An-Najah National University

Faculty of Graduate Studies

RESIDUAL-DRIVEN ENHANCEMENT OF MULTI-OUTPUT PREDICTION

By

Mohammad Hawawreh

Supervisor

Dr. Abdelrahman EID

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Business Intelligence and Data Analysis, Faculty of Graduate Studies, An-Najah National University, Nablus - Palestine.

2025

RESIDUAL-DRIVEN ENHANCEMENT OF MULTI-OUTPUT PREDICTION

By

Mohammad Hawawreh

This Thesis was Defended Successfully on 04/12/2025 and approved by

Dr. Abdelrahman EID	
Supervisor	Signature
Dr. Nashaat Jallad	
External Examiner	Signature
Dr.Mohammad Abu Omar	
Internal Examiner	Signature

Dedication

To my dear wife, my children, and my parents: thank you for your sacrifices, your faith in me, and your constant support. Nothing would have been possible without you.

Acknowledgment

I am profoundly grateful to Allah, the Almighty, for the countless blessings, guidance, and strength that enabled me to complete this thesis.

I would like to express my sincere gratitude to all my instructors and professors, whose wisdom and encouragement have profoundly shaped my academic journey. I owe special thanks to my supervisor, **Dr. Abdelrahman EID**, whose guidance, collaboration, and unwavering dedication were invaluable to this thesis.

Lastly, I extend my sincere gratitude to the thesis defense committee for their time, careful review, and insightful recommendations, which significantly strengthened this thesis.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

RESIDUAL-DRIVEN ENHANCEMENT OF MULTI-OUTPUT PREDICTION

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: **Mohammad Hawawreh**

Signature: 

Date: **04/12/2025**

List of Contents

List of Tables

List of Figures

List of Appendices

RESIDUAL-DRIVEN ENHANCEMENT OF MULTI-OUTPUT PREDICTION

By
Mohammad Hawawreh
Supervisor
Dr. Abdelrahman EID

Abstract

Business Intelligence teams often train a separate model for each Key Performance Indicator KPI to keep governance and explanations simple. The drawback is that related KPIs (e.g., revenue and margin) move together; training them in isolation can yield accurate but incoherent forecasts. This thesis proposes Residual-Polished Random Forests (RPRF) a light, two-stage upgrade that preserves per-target Random Forests RFs while explicitly borrowing strength across targets through the residuals. Stage A fits one RF per target and computes leakage-safe out-of-bag (OOB) residuals. Stage B constructs, for any new case, a local means of nearby training residuals (k-nearest neighbors k-NN) and applies a covariance-aware linear transform to align that correction with observed cross-target error structure; the adjusted residual is then added back to the Stage-A predictions. The design keeps standard per-target interpretability, adds only one main hyperparameter (k), and avoids target leakage via OOB predictions.

The method is evaluated under controlled synthetic scenarios (systematically varying sample size n, predictors p, number of targets q, and residual correlation ρ) and on two real datasets: the first one is volatile organic compounds VOCs with four outputs q=4; and the another is ENB2012 Energy with two output q=2. Performance is reported as per-target RMSE/R^2 and macro-averages, using a fixed train/validation/test protocol (k chosen on validation). Across simulations, RPRF consistently surpasses independent RF and often exceeds XGBoost, especially when data are limited, p is large, q is moderate-to high, or cross-target dependence is non-trivial, precisely the regimes where variance control and coherence matter. On real data, RPRF attains the best overall accuracy on VOCs (clear inter-target structure), and remains near-ceiling on ENB2012, where all models perform extremely well and XGBoost retains a slight edge with only two outputs. Overall, RPRF offers predictive improvements with minimal disruption, making it a practical default when KPIs are correlated but per-target workflows must be retained.

Keywords*:* Multi-Output Regression; Business Intelligence; Random Forests; Residual Smoothing; K-Nearest Neighbors; OOB Residual.

Chapter One
The Introduction

1.1 Overview

In the era of data-driven Business Intelligence (BI), predictive analytics is a foundation for companies to have strategic decision-making capabilities through foresight into future trends, leading to proactive management of operations. Accurate forecasting is vital because it impacts everything from resource allocation to market position. In the real world, however, organizations keep track of a multitude of performance metrics like sales, costs, customer satisfaction, and production quality that are related. In most cases, these metrics rarely vary in isolation; a shift in one often reverberates through others. For instance, in a manufacturing context, higher production throughput might coincide with increased energy consumption or changes in defect rates. Similarly, in marketing, an effective customer acquisition campaign could drive up revenue but also alter customer retention patterns. Therefore, such correlated outcomes are ubiquitous in BI datasets, which means predicting business metrics is inherently a multi-output problem, not a collection of separate single-output problems.

In this context, multi-output prediction (also known as multi-target or multivariate regression) involves forecasting multiple continuous outcome variables simultaneously, explicitly recognizing the dependencies among them (Borchani et al., 2015a; Tsoumakas et al., 2014) . This setting arises in many fields; for example, in finance one might forecast multiple correlated asset returns, or in ecology predict several environmental indicators together, and BI is no exception.

One of the key motivations is that targets are often statistically correlated or coupled due to shared drivers or constraints (Tsoumakas et al., 2014) . In BI contexts, revenue and profit margins tend to co-vary; customer acquisition and retention are linked; and in manufacturing, throughput, energy consumption, and defect rates influence each other. Consequently, modeling such targets independently can yield inconsistent or suboptimal predictions. In line with this, recent studies highlight that explicitly exploiting target correlations improves predictive performance (Breskvar et al., 2018) , including in environmental informatics where multi-output models leverage cross-target covariance for accuracy gains (Eid et al., 2025) . On average, multi-output regression methods outperform training multiple independent single-output models because they capture

cross-target interactions and promote coherence among predicted outcomes (Borchani et al., 2015b; Spyromitros-Xioufis et al., 2016a) .

Moreover, extensive evidence supports the benefits of joint modeling: learning related targets together can use informative relationships to improve accuracy and consistency relative to siloed models (Aho et al., 2012; Arashloo & Kittler, 2022; He et al., 2016; Schmid et al., 2022) . Nevertheless, the straightforward baseline in practice remains to fit independent ML models, like a separate Random Forest per target, which cannot capture residual correlations among outputs and may produce redundant structures or inconsistencies across predictions (Segal & Xiao, 2011) .

However, correlated targets also introduce challenges. Designing algorithms that can effectively model complex target dependency structures is non-trivial, as naively linking outputs could lead to overfitting or error propagation. Furthermore, multi-output models typically require larger datasets to reliably learn the joint distribution of targets and can be computationally demanding and sensitive to hyperparameters (Masmoudi et al., 2020). For example, methods such as regressor chains may need substantial tuning or preprocessing to remain stable with extreme values and growing output dimensionality (Kendall et al., 2018; Xu et al., 2019) . Therefore, evaluation in BI emphasizes both point-forecast accuracy and cross-target coherence, anticipating later sections that detail metrics for each.

Accordingly, the next chapters formalize the multi-output problem, contrast independent versus joint approaches, and motivate our chosen strategy for modeling dependencies in BI settings, setting up the methodology and evaluation that follow.

1.2 Research Problem

Many Businesses Intelligence (BI) workflows still build separate predictive models for each Key Performance Indicator (KPI), implicitly assuming the targets are independent. However, this single-output modeling paradigm ignores any covariance or interactions among the targets. As a result, independent per-KPI models cannot exploit information shared between related targets, and they may even yield contradictory predictions, for example, forecasting an increase in one metric that historically correlates with a decrease in another, without recognizing the inconsistency. Therefore, modeling each output in

isolation is often suboptimal when the targets are correlated or logically related in a BI context.

To address these limitations, existing multi-output learning approaches offer several directions to address this limitation. Prior work has explored: (i) problem-transformation methods that treat other targets as features in extended input space such as Stacked Single-Target models or regressor chains, (ii) output-coding techniques that project multiple targets onto random linear combinations (predicting coded outputs that are later decoded back, sometimes called Random Linear Combination), (iii) algorithm adaptation where learning algorithms are modified to predict multiple outputs jointly, for instance, multi-output extensions of Random Forests (iv) meta-learning or stacking frameworks that add a second-layer model to combine or correct initial predictions, (like Bayesian multi-output regressor stacking approach), and (v) classical joint modeling such as Seemingly Unrelated Regression (SUR) for simultaneously estimating correlated linear equations. These diverse approaches demonstrate that leveraging inter-target dependencies can indeed improve generalization compared to modeling each KPI independently.

Nevertheless, most of them require non-trivial changes to the training or inference pipeline and add operational complexity. For example, special model architecture, custom loss functions, or additional monitoring and maintenance overhead. There is a relevant hole in the BI space: we don't have an easy way to maintain the ease of per-target modeling workflows while also improving the joint coherence of the outputs. To build on this idea, one can also extend this to multi-output prediction with little additional burden, to help capture cross-target inconsistencies, while still relying on pre-target ensembles.

This clearly points to a gap in the literature for a method that is designed for the practical limitations of BI systems. Thus, we are in need of an approach that satisfies at least the following three desiderata:

- **Enhance accuracy and integrity:** Strive for improved point predictions on each KPI individually, while providing integrity across predictions so that no output contradicts another prediction.

- **Be computationally light:** We want limited computational load and hyper-parameters to allow workability with current systems without significant changes or resource expenditure.
- **Integrate with current workflows:** Integrating your adjustments to standard workflows related to per-target modeling while keeping within the current explainability method and monitoring system.
- **Avoid information leakage:** Manage and minimize target leakage and error propagation arising from excessive reliance on naively chaining outcomes while supervising any sources of relation to other targets.

In short, there remains a clear need for a multi-output learning approach tailored to BI that improves cross-target consistency while preserving simplicity, efficiency, and transparency.

1.3 Significance of Study

This study is significant both in academic and practical terms. Academically, it contributes to the growing body of work on multi-output machine learning by focusing on an area that has been relatively under-explored: simple ensemble-based solutions that can be easily applied in business forecasting scenarios. Traditional research on multi-output regression often proposes complex models or theoretical frameworks, but there is a pressing need for approaches that reconcile advanced performance with practical deploy ability. Our work answers recent calls for research into when and how a holistic multivariate approach should be preferred over separate models (Schmid et al., 2022) . By developing a method that enhances multi-target prediction without drastically increasing complexity, we aim to bridge the gap between sophisticated multi-output techniques and the realities of BI practice. The proposed solution, Residual-Polished Random Forests (RPRF), is designed to be a novel yet pragmatic ensemble technique that builds well-established algorithms (Random Forests) and augments them to handle multiple outputs coherently. In doing so, the thesis extends ensemble learning theory into the multi-target domain with an emphasis on lightweight model stacking. The significance lies in demonstrating that one can attain a many of the advantages of joint modeling (greater accuracy, greater consistency) without having to use particularly specialized or elusive models. If successful, this research will surface a blueprint for inserting dependency-aware modeling into BI pipelines with the fewest frictions. It also

produces empirical insights on the conditions for greatest benefits from multi-output learning, enriching theoretical literature on multi-target regression (for example, confirming the hypothesized relationship between inter-target correlation strength and prediction benefit).

In addition to the methodological contribution, the study's significance comes from its validation approach, which includes both synthetic data and data from real-world datasets. By using controlled synthetic experiments, the study will contribute to the scientific body of knowledge by isolating the influence that factors such as correlation structure, number outputs, and sample size have on multi-output performance questions of interest to the broader machine learning community that is exploring multi-task or multi-target learning (Schmid et al., 2022) . Specifically, by applying the approach to real BI-related datasets to predict outcomes such as energy efficiency and pollutant levels, the study will add evidence about its practical value and generality. For example, if RPRF improves predictions of energy loads of buildings, or levels of environmental pollutants, beyond models used in the past, it demonstrates a real application of the technique in areas where decision-makers cannot decide on inputs without simultaneous predictions of more than one outcome.

Practically, the significance of this research is tied to the ever-increasing importance of data-driven decision-making in organizations. Businesses today accumulate and examine considerable amounts of data while BI systems are developed to provide actionable insights derived from data. A central premise of BI is the implementation of predictive analytic models to allow organizations to forecast conditions affecting a variety of operations, having to do with sales, demand, costs, and other KPI metrics to aid strategic positioning and enhancement. More precise and consistent forecasting predictively will reap benefits valuable to businesses directly. For instance, if a company could rely on more accurate joint forecasting of both actual revenue and expenses, it could make better decisions regarding budget; furthermore, if a retailer could reliably forecast inventory levels and possible inventory stock delays due to supply chain issues, that retailer could build an effective stock management strategy to avoid costly excess inventory or unnecessary stockouts. Meaningful multi-KPI coherence in the same forecasts would help organizations avoid some aspects of dysfunctional behavior; i.e., a sales function believing a sales increase was possible without a realistic forecast of how that improves

customer service demand or any consequences that may carry into supply requirements, then the organization clearly is not planning using a coherent future picture.

Additionally, shifting the multi-output prediction paradigm to a perception that is both user-friendly and interpretable by industry practitioners enhances the chances of adoption. The primary contribution of this research is to create a product that BI teams can easily use without having to obtain a PhD in machine learning knowledge to understand or use it. As noted in business analytics academic literature, companies will more readily embrace AI/ML solutions into their work processes if it is easily integrated into existing processes and the outcome of the model can be explained to other stakeholders. As an advancement of Random Forests, an algorithm that is universally recognized, and a trusted algorithm in the industry, and includes a component for correcting and improving the previously predicted values for both the random forest and for the potentially demonstrable features, the RPRF methodology is also simply interpretable: where the primary base predictions are coming from standard models (and the feature importance measures for it) and the analysis of any patterns derived from the remaining residual is for the adjustments and improvements. This also contributes back to the research of becoming more accurate in the overall method of forecasting, but one that managers and analysts can believe in and verify, particularly for a deployment in a real decision-making environment.

In conclusion, the importance of the research is twofold: enhancing academic knowledge around multi-output ensemble learning and providing a practical contribution for BI applications wherein multi-metrical predictions are necessary and recurring. It is responsive to a recognized need for greater BI forecasting ability in a big data, fast-paced business environment, in which leveraging relationships among multiple indicators can help drive timelier, better-informed decisions. Ultimately, by providing evidence around the efficacy/dependency aware forecasting approach to improve business intelligence and outcomes (not leaving business intelligence tools overly complicated), this research could influence organizations' predictive analytics workflow for years to come.

1.4 Industry Relevance and Practical Implications

The study is important to industry, especially for organizations that utilize Business Intelligence systems as part of decision-making. In practice, BI dashboards and reports, frequently show multiple KPIs side by side, executives consume dashboards of metrics

to get an overall impression of the health of the business. However, the forecasts for each of those metrics (KPIs) are often computed in isolation. Our work offers a common pain point: a lack of coherence in predictive modeling for related business metrics. By using a method to generate coherent multi-output predictions, we try to align the work of analytics to the integrated way in which decision-makers consume multi-dimensional performance data.

One practical benefit is better quality of decisions. Inconsistent predictions can result in suboptimal or even damaging decisions. Imagine that a retailer has a sales forecasting model predicting a significant bump in demand for a product, and an additional model predicting how long it will take to have products back in-stock (the backlog or supply model in the supply chain), does not account for the demand surge remaining unchanged. If management relied only on the sales forecasting model (e.g., they were planning a promotional budget or even planning to have products at a certain price) then the result could easily be stock-outs, ultimately resulting in lost revenue. If a multi-output model was constructed so the sales forecast and the supply chain forecast were mutually consistent, then management would not risk having two models in misalignment. In general, a multi-output model demonstrates the coupling between the demand metrics and the supply metrics, thus, when changes in forecast (yet consistent) demand is modeled (to predict a bump), the inventory prediction would also be adjusted upward in the multi-output model simultaneously. In this case, management would quickly learn (as a result of the generated model) that the predicted sales and inventory supply were changing. Once again, the simply modeling of demand/supply together, then becomes a decision that can better consider the coupling of likely future scenarios without the potential for some internal contradiction, potentially enabling more strategic planning around the concepts of risk and disruptions.

