



HAL
open science

Localized Feature Ranking approach for Multi-Target Regression

Hugo Bertrand, Haytham Elghazel, Sahar Masmoudi, Emmanuel Coquery,
Mohand-Said Hacid

► **To cite this version:**

Hugo Bertrand, Haytham Elghazel, Sahar Masmoudi, Emmanuel Coquery, Mohand-Said Hacid. Localized Feature Ranking approach for Multi-Target Regression. 2022 International Joint Conference on Neural Networks (IJCNN), Jul 2022, Padua, Italy. pp.1-8, 10.1109/IJCNN55064.2022.9891892 . hal-04947569

HAL Id: hal-04947569

<https://hal.science/hal-04947569v1>

Submitted on 14 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Localized Feature Ranking approach for Multi-Target Regression

Hugo BERTRAND^{1,2}
¹ Université Lyon 1, LIRIS,
UMR CNRS 5205, F-69622, France
hugo.bertrand@univ-lyon1.fr
² CIRIL Group
49 Av. Albert Einstein
Villeurbanne, France
hbertrand@cirilgroup.com

Haytham ELGHAZEL¹
¹ Université Lyon 1, LIRIS,
UMR CNRS 5205, F-69622, France
haytham.elghazel@univ-lyon1.fr

Sahar MASMOUDI³
³ ENIS LR3E
University of Sfax
Sfax, Tunisia
sahar-masmoudi-phd@enis.tn

Emmanuel COQUERY¹
¹ Université Lyon 1, LIRIS,
UMR CNRS 5205, F-69622, France
emmanuel.coquery@liris.cnrs.fr

Mohand-Saïd HACID¹
¹ Université Lyon 1, LIRIS,
UMR CNRS 5205, F-69622, France
mohand-said.hacid@univ-lyon1.fr

Abstract—Multi-target regression (MTR) aims at designing models able to predict multiple continuous variables simultaneously. The key for designing an effective feature selection model for MTR is to develop a framework under which the feature importances are measured using the correlation between features and targets in a natural way. So far, feature importances in MTR problems were evaluated in a global sense where proposed approaches generate a single ordered list of features common for all the targets. In this work, we adapt the Ensemble of Regressor Chains algorithm in tandem with the random forest paradigm to appropriately model both dependencies among features and targets in a target-specific (localized) feature ranking process. We provide empirical results on several benchmark MTR data sets indicating the effectiveness of our strategy to perform better than selecting one global ranking for all targets with existing state-of-the-art algorithms.

Index Terms—Multi-Target Regression, Feature Importance, Local variable selection.

I. INTRODUCTION

The purpose of a Single-Target Regression (STR) model is to predict the value of a continuous target given a set of input features. STR can be generalized to predict multiple targets at the same time from the same set of features, this generalization is called Multi-Target Regression (MTR). While STR has been widely studied by the machine learning community, MTR has received a more moderate interest.

MTR is a challenging problem that emerges in several modern applications such as : medicine, for example with time-series prediction of drug efficacy [8], ecology, with vegetation condition modeling [6] or soil properties prediction [16], computer vision, with vegetation recognition [18], or data streaming, with time series forecasting [2].

On the other hand, Feature ranking FR is another significant task in machine learning. Often looked as a preprocessing

step, the purpose of this task is to estimate the importance of each feature in the input space regarding the prediction of a target, resulting in a ranking. This allows to reduce the dimensionality of the input space of the model, by selecting only the most significant variables, thus decreasing the amount of time and computing power needed to train a model, as well as reducing the potentiality of introducing noise and bias via the use of moderately to non-significant and redundant variables. Furthermore, the less variables the model uses, the easier it is to understand, an important aspect when working with domain experts.

Numerous feature ranking approaches has been developed for single-target prediction [4], [5], however it is far more limited for MTR. As far as we know, the only tangible study relies on Random Forest Predictive Clustering Tree (RFPCT) [14]. This approach is based on a global variable selection: the feature ranking is determined over all predicted targets.

In this paper, we present a feature ranking method, termed as Ensemble of Regressor Chains guided Feature Ranking ERCFR, based on a combination of Ensemble of Regressor Chains ERC [19] and random forest permutation importance measure [3]. The main advantage of our approach is its ability to lead a local feature ranking per target by appropriately modeling both dependencies among features and targets in the feature ranking process. It simultaneously estimate the importance for each feature regarding each target individually and the intrinsic inter-target relevance as well.

The inquiry here revolves around the improvement of the predictive abilities of a multi-target regressor using our feature ranking method, especially confronting global variable selection. To assess the relevance of our method, we compare it with the existing RFPCT approach on several benchmark datasets. We test several uses of feature ranking with different MTR

predictive contexts: Binary Relevance (BR), Regressor Chain (RC) and Ensemble of Regressor Chains (ERC). Experiment shows that our method demonstrates better overall results, highlighting that local variable selection is more efficient than global variable selection.

The rest of the paper is organized as follows: Section II briefly reviews recent studies on Multi-target regression techniques. Section III presents the details of proposed ERCFR approach. Section IV reports the results of our comparative studies on different real-world data sets. We raise several issues for future work in Section V and conclude with a summary of our contribution.

II. RELATED WORK

A. Multi-Target Regression

In Multi-Target Regression, a set of features is associated to a set of multiple continuous target variables. The issue is to design a model able to exploit relations between these sets, the correlations, using variables from the feature set to predict the ones of the target set.

