Projective Operator-based Explanations for Multi-Output Prediction Models in Semiconductor Industry

Amina Mević [1], **Sandor Szedmak** [2], **Senka Krivić** [1]
[1]Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina
[2] Aalto University, Helsinki, Finland
amevic@etf.unsa.ba

Abstract

In industrial applications, achieving accuracy and understanding prediction outputs are both crucial, particularly when dealing with large datasets with numerous features and multi-prediction targets. This paper presents a novel method grounded in the projective operation concept, designed to provide explanations for multi-output prediction models (ProjEx). Our approach is model-agnostic, ensuring wide applicability across various contexts. We validate the robustness and efficiency of our method within a semiconductor production virtual metrology (VM) setup. The need to explain the multi-output learning approach in VM arises due to the interrelated properties of a product that should be predicted simultaneously, and experts monitoring production must understand and trust these prediction outputs. Furthermore, we introduce a stability index to rigorously evaluate the reliability of the explanations generated by our method. ProjEx outperforms SHAP and TreeInterpreter in computation time, while the introduced stability index and correlation are comparable.

Introduction

Artificial intelligence (AI) has seen dramatic advancements recently. It is being utilized to make suggestions to humans' decisions in diverse domains, such as education, health care, news, entertainment, travel, logistics, manufacturing, law enforcement, and finance (Rai 2020). As AI systems become increasingly integrated into various aspects of society, it is crucial to understand how users perceive and trust these systems. Explainable AI (XAI) helps make AI systems transparent, understandable, and trustworthy. Arrieta et al. (2020) described explainable AI as a suite of algorithmic techniques that generate high-performance explainable models humans can easily understand and trust. Trust in the AI system depends on fairness, explainability, accountability, privacy, and user acceptance (Kaur et al. 2022).

Multi-output learning is a machine learning (ML) paradigm that aims to predict multiple outputs simultaneously given an input (Xu et al. 2019). Examples include *multi-label learning*, where multi labels are assigned to an instance, *multi-dimensional learning*, where multi-tasks are learned together, and *multi-target regression*, where multi-continuous outputs are predicted. These methods improve prediction accuracy by capturing complex relationships between outputs, essential in scenarios with multiple interrelated factors.

The need for a multi-output learning approach arises in several applications such as virtual metrology (VM) in the semiconductor manufacturing industry (Choi et al. 2024), healthcare (Cui et al. 2018), environmental science (Džeroski, Demšar, and Grbović 2000), multi-classes image classification (Dong, Zhu, and Gong 2019), text processing (Sanh, Wolf, and Ruder 2019), weather and air quality forecasting (Liang et al. 2023). However, the integration of such predictions in industrial processes is aggravated due to a lack of transparency and explainability of the prediction process and its outcomes.

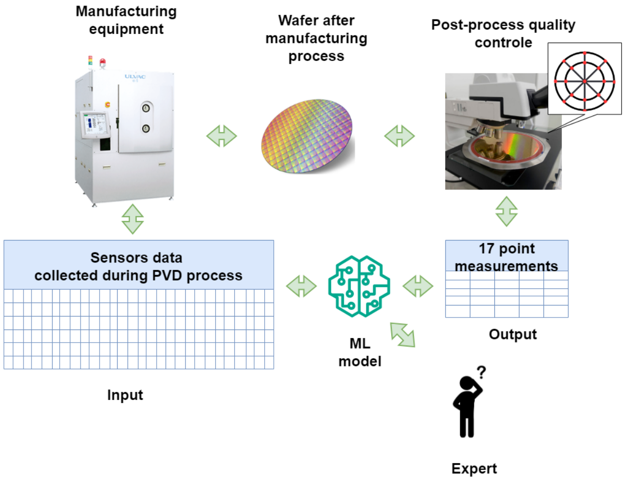

Manufacturing equipment
Wafer after manufacturing process
Post-process quality controle
Sensors data collected during PVD process
Input
ML model
17 point measurements
Output
Expert

Figure 1: An example of using VM to replace post-process quality control measurements. The VM system uses ML models to predict wafer properties at 17 characteristic points from sensor data and classify products, eliminating the need for post-process quality measurements. Experts who rely on predictions to adjust process parameters and schedule maintenance need to understand and trust the model outputs.