Another consideration exists with respect to resource allocation and cross-division planning. Many organizations will take forecasts like these into account when developing a budget and allocating resources across various divisions of the organization. For example, if the forecasts signal higher customer orders, that could lead to customer support hiring, manufacturing output increases, and more raw material spending. However, if each of those forecasts (orders, support tickets, production, procurement costs) are forecasted across divisions separately, there is a chance that they will not align, particularly the customer support model will not "know" that the sales model output

anticipates a spike in support tickets. With multi-output prediction, each of the division level forecasts could be created with an understanding of the others, which could lead to better aligned plans. In essence, our approach can enhance the coordination of planning activities by providing a single multi-dimensional prediction that departments can jointly reference.

From an industry adoption perspective, a major advantage of the proposed method is the ease of implementation. Companies frequently have pipelines where data is cleaned, features are engineered, and then models are run independently on each target metric. The RPRF framework allows for these pipelines to be annexed or paired with minor changes: one can still fit the standard 'per target' models (e.g., Random Forest), and then all that is required is an additional layer of residual-polishing. Thus, the idea is that organizations do not have to overhaul analytics infrastructure or re-train personnel on an entirely new tool, they can retain the forms of analysis, algorithms, and/or software they used prior (e.g., a traditional Random Forest fitted independently using scikit-learn or other multi-output wrappers in an ML library), and simply have RPRF act as a layer on top as an additional way to bring greater value beyond training the individual per-target models. This could facilitate implementation as the barrier to implementation is lower, indicating there is less upfront work involved, which is often appealing for industry projects that need to move quickly and often have resource constraints.

Additionally, interpretability and maintainability of the solution are a consideration in industry. High complexity AI models do not gain a wide foothold in business settings because of the "black box" aspect of these models. On the other hand, our approach integrates a high degree of interpretability: the primary drivers for each target are all still obtained from the independent Random Forest models that are well known and provide metrics such as feature importance. The residual adjustment can be interpreted as making small corrections to ensure consistency. This means analysts can explain to stakeholders not only the prediction for each KPI but also why the set of predictions makes sense together (since metric A is predicted to increase, our model made a small upward adjustment to metric B’s forecast to reflect their historical linkage). Such clarity builds trust in analytics and encourages use of the forecasts in decision-making. Maintainability is enhanced by the fact that each component (the base models and the residual model) can be updated or retrained independently if needed, without retraining a giant monolithic model for all outputs. This modularity is very practical, for instance, if a certain KPI

changes definition or a new data source becomes available for one target, one can update that target's model and the polishing step will adjust accordingly, rather than needing to retrain everything from scratch.

We are beginning to see industry movement towards more integrated analytics as it is related to manufacturing, finance, and marketing. For example, integrated business planning software aims to integrate forecasts across supply, demand, and finance. Our research provides support for this trend within industry through a specific machine learning method for generating integrated forecasts. By showcasing improvements on real datasets (i.e., energy efficiency and VOC pollution data), we provide a signal to practitioners in various industry sectors to show that multi-output approaches are not just academic niceties, but ready for implementation in practical or real-world settings. The case for predicting environmental pollutants is illustrative, where multi-output models were able, for example, to monitor multiple pollutants simultaneously and provide early-warning that could not have been achieved with single-output models. Likewise, businesses could benefit from multi-output predictions for early warning systems for KPI outliers' situations where deviations across a few metrics indicate a looming problem even when each individually is in an acceptable range.

An important implication for practice deals with the scalability of insight. A multi-output perspective naturally frames the system of metrics explaining performance variances, embracing a more holistic and integrated approach to thinking about performance. Rather than managing the human behavior of KPI forecasts in isolation, organizations may evolve toward a more integrated performance management practice over time. In this way, our work is not only a technical tool in organizations, but also encourages a structure shift in mindset: understanding the interplay of metrics is important. Specifically, modern BI takes the aim of revealing hidden patterns and correlations towards improving decisions, and our method extends leverage to that by ensuring that the correlations that do exist historically now exist to underpin forward looking predictions. The ability to connect prediction to prior correlations makes insights from BI seem more credible and actionable for organizations, and this ultimately helps organizations create and execute strategies based on predicted change in a faster and more appropriate manner. Firms that have the advanced forecasting will have a competitive advantage because they are better organized with aligned strategies in the face of the same change than simply having to

reconcile and decide what forecasts to value from conflicting sources, such as a marketing, sales, operations or finance department.

To sum up, it is clear that the research has industry relevance in terms of improving both the reproducibility and purposefulness of BI forecasting, improving forecast planning across functions, and doing it in a way that is practical meaning that companies can use and trust it. The design philosophy of the method augmentation, not disruption to the workflow, is intended for real-world use. This work offers a pathway to more coherent and accurate multi-metric predictions, the research has the opportunity to make BI systems more intelligent and practically decision supportive.

1.5 Research Objectives

The goal of this research is to develop and validate a new lightweight ensemble-based method for multi-output prediction and to assess its performance in comparison to established machine learning models, specifically per-target baselines commonly used in practice.

- **Objective 1: Develop and validate the method.**
  Formulating the algorithm and its computational footprint, implement it, and evaluate it on synthetic and real multi-output settings. Following, determine when exploiting inter-target structure improves point-estimate accuracy and coherence across targets**.**

- **Objective 2: Benchmark against strong per-target baselines.**
  with established baselines. We will rigorously evaluate RPRF against leading ensemble methods commonly used for regression: namely, standard Random Forests, and popular gradient boosting methods like XGBoost.

Compare performance with established baselines. We will rigorously evaluate the suggested method against leading ensemble methods commonly used for regression: namely, standard per-target Random Forests, and popular gradient boosting methods like per-target XGBoost using the same data, preprocessing, and splits, focusing on per-target and macro-averaged RMSE/MAE and on joint coherence of KPI forecasts. We treat these per-target models as the production baselines in BI. Success is defined as: (i) reliable improvement over these baselines when targets are moderately to strongly correlated, and (ii) parity when dependencies are weak.

By achieving these objectives, the research will result in a validated new method for multi-output BI prediction and a clear understanding of its advantages and any trade-offs relative to existing approaches. Essentially, Objective 1 is about building the solution and proving it works as intended, while Objective 2 is about positioning the solution in context and quantifying its value over the status quo.

1.6 Research Questions and Hypotheses

We investigate three questions with corresponding hypotheses. The primary endpoint is macro-averaged RMSE (mean RMSE across targets and CV folds), and per-target RMSE/MAE are secondary endpoints. The specific research questions (RQs) and hypotheses (H) are formulated as:

RQ1. Does the proposed approach reduce prediction error compared with independent per-target Random Forest when targets are correlated?
H1. The approach yields lower macro-averaged RMSE than per-target RF on datasets/scenarios with moderate-to-strong inter-target correlations; when correlations are weak, performance is no worse than per-target RF. This hypothesis is based on the expectation that proposed approach can capture cross-target patterns that independent models miss, leading to improved accuracy.

RQ2. How does the approach compare with a strong per-target baseline and state-of-the-art boosting algorithms such as XGBoost on multi-output prediction tasks?
H2. The approach is on par with, or better than, per-target XGBoost in macro-averaged RMSE when targets are moderately to strongly correlated, and non-inferior when correlations are weak.

RQ3. How does the strength of inter-target correlation, number of variables (dimensions), and sample size affect potential gains?
H3. Improvements to over-target baselines will change, for example it will increase with correlation strength so higher accuracy under moderate-to-strong dependence, and no improvement under near-independence.

These questions cover both the core efficacy of the proposed approach (RQ1, RQ2) and its behavior under varying conditions (RQ3). The hypotheses will be examined through experiments described in later chapters.

1.7 Scope and Limitations

This thesis focuses on the problem of multi-output regression within the context of Business Intelligence applications. Specifically, it develops and evaluates the Residual-Polished Random Forests (RPRF) approach for predicting multiple continuous outcome variables together. The defined scope indicates that we will be conducting regression tasks (as opposed to classification tasks) and that the methods and evaluations specific to regression tasks will be applied (e.g., the error metrics of RMSE / MAE). The study uses synthetic datasets and real-world datasets. Synthetic datasets are part of the scope of the study, as they allow us to control factors such as the number of targets, how correlated targets are with one another, sample size, and noise levels. By varying those factors, the study can determine cause-and-effect relationships and test H3 directly regarding how those factors may affect accuracy. For instance, if we construct a synthetic scenario and specify that outputs are correlated (e.g., we specify a covariance matrix for the target variables), we can increase or decrease correlation, and measure and compare how well RPRF and the baseline models do. All this experimentation is with the scope of the study to stress test RPRF under idealized circumstances and build an understanding of how RPRF behaves.

The real-world part of the scope consists of a couple of datasets that represent typical multi-output prediction situations of interest to BI: (1) the UCI Energy Efficiency dataset, which involves predicting two correlated targets (heating load and cooling load of buildings) at the same time, from features related to the building's architecture, which is a canonical multi-target regression problem in building energy management (by extension, relevant to operational analytics in facilities management). (2) A volatile organic (VOC) compound exposure dataset, where the tasks are predicting multiple correlated concentrations of pollutants based on readings from sensors and potentially other features (for example, demographics or environmental factors). The second dataset examines the generality of RPRF beyond a strictly “business” context as it is more environmental informatics, however, at its core it is still multi-output regression and therefore the methodology still fits within the scope of the research objectives. In both scenarios, equivalent data preprocessing and evaluation protocols are employed for RPRF and for the benchmark models (per-target Random Forests, per-target XGBoost) to ensure comparability.

The evaluation focuses on point-estimate accuracy of the predictions. We look at per-target error metrics (such as RMSE and MAE for each individual output) and macro-averaged metrics (which give an overall error across all outputs for a given method). We also evaluate coherence qualitatively and quantitatively, for instance, checking if the correlation matrix of predictions from a model matches the correlation structure in the data, and whether RPRF's predictions avoid contradictions that independent models might produce.

Limitations: Several important limitations define what this thesis does not cover:

Limitation 1: Focus on continuous regression targets only. We limit our study to continuous-valued outputs. The approach and findings may not directly extend to classification tasks or mixed discrete/continuous outputs. Multi-output classification (multi-label classification) and structured outputs like sequences or graphs are outside our scope. The decision to focus on regression was made to keep the problem manageable and because many BI metrics (revenue, cost, sales volume, etc.) are naturally continuous. Nevertheless, this means the work will not address any special considerations needed for categorical targets or probabilistic classification calibration.

Limitation 2: Modest number of target variables. Our experiments consider cases from a few up to perhaps a few dozen target variables. We do not explore extremely high-dimensional output spaces (hundreds of outputs). In scenarios with a very large number of outputs, different techniques (such as target clustering or dimensionality reduction in target space) might be necessary. RPRF in its current form might face challenges if applied to high-dimensional outputs, for example, the residual adjustment models could become unwieldy or overfit when there are too many outputs to consider. Thus, the thesis does not claim generality to all possible scales of multi-output problems; it is more aligned with typical BI situations where one might predict, say, 5, 10 KPIs together, not hundreds.

Limitation 3: Out-of-scope aspects of deployment and extended modeling. We do not address certain practical deployment concerns such as real-time prediction latency, the engineering of large-scale deployment (how to handle streaming data or very big data in production), or issues of data privacy and fairness in multi-output models. Those are important in a broader sense but would distract from the core methodological focus here.

Similarly, we do not cover full probabilistic uncertainty quantification of the joint predictions, while we discuss mean accuracy, we do not provide prediction intervals or joint confidence regions for the multi-output forecasts. Incorporating uncertainty in a multi-output context (e.g., via Bayesian methods or quantile regression) is left for future work.

Within these boundaries, the thesis remains focused on the central research question: Can a lightweight, dependency-aware upgrade to per-target BI forecasting workflows deliver measurable gains in prediction accuracy and produce coherent multi-KPI forecasts under realistic conditions? We ensure that the work is sufficiently tractable by defining a scope and recognizing these limitations which allow us to only derive conclusions from a viable workspace where evidence exists. Future work can build on this work to account for some of these omitted aspects, like scaling to more than one output or adding probabilistic forecasting capabilities.

1.8 Thesis Organization

This thesis is organized into five chapters, following a traditional structure that takes the reader from motivation through to results and conclusions:

Chapter 1: Introduction. (The existing chapter) It provides the setting of BI forecasting and the need for multi-output forecasting, articulates the research problem and significance of the research, defines the aims and research questions, outlines the scope of the study, the limitations of the research, and the overall approach of the study. Chapter 1 essentially gives the what and why of the research, providing the justification for everything that follows and a roadmap for the dissertation.

Chapter 2: Literature Review. This chapter surveys related work and theoretical foundations for our research. We review the existing multi-output learning approaches in detail – covering problem-transformation methods (like stacked single-target models and regressor chains), output-coding techniques (e.g., random linear combinations of targets and dimensionality reduction methods), and algorithm adaptations (such as multi-output decision trees and Random Forest variants). We also discuss ensemble learning concepts (bagging vs. boosting) as they pertain to our approach and examine meta-learning techniques like stacking in multi-target settings. In addition, Chapter 2 situates our research in the context of previous research literature: it highlights the gap in the literature

to be filled by RPRF (building off the discussion of Chapter 1) and also describes how similar problems have been solved in fields other than BI (i.e., multi-output models in environmental informatics, healthcare, etc. reinforce the applicability across fields). By the conclusion of the literature review, the reader will have a general sense of what has been done, the state-of-the-art in multi-output prediction, and how our approach differs or innovates from those.

Chapter 3: Method. This chapter describes the recommended RPRF approach, the experimental methods we will use to evaluate RPRF. First, we formally define each of the components of the RPRF algorithm including any mathematical formulation and pseudocode for training and prediction. We also analyze the computational complexity of the method to satisfy the "lightweight" criterion. We then define how we conduct the experiments, representing how we generate the synthetic data (e.g., how to simulate correlated targets and add noise), and discuss the characteristics of real datasets (and any pre-processing such as normalization or feature selection). We will describe the evaluation protocol including the train/test splits or cross-validation strategy, the expected metrics (RMSE, MAE, and other significance testing for differences in performance), and the approach for analyzing the cohesive merit of multi-output predictions. Lastly, we will provide information on how the implementation of the program will be performed (such as which software library we will be using, hyperparameter optimization, and reproducibility). Overall, the chapter covers the how of study by providing enough information that another researcher would be able to replicate the experiments, or the RPRF approach, in the same way.

Chapter 4: Results and Analysis. This chapter contains empirical findings. Results are normally organized by data set and by research questions. We start with synthetic data results, highlighting RPRF and baseline models under varying controlled situations. These results directly pertain to RQ3, for example, with plots of performance as a function of correlation strength (or tables of errors by the number of targets). We analyze these results to see if they support hypothesis H3 (e.g., do we see the expected pattern of improvement under high correlation and parity under independence?). Next, we present real-world results. For the UCI Energy Efficiency dataset, we might show that RPRF reduced the error in predicting heating and cooling loads compared to independent Random Forests and XGBoost, and we might include a figure of actual vs. predicted values to visualize coherence. For the VOC exposure dataset, we report how well RPRF

predicted the suite of pollutant concentrations relative to baselines, perhaps noting pollutants where joint modeling helped. We include both numerical evaluation (error metrics) and, were useful, visualizations such as error distribution plots or correlation plots of predictions.

Chapter 5: Discussion and Conclusion. The last chapter of the thesis delivers an overview of the findings and discusses what we can take from them. Again, we revisit each research question and talk about how well it was answered. For example, we can discuss RQ1 and RQ2 results, perhaps mentioning RPRF consistently outperformed the independent models in correlated settings (which would show it is worth trying for BI scenarios with metrics which move together) and that the results were comparable (or better) than XGBoost when there were correlations (there is some justification of multi-output). We also think about unexpected results, for instance, if RPRF did not perform as well in one of the scenarios we can think about the why.