Let X and Y be two random vectors where X consists of d input variables X_1, \dots, X_d and Y consists of m target variables Y_1, \dots, Y_m . More formally, given a set $D = [X, Y] = \{(x^1, y^1), \dots, (x^n, y^n)\}$ of n training instances (x^i, y^i) such as, $x^i = \{x_1^i, \dots, x_d^i\} \in \mathbb{R}^d$ is a feature vector and $y^i = \{y_1^i, \dots, y_m^i\} \in \mathbb{R}^m$ indicates its affected targets, the objective in multi-target regression task is to find a function $h : \mathbb{R}^d \mapsto \mathbb{R}^m$ that assigns to each input vector x^i an output vector of m continuous values $\hat{y}^i = h(x^i)$ [1].

$$h : \mathbb{R}^d \mapsto \mathbb{R}^m$$

$$x^i = \{x_1^i, \dots, x_d^i\} \mapsto \hat{y}^i = \{\hat{y}_1^i, \dots, \hat{y}_m^i\}$$

The aim of model h is to better approximate the true target vector y^i with \hat{y}^i .

MTR methods are usually separated in two groups [1]: *Problem Transformation* and *Algorithm Adaptation*. The former consists in deconstructing the problem into several less complex single-target problems, and build a model for each of them. The latter focuses on extending common single-target approaches into predicting several targets at once, exploiting the correlations between targets.

The most basic approach of problem transformation is the Binary Relevance (BR) [1], also know as Single Target (ST) approach. It consists in viewing each target as a simple single-target problem, in which the target variables are predicted independently and the potential inter-correlation between them is ignored. In order to improve prediction performance, additional approaches in this category try to model the target relationships. To do so, they rely on building models that treat other targets as additional input variables (meta-inputs). In this context, the Regressor Chain (RC) method is proposed in [19] based on the idea of chaining single-target models. RC involves m regressors linked along a chain (given by a

random permutation of the set of m target variables) where each regressor deals with the single target regression problem associated with one target. The feature space of each link in the chain is augmented with the values of previous targets in the chain. RC is naturally very sensitive to the order of prediction. To mitigate this issue, the Ensemble of Regressor Chain (ERC) [19] generalizes the concept of RC in applying ensemble training. Several chains of prediction are trained with different random target orders, and the predictions for each target is obtained by averaging the prediction of each chain for the target in question. This method has proved quite successful in MTR contexts and inspired subsequent works [10], [12].

On the other hand, Algorithm adaptation methods use existing standard regression algorithms and adapt them to deal with MTR. To illustrate this first category, we can quote Predictive Clustering Trees (PCT) [7] which are an adaptation of decision trees capable of MTR. PCT considers the decision tree as a hierarchy of clusters where multi-target data is partitioned. The induction process in PCT is a top-down generation of clusters. It uses the sum of the variances throughout all targets to identify the best separation at each node which lead the tree to predict multiple targets at once. Koccev *et al.* [14] presented a random forest based multi-target regression model named Random Forest Predictive Clustering Tree (RFPCT). The RFPCT approach is developed on the top of the PCT algorithm. In RFPCT, each tree makes multi-target regressions, and then predictions are combined by averaging the values for each target. Furthermore, RFPCT exploits different properties of the ensemble learning mechanism to estimate a global feature selection relevant to all the targets in MTR problems [14].

B. Performance metrics

Performance evaluation in a multi-target regression context is different from traditional single target models. The metrics can be associated with multiple targets simultaneously. The following performance metrics were used to analyze our experiments:

The average Relative Root Mean Square Error (aRRMSE): The aRRMSE of a model h generated from a training data set D_{train} is resulting from averaging the relative root mean squared errors (RRMSE) estimated for each target variable on the test data set D_{test} . RRMSE is equal to the Root Mean Squared Error (RMSE) for that target divided by the RMSE of predicting the average value of that target in the training set D_{train} . Calculation is summarized in equation 1, where \hat{y}_j is the predicted values of h on the target variable Y_j , \bar{Y}_j is the mean value of Y_j over the training set D_{train} , n is the number of instances and m is the number of target variables. It is worth noting that the value of the aRRMSE ranges from 0 to 1, thus comparing it between datasets is relevant. As it is an error, the smaller the value, the better the algorithm performance.

$$aRRMSE = \frac{1}{m} \sum_{j=1}^m \sqrt{\frac{\sum_{((x^i, y^i) \in D_{test})} (y_j^i - \hat{y}_j^i)^2}{\sum_{((x^i, y^i) \in D_{test})} (y_j^i - \bar{Y}_j)^2}} \quad (1)$$

$$aCC = \frac{1}{m} \sum_{j=1}^m \frac{\sum_{((x^i, y^i) \in D_{test})} (y_j^i - \bar{Y}_j)^2 (y_j^i - \hat{y}_j^i)^2}{\sqrt{\sum_{((x^i, y^i) \in D_{test})} (y_j^i - \bar{Y}_j)^2 \times \sum_{((x^i, y^i) \in D_{test})} (y_j^i - \hat{y}_j^i)^2}} \quad (2)$$

The Relative Performance (RP): RP results from dividing the aRRMSE of the MTR approach A by the corresponding of another MTR method B (3). In other words, it measures the reduction in error of an MTR approach B over another A. By Definition, an RP greater than one means that B is better than A in the evaluated problem and vice-versa [9].

$$RP = \frac{aRRMSE_A}{aRRMSE_B} \quad (3)$$

III. ENSEMBLE OF REGRESSOR CHAINS GUIDED MTR FEATURE RANKING

In this section, we discuss in details our ensemble of regressor chains guided feature ranking (ERCFR) algorithm.