Manufacturing semiconductors involves several steps, including coating and removing layers from silicon bases, known as wafers. They have a disk-like shape, are made of pure silicon, and, as the final products contain thousands of chips on average. VM introduced in 2005 in the semiconductor manufacturing industry (Chen et al. 2005), involves estimating a product's quality directly from production process data, using supervised or unsupervised ML algorithms, without physically measuring it (Dreyfus et al. 2022), and in this way reducing production times and costs. The VM process addressed within this paper is demonstrated in Fig. 1.

Multi-output approaches are overlooked in deep learning-based VM modeling, although the joint information among the process outputs can improve prediction performance. Choi et al. (2024) proposed a CNN-based multivariate VM model using multi-sensor process sensor data and evaluated the proposed model for VM modeling at an etching process. Yamaguchi and Yamashita (2024) proposed a multi-target regression method that combines Random Linear Target Combinations and PCA. However, there is a lack of research based on explainability and interoperability.

Production equipment sensors continuously monitor a large number of signals. There is also a need to use feature selection and dimensionality reduction methods in VM systems, and they are often considered as a separate element (Djedidi et al. 2022). Several feature selection methods (Boyd et al. 2018; Brouard et al. 2022; Jordan, Liu, and Ruan 2021; Song et al. 2012) have been proposed for structured output learning tasks. Still, they use separate kernels for data samples in input and output spaces and may not scale well to large sample sizes. Szedmak et al. (2023) proposed a novel approach for variable selection for vector-valued or two-view learning problems utilizing projection operators and their algebra - the ProjSe algorithm. This paper builds upon their findings, introduces Projective Operator-based Explanations (ProjEx), and provides a novel and transparent solution that enables more accurate predictions and a deeper understanding of the importance of individual variables.

Numerous methods have been proposed for interpretability and explainability in single-output prediction scenarios, including well-known techniques like SHAP (Kariyappa et al. 2024) and LIME (Ribeiro, Singh, and Guestrin 2016). These foundational methods have given rise to more advanced approaches, such as Counterfactual Shapley Additive Explanations (Albini et al. 2022). To the best of our knowledge, no methods specifically designed to explain multi-output predictions are currently available in the literature. The main contributions of this paper are:

- We present a projective operator-based framework for interpreting multi-output predictions (ProjEx) and introduce its stability measure.
- We showcase the effectiveness of the proposed method on the real-world problem of VM in the semiconductor manufacturing industry.To the best of our knowledge, this paper utilizes a feature selection method for vector-valued output as part of a VM system for the first time.

Preliminaries

Multi-Ouptput Prediction involves mapping each input (instance) to multi outputs. Given that $X \in \mathbb{R}^{n_x}$ represents a n_x-dimensional input space and $Y \in \mathbb{R}^{n_y}$ represents an n_y-dimensional output space, the goal is to learn a function $f : X \to Y$ from a training set $D = \{(x_i, y_i) \mid 1 \leq i \leq m\}$, where m is number of samples. Here, each training example (x_i, y_i) consists of a n_x- dimensional feature vector $x_i \in X$, and $y_i \in Y$ represents the corresponding output linked to x_i (Borchani et al. 2015; Tsochantaridis et al. 2005; Xu et al. 2019). The general framework for multi-output learning can be described as follows: the task is to find a function $F : X \times Y \to \mathbb{R}$ using a training set of input-output pairs, where $F(x, y)$ serves as a compatibility function that measures how well the input x and output y align. When presented with a new instance x during testing, the output is predicted to be the one that maximizes the compatibility score, specifically $f(x) = \hat{y} = \arg\max_{y \in Y} F(x, y)$ (Borchani et al. 2015).