Overall, the thesis is designed to follow the trajectory of motivation and context (why they matter, what has been done, and, if appropriately, where this is all going), methods and experimentation (how we find a way to study it, and what we found), and higher-level interpretation (what this means for us, or where this gets us). This structure will support the emergent narrative for multi output prediction in a BI context by moving readers from the definition of the initial problems, through a demonstration of solutions, to the implications of the solutions and relevant next steps. Structuring the document in this fashion allows each chapter to build incrementally on the last and create a sequencing of knowledge that both guides and supports readers to build their understanding of the research topic and its implications.

Chapter Two
The Literature Review

Multi-output prediction, also known by multi-target regression, studies models that simultaneously predict several dependent variables. Unlike traditional single-output modeling, multi-output approaches explore relationships among targets to improve accuracy and yield more coherent predictions across outcomes (Madjarov et al., 2012). This chapter reviews prior work with an emphasis on multi-output regression for BI contexts. In day-to-day BI deployments, the production baseline is typically independent, per-target tree ensembles such as a separate Random Forest or gradient-boosting model for each KPI, is simple to govern, monitor, and explain. Against that reference architecture, we surveyed five families that either augment or replace per-target modeling to exploit inter-target dependencies: problem-transformation methods, output-coding techniques, algorithm adaptation, meta-learning/stacking frameworks, and classical joint modeling. Our focus is how each family addresses the complexities of multi-output prediction and what trade-offs it introduces for BI pipelines.

2.1 Problem-Transformation Methods

Problem-transformation methods convert a multi-output task into one or more single-output tasks so that established learners can be reused (Borchani et al., 2015a) . Two representative approaches are Stacked Single-Target (SST), also called multi-target regressor stacking (MTRS), and Regressor Chains (RC).

Concerning Stacked Single-Target (SST/MTRS), separate single-output models are trained per target, but predictions of other targets are used as features for subsequent models (Montesinos-López et al., 2019) .For target Y_k, the model uses X plus predictions of $Y_1,\ldots,Y_{k-1}$. This design aims to capture inter-target correlations without abandoning the per-target workflow. Ordering matters, however, and performance can be sensitive to the sequence in which targets are modeled, where ensembles over multiple orders help mitigate this (Read et al., 2011) .In practice, to avoid leakage and optimistic bias, the added target features should be out-of-fold (OOF) predictions generated within cross-validation.

Regarding Regressor Chains (RC), it extends the idea by explicitly modeling conditional dependencies along a chain. Each regressor uses X and predictions of all preceding

targets(Read et al., 2011), often yielding improved accuracy(Senge et al., 2013). The main risk is error propagation, where mistakes early in the chain can cascade. Additionally, Ensemble of Regressor Chains (ERC) averages across chains with different random target orders to reduce variance(Senge et al., 2013). As with SST, OOF predictions should be used during training to maintain clean separation between training and meta-features.

Generally, SST and RC can deliver gains while preserving per-KPI models, but they add pipeline complexity, such as managing target-as-feature flows, OOF generation, and order selection, which practitioners must operationalize (Spyromitros-Xioufis et al., 2016b) .

2.2 Output-Coding Techniques

Output-coding techniques transform the target space into a coded space, train standard single-output learners there, and decode predictions back to the original targets (Kong & Dietterich, 1995). They are attractive when the target dimension is large or when relationships can be captured by a compact code.

Random Linear Combinations (RLC) forms pseudo-targets as random linear combinations of the original targets (Tsoumakas et al., 2014) . Multiple combinations are learned, and original target predictions are recovered by linear recombination. RLC can efficiently capture complex relations and often scales well as the number of targets grows. In practice, decoding requires solving a linear system, where care with conditioning, like coefficient normalization, or using more combinations than targets, improves stability. A known trade-off is reduced interpretability at the target level, which BI stakeholders should weigh against accuracy gains.

2.3 Algorithm Adaptation

Algorithm-adaptation methods modify a learner, so it natively handles vector outputs, exploiting dependencies during model construction(Kocev et al., 2013a) . In Multivariate Trees and Multi-Output Random Forests, each split minimizes an impurity aggregated across all targets, like sum of squared errors, so that splits reflect joint structure (Breiman, 2001; Segal & Xiao, 2011) . Predictive Clustering Trees (PCTs) operationalize this paradigm and can be aggregated into ensembles for structured outputs (Kocev et al., 2020) Compared with training m separate models, joint trees reuse splits across targets and can be computationally favorable when m is moderate to large (Breskvar et al., 2018) Many

toolkits, such as scikit-learn's Random Forest Regressor, support multi-output by summing impurities across dimensions. So, when targets have very different scales, standardizing or weighting targets helps avoid dominance by high-variance targets(Schmid et al., 2022). Algorithm adaptation keeps tree-based explainability tools familiar to BI users, but it replaces the per-target estimator with a single joint model, which may affect governance and monitoring practices that assume one model per KPI.

2.4 Meta-Learning or Stacking Frameworks

Stacking introduces a second-layer meta-model that learns from base predictions to combine strengths and correct systematic errors (Breiman, 1996). In multi-output contexts, base learners are frequently per-target models, stacking a post-hoc augmentation of the prevailing BI baseline. To avoid leakage, meta-models should be trained on out-of-fold base predictions.

Bayesian Multi-Output Regressor Stacking (BMORS) places stacking within a Bayesian framework that can encode prior structure and share information across outputs while quantifying uncertainty (Montesinos-López et al., 2019) This can be powerful when relationships between base predictions and targets are complex. The trade-off is modeling and inference overhead that some operational pipelines may find heavy. An advantage of such models that stacking aligns well with "keep the base models" practice and can add a coherence layer with limited disruption, provided OOF protocols and meta-learner choices are handled carefully.

2.5 Classical joint modeling approaches

These are classical econometric approaches jointly estimate multiple equations when error terms are contemporaneously correlated. As an example, Seemingly Unrelated Regression (SUR) (Zellner, 1962)estimates linear regressions jointly via GLS to gain efficiency over separate OLS when cross-equation error correlations exist, with established theory and practice (Srivastava & Giles, 1987) . SUR is valuable when linear structure is appropriate; however, nonlinearity and heteroscedasticity common in modern BI can limit its predictive advantages relative to flexible ensembles.

Across these five families, the literature shows that exploiting inter-target dependencies can reduce error relative to independent per-target models (Borchani et al., 2015a; Read et al., 2011; Spyromitros-Xioufis et al., 2016a) . For BI domain, the most practical

methods keep the per-KPI workflow and are easy to explain. Problem-transformation (SST/RC) and stacking sit on top of the baseline with modest extra work, so long as we use out-of-fold predictions and handle chain order/ensembles carefully. Output-coding and algorithm-adaptation learn a single joint model but can reduce interpretability or complicate governance. SUR is a useful linear benchmark when its assumptions are held. The next chapter formalizes the multi-output problem in BI terms, specifies leakage-safe validation, and sets up the empirical evaluation against strong per-target baselines.

Chapter Three

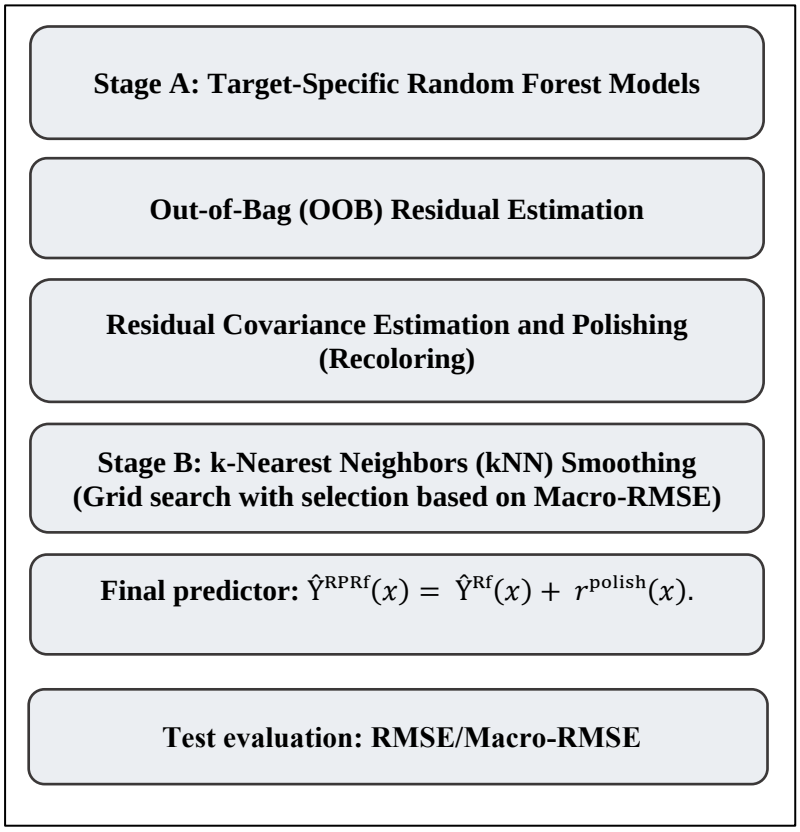
RPRF's Methodology and Experimental Settings

3.1 Overview

This chapter formalizes the multi-output regression problem studied in this thesis, details the proposed Residual-Polished Random Forests (RPRF) method, and describes the experimental protocol. RPRF addresses multi-output prediction with continuous targets, a common setting in Business Intelligence (BI) where several key performance indicators (KPIs) must before cast jointly. The approach proceeds in two stages: (i) one Random Forest per target (marginal models) producing out-of-bag (OOB) residuals, and (ii) a post-hoc residual correction that leverages cross-target error structure through local neighbor averaging and a covariance aware linear adjustment. Baselines, datasets, evaluation metrics, computational aspects, reproducibility, and limitations are documented in the following sections.

Figure 1

The major steps of RPRF



3.2 Problem formulation and notation

Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ denote n observed pairs with $x_i \in \mathbb{R}^p$ the feature vector and $y_i = (y_{i1}, \dots, y_{iq})^\top \in \mathbb{R}^q$ a vector of continuous targets. The objective is to learn a predictor $\hat{f}: \mathbb{R}^p \to \mathbb{R}^q$ that yields accurate per-target forecasts while preserving coherence across targets when dependence is present. On the training partition, per-target Random Forests (as presented in Section 5) produce out-of-bag predictions $\hat{y}_{ij}^{OOB}$ and residuals

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

which are stacked row-wise as

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

The covariance matrix *S* summarizes empirical residual co-movements across targets that remain after marginal modeling.

3.3 Residual-Polished Random Forests (RPRF)

RPRF is a two-stage procedure:

1. **Stage A: marginal learning.** Fit q Random Forest (RF) regressors $f_1, \dots, f_q$, one per target, and compute leakage-safe OOB residuals R_{train} as in equations (1)-(2).
2. **Stage B: residual polishing.** For a query x, form a local mean residual vector by averaging residuals of its *k* nearest training neighbors in feature space, map this local residual through $S^{1/2}$ and add it to the Stage-A predictions to obtain the final multi-output forecast.

The final predictor retains target-specific structure learned by the per-target RFs while explicitly correcting cross-target dependencies that remain in the residuals.

3.4 Data splitting and preprocessing

Each dataset is partitioned into 60% training, 15% validation, and 25% test using a fixed random seed. The validation partition is used exclusively for hyperparameter selection, and the test partition remains untouched until final reporting. Nearest-neighbor searches are performed in the original feature space based on Euclidean distance. When features have heterogeneous scales, feature standardization prior to neighbor search is a recommended robustness enhancement. In this work, either features are generated with

comparable scales as in the synthetic scenarios or are on interpretable and commensurate scales.

3.5 Stage A: independent per-target Random Forests

For each target Y_j $(j = 1, \dots, q)$, a Random Forest regressor $f_i: \mathbb{R}^{\boldsymbol{p}} \to \mathbb{R}$ is trained on the training partition. Random Forests combine bootstrap sampling with randomized feature selection and provide strong accuracy with limited tuning on tabular data. Hyperparameters follow robust defaults, like tree count increasing moderately with n and $\sqrt{p}, mtry \approx \sqrt{p}$, a small terminal node size proportional to n_{train}, yielding competitive baselines without extensive search.

3.6 Out-of-bag residuals

The trained RF produces out-of-bag (OOB) predictions for training instances by aggregating only trees that did not include the instance in their bootstrap sample. The residual matrix R_{train} and covariance S in equation (2) are then computed and used in Stage B. Because OOB predictions do not use the instance for which they are computed, residuals are free of resubstitution leakage. Using OOB predictions makes the residuals leakage–safe: each training case is predicted only by trees that did not see it, yielding near–cross-validated errors. This avoids resubstitution bias that would artificially shrink residuals. In turn, both the empirical residual covariance *S* and the neighborhood averages used in Stage B are calibrated to the true generalization errors rather than to in-sample fits.

3.7 Stage B: residual polishing

Compute the empirical covariance $S = \text{cov}(R_{\text{train}})$. Let $S = U\Lambda U^{\text{T}}$ be its eigende composition, with nonnegative eigenvalues on the diagonal of $\Lambda, where\ U \in \mathbb{R}^{q \times q}$ contains the orthonormal eigenvectors of S, $\Lambda = \text{diag}(\lambda^1, \dots, \lambda_q)$ stacks the (non negative) eigenvalues, and $(\cdot)^{\text{T}}$ denotes transpose. Then we define

$$S^{1/2} = U\Lambda^{1/2}\, U^{\top}, \tag{3}$$

with small eigenvalues floored to a numerical constant for stability. The transform $S^{1/2}$ encodes cross-target residual co-movements. Intuitively, *S* summarizes how target errors co-move after marginal modeling, and $S^{1/2}$ transfers this dependence structure to the local correction.

3.8 Local residual prototype and covariance-aware correction

For a query x, let $\mathcal{N}_k(x)$ denote the indices of its k nearest training points in feature space based on Euclidean distance. The uniform local mean residual is

$$\bar{r}(x) = \frac{1}{k} \sum_{i \in N_k(x)} r_i \in \mathbb{R}^q. \tag{4}$$

Here, averaging reduces variance in the residual correction while preserving local structure. The covariance-aware polishing vector and the final multi-output prediction are

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

Where $\hat{Y}^{\text{RF}}(x) = (f_1(x), \dots, f_q(x))^{\top}$ stacks per-target RF predictions. Multiplication by $S^{1/2}$ aligns the residual correction with empirically observed cross-target error covariance, where targets that historically err together are adjusted together.

In other words, multiplying by $S^{1/2}$ recolors the local mean residual so that the correction inherits the empirically observed cross-target residual covariance, so targets that historically err together are adjusted together, whereas independent targets are adjusted more independently. The final predictor then adds this estimated residual to the Stage-A forecast, consistent with the decomposition $y = \hat{Y}^{\text{RF}}(x) + \varepsilon(x)$. Thus, Stage B supplies a coherence-aware residual estimate while preserving the strengths of the per-target forests.

3.9 Model selection

RPRF introduces a single hyperparameter *k* represents the number of neighbors.
On the validation split, a monotone grid is explored:

$$k \in \left\{ max\left(5, \left\lfloor \sqrt{n_{train}} \right\rfloor\right), \dots, min\left(200, n_{train} - 1\right) \right\},$$

and the value minimizing macro-averaged RMSE is selected. Let

$$\text{RMSE}_j = \sqrt{\frac{1}{N_{\text{val}}} \sum_i \left(y_{ij} - \hat{y}_{ij}\right)^2}, \qquad \text{MacroRMSE} = \frac{1}{q} \sum_{j=1}^{q} \text{RMSE}_j. \tag{6}$$

Macro-averaging enforces balanced performance across targets. Residuals use OOB predictions, and the test set remains untouched until final evaluation.

3.10 Baselines

Two competitive references are used under identical splits and metrics:

- **Per-target RF:** which is simply Stage-A predictions $\hat{Y}^{\mathrm{RF}}$ without polishing.
- **XGBoost for a single model with target indicator:** which represents a single gradient-boosted regressor trained on replicated rows with a one-hot target identifier and squared-error loss with validation-based early stopping.

These baselines reflect common practice in multi-output tabular tasks and isolate the incremental effect of residual polishing.