Our approach adapt the ensemble of regressor chains algorithm in tandem with random forest (RF) paradigm, contributing a new feature selection method for MTR. Indeed, the way internal estimates are used to measure variable importance in the random forest paradigm [3] have been influential in our thinking. The key idea consists to use random forest as the base models in ERC to calculate the variable importance. RF has several characteristics that make it ideal for features selection (robust with high-quality predictive performance, does not over fit, handle simultaneously categorical and continuous feature [3], [15]). Furthermore, RF have proved to be efficient for features selection in different application domains.

Before introducing our ERCFR framework, we describe in the sequel how RF provide measures of feature importance. The RF method creates multiple trees using regression trees (CART) [3]. The variable importance measure in RF is based on the decrease of predictive performance when values of a descriptive variable in a node of a tree are permuted randomly. Basically, decision trees in the forest are developed using a random selection of data (bootstrapping) and random selection of variables. Each bootstrap replicate is used as training set to create trees in the forest. In each bootstrapped data set, almost 33% are left out of bag (oob_i), i.e., they are not used for the construction of the i th corresponding tree t_i ($i \in \{1, \dots, T\}$ where T is the size of the forest). Thus, these patterns can be used to estimate unbiased feature relevance. Once a tree t_i is grown, the algorithm evaluates the performance of t_i in terms of mean squared error (MSE) by using its corresponding oob_i examples. This results in the predictive error $MSE(oob_i)$. To assess the importance of an input feature f for the tree t_i , we randomly permute the values of f in the set oob_i resulting in the set oob_i^f . Then, the tree t_i is used to predict the targets of the new out of bag patterns and the mean squared error $MSE(oob_i^f)$ is computed. The importance of the variable f is then calculated as the relative increase of the MSE after its values are randomly permuted. The random forest importance of the feature f is given by the average of its importance

values across all T trees in the forest (c.f. Equation. 4). We note that the greater the value of the importance measure, the more relevant is the feature.

$$I(f) = \frac{1}{T} \sum_{i=1}^T \frac{MSE(oob_i^f) - MSE(oob_i)}{MSE(oob_i)} \quad (4)$$

In our ERCFR approach, the multi target feature selection problem is handled using the Ensemble of Regressors Chain strategy (ERC). As mentioned before, ERC works by building a set of L random regressor chains RC_1, \dots, RC_L .

In the training of a Regression Chain model RC_k , a random chain, of the set of target variables is selected and for each target in the chain, models are built sequentially by using the output of the previous model as input for the next. Assuming that the random chain $C_k = \{y_1, y_2, \dots, y_m\}$ is selected, new m separate regression models are induced. Each model deals with the single target regression problem associated with one target y_j ($1 \leq j \leq m$). In our approach, each regressor is created using the RF algorithm. The first random forest model concerns the prediction of y_1 , has the form $RF_1 : X \rightarrow \mathbb{R}$. The subsequent models $RF_{j, j>1}$ are learned on modified training sets $D_j = \{(x_j^{*1}, y_1^j), \dots, (x_j^{*n}, y_n^j)\}$, where the original input vectors x^i ($i \in \{1, \dots, n\}$) of the n training examples have been augmented by the actual values of all previous targets of the chain to form expanded input vectors $x_j^{*i} = \{x_1^i, \dots, x_d^i, y_1^i, \dots, y_{j-1}^i\}$. Thus, the Random forests built for targets y_j ($j > 1$) have the form $RF_j : X \times \mathbb{R}^{j-1} \rightarrow \mathbb{R}$. Each Random forest RF_j ($1 \leq j \leq m$) in the chain allows us to assess the relevance of each feature in the expanded space (containing both original features and previous targets) to the individual target y_j . The procedure is repeated for all L regressor chains (RC_1, \dots, RC_L). Finally, the average values of the scores of both features and targets across all targets is considered. The main advantage of our approach is its ability to appropriately modeling both dependencies among features and targets in the target-specific feature ranking process.

Thus, our approach relies on a local variable selection. The importance value for each feature towards each target and the importance between the targets are considered, as it allows to apply MTR approaches that use targets as features, like in RC or ERC. Algorithm 1 gives a formal description of the ERCFR training procedure.

IV. EXPERIMENTAL EVALUATION

This section presents the experimental protocol used to assess the effectiveness of our proposed ERCFR feature ranking approach. We first establish the questions driving the approach. Next, we summarize the MTR datasets used for the experiment, as well as the evaluation metrics. We finally develop the established evaluation procedure.

Algorithm 1 Training of ERCFR

Require: A multi-target regression training Data set $D_{train} = [X_{train}, Y_{train}]$, number of targets (m), ensemble size of ERC (L), number of trees performed by the base random forest learner (T).