Feature Selection is the process of selecting a subset of relevant features from a larger set $X_s \subset X$ for use in model construction thereby improving model performance and interpretability. For given a dataset represented as X with n_x features and m samples, the goal of feature selection is to find a subset of features $X_s \subset X$ such that the predictive model built using X_s achieves the highest possible accuracy. This can be mathematically expressed as: maximize $f(X_s)$ subject to $|X_s| \leq k$ where:$f(X_s)$ is a performance metric (e.g., accuracy, F1 score) evaluated on the model trained with features in X_s, $|X_s|$ denotes the number of features in subset X_s, k is a predefined limit on the number of features to select (Lutu et al. 2010).

Least-Square Regression is a fundamental method in statistics and ML used to find the best-fitting line or hyperplane that minimizes the sum of the squared differences between the observed values and the values predicted by the model (Farebrother 2018). This method is widely used for linear regression analysis to model the relationship between a dependent variable and one or more independent variables. Given a dataset with m samples and n_x features (or variables), let $\mathbf{X} \in \mathbb{R}^{m \times n_x}$ represent the matrix of input features, and let $\mathbf{y} \in \mathbb{R}^m$ represent the vector of observed outcomes. The goal of least squares regression is to find the vector of coefficients $\mathbf{w} \in \mathbb{R}^{n_x}$ that minimizes the residual sum of squares: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} \left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2$ where: y_i is the observed outcome for the i-th instance, $\mathbf{x}_i^\top$ is the transpose of the i-th row of the matrix $\mathbf{X}$, representing the feature values for the i-th instance, $\mathbf{w}$ is the vector of coefficients to be estimated, $\hat{\mathbf{w}}$ denotes the estimated coefficients that minimize the sum of squared residuals. The least squares solution can also be expressed in matrix form as: $\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ where $\mathbf{X}^\top \mathbf{X}$ is the Gram matrix, and $\mathbf{X}^\top \mathbf{y}$ is the vector of correlations between the features and the output variables.

Methodology

Szedmak et al. (2023) proposed a novel approach for variable selection for vector-valued or two-view learning problems utilizing projection operators and their algebra. The method incorporates a kernel-based representation of the variables, enabling the capture of complex and nonlinear relationships. The proposed method is scalable and can handle large-scale selection tasks with millions of data samples. Due to the properties of the projection operators, ProjSe ensure invariance: the selected variables depend only on the subspace spanned by the target variables and are independent of any transformation on the response variables that would span the same subspace. We build upon these findings and create important values of features based on the model-trained data to provide explanations for end users.

Features importance The properties of projection operators into subspaces of a Hilbert space can be used to measure the correlation between input features n_x and a set of output variables n_y. As Szedmak et al. (2023) did, we observe all potential variables that can be used as features and selecting the most significant ones. The correlation between an unselected input variable $\mathbf{x}$ and the subspace spanned by the outputs $\mathbf{Y}$ after selecting t input variables, where $(t = 0, \ldots, min(n_y, n_x))$, is given by

$$\begin{aligned} \mathrm{corr}_{\mathbf{X}_t}(\mathbf{x}, \mathbf{Y}) &= \left\| \mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\mathbf{X}_t^\perp}} \frac{\mathbf{x}}{||\mathbf{x}||} \right\| \\ &= \left\langle \mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\mathbf{X}_t^\perp}} \frac{\mathbf{x}}{||\mathbf{x}||}, \mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\mathbf{X}_t^\perp}} \frac{\mathbf{x}}{||\mathbf{x}||} \right\rangle^{\frac{1}{2}} \\ &= \left\langle \frac{\mathbf{x}}{||\mathbf{x}||}, \mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\mathbf{X}_t^\perp}} \frac{\mathbf{x}}{||\mathbf{x}||} \right\rangle^{\frac{1}{2}} \end{aligned} \tag{1}$$

where $\mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\mathbf{X}_t^\perp}}$ is the orthogonal projection operator into the intersection of the subspace $\mathcal{L}_Y$ spanned by output variables and the orthogonal complement of the subspace $\mathcal{L}_{\mathbf{X}_t^\perp}$ spanned by the input variables selected earlier.