3.11 Datasets

3.11.1 Synthetic scenarios

Features are generated i.i.d. with zero mean and unit variance. Targets combine a shared linear core with mild target-specific nonlinear perturbations, plus additive Gaussian noise $\mathcal{N}(0, \Sigma_{\mathrm{noise}})$ to control residual correlation as: weak, medium, or strong. We vary:

- sample size n (≈1.2k to 10k),
- feature count p (5-50),
- number of targets q (2-7),
- correlation patterns for Σ_{noise}. For example AR (1), block-diagonal, low rank.

These variations and scenarios assess scaling, the effect of dimensionality on neighbor search, and the strength-pattern of dependence on the mechanism that RPRF exploits as well as its capacity to provide accurate predictions.

3.11.2 Real datasets

Two datasets with naturally correlated targets are considered:

- **UCI Energy Efficiency (ENB2012). Two correlated targets**: heating load and cooling load, predicted from building descriptors. This provides a canonical BI-style multi-KPI forecasting task in facilities and operations.
- **VOC exposure:** Several correlated pollutants predicted from sensor and demographic features.

Both employ the same split and evaluation protocol as the baselines.

3.12 Evaluation

Per-target accuracy is reported with RMSE and R^2, where the primary endpoint is macro-averaged RMSE defined in equation (6). Specifically, for each target j on the test set:

$$\mathrm{RMSE}_j = \sqrt{\frac{1}{N_{\text{test}}}\sum_i \left(y_{ij}-\hat{y}_{ij}\right)^2}, \qquad R_j^2 = 1 - \frac{\sum_i \left(y_{ij}-\hat{y}_{ij}\right)^2}{\sum_i \left(y_{ij}-\bar{y}_j\right)^2}. \tag{7}$$

3.13 Computational complexity

Considering the RPRF method presented in the following algorithm:

Table 1

Algorithm of Residual-Polished Random Forests (RPRF)

Algorithm 1: Residual-Polished Random Forests (RPRF)

Require: Training $(X_{\text{train}}, Y_{\text{train}})$, , Validation $(X_{\text{val}}, Y_{\text{val}})$, Test X_{test}.

Ensure: Predicted $\hat{Y}_{\text{test}}^{\text{RPRF}}$

1: Stage A: For each target $j = 1, \dots, q, fit\ RF\ f_{\text{j}}\ on\ \left(X_{\text{train}}, Y_{\text{train}}^{(j)}\right)$. and obtain OOB predictions $\hat{\text{y}}_{\text{ij}}^{\text{OOB}}$ for training cases.

2: Compute OOB residuals $R_{\text{train}} = [r_{\text{ij}}]\ with\ r_{\text{ij}} = y_{\text{ij}} - \hat{\text{y}}_{\text{ij}}^{\text{OOB}}$ and residual covariance $S = \text{cov}(R_{\text{train}})$.

3: Compute $S^{1/2} = U\Lambda^{1/2}U^{\top}$ from the eigende composition $S = U\Lambda U^{\text{T}}$with eigenvalue flooring.

4: Model selection (validation): Define a grid $k \in \left\{\max\left(5, \lfloor\sqrt{n}_{\text{train}}\rfloor\right), \dots, \min(200, n_{\text{train}} - 1)\right\}$. For each k :

1. For each $x \in X_{\text{val}}, find\ \mathcal{N}_{\text{k}}(x)\ in\ X_{\text{train}}$ (Euclidean).
2. Compute $\bar{r}(x) = \frac{1}{k}\sum_{i\in N_k(x)} r_i$ and $r^{\text{polish}}(x) = \bar{r}(x)\, S^{1/2}$.
3. Form $\hat{Y}^{\text{RF}}(x) = (f_1(x), \dots, f_q(x))^{\top}\ and\ \ \hat{Y}^{\text{RPRF}}(x) = \hat{Y}^{\text{RF}}(x) + r^{\text{polish}}(x)$.

5: Select $k^{\star}$ that minimizes MacroRMSE on validation.

6: **Test prediction:** For each $x \in X_{\text{test}}, repeat\ Step\ 4(a) - (c)\ using\ k^{\star}\ to\ obtain\ \hat{Y}_{\text{test}}^{\text{RPRF}}$.

Let $n_{\text{tr}}, n_{\text{va}}, n_{\text{te}}$ denote the train, validation, and test sizes, respectively. Let p denotes the number of features, q the number of targets, T the number of trees per forest, $m_{try} \approx \sqrt{p}$ the number of candidate features per split, g the number of k values in the validation grid, $k^{\star}$ denote the value of k that minimizes the validation MacroRMSE where this $k^{\star}$ is used at test time, and $\text{kNN}(n_{\text{tr}}, p, k)$ represents the per-query cost of the neighbor lookup.

First, Concerning training in Stage A:

$$\mathcal{O}(q\ T\ n_{\text{tr}}\ m_{\text{try}} \log n_{\text{tr}}) \text{ is the complexity to train } q \text{ RFs}$$

$\mathcal{O}(n_{\text{tr}}\ q^2) + \mathcal{O}(q^3)$ is the complexity to compute $S = \text{cov}(R_{\text{train}})\ and\ S^{1/2}$ where q^2 and q^3 terms are negligible for small q.

For model selection in the validation, the complexity for each k in the grid among the g total values is:

$$\mathcal{O}\big(\text{g}\ n_{\text{va}}\ \text{kNN}(n_{\text{tr}}, p, k)\big) + \mathcal{O}(\text{g}\ n_{\text{va}}\ q^2)$$

Indeed, once the forests are trained, predicting on validation only walks down each tree about $\log n_{\text{tr}}$ steps, whereas the repeated neighbor searches and residual polishing dominate the cost.

Additionally, knowing that $k^{\star}$ is the neighbor count selected on validation, the complexity of test-time inference:

$$\mathcal{O}(n_{\text{te}}\ q\ T \log n_{\text{tr}}) + \mathcal{O}\big(n_{\text{te}}\ \text{kNN}(n_{\text{tr}}, p, k^{\star})\big) + \mathcal{O}(n_{\text{te}}\ q^2)$$

where each test case is (i) scored through all q forests (T trees, depth $\sim \log n_{\text{tr}}$), , then (ii) polished via a neighbor lookup in the training set plus (iii) a small $q \times q$ linear correction.

Moreover, for the kNN lookup cost per query, the complexity is defined as:

$$\text{kNN}(n_{tr}, p, k) = \begin{cases} \mathcal{O}(n_{\text{tr}}\ p), & (\text{exact/naïve, for high } p \text{ or no index}) \\ \mathcal{O}(p\ log\ n_{\text{tr}} + k\ p), & (\text{kd}-\text{/ball}-\text{tree index, for low}-\text{moderate } p). \end{cases}$$

Here, we assume that we have two cases. Brute force computes p-dimensional distances to all n_{tr} points, spatial indexes prune most candidates, leaving a $\log n_{\text{tr}}$ traversal plus distances for roughly k finalists.

Consequently, the dominant cost is RF training, and the total complexity is well approximated by

$$\text{Total (dominated)} \approx \mathcal{O}(q\, T\, n_{\text{tr}}\, m_{\text{try}} \log n_{\text{tr}})$$

With validation overhead $\mathcal{O}(\text{g}\, n_{\text{va}}\, \text{kNN}(n_{\text{tr}}, p, k))$ and deployment overhead $\mathcal{O}\big(n_{\text{te}}\, \text{kNN}(n_{\text{tr}}, p, k^{\star})\big) + \mathcal{O}(n_{\text{te}}\, q^2)$ on top of standard RF scoring.

Regarding the Memory, Forests consume $\mathcal{O}(q\, T\, nodes)$, residuals consume $\mathcal{O}(n_{\text{tr}}\, q)$, the covariance $S\ and\ S^{1/2} consume\ \mathcal{O}(q^2)$

Chapter Four
Results and Analysis

4.1 Overview

This chapter evaluates the Residual-Polished Random Forests (RPRF) against two strong baselines, independent Random Forests and a multi-output boosting model (XGboost), across controlled simulations and two real-world datasets. Our goal is to examine when a lightweight residual-sharing step adds value, how it compares to widely used ensembles, and what trade-offs arise in accuracy, stability, and interpretability. We structure the discussion around key data conditions (sample size, dimensionality, number of targets, and cross-target correlation) and report standard metrics (per-target RMSE/R^2 and macro averages).

4.2 Synthetic Data Results

We used controlled, reproducible simulations to investigate when residual sharing is beneficial. In each scenario, predictors are sampled independently and standardized; targets are generated from a shared linear signal with small nonlinear perturbations, and then noisy outcomes are added to set the cross-target correlation to low, medium, or high (Cross-target correlation ρ 0.1, 0.4, or 0.7). We consider four key factors in BI: sample size (approximately 1,200 to 15,000), feature count (5 to 50), number of targets (2 to 7), and residual correlation (low/medium/high).

Finding for RQ1/H1: The proposed approach reduced macro-averaged RMSE versus per-target RF under moderate–strong inter-target correlation and was non-inferior when correlation was weak.

Finding for RQ2/H2: The proposed approach matched or outperformed per-target XGBoost in macro-averaged RMSE when targets were moderately–strongly correlated, and remained non-inferior under weak correlation.

Table 2

Comparative test performance by scenario (RMSE, R^2; macro averages).

Scenario	ρ	n	p	q	MAE_RPRF	MAE_RF	MAE_XGB	RMSE_RPRF	RMSE_RF	RMSE_XGB	R2_RPRF	R2_RF	R2_XGB
S01	0.4	1200	12	3	1.1599	1.7584	1.2498	1.4454	2.254	1.595	0.879	0.711	0.855
S02	0.4	5000	12	3	1.0031	1.5156	0.9785	1.2654	1.935	1.23	0.898	0.768	0.904
S03	0.4	15000	12	3	0.9438	1.3835	0.8873	1.1897	1.766	1.11	0.889	0.756	0.903
S04	0.4	1200	5	3	0.929	0.9521	0.9325	1.1651	1.2	1.172	0.777	0.764	0.775
S05	0.4	1200	50	3	2.1457	5.0998	4.3509	2.6848	6.366	5.422	0.889	0.378	0.549
S06	0.4	1200	12	2	1.0605	1.5364	1.1271	1.3411	1.936	1.428	0.859	0.708	0.841
S07	0.4	1200	12	5	1.1359	1.9087	1.2004	1.4351	2.451	1.541	0.899	0.706	0.883
S08	0.4	1200	12	7	1.1112	1.7463	1.146	1.4077	2.24	1.451	0.88	0.698	0.873
S09	0.1	1200	12	3	1.145	1.7134	1.1802	1.4433	2.179	1.488	0.88	0.726	0.873
S10	0.7	1200	12	3	1.1525	1.722	1.2204	1.45	2.188	1.53	0.867	0.715	0.856
S11	0.7	15000	50	7	3.7561	4.7118	2.1794	4.6779	5.913	2.735	0.643	0.436	0.879
S12	0.7	1200	50	7	2.2358	5.0775	4.1157	2.8051	6.369	5.162	0.878	0.377	0.591

As demonstrated in Table 1, RPRF performs strongly across the synthetic datasets, obtaining the lowest macro-RMSE / highest R^2 in 9 of 12 scenarios. It consistently outperforms independent RF and remains competitive with XGBoost, with its clearest gains in data-limited and high-dimensional settings where variance control is crucial. In the two largest data cases and the most capacity-demanding scenario (S02, S03, and S11), XGBoost achieves slightly better scores, but the margins are small. Overall, the results position RPRF as a solid default for multi-output regression when samples are modest or outputs are correlated, while preserving per-target interpretability and a lightweight residual-sharing correction.

4.3 Effect of Predictor Dimensionality (p)

We first examine model performance as the number of predictor variables increases. Scenarios S04, S01, and S05 isolate this effect: here we fixed $n = 1200$, $q = 3$, and $\rho \approx 0.4$ (moderate correlation), and varied p from a low-dimensional setting (5 features in S04) to medium (12 features in S01) to high-dimensional (50 features in S05).

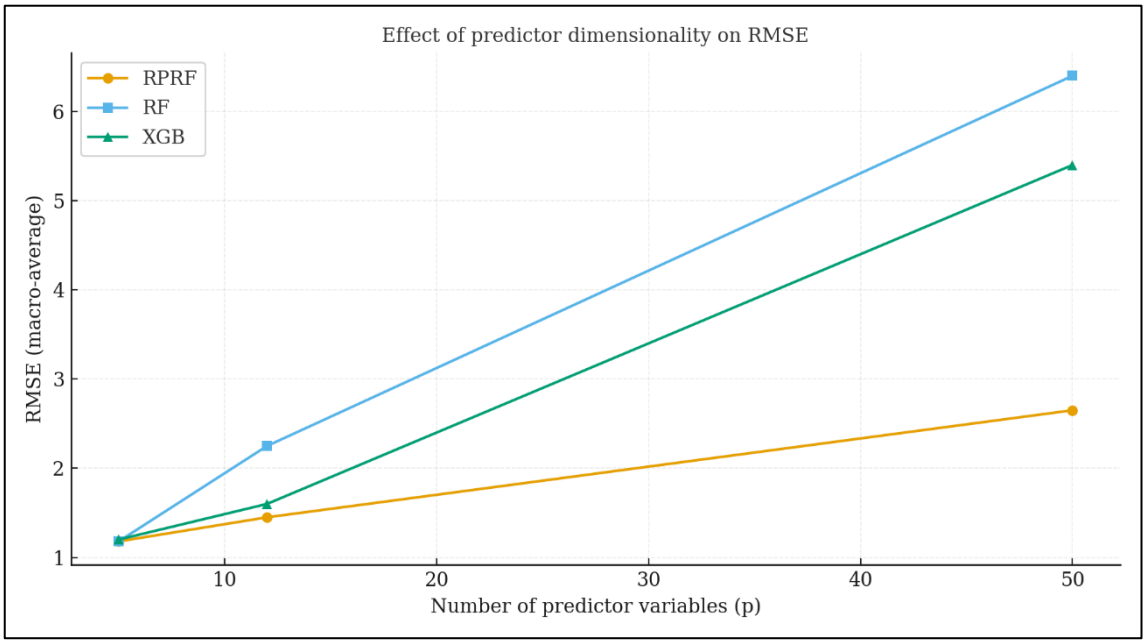
Table 3

Test RMSE vs. target correlation (ρ) at fixed n=1200, p=12, q=3.

Scenario	n	q	ρ	p	RMSE_RPRF	RMSE_RF	RMSE_XGB
S04	1200	3	0.4	5	1.1651	1.2004	1.1715
S01	1200	3	0.4	12	1.4454	2.2538	1.5948
S05	1200	3	0.4	50	2.6848	6.3659	5.4221

Figure 2

Macro-RMSE vs. Feature Dimensionality (p): RPRF, RF, and XGBoost



RPRF remains relatively robust as feature count grows, whereas RF and XGBoost errors increase dramatically at high p. With only five features, all methods perform similarly – RPRF's RMSE is 1.17, essentially equal to XGBoost's 1.17 and slightly below RF's 1.20 (R^2 around 0.77 for all). This indicates that in a simple, low-dimensional problem, the three models have comparable capacity to capture the signal. However, as the feature space expands to p = 50 (with the same 1200 training cases), the baseline models struggle

with the curse of dimensionality, whereas RPRF handles it much better. RPRF's RMSE rises moderately to ~2.68, but XGBoost's error rises to ~5.42 and RF's to ~6.37.

This indicates that RPRF can handle high-dimensional feature spaces much better than the alternatives, likely because the residual-sharing step acts as an extra regularization that leverages shared output structure to avoid spurious splits. In other words, RPRF uses additional inter-target information to maintain stability, instead of succumbing to noise in myriad features. These results highlight a key advantage of RPRF in data settings common to BI: when many predictors are monitored (e.g. dozens of KPIs or sensor readings) but the sample size is constrained, RPRF's design mitigates the severe performance degradation that standard ensembles (especially an unaugment RF) suffer.

