2.7007	5.9196	4.3572	5.7350
S12	0.1	1200	50	7	200	2.9894	6.3712	5.1685	6.4119	4.8552	6.2641

Table 7 reports the paired RMSE comparison between RPRF and independent RF for the synthetic scenarios. The positive paired differences show that the RMSE reduction relative to independent RF was stable across repeated splits in all 12 scenarios.

Table (7): Paired RMSE comparison between RPRF and independent RF across repeated splits for the existing synthetic scenarios. Positive differences indicate lower RMSE for RPRF.

Scenario	**RPRF RMSE**	**RF RMSE**	Difference	95% CI	**paired** t**-test** p
S01	1.4460	2.2303	0.7843	[0.7468, 0.8218]	< 0.001
S02	1.2957	1.9798	0.6841	[0.6706, 0.6977]	< 0.001
S03	1.1974	1.7642	0.5668	[0.5569, 0.5767]	< 0.001
S04	1.1446	1.1897	0.0450	[0.0364, 0.0536]	< 0.001
S05	2.7996	6.3060	3.5064	[3.3998, 3.6129]	< 0.001
S06	1.3598	1.9362	0.5764	[0.5338, 0.6190]	< 0.001
S07	1.4433	2.4419	0.9986	[0.9655, 1.0317]	< 0.001
S08	1.4137	2.1835	0.7698	[0.7469, 0.7927]	< 0.001
S09	1.4136	2.1305	0.7170	[0.7023, 0.7316]	< 0.001
S10	1.4182	2.2746	0.8564	[0.8224, 0.8903]	< 0.001
S11	4.7417	5.8834	1.1416	[1.1100, 1.1733]	< 0.001
S12	2.9894	6.3712	3.3818	[3.2957, 3.4679]	< 0.001

To evaluate the method beyond the original compound-symmetry Gaussian setting, Table 8 reports additional synthetic scenarios with nonlinear signal, heteroscedastic errors, heavy-tailed errors, block covariance, and null residual correlation. In all five scenarios, RPRF reduced macro-RMSE relative to independent RF, supporting the residual-polishing mechanism under more complex controlled conditions.

Table (8): Additional synthetic scenarios: mean macro-RMSE across repeated splits where $(n = 1200,\ \&p = 12)$.

Scenario	Setting	q	Selected k	RPRF	RF	XGBoost MiMO	RC-RF	SST-RF	RLC-RF
S13	Nonlinear signal	3	59	1.576	2.228	1.591	2.242	1.774	2.253
S14	Heteroscedastic errors	3	113	1.820	2.569	1.940	2.596	2.084	2.610
S15	Heavy-tailed errors	3	91	1.906	2.455	1.991	2.476	2.099	2.398
S16	Block covariance	6	91	1.462	2.293	1.510	2.359	1.819	2.195
S17	Null residual correlation	3	70	1.343	2.120	1.550	2.144	1.614	2.140

Table 9 reports the paired RMSE comparison for the additional synthetic scenarios. The paired differences were positive in all five scenarios, indicating stable reductions in RMSE relative to independent RF.

One-factor-at-a-time (OFAT) synthetic sweeps

In addition to the full scenario grid in Table 6, the OFAT sweeps examine the effect of changing one design factor at a time while keeping the remaining factors fixed. These sweeps provide a compact diagnostic view of how performance changes with the input dimension p, output dimension q, dependence strength ρ, and sample size n. The tables focus on the core comparison among RPRF, independent RF, and XGBoost MiMO, while the full six-model comparison is reported in Table 6.

Table (9): Paired RMSE comparison between RPRF and independent RF across repeated splits for the additional synthetic scenarios. Positive differences indicate lower RMSE for RPRF.