Determination of the importance of variables and their selection is based on efficient iterative computation of projections of input variables into the intersection of the space spanned by all output variables and the orthogonal complement of the space of the previously selected input variables. This projection guarantees that the selected input variable has a high correlation to all output variables, but in contrast, the correlation between a newly selected input and the previously selected ones is minimized. This is the foundation of the ProjSe selection algorithm given within Algorithm 1 while the graphical demonstration is shown in Fig. 2.

Kernel-based representation of the feature selection problem allows exploring complex, nonlinear relationships between all the variables appearing in the available data set, (Szedmak et al. 2023). A kernel can be interpreted as a function which can express the similarity between pairs of vectors in a high-dimensional reproducing kernel Hilbert space without explicitly determining the coordinates of those vectors (Hofmann, Schölkopf, and Smola 2008). The usage of a kernel function allows including non-linearity to the models implicitly via a feature map $\varphi : \mathcal{X} \rightarrow \mathcal{F}_k$: a kernel evaluated with two samples corresponding to an inner product in this so-called feature space : $k(x, z) = \langle \varphi(x), \varphi(z) \rangle_{\mathcal{F}_k}$. We investigate the difference between selections where linear and Gaussian kernel is applied, and the stability of the variable selection process regarding Gaussian kernel parameters is discussed. The linear kernel calculates the dot product between pairs of data points in the original feature space. The N-dimensional Gaussian kernel is defined as

$$G_{ND}(\vec{x}; \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp^{-\frac{|\vec{x}|^2}{2\sigma^2}}, \tag{2}$$

where σ determines the width of the kernel, σ^2 the variance, N the dimensionality of the space, and $\vec{x}$ is an N-dimensional vector, (Shawe-Taylor and Cristianini 2004).

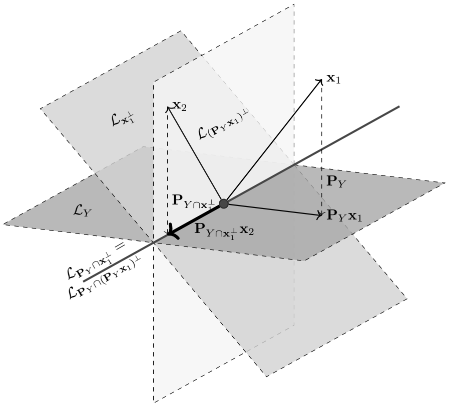

$\mathcal{L}_{\mathbf{x}_1^\perp}$
$\mathbf{x}_2$
$\mathcal{L}_{(\mathbf{P}_Y\mathbf{x}_1)^\perp}$
$\mathbf{x}_1$
$\mathbf{P}_Y$
$\mathcal{L}_Y$
$\mathbf{P}_{Y\cap\mathbf{x}_1^\perp}$
$\mathbf{P}_Y\mathbf{x}_1$
$\mathbf{P}_{Y\cap\mathbf{x}_1^\perp}\mathbf{x}_2$
$\mathcal{L}_{\mathbf{P}_Y\cap\mathbf{x}_1^\perp} =$
$\mathcal{L}_{\mathbf{P}_Y\cap(\mathbf{P}_Y\mathbf{x}_1)^\perp}$

Figure 2: The first step in the variable selection process is maximizing the projection of a feature $\mathbf{x}_1$ onto the subspace spanned by the vectors of output (prediction) variables $\mathcal{L}_Y$. Next feature, $\mathbf{x}_2$ is projected on the intersection of $\mathcal{L}_Y$ and $\mathcal{L}_{\mathbf{x}_1^\perp}$, subspace orthogonal to $\mathbf{x}_1$, and the correlation between $\mathbf{Y}$ and $\mathbf{x}_2$ is computed via the formula (1) which ensures a deterministic selection.