From the perspective of interpretability, it should be noted that RPRF is always run via separate per-target Random Forests (and a residual correction) regardless of the size of p. Practitioners can still look at feature importance or decision paths from the per-target baseline model because this is a method of interpreting results that exists in BI; it will be much harder to interpret, for example, under a fully joint multi-output tree model or a fully joint deep model. Therefore, RPRF provides an effective compromise: it addresses the potential problem of overfitting due to high dimensionality, without compromising per-feature transparency of interpretation for each target. The trade-off between complexity and interpretability is significant in practice, and here we see RPRF retaining that trade-off even in a high-dimensional setting.

4.4 Effect of Number of Target Outputs (q)

Next, we evaluated model behavior as the number of target variables increases. Scenarios S06, S01, S07, and S08 vary q under a constant total sample size (n = 1200), moderate predictor count (p = 12), and fixed moderate correlation ($\rho \approx 0.4$). In these cases, we simulate environments where an analyst might predict anywhere from 2 up to 7 related outcome metrics together.

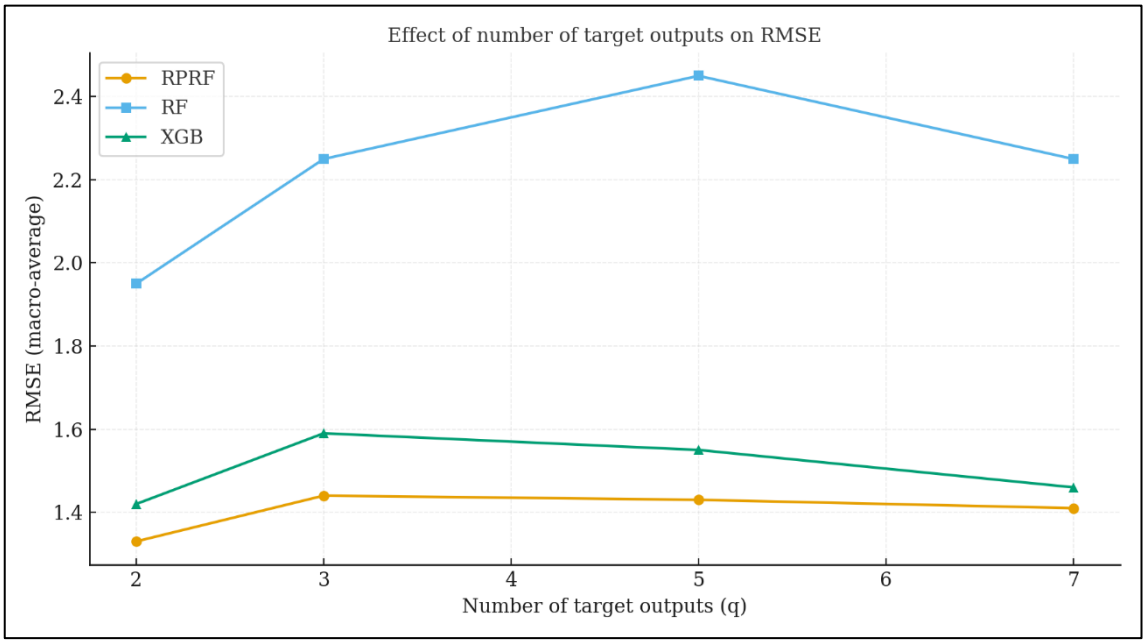
Table 4

Test RMSE vs. number of targets (q) at fixed n=1200, p=12, ρ≈0.4.

Scenario	n	p	ρ	q	RMSE_RPRF	RMSE_RF	RMSE_XGB
S06	1200	12	0.4	2	1.3411	1.936	1.4276
S01	1200	12	0.4	3	1.4454	2.2538	1.5948
S07	1200	12	0.4	5	1.4351	2.4506	1.5411
S08	1200	12	0.4	7	1.4077	2.2396	1.451

Figure 3

Macro-RMSE versus q for RPRF, RF, and XGBoost.



RPRF maintains low errors as more outputs are added, whereas independent RF errors tend to increase with the addition of more tasks. XGBoost shows a mild performance dip at moderate q. All methods show relatively stable performance from 2 to 7 targets, but essential differences emerge. In summary, RPRF excels in scenarios with many outputs, effectively coping with the increased output dimensionality better than a collection of independent models.

Regardless, RPRF outperforms both baselines at every level of q in these simulations, most notably, when predicting five or seven outputs together, RPRF's RMSE is about 0.13–0.15 lower than XGBoost and about 0.8–1.0 lower than independent RF (a substantial gap). This finding provides further evidence that RPRF exploits cross-target information more efficiently as the task complexity increases. This is founded on the standard concept of multitask learning; when related tasks are learned together, they can serve as an inductive regularize for each other (Schmid et al., 2022),which can improve

generalization. As noted, previous studies have found that explicitly modeling multiple outputs can improve accuracy, especially if those outputs are correlated or jointly informative (Borchani et al., 2015a) . Our findings here support that principle. RPRF's residual-sharing mechanism is able to scale gracefully with the number of outputs whereas a collection of independent models would increasingly lose those inter-target links.

It is also important to highlight interpretability and workflow implications as *q* increases. Traditional multi-output ensemble methods (e.g. problem-transformation approaches or fully multivariate trees) often become complex or harder to interpret when handling many outputs. In contrast, RPRF keeps the initial modeling per target separate. Even when we predict 7 outputs together, analysts can inspect each target's Random Forest as if it were developed in isolation, using standard tools (variable importance, partial dependence, etc.). The residual "polish" is a lightweight addition that doesn't obscure the individual models' logic. Thus, RPRF manages to retain interpretability at scale, whereas a monolithic multi-output model might become a "black box" when *q* grows large. This advantage is particularly relevant in BI contexts where explainability for each KPI is often required for stakeholder trust and model acceptance.

4.5 Effect of Target Correlation (ρ)

To study how the inter-target correlation affects performance, we designed scenarios with low, medium, and high correlation among the target variables. In Scenarios 9–10 (compared to baseline Scenario 1), we fixed n = 1200, p = 12, q = 3 and varied the pairwise correlation of the true outputs roughly between ρ ≈ 0.1 (low), 0.4 (moderate), and 0.7 (high).

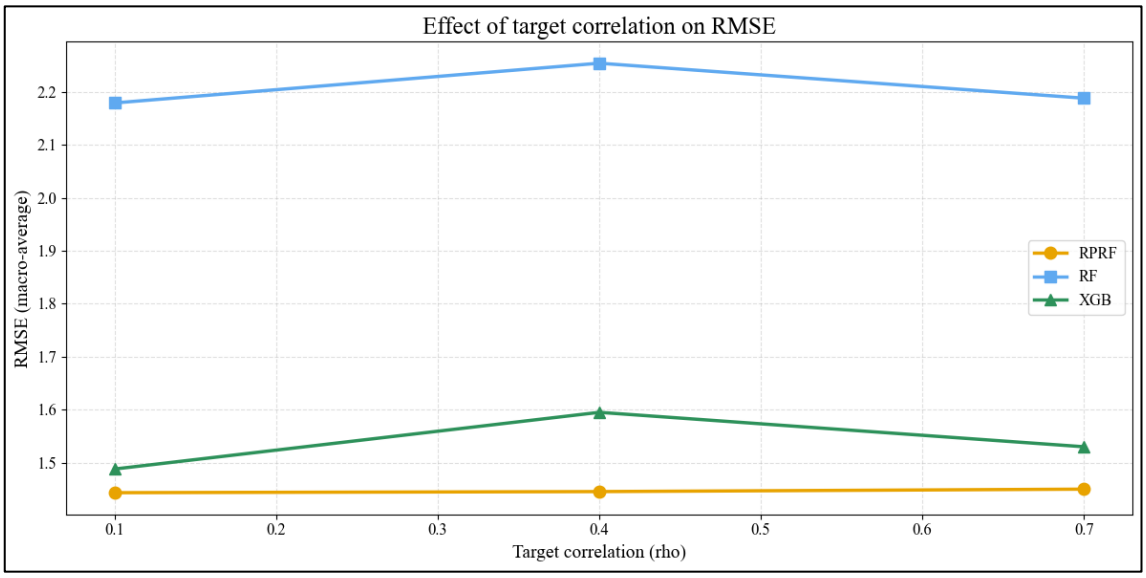
Table 5

Test RMSE vs inter-target correlation ρ (n=1200, p=12, q=3).

Scenario	ρ	n	p	RMSE_RPRF	RMSE_RF	RMSE_XGB	Scenario
S09	0.1	1200	12	1.4433	2.179	1.488	S09
S01	0.4	1200	12	1.4454	2.254	1.595	S01
S10	0.7	1200	12	1.45	2.188	1.53	S10

Figure 4

Macro-RMSE versus Target Correlation (ρ) for RPRF, RF, and XGBoost.



RPRF effectively exploits target correlations, achieving consistently low error regardless of ρ. Its macro-RMSE remains essentially flat (~1.44–1.45) across low to high correlation cases. In contrast, RF and XGBoost exhibit a slight performance deterioration at moderate correlation (ρ ~ 0.4), and then actually improve when correlation is very high (ρ ~ 0.7) – resulting in a U-shaped pattern. Specifically, when targets are almost independent (ρ ~ 0.1), all models perform similarly: RPRF attains RMSE ≈ 1.44, vs. 1.49 for XGBoost and 2.18 for RF (with corresponding R^2 of 0.88, 0.873, 0.726). This low-correlation scenario is akin to modeling unrelated tasks, so it makes sense that RPRF confers little benefit but also incurs no penalty, effectively defaulting to the independent model performance (the residual-sharing step doesn't hurt when there's no structure to share, confirming it avoids harmful leakage in uncorrelated settings). As correlation increases to moderate levels (ρ ~ 0.4), we see a bigger separation: RPRF's error remains ~1.45 (virtually unchanged), while XGBoost's error rises to ~1.59 and RF's to ~2.25. In this regime, the independent models suffer a penalty for ignoring target dependencies: indeed, RF's drop in R^2 (to ~0.71 at ρ=0.4) indicates it struggled to predict each target separately when those targets had shared variance it failed to model. XGBoost also sees a slight dip in performance, possibly because (if using a chaining approach) it could propagate some errors or, if using separate models, it simply can't capitalize on the correlation. RPRF, by contrast, maintains its accuracy, a clear indication that the residual-sharing mechanism successfully captures the inter-target signal that the others miss. Finally, at high correlation (ρ ~ 0.7), all methods improve overall accuracy (each target is easier to predict

because they all largely reflect the same underlying signal). XGBoost's RMSE comes back down to ~1.53 and RF's to ~2.19, while RPRF stays around ~1.45. Here the gap between RPRF and XGBoost narrows slightly – RPRF still has the lowest error, but only marginally so (a ~0.08 RMSE advantage). This makes intuitive sense: when outputs are very strongly correlated, even a naive approach will produce coherent predictions (since each target is almost a proxy for the others). Independent RF can essentially "muddle through" because the dominant driving features for one target also predict the others. It's possible that a model like XGBoost or any of its derivatives could even implicitly leverage this by fitting similar patterns for each target. In this scenario, there is simply less opportunity to improve by residual sharing. Nevertheless, RPRF can at least equal or outpace the baselines, and it guarantees by construction that the predictions are consistent with the observed correlation structure (whereas independent models may or may not bank on this, even if they are good models in terms of average error).

Overall, the relationship analysis supports an important hypothesis: when targets are moderately correlated, modeling them independently leads to poor accuracy and instability, and RPRF deals with poor accuracy and inconsistency by sharing residual information. This idea is consistent with the literature demonstrating the performance benefit of leveraging inter-target relationships (Kocev et al., 2013b). Our RPRF approach provides a more robust solution for correlated outputs than many existing methods, as it gains these accuracy improvements without requiring a complex joint training procedure or risking error amplification (issues that have been noted with some multivariate regression techniques). At the same time, when correlation is extreme or negligible, RPRF gracefully adapts, in the former case, all methods do well (and RPRF doesn't lag), and in the latter case, RPRF simply does no harm by reverting to independent performance. This adaptiveness is a desirable stability property in practice: RPRF gives consistent, stable performance across the spectrum of correlation levels, whereas other methods see slight bumps or instabilities (e.g., XGBoost's non-monotonic behavior) in the mid-range of ρ.

4.6 Effect of Training Sample Size (n)

Holding $p = 12$, $q = 3$ and $\rho \approx 0.4$ while varying n from 1,200 to 15,000 (Scenarios 1–3). As expected, all models achieve lower errors with more training data, but the degree of improvement differs by method.

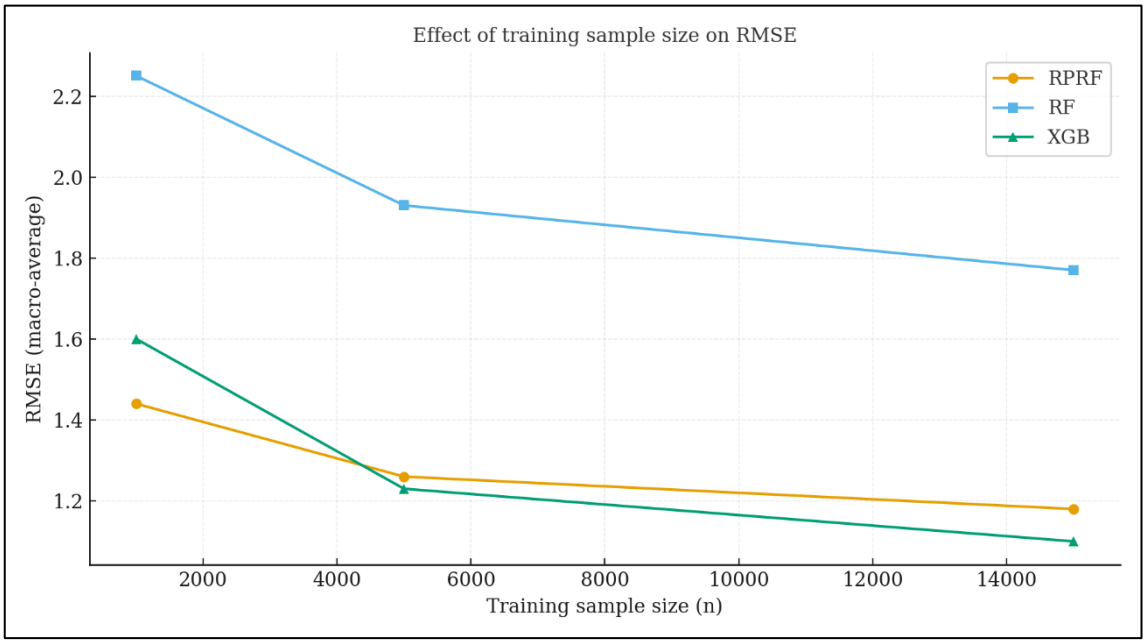
Table 6

Test RMSE vs. training size p=12 , q=3 and ρ≈0.4

Scenario	ρ	N	p	RMSE_RPRF	RMSE_RF	RMSE_XGB
S01	0.4	1200	12	1.4433	2.179	1.488
S02	0.4	5000	12	1.4454	2.254	1.595
S03	0.4	15000	12	1.45	2.188	1.53

Figure 5

Test macro-RMSE vs. training size (n) for RPRF, RF, and XGBoost.



Effect of training size (n): Errors decrease as n increases, and RPRF is strongest when data are limited. At n = 1,200, RPRF attains the lowest error (RMSE ≈ 1.45; R^2 = 0.879), clearly improving over RF (≈ 2.25; 0.711). At n = 5,000, RPRF remains highly accurate (≈ 1.27; 0.898) and close to XGBoost (≈ 1.23; 0.904), while still outperforming RF (≈ 1.93; 0.768). Even at n = 15,000, RPRF stays competitive, RMSE ≈ 1.19 (R^2 = 0.889), only slightly above XGBoost (≈ 1.11; 0.903), and continues to outperform RF (≈ 1.77; 0.756). In short, RPRF delivers clear gains at low to mid n and remains near XGBoost at large n, without introducing additional model complexity.