- 1: **Local Importance Initialization**
- 2: **for** $f \in \text{columns}(X_{train}UY_{train})$ **do**
- 3: **for** $\text{target} \in \text{columns}(Y_{train})$ **do**
- 4: $\text{imp}(f, \text{target}) \leftarrow 0$
- 5: **end for**
- 6: **end for**
- 7: **Models building and Feature Relevance estimate**
- 8: **for** $i = 1 \rightarrow L$ **do**
- 9: $C_i \leftarrow \{y_1, \dots, y_m\}$ a chain of targets in a random order
- 10: $RC_i \leftarrow \emptyset$
- 11: $X'_{train} \leftarrow X_{train}$
- 12: **for** $j = 1 \rightarrow m$ **do**
- 13: $D'_{train} \leftarrow [X'_{train}, y_i]$
- 14: Train a random forest RF_j of T trees on D'_{train}
to predict y_i
- 15: $RC_i \leftarrow RC_i \cup RF_j$
- 16: **for** $f \in \text{columns}(X'_{train})$ **do**
- 17: $\text{imp}(f, y_j) \leftarrow \text{imp}(f, y_j) + \text{FeatureImportance}(RF_j, D'_{train})$
- 18: **end for**
- 19: $\hat{y}_j \leftarrow \text{predict}(RF_j, y_i)$
- 20: $X'_{train} \leftarrow (X'_{train} \cup \hat{y}_j)$
- 21: **end for**
- 22: **end for**
- 23: **for** $f \in \text{columns}(X_{train}UY_{train})$ **do**
- 24: **for** $\text{target} \in \text{columns}(Y_{train})$ **do**
- 25: $\text{imp}(f, \text{target}) \leftarrow \text{imp}(f, \text{target})/L$
- 26: **end for**
- 27: **end for**
- 28: **return** RC, imp

A. Experimental questions

This study is driven towards answering the following questions:

- 1) Can feature importance bring improvement to the predictive abilities of a multi-target regressor ?
- 2) Is local variable selection more relevant than global variable selection in a MTR context ?
- 3) Can importance between target variables bring additional improvement to the performance of a multi-target regressor ?

B. Datasets

To evaluate the performance of our approach, we test it on several MTR datasets from the Mulan Repository [17]. These datasets encompass real information from various domains: water quality (*andro*, *enb*), airline ticket pricing (*atp1d*, *atp7d*), occupational employment survey (*oes10*, *oes97*), etc. The number of features per dataset ranges from 7 to 411, all

TABLE I: Description of the Multi-target regression data sets used in the experiments.

Data	#Examples	#Features	#Targets
atp1d	337	411	6
atp7d	296	411	6
oes97	334	263	16
oes10	403	298	16
edm	154	16	2
wq	1060	16	14
enb	768	8	2
slump	103	7	3
andro	49	30	6

being numeric. The number of target varies from 2 to 16, and the number of examples ranges between 49 and 1060. The statistics of these data are summarized in the Table I.

C. Performance evaluation

The quality of the feature ranking returned by ERCFR on all data sets were evaluated through executing three multi-target regression approaches on them: the single-target (denoted by ST_{ERCFR}) approach which transforms the multi-target regression task into a series of single learning problems, the Regressor chain approach (denoted by RC_{ERCFR}) and its ensemble variant denoted by ERC_{ERCFR} . These algorithms were used here as they have shown considerable performance in training multi-target models in the literature [1], [11].

Feature importances given by ERCFR are incorporated in ST_{ERCFR} , RC_{ERCFR} and ERC_{ERCFR} by adopting the same methodology in [14]. In [13], the authors propose to use the weighted distance-based k-nearest neighbor algorithm (kNN) as a base learner when inducing the MTR approaches. kNN is a simple and most intuitive technique that has been widely considered as a powerful tool to evaluate the effectiveness of feature importance approaches. In the case of regression problems, the target of a test instance is predicted with the traditional kNN by averaging the targets associated to the k nearest neighbors in the training set. Following [14], we propose to directly use the feature relevance values, produced by ERCFR, in a weighted version of the Euclidean distance instead of the original one in the process of finding the closest instance to the new test observation. The weighted distance d_w between two instances x^1 and x^2 is simply formulated as:

$$d_w(x^1, x^2) = \sqrt{\sum_{j=1}^d w_j (x_j^1 - x_j^2)^2} \quad (5)$$

where w_j is the feature importance of feature j produced by ERCFR.

The idea behind using weighted distance-based kNN is to measure of how the usage of the feature ranking methods can benefit performances of multi-target regression methods.

To better assess the feature relevance given by our ERCFR approach in the RC context, we propose the use of a Regressor-based Importance Chain (RCI) that builds a unique chain of targets, capturing the maximum relevance among target outputs. To bypass the problem of randomly selected chain

order with RC, RCI uses the inter-targets importance matrix returned by ERCFR and forms a maximum importance chain, which is used to build a single chained regression model. To determine this optimal order, the importance of a target towards the other ones are summed for each target, and the order is given by decreasing sum. Consequently, the most impactful target will be predicted first, then becoming a significant variable in the prediction of the following targets. This order must demonstrate that the target ranking obtained thanks to ERCFR further improve performance. The Feature importance given by ERCFR are also incorporated in RCI leading to RCI_{ERCFR}

To better assess the quality of our ranking, we built the above MTR methods using all Features (*i.e.*, without feature ranking). We refer to them respectively as ST, RC and ERC in the remaining of the paper.

In the sequel, we also compared our feature ranking given by ERCFR with the recently proposed ensemble feature ranking method RFPCT [14]. Compared to ERCFR which leads specific feature pertinence for each target (including both predictive features and targets), RFPCT generates only a single ordered common list of predictive features toward all targets. It is worth noting that this property is convenient for only executing the single target regression approach (denoted by ST_{RFPCT}) and not for RC and ERC approaches.

We evaluated the performance of the methods using a 10-fold cross validation. Specifically, around 90% of instances randomly chosen from the complete data set were used for the training process (D_{train}), and the remaining 10% were used as a test data set (D_{test}) for measuring the performance of all compared methods. This process was repeated 10 times. The overall performance measures was computed by averaging over those 100 iterations.