Projective Operator-based Explanations

The proposed framework for interpreting multi-output predictions, ProjEx, is based on projective operator algebra and utilizes the results from feature selection by ProjSe to provide local and global explanations of behavior ML models. Local explanations focus on understanding individual predictions by analyzing the specific features that influence a single instance's outcome, providing insights into why a model made a particular decision for a given input (Mor, Belinkov, and Kimelfeld 2024). Global explanations aim to provide an overarching understanding of the model's behavior across all predictions, analyzing the model's overall structure and feature importance to identify patterns and trends that apply to the entire dataset (Arya et al. 2023). This framework comprises four main components, illustrated in Fig. 3: a heatmap of the correlation matrix between input and output variables, global explanations, local explanations and predicted target values.

The procedure for generating explanations is as follows. First, all features are used to predict the

Algorithm 1: Variable Selection by Projection

```
Input:
    • A set of output variables $\{\mathbf{y}_1, \dots, \mathbf{y}_{n_y}\}$ in $\mathbb{R}^m$, represented by $\mathbf{Y} \in \mathbb{R}^{m \times n_y}$.
    • A set of input variables $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_x}\}$ in $\mathbb{R}^m$, collected into $\mathbf{X} \in \mathbb{R}^{m \times n_x}$.
    • $D \leq \min(n_y, n_x)$: number of variables to be chosen from $\mathbf{X}$.
Output: Set $\mathcal{I}_D$ of indices of selected variables from $\mathbf{X}$ in the selection order.
Initialize: Let $t = 0$ and $\mathcal{I}_t = \emptyset$. Set $\tilde{\mathbf{X}}_t = \mathbf{X}[:, \mathcal{I}_t]$; since $\mathcal{I}_t = \emptyset$, $\tilde{\mathbf{X}}_t^\perp = \mathbb{R}^m$.
while $t < D$ do
    Let $\mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\tilde{\mathbf{X}}_t^\perp}}$ be the projection into the intersection of $\mathcal{L}_Y$ and $\mathcal{L}_{\tilde{\mathbf{X}}_t^\perp}$.
    Choose an index $k_*$ by:
        $k_* = \arg \max_{k \in \{1, \dots, n_x\} \setminus \mathcal{I}_t} \left\| \mathbf{P}_{\mathcal{L}_Y \cap \mathcal{L}_{\tilde{\mathbf{X}}_t^\perp}} \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \right\|^2$
    Update $\mathcal{I}_{t+1} = \mathcal{I}_t \cup \{k_*\}$ and $\tilde{\mathbf{X}}_{t+1} = \mathbf{X}[:, \mathcal{I}_{t+1}]$.
    Increment $t = t + 1$.
end while
```

target variables using a regression model, such as `DecisionTreeRegressor (DTR)` (Clark and Pregibon 2017). Next, ProjSe is applied to select the features most correlated with the predicted target variables. For each selected feature $\mathbf{x}_\mathbf{s} \in \mathbf{X}_\mathbf{s}$, we find the projection of the vector $\mathbf{x}_\mathbf{s}$ onto the plane spanned by the predicted target variables $\mathbf{y}_\mathbf{p} \in \mathbf{Y}_\mathbf{p}$. The projection $\mathbf{x}_{\mathbf{s}_\text{proj}}$ of a $\mathbf{x}_\mathbf{s}$ onto the plane spanned by $\mathbf{y}_\mathbf{p}$ is given by: $\mathbf{x}_{\mathbf{s}_\text{proj}} = \mathbf{Y}_\mathbf{p}(\mathbf{Y}_\mathbf{p}{}^\top \mathbf{Y}_\mathbf{p})^{-1} \mathbf{Y}_\mathbf{p}{}^\top \mathbf{x}_\mathbf{s}$.

We calculate the correlation coefficient ρ between $\mathbf{x}_\mathbf{s}$ and $\mathbf{x}_{\mathbf{s}_\text{proj}}$ using: $\rho = \frac{\mathbf{x}_\mathbf{s} \cdot \mathbf{x}_{\mathbf{s}_\text{proj}}}{\|\mathbf{x}_\mathbf{s}\| \|\mathbf{x}_{\mathbf{s}_\text{proj}}\|}$. These correlation coefficients are presented as a bar plot in Fig. 3, with feature indices on the y-axis and coefficients on the x-axis. This bar plot provides a global explanation of the model by indicating the influence of features on the predicted targets through the projection coefficients.