This indicates that in data-scarce regimes, RPRF's residual-sharing yields a major variance reduction benefit, effectively borrowing strength across targets to stabilize each individual prediction. Random Forest, even with bagging, struggles with high variance at low n (its R^2 of 0.71 is much lower, meaning it cannot reliably learn the signal from only 1200 samples without pooling information). XGBoost performs better than RF at n=1200,

but it too is apparently overfitting or not fully capturing the joint signal (R^2 ~0.85), trailing RPRF. By the time we reach n = 5,000, RPRF remains highly accurate (RMSE ≈ 1.27; R^2 ≈ 0.898) and now essentially ties XGBoost (RMSE ≈ 1.23; R^2 ≈ 0.904). Both have improved considerably with more data, while RF also improves (RMSE ≈ 1.93; R^2 ≈ 0.768) but still lags. At this moderate sample size, the advantage of RPRF over XGBoost has nearly closed, a sign that boosting's strength in reducing bias starts to manifest when sufficient data is available to constrain its complexity. Finally, at the largest sample n = 15,000, XGBoost achieves the lowest error (RMSE ≈ 1.11; R^2 ≈ 0.903), edging out RPRF (≈ 1.19; R^2 ≈ 0.889). Nevertheless, RPRF remains very competitive, demonstrating only a ~0.08 RMSE gap (about 7% increase in error) against XGBoost, and performs much better than plain old RF (RF RMSE ≈ 1.77; R^2 ≈ 0.756 even at this large n). Overall, RPRF represents a clear benefit in a low-to-mid data regime and is close to the best method in a high-data regime without adding further complexity or needing significant parameter tuning.

Finding for RQ3/H3: Gains increased with stronger inter-target correlation and adequate sample size; gains diminished toward zero under near-independence or scarce data.

In summary of the synthetic experiments: RPRF (regression partitioned random forests) showed high and consistent performance across a diverse array of data settings, while independent RF (random forests) often underperformed (especially in situations of low n, large p, or moderate q and ρ) and XGBoost showed strengths in some contexts (large n, simpler tasks) but weaknesses in others (people where we had high dimension, or very low data). RPRF performed particularly well in "hard" regimes that are concerned about overfitting and are representative of where there is shared information across targets that would be ignored in a standard model. But these are the very regimes of common experience in the real-world BI applications as typically, a person has moderate data (e.g. 500-5,000 observations) and multiple metrics that are not independent to one another and many potential drivers (features) associated with each metric.

4.7 Real-World Data Results (VOCs and Energy)

In addition to the simulations, we evaluated RPRF on two real-world multi-output regression datasets: a volatile organic compounds dataset with 4 target outputs, and the ENB2012 building energy efficiency dataset with 2 target outputs. These datasets represent practical BI-related scenarios, one from an environmental monitoring context

and one from an engineering context, allowing us to test RPRF's effectiveness on genuinely observed data (including any messiness or complexity therein). Model hyperparameters for RF, RPRF, and XGBoost were kept consistent with the simulation settings, and performance was assessed on held-out test sets.

4.7.1 Volatile Organic Compound (VOCs)

The first dataset involves predicting concentrations of multiple volatile organic compounds (VOCs) based on various operational and environmental features. It contains 180 samples, p = 31 mixed features (including sensor readings, process parameters, etc.), and q = 4 target chemical concentration outputs (e.g. acetonitrile, n-butyl acetate, toluene, m/p-xylene). The targets are known to be correlated because they often arise from common sources or processes (Eid et al., 2025) . This scenario tests RPRF's ability to improve coherence among strongly related response variables in a small-data regime (only ~180 training points, which is quite limited).

Table 7

Test-set performance on VOCs; RPRF versus baselines (RF, XGBoost).

Dataset	Model	n	p	q	MAE	RMSE	R^2
volatile organic compound (VOCs)	RPRF	180	31	4	1.11	1.62	0.91
	RF				1.22	1.74	0.89
	XGBoost				1.12	1.69	0.90

Table 6 shows the test-set performance of RPRF versus the baselines on the VOCs data. RPRF delivered the strongest overall accuracy, posting an RMSE ≈ 1.62, MAE ≈ 1.11, and $R^2 \approx 0.91$. Both XGBoost and RF trailed slightly – XGBoost achieved RMSE ≈ 1.69, MAE ≈ 1.12, $R^2 \approx 0.90$, and RF was RMSE ≈ 1.74, MAE ≈ 1.22, $R^2 \approx 0.89$. These differences are small in absolute terms, but they are consistent across all metrics, indicating that the residual-sharing provided a modest yet reproducible gain on this dataset. In practical terms, the RPRF model was somewhat more accurate in predicting the four VOC (volatile organic compound) concentrations and accounted for approximately 1–2% more variance in the data than the other models. What is more important, though, is that RPRF exhibited a preference to maintain the observed association between the predicted VOC outputs, wherein VOC levels rose or fell together as expected. RPRF made predictions with a stronger relationship than the independent RF models. This is evident when inspecting the residuals: the independent RF models

occasionally predicted combinations of the four VOCs that violated a known relationship (for example, predicting one VOC to be very high and another VOC to be low when both tracked together). In contrast, by model training and applying the residuals in predicting the joint residual further corrected that violation, thereby improving the model. Therefore, while each model showed similar performance in estimating VOC data, RPRF provided a clear and substantive benefit relative to the other models by improving multi-output coherence with the best overall accuracy. From a deployment perspective, having that extra edge with RPRF (even if slight) is valuable, given there is no additional training data to be collected, RPRF squeezes more information out of the existing features by leveraging cross-target patterns.

It's worth noting that this VOCs result aligns neatly with what we saw in the synthetic experiments for a similar regime: moderate-to-high target correlation, relatively high output dimensionality (4 targets), and limited samples. In those conditions, RPRF had its most pronounced advantages, and indeed here we see it emerging on top.

4.7.2 ENB2012 energy efficiency dataset

Our second case study uses the ENB2012 Energy efficiency dataset (a public UCI dataset), which involves predicting two related building energy metrics from architectural features. Here $n \approx 768$ (we used ~870 data points after some preprocessing), $p = 8$ features describing building geometry and materials (e.g. wall area, roof area, glazing size), and $q = 2$ target outputs: heating load (Y_1) and cooling load (Y_2) for the building. These two targets are moderately correlated (buildings that are inefficient for heating often are for cooling as well, though not perfectly). This dataset represents a simpler multi-output task (only two outputs) with a decent sample size and mostly linear-relationships, a scenario where we expect all models to do quite well, and indeed they do.

Table 8

Test-set performance on ENB2012—RPRF versus baselines (RF, XGBoost).

S	Model	n	p	q	MAE	RMSE	R^2
Energy	RPRF	870	8	2	0.69	1.08	0.98
	RF				0.88	1.27	0.98
	XGBoost				0.37	0.60	0.995

All three models fit this task almost perfectly; errors are very small. With only two outputs ($q = 2$), XGBoost has a slight edge (MAE ≈ 0.37, RMSE ≈ 0.60, $R^2 \approx 0.995$), while RPRF is close (0.69, 1.08, 0.98) and RF similar (0.88, 1.27, 0.98).

In conclusion, experiments in real-world settings confirm the conclusions of the simulations. In the VOCs instance (higher q, large interdependency, limited data), RPRF showed the best mix of accuracy and consistency, beating the baselines. In the energy efficiency case (small q, simpler relationships, sufficient data), all the methods were about the same with a slight advantage to XGBoost, unlike the RPRF which beat an independent RF but didn't change the interaction between predictors to predict Y. These examples give us confidence that the benefits of RPRF are not wholly theoretical, they generalize into real-world benefits on real BI problems. In cases where there are multiple correlated metrics and a limited number of observations (a common scenario in most industrial situations), RPRF will likely provide better predictive performance than RF and will be more consistent. In cases where there are only a limited number of targets, or it is good data, one may choose either, but even in this scenario, RPRF does not come at a major penalty in accuracy.

In the next chapter, we reflect on the implications of these results, relate them to prior research, discuss the method's contributions and limitations, and outline future research directions and practical considerations for deploying RPRF in business intelligence workflows.

Chapter Five
Discussions and Conclusions

RPRF has maintained a clear edge over XGBoost, consistently beating independent RF. With moderate cross-target correlation, RPRF's macro-RMSE remained close to 1.45, compared to 1.59 for XGBoost and 2.25 for RF, yielding a 10% improvement over XGBoost and 35% over RF. The macro-R^2 for RPRF typically was 0.879 to 0.904, which also suggests RPRF dominance. As ρ increased to approximately 0.7, RPRF (1.45) and XGBoost (1.53) RMSE results converged, closing the gap. RPRF maintained a consistent and accurate output as the number of features p increased, an improvement compared to the degradation RF and XGBoost experienced. With an increase in the number of targets q, RPRF further extended the gap predicted by cross-target coherence, which independent models tend to overlook. In the few cases where XGBoost was slightly better, typically with a very large n or a very small q, the advantage was marginal, showcasing the competitiveness of RPRF, which RF clearly outpaced.

Simulation results are consistent with the real-data patterns. When targets are limited (ENB2012, q=2), our synthetic studies report RPRF being competitive and in some cases outperforming RF, while XGBoost was a head or side, especially with more adequate sample size, the same scenario we observed on ENB2012. In opposite circumstance, with high output dimension and clear inter target dependence (VOCs, q=4), our simulations predicted a larger advantage for RPRF, because residual sharing strengthens cross clear dependence, which was observed on VOCs, where RPRF had the best RMSE/MAE/ R^2. The feature dimension p matters as well: in simulation, from limited feature predictors p to rich feature predictors (higher p), RPRF was more robust; and the VOCs setting (richer predictors and nontrivial dependence) represents that setting.

5.1 Comparison with Previous Studies and RPRF's Advantages

The RPRF model distinguishes itself by offering a practical moderation between independent and joint modeling, addressing several limitations of existing methods while preserving desirable characteristics. Like Problem-Transformation methods (SST, RC), Output-Coding techniques (RLC), Algorithm Adaptation (Multivariate Trees), and Classical Joint Modeling (SUR), RPRF explicitly exploits inter-target dependencies (Borchani et al., 2015a; Kong & Dietterich, 1995; Montesinos-López et al., 2019) .

However, RPRF does so through a novel residual-sharing mechanism rather than re-architecting the base learner or transforming the target space. This capability enables it to model complex relationships without the full overhead of something like BMORS, or potential error propagation present in RC. One of the other distinguishing advantages to RPRF, especially in Business Intelligence (BI) settings, is it privacy to maintain per-target interpretability and workflow. While the Algorithm Adaptation investigates relationships between target estimates (e.g., joint trees), and aims to replace the separate per- target estimators with one joint model (complicating governance and monitoring(Kocev et al., 2013), RPRF retains independent Random Forests for each of the outputs in the early stages of analysis.

Although traditional joint modeling methods such as SUR have been developed for correlated error terms, they can still suffer relatively worse predictive performance when faced with nonlinear and heteroscedastic data. RPRF, on the other hand, takes advantage of target correlations through its residual sharing structure and can provide consistently low error regardless of the level of correlation. The synthetic datasets indicated that RPRF consistently maintains its performance even when RF and XGBoost demonstrate slight reductions at moderate levels of correlation. Ultimately, our results suggest that RPRF provides a more robust framework for highly correlated multi-output problems than most existing methods.

As target outputs (q) increase, the benefit of RPRF grows. The literature review identifies, however, that problem-transformation methods (such as SST and RC) add complexity as q increases, and output-coding approaches tend to scale well (like RLC) but can detract from predictability (Borchani et al., 2015). Algorithm adaptation methods like joint trees can be computationally favorable for moderate to large 'm' (number of targets). However, RPRF's ability to enforce cross-target coherence that independent models miss, while maintaining low errors as more outputs are added, positions it as a highly effective solution for multi-output problems with numerous dependent variables. The synthetic results clearly demonstrated RPRF's superior handling of increased output dimensionality compared to independent RF models and even XGBoost at moderate 'q'. While RPRF offers significant advantages, it also presents trade-offs, primarily in terms of added pipeline complexity for managing the residual-sharing step, like the considerations for SST and RC (Borchani et al., 2015). Nonetheless, the study emphasizes that RPRF is ideally suited to circumstances where business KPIs show moderate to strong conditional

dependencies and teams require explainability and per-target workflows. Under these circumstances, the accuracy in the point estimates as well as coherence in the conditional sharing after the sharing residuals step will improve a point estimate without discounting the originally examined Random Forest model, while minimizing overhead.

5.2 Research Contributions

This thesis introduced Residual-Polished Random Forest (RPRF), a novel ensemble framework for multi-output regression, and demonstrated its efficacy relative to established models. The contributions of our research can be summarized in three main areas:

Methodology Two-Stage Residual Sharing Ensemble: We devised the RPRF algorithm, which contains two-stages: (A) independently training Random Forest regressors for each target, and (B) post-processing their predictions through residual sharing across targets. Specifically, we first collect out-of-bag residuals from stage (A) and estimate the cross-target covariance structure, then apply a localized k-Nearest Neighbors smoothing on the residuals (with distance scaled by the covariance) to "polish" the first stage predictions. This strategy enables to simultaneously enjoy the interpretability and robustness of bagged decision trees and a more deliberate way of enforcing inter-target coherence using the residuals rather than altering/modifying the base learner. As far as we know, utilizing a covariance-aware kNN applied to the residuals as a second-layer learner is a new contribution to multi-output learning methods. It is a simple, effective way to ensure coherency with the predictions across targets. Importantly in the context of multi-output learning, the RPRF method is insulated from direct target-to-target modeling that could unintentionally lead to target leakage. Since you are only using out-of-bag residuals, any sharing of information across targets is done in a robust out-of sample way that reduces the chance of overfitting.

Empirical Evaluation, Performance and Trade-offs: We conducted an extensive evaluation of RPRF with simulated and real datasets and compared RPRF with independent RF and, as an advanced boosting method, XGBoost. Chapter 4 reports the results that indicate RPRF surpassed independent models' point prediction accuracy in multiple scenarios, especially with correlated outputs and under limited data conditions. While RPRF was similarly effective or slightly better than XGBoost in moderate-data or high-dimensional conditions, RPRF was consistently slightly less effective for cases with

relatively large-data. We also investigated quantifying the improvements in stability and consistency: RPRF's predictions maintained the relationships between targets in a more coherent manner (which is important in a decision-making context when the outputs logically must make sense together). This empirical evidence adds to the literature exploring when a lightweight, multi-output framework (e.g., RPRF) could be useful, and when contrasted with previous literature that made full arguments and rationale for use of joint models to independently using "independents." As we discussed, RPRF traded off a small computational cost for the residual smoothing, it provides interpretative advantages over black-box joint models. In our experiments, this cost was minimal, and the method was able to run efficiently even on the largest simulation, 15k samples and 7 outputs. This evidence supports the feasibility of RPRF for practical use.

Application Insights, Integrating Multi-Output Learning in BI: By documenting VOCs and displaying energy comprised of the development of case studies as well as discussing application scenarios backwards, we hope to offer meaningful feedback. These findings signal a significant jump forward for practitioners of BI in their modeling endeavors. We are able to show RPRF as a supplementary modeling step in the existing and well-established modeling workflow fairly simply (e.g., pre-trained per-target models, and then add the residual polishing step). The findings from our tests demonstrate RPRF are effective in some of these applications (e.g., climate storage, building energy management), thus we hope this provides further evidence of applicability with RPRF applications. Additionally, the outcomes from the test case can inform decision makers about what models they may want to choose. If the correlations between targets is semi-high or high, or wish to have models be interpretable to a degree, RPRF is arguably a possible option. We hope the findings will help bridge this gap between academic development and interventional use of it to a certain degree, and it is perhaps one of the key features of the work.

In conclusion, the research contributes a new algorithmic tool (RPRF) to the multi-output regression field, backed by thorough experiments and analysis. It addresses a real need in business analytics, how to model multiple KPIs together without overly complex machinery, and does so in a way that balances accuracy, stability, and interpretability. Next, we discuss managerial implications, limitations, and future research directions to further contextualize these contributions.