The ensemble size in ERC is set to $L = 20$. The random forest algorithm used in ERCFR and RFPCT are tuned similarly. The number of variables to split on at each node and the number of trees T are set to \sqrt{d} and 100, respectively [14]. To make fair comparisons, the number of nearest neighbors for KNN was set to 5 as suggested by the authors in [14].

D. Discussion

1) *Feature Importance evaluation:* In this subsection, the comparative results between the proposed methods ST_{ERCFR} , RC_{ERCFR} , RCI_{ERCFR} , ERC_{ERCFR} and the others classic MTR approaches ST, RC and ERC as well as ST_{RFPCT} are presented.

Detailed performances in terms of aRRMSE and aCC of each MTR algorithm for all datasets are reported in Tables II, III and IV. They also presents the obtained Relative performance (RP) in terms of aRRMSE when comparing standard MTR approaches with the one incorporating feature relevance values given by ERCFR in the last column of each Table.

Several conclusions may be drawn from these results:

- First, as stated in [14], the results indicate that the use of Feature Ranking tends to **increase the predictive performance** of an approach. Indeed, for each MTR

method without feature ranking, their FR-based counterparts, whether RFPCT or ERCFR, outperform the former by a noticeable margin.

- We found feature ranking given by ERCFR to be remarkably effective at improving the performance of multi-target regression methods including single target (ST as well as Regressor chain based methods (RC and ERC over all the metrics. ST_{ERCFR} , RC_{ERCFR} and ERC_{ERCFR} respectively outperform the ST, RC and ERC approaches (without feature ranking) by a noticeable margin.
- ERCFR exhibits the highest performances in terms of aRRMSE and aCC measures than the other state-of-the-art feature ranking method RFPCT. As may be observed in Table II, the incorporation of feature relevance values produced by ERCFR in ST_{ERCFR} has a stronger effect on significantly improving the predictive performance for all datasets, compared to RFPCT in ST_{RFPCT} . The strategy proposed in ERCFR to lead a separate feature ranking per target seems to perform better than selecting one global ranking for all targets in RFPCT.
- Regarding the Regressor Chain approaches in Table III, the use of the importance as weights for the prediction offers little advantage when the target order is randomly determined. However, the order of prediction is key to an accurate prediction for a Regressor Chain. The real strength of the local variable importance appears when considering the **importance of the targets between each other**. These importance allows to determine an optimal order, thus achieving a more significant relative performance.

2) *Feature ranking evaluation:* At a later stage, we want to confront our approach to RFPCT once again, to observe if the usage of local variable selection appears more relevant than global variable selection in terms of performance.

In the sequel, we evaluate the feature ranking of our ERCFR approach in comparison to RFPCT. To do so, we perform a new Single target approach, with a standard KNN as a base predictor. The base predictor is not really important in this context, as it does not use feature importance directly, as so the predictor could be any standard simple regression algorithm, but we stick to kNN for consistency and simplicity. The feature variables must first be ordered by decreasing relevance towards the predicted target. As RFPCT achieves variable selection in a global way, the order will never change, whatever the target. Conversely, ERCFR is considered locally: each feature has an importance value regarding each target. As so, ERCFR appears to be more flexible and more adaptable to the prediction context. To assess the performance of each paradigm, we track the evolution of aRRMSE and aCC for each method as the number of variables used for prediction increases.

Figures 1 and 2 plot the performance of the standard single target approach (ST) in terms of aRRMSE and aCC averaged over the 100 runs against different numbers of selected features obtained by the above compared approaches. Blue curves represent the ERCFR approach, and red curves are for RFPCT.

TABLE II: Performance metrics for Single Target approaches

Dataset	Average aRRMSE						Average aCC						RP	
	ST		ST _{ERCFR}		ST _{RFPCT}		ST		ST _{ERCFR}		ST _{RFPCT}		ST _{ERCFR} / ST	ST _{RFPCT} / ST
	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.		
atp1d	0,456	0,081	0,408	0,080	0,444	0,079	0,883	0,060	0,904	0,050	0,889	0,056	1,118	1,027
atp7d	0,621	0,122	0,546	0,105	0,610	0,128	0,782	0,094	0,828	0,069	0,788	0,092	1,137	1,018
oes97	0,631	0,122	0,595	0,113	0,655	0,144	0,770	0,132	0,798	0,125	0,770	0,135	1,061	0,963
oes10	0,442	0,042	0,406	0,036	0,430	0,043	0,909	0,040	0,922	0,033	0,913	0,036	1,089	1,028
wq	0,930	0,014	0,928	0,014	0,931	0,014	0,375	0,027	0,380	0,025	0,375	0,026	1,002	0,999
enb	0,210	0,059	0,125	0,020	0,144	0,017	0,975	0,013	0,989	0,004	0,989	0,004	1,680	1,458
slump	0,740	0,136	0,684	0,131	0,736	0,131	0,630	0,196	0,685	0,167	0,644	0,187	1,082	1,005
andro	0,543	0,299	0,486	0,256	0,506	0,269	0,783	0,214	0,830	0,152	0,808	0,189	1,117	1,073
edm	0,824	0,182	0,798	0,166	0,798	0,174	0,626	0,126	0,655	0,110	0,654	0,118	1,033	1,033
mean	0,600	0,117	0,553	0,102	0,584	0,111	0,748	0,100	0,777	0,082	0,759	0,094	1,146	1,067