To assess the impact of each feature value on the predicted target variables, we calculate $\mathbf{x}_\mathbf{i}$ vectors by scaling the selected feature with the correlation coefficient: $\mathbf{x}_\mathbf{i} = \mathbf{x}_\mathbf{s} \cdot \rho$. The procedure is shown within Algorithm 2. The bar plot in Fig. 3 shows the values of $\mathbf{x}_\mathbf{i}$ for a randomly selected sample. The x-axis represents the values of $\mathbf{x}_\mathbf{i}$, indicating the impact of the feature on the prediction, while the y-axis shows the selected features and their normalized values. Additionally, the normalized values of $\mathbf{y}_\mathbf{p}$ for the given point are provided. This approach creates local explanations, revealing the influence of features on the predicted targets for a single sample.

Based on projective operator algebra, ProjEx is exact and does not rely on probabilistic assumptions, ensuring that the explanations are deterministic and free from variance caused by stochastic elements. Furthermore, ProjEx is model-agnostic and can be applied to any predictive model.

Algorithm 2: Generating Explanations

```
Input: Matrix $\mathbf{X}$ (features), matrix $\mathbf{Y}$ (predicted target variables)
Select $\mathbf{X}_\mathbf{s}$ by ProjSe
for $\mathbf{x}_\mathbf{s} \in \mathbf{X}_\mathbf{s}$ do
    Compute the projection of $\mathbf{x}_\mathbf{s}$ onto the plane spanned by $\mathbf{y}_\mathbf{p} \in \mathbf{Y}_\mathbf{p}$:
        $\mathbf{x}_{\mathbf{s}_\text{proj}} = \mathbf{Y}_\mathbf{p}(\mathbf{Y}_\mathbf{p}{}^\top \mathbf{Y}_\mathbf{p})^{-1} \mathbf{Y}_\mathbf{p}{}^\top \mathbf{x}_\mathbf{s}$
    Calculate
        $\rho = \frac{\mathbf{x}_\mathbf{s} \cdot \mathbf{x}_{\mathbf{s}_\text{proj}}}{\|\mathbf{x}_\mathbf{s}\| \|\mathbf{x}_{\mathbf{s}_\text{proj}}\|}$
end for
for $\mathbf{x}_\mathbf{s} \in \mathbf{X}_\mathbf{s}$ do
    Calculate
        $\mathbf{x}_\mathbf{i} = \mathbf{x}_\mathbf{s} \cdot \rho$
end for
```

The global explanations, derived from projection coefficients, provide a understanding of how each feature contributes to the overall prediction while the local explanations allow analysis of how individual feature values impact specific predictions. This dual approach ensures that the explanations offer a broad view of model behavior, and granular, enabling detailed insight into individual predictions.

Stability

The "stability" of a feature selection algorithm refers to the robustness of its feature preferences, with respect to data sampling and to its stochastic nature. An algorithm is 'unstable' if a small change in data leads to large changes in the chosen feature subset. We use rigorous statistical treatment proposed in (Kuncheva 2007), (Nogueira, Sechidis, and Brown 2018) and (Hamer and Dupont 2021). The stability index measures the consistency of a feature selection process by determining whether two selections are considered identical if their unordered sets of selected indices are equal. In contrast, the stability correlation also applies weights on the variables to capture their importance, for example, the position in the selection. In our analysis, those weights are given by $1/n_i$, where a n_i is the position of variable i in the selection, thus, variables selected earlier have higher weights. The dependence on the selection order distinguishes the stability index from correlation; the former is insensitive to the order, but the latter is.