Scenario	RPRF RMSE	RF RMSE	Diff.	95% CI	paired t-test p
S13	1.5761	2.2282	0.6521	[0.6338, 0.6704]	< 0.001
S14	1.8197	2.5686	0.7490	[0.7207, 0.7772]	< 0.001
S15	1.9063	2.4548	0.5486	[0.5337, 0.5634]	< 0.001
S16	1.4616	2.2928	0.8312	[0.8147, 0.8477]	< 0.001
S17	1.3428	2.1204	0.7776	[0.7575, 0.7978]	< 0.001

Table (10): OFAT sweep over feature dimension p with $(n, q, \rho) = (1200, 3, 0.4)$ fixed.

n	q	ρ	p	RMSE RPRF	RMSE RF	RMSE XGBoost MiMO
1200	3	0.4	5	1.1446	1.1897	1.1560
1200	3	0.4	12	1.4460	2.2303	1.5609
1200	3	0.4	50	2.7996	6.3060	5.3325

Table (11): OFAT sweep over output dimension q with $(n, p, \rho) = (1200, 12, 0.4)$ fixed.

n	p	ρ	q	RMSE RPRF	RMSE RF	RMSE XGBoost MiMO
1200	12	0.4	2	1.3598	1.9362	1.4197
1200	12	0.4	3	1.4460	2.2303	1.5609
1200	12	0.4	5	1.4433	2.4419	1.5560
1200	12	0.4	7	1.4137	2.1835	1.4374

Table (12): OFAT sweep over dependence strength ρ with $(n, p, q) = (1200, 12, 3)$ fixed.

ρ	n	p	q	RMSE RPRF	RMSE RF	RMSE XGBoost MiMO
0.1	1200	12	3	1.4136	2.1305	1.5191
0.4	1200	12	3	1.4460	2.2303	1.5609
0.7	1200	12	3	1.4182	2.2746	1.5771

Table (13): OFAT sweep over sample size n with $(p, q, \rho) = (12, 3, 0.4)$ fixed.

ρ	n	p	q	RMSE RPRF	RMSE RF	RMSE XGBoost MiMO
0.4	1200	12	3	1.4460	2.2303	1.5609
0.4	5000	12	3	1.2957	1.9798	1.2489
0.4	15000	12	3	1.1974	1.7642	1.1121

The OFAT sweeps show that RPRF was most favorable when the independent RF baseline left larger residual error, especially in higher-dimensional feature and output settings. RPRF was best in the p, q, and ρ sweeps, while increasing the sample size favored XGBoost MiMO in this simulation design, as XGBoost MiMO achieved the lowest RMSE at $n = 5000$ and $n = 15000$.

Discussion

The experiments show that RPRF can provide a residual-polishing benefit over the independent per-target RF baseline, especially in controlled synthetic settings where residual structure is deliberately introduced. In the simulated scenarios, RPRF reduced macro-RMSE relative to independent RF, RC-RF, and RLC-RF in all 12 scenarios. It also reduced macro-RMSE relative to SST-RF in 11 scenarios, with S11 as the only exception. Against XGBOOST MIMO, RPRF achieved lower macro-RMSE in 9 of the 12 scenarios. These results support the usefulness of a post-hoc residual-correction layer when the first-stage per-target forests leave recoverable local and cross-target residual structure. However, the method should not be interpreted as uniformly superior to all multi-output alternatives.

The two real datasets provide a more practical view of this behaviour. In VOCs and Energy, RPRF reduced the error of independent RF, but the gains were modest and dataset-dependent. In both real-data analyses, XGBOOST MIMO achieved the lowest error, and in Energy, SST-RF also achieved lower RMSE than RPRF. This pattern clarifies the intended role of the proposed method. RPRF is not designed to replace highly optimized boosted-tree multi-output models. Rather, it is designed to strengthen the independent per-target RF workflow when maintaining separate target-specific models is useful and when residual dependence remains after marginal fitting.

This interpretation is consistent with the mechanism of the method. Stage A estimates the main marginal signal using one Random Forest per target. Stage B then uses local residual information and the empirical residual covariance to adjust the prediction. Therefore, the benefit of RPRF depends on the amount of residual structure left after Stage A. When nearby observations have similar residual patterns and the residual covariance remains informative, the correction can reduce error. When the first-stage learner or a competing model already captures most of the systematic structure, the additional gain from residual polishing is expected to be smaller. This explains why RPRF performed strongly in several simulated settings while showing more moderate gains in the real datasets.