TABLE III: Performance metrics for Regressor Chain approaches

Dataset	Average aRRMSE						Average aCC						RP	
	RC		RC _{ERCFR}		RC _{RFPCT}		RC		RC _{ERCFR}		RC _{RFPCT}		RC _{ERCFR} / RC	RC _{RFPCT} / RC
	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.		
atp1d	0,544	0,091	0,548	0,100	0,502	0,087	0,837	0,076	0,830	0,079	0,860	0,069	0,993	1,084
atp7d	0,764	0,150	0,749	0,150	0,740	0,160	0,665	0,141	0,676	0,122	0,699	0,113	1,020	1,032
oes97	0,631	0,122	0,651	0,144	0,628	0,124	0,770	0,132	0,774	0,129	0,773	0,132	0,969	1,005
oes10	0,590	0,079	0,567	0,073	0,510	0,059	0,835	0,067	0,844	0,057	0,873	0,046	1,041	1,157
wq	0,945	0,013	0,934	0,013	0,935	0,014	0,344	0,027	0,368	0,024	0,364	0,026	1,012	1,011
enb	0,210	0,059	0,142	0,018	0,136	0,023	0,975	0,013	0,986	0,004	0,989	0,004	1,479	1,544
slump	0,829	0,143	0,765	0,121	0,800	0,103	0,524	0,212	0,605	0,182	0,603	0,154	1,084	1,036
andro	0,543	0,299	0,487	0,266	0,499	0,275	0,783	0,214	0,831	0,171	0,823	0,184	1,115	1,088
edm	0,825	0,182	0,795	0,168	0,795	0,172	0,624	0,130	0,657	0,111	0,659	0,111	1,038	1,038
mean	0,653	0,126	0,626	0,117	0,616	0,113	0,706	0,112	0,730	0,098	0,738	0,093	1,083	1,111

TABLE IV: Performance metrics for Ensemble of Regressor Chains approaches

Dataset	Average aRRMSE				Average aCC				RP
	ERC		ERC _{ERCFR}		ERC		ERC _{ERCFR}		
	mean	std.dev.	mean	std.dev.	mean	std.dev.	mean	std.dev.	
atp1d	0,488	0,075	0,454	0,068	0,868	0,063	0,888	0,060	1,075
atp7d	0,640	0,122	0,620	0,096	0,771	0,102	0,793	0,086	1,032
oes97	0,560	0,104	0,528	0,094	0,811	0,117	0,838	0,104	1,061
oes10	0,477	0,040	0,476	0,038	0,908	0,038	0,907	0,037	1,002
wq	0,919	0,011	0,917	0,012	0,389	0,025	0,395	0,024	1,002
enb	0,195	0,035	0,118	0,016	0,980	0,007	0,991	0,003	1,653
slump	0,778	0,118	0,734	0,106	0,633	0,197	0,666	0,166	1,060
andro	0,464	0,234	0,440	0,259	0,810	0,180	0,862	0,149	1,055
edm	0,812	0,166	0,795	0,169	0,635	0,119	0,658	0,111	1,021
mean	0,593	0,101	0,565	0,095	0,756	0,094	0,778	0,082	1,107

In an effort of concision, we do not display the metrics of all tested datasets, we focus on those with a great number of features, as we think they illustrate better the phenomenon we want to observe.

We want to describe the general shape of the metrics plots. Whether it is ascending (aCC) or descending (aRRMSE) according to the metrics, the curves of both methods follow a same pattern: there is first a steep progression for the most significant variables, towards a progressive softening of the slope or even often an optimum as for ERCFR, in the case of *atp1d*, *atp7d* or *oes10*. Then, as variables become less and less relevant towards the prediction, the curves move to a plateau, meaning that the further inclusion of variables is no longer relevant towards the prediction, if not introducing noise and/or bias.

The most important part of the curve is this first steep progression we described. It depicts the ability of the feature ranking method to select the most relevant features. The steeper the slope, the more relevant the features. As expected, ERCFR has most of the time a steeper curve than RFPCT, meaning that it goes further and quicker into optimality. The performance of ERCFR generally decreases (respectively increases) swiftly at the beginning (the number of selected feature is small) in terms of aRRMSE (respectively aCC) and slows down at the end. This characteristic suggests that ERCFR ranks the features properly and that a single target regression algorithm can achieve a very good MTR performances

Consequently, it means that ERCFR generally outperforms RFPCT in terms of MTR metrics value. In most cases and for

every number of features, ERCFR achieves a better metrics value, meaning its prediction is most of the time accurate. In other words, for a same number of used features, ERCFR offers a better variable selection regarding the predicted variables.

These findings show that local variable selection outperforms global variable selection in a MTR context, when dealing with a consequent number of features. This is important in view of the current state of the art global variable selection methods. The difference is however less visible when having a smaller number of variables.