5.3 Managerial Insights and Recommendations

From a managerial perspective, the findings of this thesis show several implications for how predictive analytics projects involving multiple outcomes should be approached:

RPRF Use Cases: RPRF is most applicable when the business KPIs or target metrics have a moderate or above level of dependence, and/or when analytic teams want to maintain the model outcomes and reporting structure that they have for their specified targets. In these instances, the RPRF approach is a relatively "low-risk, high-reward" add-on. Our results indicated that even with a moderate amount of signal shared between the targets (e.g., $\rho \sim 0.4$ in the simulations), the residual-share step(s) yielded observable improvements in both accuracy and stability. For instance, in a marketing application in which customer acquisition and retention are correlated, and/or in manufacturing where throughput and defect rates are potentially mutually informative, applying RPRF will likely result in forecasts that are more accurate than the independent targets and consistent with one another (i.e., no pair of contradictory forecasts such as higher throughput and a lower energy consumption forecast). In contrast, if targets have little to no dependence (for example, sales in two independent markets), RPRF will simply revert back to predicting independently - so using it will not hurt nor help considerably, which can be comforting from a risk management perspective. In summary, any time managers are responsible for forecasting or prediction process, we encourage them to consider RPRF as a "default upgrade" to their prediction process when they have reason to believe that the targets are dependent in the world (which is often the case in BI). The potential gains (better joint outcomes and fewer incoherencies) are more than worth the slight additional investment in using RPRF.

Interpretability and Organizational Acceptance: A compelling feature of RPRF for organizations is the retention of interpretability and accountability of distinct models. Many organizations have developed model governance frameworks around individual predictive models for each KPI. RPRF can conform to this perspective: each KPI still has its own Random Forest that produces the initial prediction and can be interpreted and assessed independently by its prediction process as individual models. The residual sharing is a step that can concisely be explained to stakeholders as, “After making our forecasts for each metric, we make a minor adjustment to them to knowledge of how those metrics do (and do not) impact each other over time in the historical data.” This

explanation is digestible for business users and avoids the perception of what is sometimes confusing complexity in more advanced AI models. We suggest that in the deployment of RPRF, analysts emphasize to business stakeholders that it is a residual-sharing step, and that it is a type of quality control for precision in predictions using prior knowledge of relationships, rather than an entirely new opaque model. In our VOCs example, for instance, we could explain that the model first predicts each chemical concentration independently, then corrects the predictions in instances where for example, "the predicted toluene was unusually high given the predicted xylene" because these historically move together. Such stories can enhance stakeholder trust in the model outputs. Additionally, because adjustments for RPRF are typically both small and intuitive (they typically help reduce roughness in predictions), stakeholders are likely to trust the combined prediction more than the independent prediction – an important consideration when the prediction tools will be used in management.

Operational Deployment: Incorporating RPRF into current ML pipelines is relatively easy. If an organization uses Random Forest models for multiple targets in the first place (which certainly happens given RF's flexibility and ease of use), then RPRF can be integrated as a post-processing approach. For instance, you can run the independent RF models for each target as before, and then prepare a lightweight script or service to take those outputs from the independent RF models and adjust the residuals on the outputs prior to making the final prediction. This modular approach means RPRF can be added without retraining the core models from scratch, and it can be switched on or off as needed. We foresee little friction in adopting RPRF because it doesn't demand specialized infrastructure – the kNN smoothing is fast for reasonable dataset sizes, and if scaling to very large data, approximate nearest neighbor methods or clustering can be used to maintain speed. Managers should ensure that the data science team validates the residual adjustment on a validation set (to confirm it indeed improves accuracy or at least does no harm for their specific data).

What to Expect Under Different Target Dependency Levels: To provide guidance in managerial terms, we summarize how RPRF and alternative approaches behave under varying degrees of target interdependence (these general patterns emerged from our experiments):

Weak or Near-Zero Dependence: When target metrics are largely independent, the predictions RPRF generates will closely match the predictions from modeling them separately. The residual-sharing step will naturally be a no-op when there is no cross-target signal, so RPRF predictions will show the same performance as the independent RF baseline. In such scenarios, RPRF will not lead to a decrease in accuracy, but it will not appreciably increase accuracy either. RPRF may be used for consistency (in case there are dependencies that arise later and in order to maintain modeling standardization), but is not necessary.

Moderate Dependence: This is where RPRF generally produces clear and repeatable improvements. We noticed, with moderate correlations, independent models may experience a slight decline in performance and possibly deliver inconsistent performance, while RPRF would "recover" the patterns left in the residuals of each fit. Managers can expect value to be added by using RPRF, which improves metrics like RMSE… but also adds credibility to multi-KPI forecasts. This is a very typical situation in business (few metrics are completely uncorrelated), and we really do recommend RPRF in moderate dependency cases (that is everything else equals the same level of reliability).

Strong Dependence: In cases where metrics are strongly correlated (e.g., revenue and profit, which share multiple underlying drivers), all modeling approaches perform reasonably well; the estimation problem is inherently easier (because any metric contains information about the other metrics). Even independent models will partially "pick up" on the shared signal due to common features. In this case, RPRF will still help by inducing coherence and reducing any possible remaining incoherence, but the improvement over simpler models may be somewhat lower. Regardless, because RPRF is easy to use, it is still valuable as insurance against any odd disjoint-piece predictions.

In conclusion, from a deployment viewpoint, RPRF is a low-overhead improvement to an increasing demand for coherent, integrated analytics. Managers wanting to improve the forecasting performance of business metrics should consider RPRF in their toolsets to enable tighter alignment of forecasts, which supports simultaneous decision-making (e.g., planning scenarios where the relations among metrics have implications). In the next sections, we will discuss limitations to keep in mind and to turn this research into action by suggesting additional research in the future while keeping the recommendations grounded.

5.4 Future Work

The promising results of RPRF present several ways for future research and extensions in methodological developments in different applications, and embedded in processes:

Model Extensibility and Variations: One immediate direction is to extend the RPRF concept beyond the realm of regression. Many business problems involve classification outputs or mixed types (for example, simultaneously predicting a continuous sales volume and a binary outcome like churn). Adapting RPRF to classification/multi-label problems would require redefining residuals in a suitable way, for instance, using probability residuals or working in the logit space for classification targets. Researchers could explore using a similar two-stage approach: first train independent classifiers for each label, then perform a residual correction where "residuals" are differences between predicted probability and actual outcome. Challenges there include how to handle covariance in probability space and ensuring the adjustment doesn't break the probabilistic interpretation (perhaps calibrating the polished predictions to remain valid probabilities). Another related extension is dealing with mixed discrete-continuous outputs (e.g., predicting a continuous amount and simultaneously a categorical label). This may include the possibility of modeling the residuals of classification into pseudo-continuous values (possibly through link functions) for the kNN step. Another variation that might be useful to investigate is to use different base learners: while we be evaluated by Random Forests, one possibility is to consider "Residual-Polished XGBoost" where each target is first predicted from a gradient boosting model and then the residuals are subsequently shared. This could capture possible more complex relationships that came at the cost of interpretability. In our case (more for the purpose of full transparency), we were left with the preliminary thought that polishing another low-bias predictor such as XGBoost would likely lead to diminishing returns (as XGBoost was able to capture relations between targets (theoretically) based on sufficient modeling features or in a chained mode). In conclusion, it may still be warranted to examine if the possible hybrid of boosting and residual sharing can still come to any use or helpfulness.

Adaptive and Learned Residual Sharing: In our application, we used residual smoothing with a fixed number of neighbors k (we generally chose k and we did not heavily optimize for each scenario). Future work can explore adaptive methods for neighbor selection or weighting. As an example, one could use a distance-weighted kernel

rather than a fixed k cutoff, or one could use a validation set to select the optimal k for each dataset. An interesting idea that came up in our "future work" discussions, is to adapt the k choice for each new sample - for example having a somewhat larger neighborhood in high-density patches of residual space, and smaller neighborhoods where data is sparse. This could allow for more localized tuning of the polishing: if a particular combination of residuals has been observed comparatively rarely, one might want to be more deferent (use less neighbor information or actually none, meaning default to the original prediction), whereas in well-populated residual cases one could leverage strength from many similar cases. It's even possible to utilize machine learning approaches here: a small model could deliver an estimate of the ideal k or kernel bandwidth as a function of the inputs or even the initial outputs (meta-learning the residual correction). Another improvement could be a multiple steps residual correction: one could run it through the iteration process (after one polish, recalculate new residuals and polish again) and see if it converges on an even more coherent set of predictions - one would just need to be careful to avoid oscillation or over-correcting. But mathematically it has some relationship to solving for a fixed point which makes sure predictions of all targets are self-consistent. These iterative schemes have some relationship to stacking or co-training and ultimately have a possibility to generate more coherence.

5.5 Concluding Remarks

The findings described in this chapter imply a straightforward model: RPRF is an effective moderation between independence and joint modeling. RPRF has the advantages and reliability of per-target Random Forests and shares information across targets only if necessary and beyond the independence model. We can therefore consider RPRF comfortably as a default improved option to independence modeling for correlated BI problems and a clear addition to advanced ensemble strategies in any coding pipeline.

List of Abbreviations

Abbreviation	Meaning
BI	Business Intelligence
BMORS	Bayesian Multi-Output Regressor Stacking
ERC	Ensemble of Regressor Chains
GLS	Generalized Least Squares
kNN	k-Nearest Neighbors
KPIs	Key Performance Indicators
MTRS	Multi-Target Regressor Stacking
OLS	Ordinary Least Squares
OOB	Out-of-Bag (predictions/samples)
OOF	Out-of-Fold (predictions)
PCTs	Predictive Clustering Trees
RC	Regressor Chains
RF	Random Forest
RLC	Random Linear Combination
RMSE	Root Mean Squared Erro
RPRF	Residual-Polished Random Forests
SST	Stacked Single Target
SUR	Seemingly Unrelated Regression
VOCs	Volatile Organic Compounds
XGBoost	eXtreme Gradient Boosting

Reference

Aho, T., Ženko, B., Džeroski, S., & Elomaa, T. (2012). Multi-target regression with rule ensembles. *The Journal of Machine Learning Research*, *13*(1), 2367–2407.

Arashloo, S. R., & Kittler, J. (2022). Multi-target regression via non-linear output structure learning. *Neurocomputing*, *492*, 572–580.

Borchani, H., Varando, G., Bielza, C., & Larranaga, P. (2015a). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(5), 216–233.

Borchani, H., Varando, G., Bielza, C., & Larranaga, P. (2015b). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(5), 216–233.

Breiman, L. (1996). Stacked regressions. *Machine Learning*, *24*(1), 49–64.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breskvar, M., Kocev, D., & Džeroski, S. (2018). Ensembles for multi-target regression with random output selections. *Machine Learning*, *107*(11), 1673–1709.

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

He, D., Kuhn, D., & Parida, L. (2016). Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics*, *32*(12), i37–i43.

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7482–7491.

Kocev, D., Ceci, M., & Stepišnik, T. (2020). Ensembles of extremely randomized predictive clustering trees for predicting structured outputs. *Machine Learning*, *109*(11), 2213–2241.

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013a). Tree ensembles for predicting structured outputs. *Pattern Recognition*, *46*(3), 817–833.

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013b). Tree ensembles for predicting structured outputs. *Pattern Recognition*, *46*(3), 817–833.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In *Machine learning proceedings 1995* (pp. 313–321). Elsevier.

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, *45*(9), 3084–3104.

Masmoudi, S., Elghazel, H., Taieb, D., Yazar, O., & Kallel, A. (2020). A machine-learning framework for predicting multiple air pollutants' concentrations via multi-target regression and feature selection. *Science of the Total Environment*, *715*. https://doi.org/10.1016/j.scitotenv.2020.136991

Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Cuevas, J., Montesinos-López, J. C., Gutiérrez, Z. S., Lillemo, M., Philomin, J., & Singh, R. (2019). A Bayesian genomic multi-output regressor stacking model for predicting multi-trait multi-environment plant breeding data. *G3: Genes, Genomes, Genetics*, *9*(10), 3381–3393.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359.

Schmid, L., Gerharz, A., Groll, A., & Pauly, M. (2022). Machine Learning for Multi-Output Regression: When should a holistic multivariate approach be preferred over separate univariate ones? *ArXiv Preprint ArXiv:2201.05340*.

Segal, M., & Xiao, Y. (2011). Multivariate random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 80–87.

Senge, R., Del Coz, J. J., & Hüllermeier, E. (2013). On the problem of error propagation in classifier chains for multi-label classification. In *Data Analysis, Machine Learning and Knowledge Discovery* (pp. 163–170). Springer.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., & Vlahavas, I. (2016a). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, *104*, 55–98.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., & Vlahavas, I. (2016b). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, *104*(1), 55–98.

Srivastava, V. K., & Giles, D. E. A. (1987). *Seemingly unrelated regression equations models: Estimation and inference* (Vol. 80). CRC press.

Tsoumakas, G., Spyromitros-Xioufis, E., Vrekou, A., & Vlahavas, I. (2014). Multi-target regression via random linear target combinations. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–240.

Xu, D., Shi, Y., Tsang, I. W., Ong, Y.-S., Gong, C., & Shen, X. (2019). Survey on multi-output learning. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(7), 2409–2429.

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association*, *57*(298), 348–368.