The comparison with the added multi-output baselines also helps position the proposed approach. Regressor Chains and stacked single-target learning use target-derived information as additional input, whereas random linear target combinations transform the output space before fitting the base learner [2,5]. RPRF follows a different construction: it first estimates the marginal prediction functions and then models the residual field left by these functions. The method therefore introduces target dependence through a post-hoc residual layer rather than through target augmentation or output-space transformation. This distinction is important because it preserves the interpretability and operational simplicity of separate target-specific RF models.

A practical strength of RPRF is that it keeps the model-per-target Random Forest workflow intact. This is useful when individual target models must be inspected, monitored, or deployed separately, while the prediction system still aims to use residual association across targets. The additional computation is limited to validation-based selection of k, nearest-neighbor residual averaging, and multiplication by $S^{1/2}$. Thus, RPRF is best viewed as a lightweight residual-polishing layer for improving independent RF predictions, especially when maintaining separate target models is desirable.

Several limitations remain. First, the real-data evaluation is limited to two datasets, both with positively associated targets, so broader claims about general applicability require additional external datasets with different sizes and correlation structures. Second, although the synthetic study includes additional controlled mechanisms, it cannot cover all forms of nonlinear, clustered, heteroscedastic, or target-specific dependence that may occur in practice. Third, the empirical residual covariance used in the polishing step may be unstable in small samples, when targets have very different scales, or when the number of targets is large relative to the training size. Future work should examine target-standardized residual correction, correlation-based residual shaping, shrinkage covariance estimation, alternative neighborhood metrics, and scalable nearest-neighbor search.

Conclusion

We introduced RPRF, a residual-polishing extension of per-target Random Forest regression for multi-output prediction. The method first fits independent RF models and then applies a second-stage correction based on out-of-bag residuals, local residual averaging, and covariance-based target coupling. This design keeps the marginal Random Forest models unchanged while allowing residual cross-target structure to be used after the main target-specific predictions have been learned.

The experimental evaluation supports RPRF as a practical residual-correction layer for the independent RF workflow. The strongest gains were observed in controlled synthetic settings, where residual structure can be explicitly varied. In the real datasets, RPRF improved independent RF but did not outperform all benchmark methods, particularly XGBOOST MIMO. These findings

support a focused interpretation of the method: RPRF is useful when recoverable local and cross-target residual structure remains after marginal fitting, but it is not intended as a universally dominant multi-output regression model.

Future work should include larger real multi-output datasets with varied target-correlation structures, further nonlinear and heteroscedastic simulation designs, target-standardized and correlation-based residual corrections, alternative distance metrics for residual neighborhoods, regularized covariance estimation, and scalable nearest-neighbor search.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The datasets and analysis code used in this study are available from the corresponding author upon reasonable formal request, subject to any applicable data-sharing restrictions.

Author's contribution

A.E. conceived the study. A.E. and M.H. developed the proposed methodology, formulated the theoretical framework, implemented the computational experiments, analyzed and interpreted the results, and prepared the original draft. Both authors reviewed and approved the final version of the manuscript.

Funding

This research received no external funding.

Disclosure Statement

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgements

The authors thank An-Najah National University (Nablus, Palestine), `www.najah.edu`, for its support and for the logistical and technical assistance provided for this work.

References

[1] Borchani H, Varando G, Bielza C, Larrañaga P. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2015;5(5):216–233. doi:10.1002/widm.1157

[2] Spyromitros-Xioufis E, Tsoumakas G, Groves W, Vlahavas I. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*. 2016;104(1):55–98. doi:10.1007/s10994-016-5546-z

[3] Kocev D, Vens C, Struyf J, Džeroski S. Tree ensembles for predicting structured outputs. *Pattern Recognition*. 2013;46(3):817–833. doi:10.1016/j.patcog.2012.09.023

[4] Wolpert DH. Stacked generalization. *Neural Networks*. 1992;5(2):241–259. doi:10.1016/S0893-6080(05)80023-1

[5] Tsoumakas G, Spyromitros-Xioufis E, Vrekou A, Vlahavas I. Multi-target regression via random linear target combinations. In: *Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2014, Lecture Notes in Computer Science*. Vol 8726. Springer; 2014:225–240. doi:10.1007/978-3-662-44845-8_15