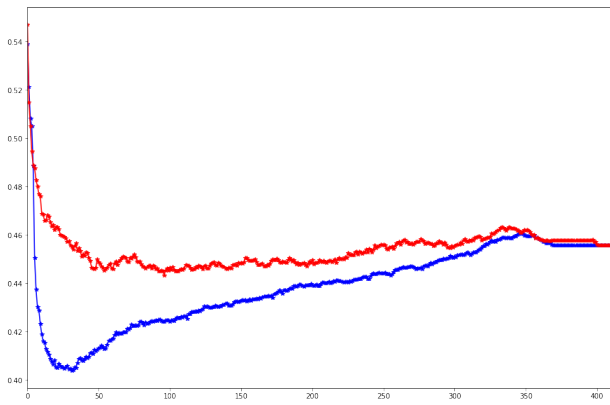
V. CONCLUSION

In this paper, we introduced a feature ranking method, termed as ensemble of regressor chains guided by feature ranking, based on a combination of Ensemble of Regressor Chains and Random Forest permutation importance measure. Our approach leads to a local feature ranking per target by appropriately modeling both dependencies among features and targets in the feature ranking process. The main advantage of this approach is that it simultaneously estimates the importance for each feature regarding each target individually and the intrinsic inter-target relevance as well. The information given by this importance is the core of our approach, and can be applied in different MTR contexts. First, it can be used as weights when computing distances for a distance-based model, like kNN. The model thus takes into account the relevance of each feature regarding the prediction of each target. Furthermore, when the order of prediction is an impactful parameter, as it is in a Regressor Chain context, the importance between targets allows to determine an optimal order of prediction. Last but not least, importance can be used for variable selection when choosing the most relevant features is decisive. Our approach is thus based on local variable selection, as opposed to global variable selection methods, like RFPCT. Experiment showed that the use of local variable selection almost always leads to better predictive performance. This is particularly true when dealing with a great amount of features.

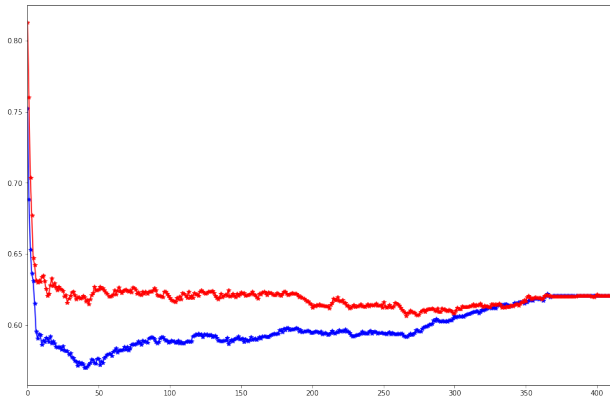
While the emphasis in this paper was on estimating feature importance in supervised multi-target regression, it is worth mentioning that the idea underlying our approach may also be extended to semi-supervised learning feature importance evaluation in the MTR context.

REFERENCES

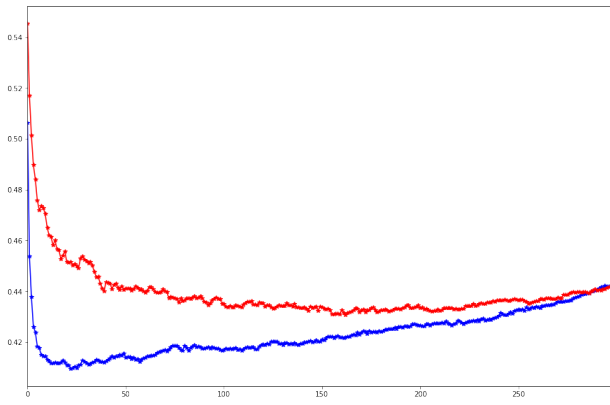
- [1] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 5(5):216–233, 2015.
- [2] Dihia Boulegane, Albert Bifet, Haytham Elghazel, and Giyyarpuram Madhusudan. Streaming time series forecasting using multi-target regression with dynamic ensemble selection. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2170–2179. IEEE, 2020.
- [3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [4] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [5] Dragi Kocev, Sašo Džeroski, Matt D. White, Graeme R. Newell, and Peter Griffioen. Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, 220(8):1159–1168, 2009.
- [6] Dragi Kocev, Celine Vens, Jan Struyf, and Saso Dzeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.
- [7] Haiqing Li, Wei Zhang, Ying Chen, Yumeng Guo, Guo-Zheng Li, and Xiaoxin Zhu. A novel multi-target regression framework for time-series prediction of drug efficacy. *Scientific reports*, 7(1):1–9, 2017.
- [8] Saulo Martiello Mastelini, Victor Guilherme Turrissi da Costa, Everton Jose Santana, Felipe Kenji Nakano, Rodrigo Capobianco Guido, Ricardo Cerri, and Sylvio Barbon Junior. Multi-output tree chaining: An interpretative modelling and lightweight multi-target approach. *Signal Processing Systems*, 91(2):191–215, 2019.
- [9] Gabriella Melki, Alberto Cano, Vojislav Kecman, and Sebastián Ventura. Multi-target support vector regression via correlation regressor chains. *Information Science*, 415:53–69, 2017.
- [10] Gabriella Melki, Alberto Cano, Vojislav Kecman, and Sebastián Ventura. Multi-target support vector regression via correlation regressor chains. 415-416:53–69.
- [11] Jose M. Moyano, Eva Lucrecia Gibaja Galindo, and Sebastián Ventura. An evolutionary algorithm for optimizing the target ordering in ensemble of regressor chains. In *2017 IEEE Congress on Evolutionary Computation, CEC 2017, Donostia, San Sebastián, Spain, June 5-8, 2017*, pages 2015–2021, 2017.
- [12] Matej Petkovic, Saso Dzeroski, and Dragi Kocev. Feature ranking for multi-target regression with tree ensemble methods. In *Discovery Science - 20th International Conference, DS 2017, Kyoto, Japan, October 15-17, 2017, Proceedings*, pages 171–185, 2017.
- [13] Matej Petkovic, Dragi Kocev, and Saso Dzeroski. Feature ranking for multi-target regression. *Machine Learning*, 109(6):1179–1204, 2020.
- [14] Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. In *ECML PKDD*, pages 313–325, 2008.
- [15] Everton Jose Santana, Felipe Rodrigues dos Santos, Saulo Martiello Mastelini, Fabio Luiz Melquiades, and Sylvio Barbon Jr. Improved prediction of soil properties with multi-target stacked generalisation on EDXRF spectra. *arXiv preprint arXiv:2002.04312*, 2020.
- [16] G. Tsoumakas, E.S. Xioufis, J. Vilcek, and I.P. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 2011.
- [17] Devis Tuia, Jochem Verrelst, Luis Alonso, Fernando Pérez-Cruz, and Gustavo Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 8(4):804–808, 2011.
- [18] Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis P. Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.