Relation with TreeInterpreter and SHAP

SHAP (SHapley Additive exPlanations) is a model-agnostic method that provides both local and global explanations using game-theoretic Shapley values (Shapley et al. 1953). By measuring the marginal contributions of features across different coalitions, SHAP assigns a score to each feature that reflects its contribution to the final prediction of the model. For ProjEx, the number of selected features matches the number of output variables, whereas in SHAP, the user must set a selection threshold, which can be subjective and may

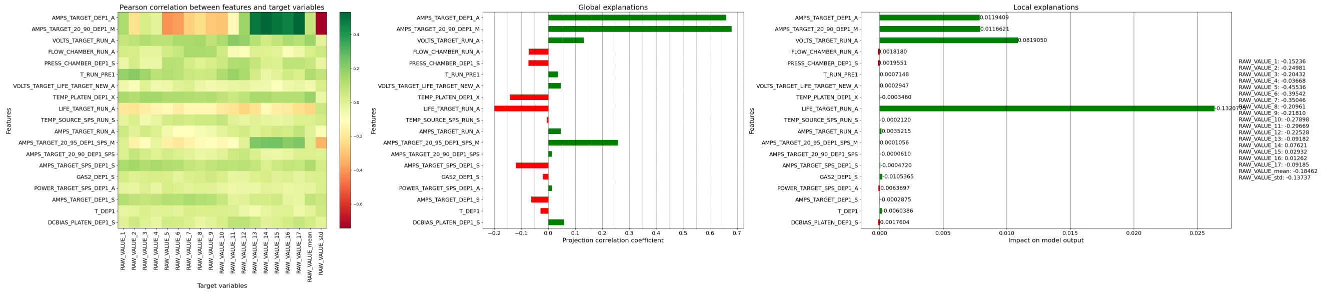


Figure 3: ProjEx Explanations: The image includes a heat map of the correlation matrix between input and output variables, with positive correlations shown in green and negative correlations in red. It also presents global explanations of the model through projection coefficients, indicating the influence of features on predicted targets. Features with red values decrease the target values, while green features increase them. Additionally, the image provides local explanations, showing the influence of features on predicted targets for one sample in a bar plot. In this plot, negative influences are highlighted in red, and positive influences in green. Normalized values of both the features and predicted target values are also shown.

require experimentation to gain optimal subset for prediction. Additionally, SHAP is resource-intensive with high-dimensional data, making the analysis impractical due to the vast number of potential feature combinations. SHAP is not suited for multi-output learning tasks.

TreeInterpreter is a model explanation method tailored for tree-based models, including decision trees, random forests, and gradient boosting machines (Sharma et al. 2020). It reveals each feature's contribution to a prediction by decomposing it based on decision paths in tree-based models. TreeInterpreter does not support multi-output prediction settings, despite the capability of tree-based models to handle multi-output tasks.