[6] Geiß C, Brzoska E, Aravena Pelizari P, Lautenbach S, Taubenböck H. Multi-target regressor chains with repetitive permutation scheme for characterization of built environments with remote sensing. *International Journal of Applied Earth Observation and Geoinformation*. 2022;106:102657. doi:10.1016/j.jag.2021.102657

[7] Wei H, Wang X, Wen Z, Li E, Wang H. An ensemble-adaptive tree-based chain framework for multi-target regression problems. *Information Sciences*. 2024;653:119769. doi:10.1016/j.ins.2023.119769

[8] Safari MJS. Multi-Output Random Forest Model for Spatial Drought Prediction. *Sustainability*. 2026;18(2):1130. doi:10.3390/su18021130

[9] Breskvar M, Kocev D, Džeroski S. Ensembles for multi-target regression with random output selections. *Machine Learning*. 2018;107(11):1673–1709. doi:10.1007/s10994-018-5744-y

[10] Kocev D, Ceci M, Stepišnik T. Ensembles of extremely randomized predictive clustering trees for predicting structured outputs. *Machine Learning*. 2020;109:2213–2241. doi:10.1007/s10994-020-05894-4

[11] Aho T, Ženko B, Džeroski S, Elomaa T. Multi-target regression with rule ensembles. *Journal of Machine Learning Research*. 2012;13(78):2367–2407.

[12] Breiman L. Stacked regressions. *Machine Learning*. 1996;24:49–64. doi:10.1007/BF00117832

[13] Džeroski S, Ženko B. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning*. 2004;54(3):255–273. doi:10.1023/B:MACH.0000015881.36452.6e

[14] van der Laan MJ, Polley EC, Hubbard AE. Super Learner. *Statistical Applications in Genetics and Molecular Biology*. 2007;6(1):Article 25. doi:10.2202/1544-6115.1309

[15] Son NM, Truong DS, Nguyen TQ. Hybrid multi-output regression with residual correction for smart agriculture: a scalable and interpretable approach. *Applied Intelligence*. 2025;55:1081. doi:10.1007/s10489-025-06876-6

[16] Breiman L. Random Forests. *Machine Learning*. 2001;45(1):5–32. doi:10.1023/A:1010933404324

[17] Probst P, Wright MN, Boulesteix AL. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2019;9(3):e1301. doi:10.1002/widm.1301

[18] Cover TM, Hart PE. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*. 1967;13(1):21–27. doi:10.1109/TIT.1967.1053964

[19] Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer; 2009. doi:10.1007/978-0-387-84858-7

[20] Higham NJ. *Functions of Matrices: Theory and Computation*. Philadelphia: Society for Industrial and Applied Mathematics; 2008. doi:10.1137/1.9780898717778

[21] Ledoit O, Wolf M. A Well-Conditioned Estimator for Large-Dimensional Covariance Matrices. *Journal of Multivariate Analysis*. 2004;88(2):365–411. doi:10.1016/S0047-259X(03)00096-4

[22] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016:785–794. doi:10.1145/2939672.2939785

[23] Jodeh S, Chakir A, Hanbali G, Roth E, Eid A. Method Development for Detecting Low Level Volatile Organic Compounds (VOCs) among Workers and Residents from a Carpentry Work Shop in a Palestinian Village. *International Journal of Environmental Research and Public Health*. 2023;20(9):5613. doi:10.3390/ijerph20095613

[24] Eid A, Jodeh S, Hanbali G, Hawawreh M, Chakir A, Roth E. Multi-Output Machine-Learning Prediction of Volatile Organic Compounds (VOCs): Learning from Co-Emitted VOCs. *Environments*. 2025;12(7):216. doi:10.3390/environments12070216

[25] Tsanas A, Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*. 2012;49:560–567. doi:10.1016/j.enbuild.2012.03.003

[26] Tsanas A, Xifara A. Energy Efficiency [Dataset]. UCI Machine Learning Repository; 2012. doi:10.24432/C51307