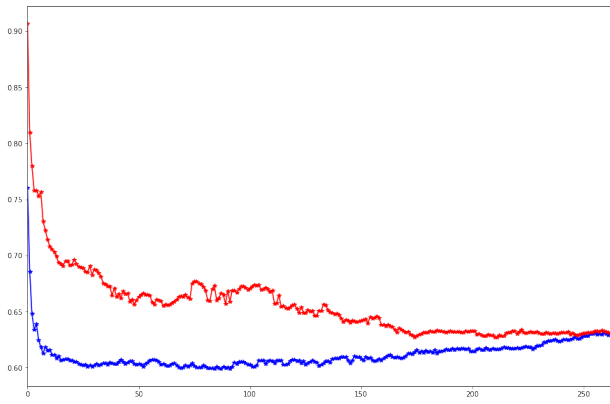
(a) atp1d



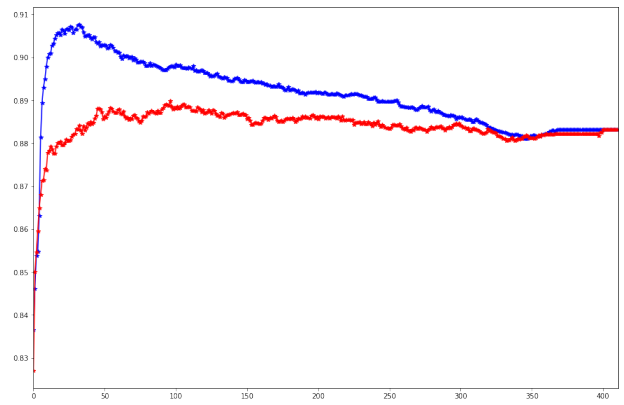
(b) atp7d



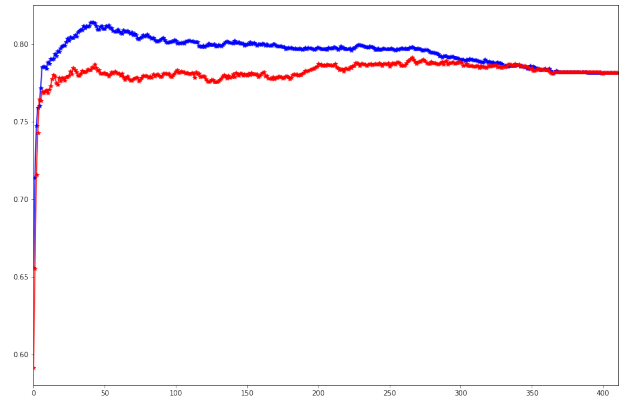
(c) oes10



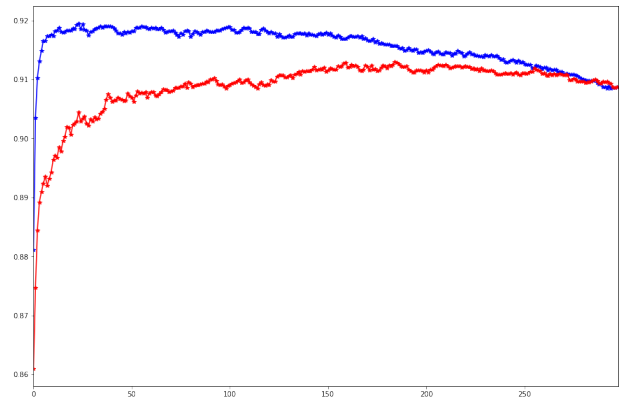
(d) oes97



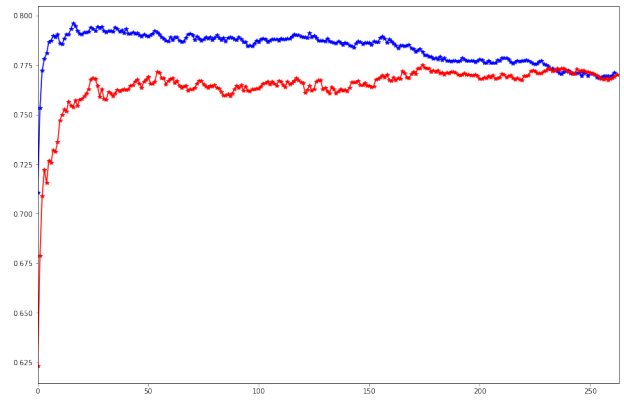
(a) atp1d



(b) atp7d



(c) oes10



(d) oes97

Fig. 1: Evolution of aRRMSE as the number of most relevant features used for prediction increases.

Fig. 2: Evolution of aCC as the number of most relevant features used for prediction increases.