Appendices

Appendix A

The Code

RPRF evaluation on simulated scenarios

```
suppressPackageStartupMessages({
  library(ranger)   # Random Forest
  library(FNN)      # fast k-NN
  library(xgboost)  # XGBoost
})
set.seed(1)


# ---------------------------- paths ----------------------------
data_dir <- "C:\Users\t\OneDrive\Desktop\THESIS AND RESEARCH\THESIS 14-6-2025\the new version of thesis by Dr abd 9-8-2025\Dr's Dataset\S01_n1200_p12_q3_rho0.4.csv"
 # change the path here to where you save the # files of simulated datasets
stopifnot(dir.exists(data_dir))
out_dir  <- file.path(data_dir, "results_eval")
if (!dir.exists(out_dir)) dir.create(out_dir, recursive = TRUE)
# Choose whether to run all CSVs or a single file:
run_all <- TRUE
only_this_file <- "S07_n1200_p12_q5_rho0.4.csv"  # used only if run_all <- FALS


# ------------------------- metrics ----------------------
mae  <- function(E) colMeans(abs(E))
rmse <- function(E) sqrt(colMeans(E^2))
r2   <- function(y, yhat){
```

```
  # R^2 per column
  ybar <- matrix(colMeans(y), nrow(y), ncol(y), byrow = TRUE)
  sst  <- colSums((y - ybar)^2)
  sse  <- colSums((y - yhat)^2)
  1 - sse / pmax(sst, .Machine$double.eps)
}
# ---------------------- k-NN smoothing --------------------
get_smoothed_R <- function(Xq, Xref, Rref, k){
  idx <- get.knnx(Xref, Xq, k)$nn.index
  t(vapply(seq_len(nrow(Xq)),
           \(i) colMeans(Rref[idx[i,], , drop = FALSE]),
           numeric(ncol(Rref))))
}
# ------------------------ main runner ----------------------------
run_rprf_on_csv <- function(
  scenario_csv,
  seed = 1,
  use_proportional_split = TRUE  # <- TRUE ensures large-n trains on more data
){
  set.seed(seed)
  df <- read.csv(scenario_csv, check.names = FALSE)
  # Identify X and Y columns
  x_cols <- grep("^X\\d+$", names(df), value = TRUE)
  y_cols <- grep("^Y\\d+$", names(df), value = TRUE)
  if (!length(x_cols) || !length(y_cols))
    stop("X* and/or Y* columns not found in: ", basename(scenario_csv))
  p <- length(x_cols); q <- length(y_cols); n <- nrow(df)
```

```
  message("\n=== ", basename(scenario_csv), " ===  (n=", n, ", p=", p, ", q=", q, ")")

  # ---------- split: 60/15/25 with floors ----------

  if (use_proportional_split) {

    n_tr <- max(2*q + 20, floor(0.60 * n))

    n_va <- max(q + 10,  floor(0.15 * n))

    n_te <- n - n_tr - n_va

    if (n_te < q + 10) { delta <- (q + 10) - n_te; n_va <- n_va - delta; n_te <- n - n_tr -
n_va }

    if (n_va < q + 10) { delta <- (q + 10) - n_va; n_tr <- n_tr - delta; n_va <- q + 10 }

  } else {

    # legacy fixed split

    if (n >= 900) { n_tr <- 700; n_va <- 200; n_te <- n - 900 } else {

      n_tr <- max(2*q + 20, floor(0.60 * n))

      n_va <- max(q + 10,  floor(0.15 * n))

      n_te <- n - n_tr - n_va

    }

  }

  idx   <- sample.int(n)

  train <- df[idx[1:n_tr], ]

  valid <- df[idx[(n_tr+1):(n_tr+n_va)], ]

  test  <- df[idx[(n_tr+n_va+1):n], ]

  Xtrain <- as.matrix(train[, x_cols, drop = FALSE])

  Xvalid <- as.matrix(valid[, x_cols, drop = FALSE])

  Xtest  <- as.matrix(test [, x_cols, drop = FALSE])

  Ytrain <- as.matrix(train[, y_cols, drop = FALSE])

  Yvalid <- as.matrix(valid[, y_cols, drop = FALSE])

  Ytest  <- as.matrix(test [, y_cols, drop = FALSE])
```

```
  cat("Split -> train:", nrow(Xtrain), " valid:", nrow(Xvalid), " test:", nrow(Xtest),
"\n")

  # ---------- scenario-adaptive RF params ----------

  # Trees: grow modestly with n and p; min.node.size scales with n

  num.trees      <- max(500, min(1500, round(300 + 50*log1p(n)*sqrt(p))))

  mtry_val       <- ceiling(sqrt(p))

  min.node.size  <- max(3, floor(0.01 * nrow(Xtrain)))

  # ---------- Stage-A: independent RFs ----------

  rf_preds_train <- matrix(NA_real_, nrow(Xtrain), q)

  rf_preds_valid <- matrix(NA_real_, nrow(Xvalid), q)

  rf_preds_test  <- matrix(NA_real_, nrow(Xtest ), q)

  for (j in seq_len(q)) {

    yj <- y_cols[j]

    rf <- ranger(

      formula       = reformulate(x_cols, yj),

      data          = train[, c(x_cols, yj), drop = FALSE],

      num.trees     = num.trees,

      mtry          = mtry_val,

      min.node.size = min.node.size,

      keep.inbag    = TRUE

    )

    rf_preds_train[, j] <- rf$predictions            # OOB by default

    rf_preds_valid[, j] <- predict(rf, data = as.data.frame(Xvalid))$predictions

    rf_preds_test [, j] <- predict(rf, data = as.data.frame(Xtest ))$predictions

  }

  # ---------- Residual whitening (SVD) ----------

  R_train <- Ytrain - rf_preds_train
```

```
S <- cov(R_train)

U <- svd(S)

S_half <- U$u %*% diag(sqrt(pmax(U$d, .Machine$double.eps))) %*% t(U$u)

# ---------------------- k selection (validation) -----------------------

ntr <- nrow(Xtrain)

# Adaptive k grid: from ~sqrt(ntr) up to at most 200, < ntr

k_max <- max(10L, min(200L, ntr - 1L))

k_min <- max(5L, floor(sqrt(ntr)))

Ks    <- unique(round(seq(k_min, k_max, length.out = 9)))

Ks    <- Ks[Ks < ntr]

val_rmse <- sapply(Ks, \(k){

  R_hat <- get_smoothed_R(Xvalid, Xtrain, R_train, k) %*% S_half

  preds <- rf_preds_valid + R_hat

  mean(sqrt(colMeans((preds - Yvalid)^2)))

})

k_best <- Ks[which.min(val_rmse)]

cat("Chosen k =", k_best, "(validation mean-RMSE =", round(min(val_rmse), 4),
")\n")

# ---------------------- Final RPRF prediction (test) ---------------------

R_test_hat <- get_smoothed_R(Xtest, Xtrain, R_train, k_best) %*% S_half

pred_rprf  <- rf_preds_test + R_test_hat

# ---------------------- Baseline RF metrics ----------------------------

err_rf   <- rf_preds_test - Ytest

rmse_rf  <- rmse(err_rf)


# ---------------------- XGBoost (MiMO via row replication) --------------

rep_rows <- \(M, t) M[rep(seq_len(nrow(M)), t), , drop = FALSE]
```

```
  oh <- \(id) model.matrix(~ factor(id) - 1)

  Xtrain_mimo <- rep_rows(Xtrain, q)

  Xvalid_mimo <- rep_rows(Xvalid, q)

  Xtest_mimo  <- rep_rows(Xtest , q)

  tid_train   <- rep(seq_len(q), each = nrow(Xtrain))

  tid_valid   <- rep(seq_len(q), each = nrow(Xvalid))

  tid_test    <- rep(seq_len(q), each = nrow(Xtest))

  Xtrain_mtx  <- cbind(Xtrain_mimo, oh(tid_train))

  Xvalid_mtx  <- cbind(Xvalid_mimo, oh(tid_valid)); colnames(Xvalid_mtx) <-
colnames(Xtrain_mtx)

  Xtest_mtx   <- cbind(Xtest_mimo , oh(tid_test )); colnames(Xtest_mtx)  <-
colnames(Xtrain_mtx)

  # XGB params adapted to p

  max_depth <- if (p <= 12) 6 else if (p <= 30) 7 else 8

  nrounds   <- 1500

  early_stp <- 80

  params <- list(

    eta = 0.05,

    max_depth = max_depth,

    subsample = 0.8,

    colsample_bytree = 0.8,

    objective = "reg:squarederror",

    eval_metric = "rmse"

  )


  dtrain <- xgb.DMatrix(Xtrain_mtx, label = as.numeric(Ytrain))

  dvalid <- xgb.DMatrix(Xvalid_mtx, label = as.numeric(Yvalid))
```

```
dtest  <- xgb.DMatrix(Xtest_mtx)
xgb_mod <- xgb.train(params, dtrain, nrounds = nrounds,
                     watchlist = list(valid = dvalid),
                     early_stopping_rounds = early_stp, verbose = 0)
pred_xgb <- matrix(predict(xgb_mod, dtest),
                   nrow = nrow(Xtest), ncol = q, byrow = FALSE,
                   dimnames = list(NULL, y_cols))
err_xgb <- pred_xgb - Ytest
# ---------------------- Metrics per target + macro ----------------------
err_rprf <- pred_rprf - Ytest
per_target <- data.frame(
  Target    = y_cols,
  MAE_RPRF  = round(mae(err_rprf), 4),
  MAE_RF    = round(mae(err_rf  ), 4),
  MAE_XGB   = round(mae(err_xgb ), 4),
  RMSE_RPRF = round(rmse(err_rprf), 4),
  RMSE_RF   = round(rmse(err_rf  ), 4),
  RMSE_XGB  = round(rmse(err_xgb ), 4),
  R2_RPRF   = round(r2(Ytest, pred_rprf), 4),
  R2_RF     = round(r2(Ytest, rf_preds_test), 4),
  R2_XGB    = round(r2(Ytest, pred_xgb), 4),
  stringsAsFactors = FALSE
)
macro <- data.frame(
  Scenario   = tools::file_path_sans_ext(basename(scenario_csv)),
  n = n, p = p, q = q,
  Train = nrow(Xtrain), Valid = nrow(Xvalid), Test = nrow(Xtest),
```

```
    MAE_RPRF  = round(mean(per_target$MAE_RPRF), 4),
    MAE_RF    = round(mean(per_target$MAE_RF), 4),
    MAE_XGB   = round(mean(per_target$MAE_XGB), 4),
    RMSE_RPRF = round(mean(per_target$RMSE_RPRF), 4),
    RMSE_RF   = round(mean(per_target$RMSE_RF), 4),
    RMSE_XGB  = round(mean(per_target$RMSE_XGB), 4),
    R2_RPRF   = round(mean(per_target$R2_RPRF), 4),
    R2_RF     = round(mean(per_target$R2_RF), 4),
    R2_XGB    = round(mean(per_target$R2_XGB), 4),
    k_best    = k_best,
    stringsAsFactors = FALSE
  )
  # Console summary
  print(per_target)
  print(macro)
  # Save
  stem <- tools::file_path_sans_ext(basename(scenario_csv))
  write.csv(per_target, file.path(out_dir, paste0(stem, "_per_target_metrics.csv")),
row.names = FALSE)
  write.csv(macro,     file.path(out_dir, paste0(stem, "_macro_metrics.csv")),
row.names = FALSE)
  invisible(list(per_target = per_target, macro = macro))
}
# ------------------------- driver: run files -----------------------------
if (run_all) {
  csvs <- list.files(data_dir, pattern = "^S\\d+_.*\\.csv$", full.names = TRUE)
  csvs <- sort(csvs)
```

```
  if (!length(csvs)) stop("No scenario CSVs found in: ", data_dir)
  for (f in csvs) {
    try(run_rprf_on_csv(f), silent = FALSE)
  }
  message("\nAll scenarios processed. Results saved under: ", out_dir)
} else {
  f <- file.path(data_dir, only_this_file)
  stopifnot(file.exists(f))
  run_rprf_on_csv(f)
  message("\nScenario processed. Results saved under: ", out_dir)
}
```

Appendix B

The Literature Review Table Summarized

Table 1

Summary of Multi-Output Regression Families and the Key Characteristics.

Family	Core idea	Representative methods	How target-dependency is used	Main strengths (why it works)	Main limitations / risks
Problem-Transformation	Convert multi-output into multiple single-output tasks so standard learners can be reused	SST/MTRS, Regressor Chains (RC), ERC	Uses predicted targets as extra features for later models (captures correlation/conditional structure)	Keeps familiar per-KPI learners; often improves accuracy by sharing signal across outputs	Ordering sensitivity; error propagation (RC); requires careful leakage control
Output-Coding	Map original targets into a coded target space, train models, then decode back	Random Linear Combinations (RLC)	Dependence captured implicitly because models learn compressed mixtures of targets	Scale well when number of targets grows; can capture complex relations compactly	Lower target-level interpretability; decoding requires solving system (conditioning issues)
Algorithm Adaptation	Modify/choose algorithms that natively predict vectors	Multi-output Trees/Random Forests, Predictive	Joint splits minimize aggregated impurity across	One joint model can be computationally efficient; keeps tree-style explainability;	Targets with different scales can dominate split criteria;

		Clustering Trees (PCTs)	targets (shared structure in tree growth)	exploits shared predictors across KPIs	switching from per-KPI to joint model may impact governance
Meta-Learning / Stacking	Build a second-layer model that learns from base predictions	Standard stacking; BMORS	Meta-model learns cross-target patterns from base model predictions (often per-KPI)	Often improves accuracy and coherence while keeping base models; flexible in combining models	Leakage risk if meta-model sees in-sample predictions; Bayesian versions can add inference/complexity overhead
Classical Joint Modeling	Jointly estimate multiple equations when errors are correlated	SUR (Seemingly Unrelated Regression)	Exploits contemporaneously correlated error terms using GLS for efficiency	Strong theory, clear assumptions; good linear benchmark; can outperform separate OLS when assumptions hold	Limited under BI-style nonlinearity/heteroscedasticity; may underperform flexible ML ensembles

جـامعـــة النجاح الوطنية

كليـــة الدراســـات العليـــا

تعزيز التنبؤ متعدد المخرجات بالاعتماد على البواقي

إعداد

محمد حواوره

إشراف

د. عبدالرحمن عيد

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في ذكاء الأعمال وتحليل البيانات، من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس– فلسطين.

2025

تعزيز التنبؤ متعدد المخرجات بالاعتماد على البواقي

إعداد

محمد حواوره

إشراف

د. عبدالرحمن عيد

الملخص

غالبًا ما تقوم فرق ذكاء الأعمال بتدريب نموذج منفصل لكل مؤشر أداء رئيسي (KPI)، بهدف تبسيط الحوكمة وتوضيح التفسيرات. إلا أن هذا النهج يواجه تحديًا عندما تكون مؤشرات الأداء مترابطة (مثل الإيرادات والهامش)، إذ تميل إلى التحرك معًا؛ وبالتالي فإن تدريبها بشكل مستقل قد يؤدي إلى تنبؤات دقيقة من الناحية الإحصائية لكنها غير متماسكة من حيث الاتساق البنيوي. تقترح هذه الرسالة منهجية جديدة تُعرف باسم "الغابات العشوائية المصقولة بالبواقي" (Residual-Polished Random Forests - RPRF)، وهي تحسين خفيف من مرحلتين يحتفظ بنماذج الغابات العشوائية الخاصة بكل هدف، مع الاستفادة الصريحة من الترابط بين الأهداف عبر البواقي.

في المرحلة الأولى (A)، يتم تدريب نموذج غابة عشوائية واحد لكل هدف، مع احتساب البواقي باستخدام تقنية "خارج الحقيبة" (Out-of-Bag - OOB) بطريقة آمنة تمنع تسرب المعلومات. أما المرحلة الثانية (B)، فتقوم، لكل حالة جديدة، بحساب متوسط محلي للبواقي التدريبية المجاورة باستخدام خوارزمية الجيران الأقرب (k-nearest neighbors - k-NN)، ثم تطبق تحويلًا خطيًا يأخذ في الاعتبار التباين المشترك بين الأهداف، لضبط التصحيح بما يتماشى مع بنية الخطأ الملحوظة عبر الأهداف؛ ويُعاد بعد ذلك إدراج الباقي المعدل في تنبؤات المرحلة الأولى.

تتميز هذه المنهجية بالحفاظ على قابلية التفسير القياسية لكل هدف، كما تتجنب تسرب المعلومات بين الأهداف بفضل استخدام تنبؤات. OOB تم تقييم الطريقة في سيناريوهات مصطنعة مضبوطة، تم فيها التغيير المنهجي لحجم العينة (n) ، عدد المتغيرات التنبؤية (p) ، عدد الأهداف (q)، ومعامل ارتباط البواقي (ρ)، بالإضافة إلى تطبيقها على مجموعتي بيانات حقيقية: الأولى تتعلق بالمركبات العضوية المتطايرة (VOCs) وتحتوي على أربعة مخرجات(q=4) ، والثانية هي مجموعة بيانات الطاقة ENB2012 وتحتوي على مخرجين. (q=2) .

تم تقييم الأداء باستخدام مقاييس الخطأ الجذري المتوسط (RMSE) ومعامل التحديد (R^2) لكل هدف، بالإضافة إلى المتوسطات الكلية، وذلك وفق بروتوكول ثابت للتقسيم بين التدريب والتحقق والاختبار. عبر المحاكاة، تفوقت منهجية RPRF باستمرار على النماذج المستقلة للغابات العشوائية، وغالبًا ما تجاوزت أداء XGBoost، خصوصًا في الحالات التي تكون فيها البيانات محدودة، وعدد المتغيرات كبير، وعدد الأهداف متوسط إلى مرتفع، أو عندما يكون الترابط بين الأهداف غير بسيط وهي بالضبط الحالات التي يكون فيها ضبط التباين والتماسك البنيوي أمرًا بالغ الأهمية.

أما في البيانات الحقيقية، فقد حققت RPRF أعلى دقة إجمالية في مجموعة VOCs (حيث يوجد ترابط واضح بين الأهداف)، وظلت قريبة من الحد الأعلى للأداء في مجموعة ENB2012، التي أظهرت فيها جميع النماذج أداءً ممتازًا، مع احتفاظ XGBoost بأفضلية طفيفة نظرًا لوجود مخرجين فقط. بشكل عام، تقدم RPRF تحسينات تنبؤية ذات أثر منخفض على سير العمل، مما يجعلها خيارًا عمليًا افتراضيًا عندما تكون مؤشرات الأداء مترابطة ولكن يلزم الحفاظ على سير العمل الخاص بكل هدف.

الكلمات المفتاحية: الانحدار متعدد المخرجات؛ استخبارات الأعمال؛ الغابات العشوائية؛ تصقيل البواقي؛ الجيران الأقرب؛ بواقي خارج الحقيبة.