Dataset

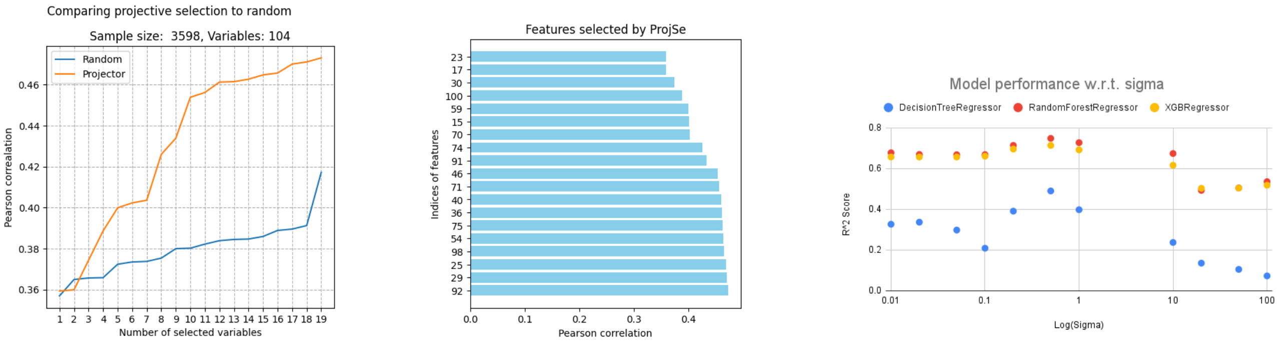
We use a real-world dataset from semiconductor manufacturing, focusing on the physical vapor deposition (PVD) process—one of the key production steps for creating thin layers by depositing metal vapor onto a substrate (Powell and Rossnagel 1999). The crucial physical properties of the film, thickness and resistance, depend mostly on deposition time, power, and temperature. Numerous sensors monitor these parameters throughout the PVD process to ensure optimal deposition conditions. After a process the product's physical properties are measured at 17 different points (Fig. 1). The dataset includes data collected from 16 chambers of six PVD machines at the same semiconductor manufacturing fab, Infineon Technologies AG, from 2021 to 2023. For over 3 years, 3598 procedures have been performed and considered as samples for this dataset. For each PVD sample procedure, 1007 attributes were collected: logistics attributes, process parameters defined by recipe, and values of these parameters during a process. After removing features with missing or constant values$n_x = 104$ features were remained. For each of the 17 points on the wafer resistance and thickness are measured and their product (resistivity) is calculated. In our multi-output learning problem, the outputs include resistivity values at all 17 points, their average, and their standard deviation, resulting in a total of $n_y = 19$ target variables.

Evaluation

Comparison of ProjSe with random selection We compared the prediction results using features selected by ProjSe with those selected randomly. To ensure unbiased evaluation, the data are centralized and normalized. We iteratively recalculate the least-squares regression model (Farebrother 2018) as features are added for features selected randomly and by ProjSe based on the introduced feature importance. Prediction accuracy is measured using Pearson correlation between actual Y and predicted outputs Y_p: $\rho_{Y,Y_p} = \frac{\text{Cov}(Y,Y_p)}{\sigma_Y \sigma_{Y_p}} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(y_{p_i} - \bar{y}_p)}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(y_{p_i} - \bar{y}_p)^2}}$.

Fig. 4a demonstrates how the number of selected features enhances prediction accuracy compared to random selection. The projective operator-based selection yields higher Pearson correlation coefficients, indicating a correlation between selected features and outputs. Fig. 4b shows the changes in Pearson correlation coefficients as features selected by ProjSe are added incrementally. The first 5 features capture the most relevant information for predicting the output, and subsequent features contribute minimally.

Stability and computation time w.r.t. dataset size Algorithm 1 was applied to the entire dataset (3,598 samples), the training dataset (2,878 samples), the test dataset (720 samples), and the test dataset with outputs predicted by a DTR (720 samples). Experiments were performed on a system equipped with dual Intel(R) Xeon(R) Gold 6448Y processors (32 cores), an NVIDIA H100 80GB HBM3 graphics card, and 503GB of RAM. In each case, there were selected 19 features. Table 1 presents the first 5 selected variables and the computation time for each dataset. The first three most important variables were consistent across the whole, training, and test datasets, but significant differences were observed between selections on actual and predicted outputs. The stability index was 0.84 for selections on the actual datasets and 0.82 when the dataset with predicted outputs was included. The stability correlation was 0.94 for datasets with actual outputs but dropped significantly to 0.49 when the dataset with predicted outputs was included. Execution



(a) The blue line shows how prediction accuracy changes when variables are added randomly, whereas the orange line demonstrates the performance when variables are chosen using the projective selection method.

(b) Features with indices 23, 17, 30, 100, and 59 contribute most significantly to prediction accuracy measured by Pearson correlation, with the correlation increasing with each added feature until saturation after the fifth feature (index 59).

(c) The plot illustrates the oscillatory behavior of the R^2 scores of DTR, RFR, and XGB models as a function of the Gaussian kernel parameter σ for different feature selection. The x-axis represents the σ values on a logarithmic scale. All models' highest R^2 scores were observed when σ was 0.5.

Figure 4: Examination of feature importance

time was measured 5 times, with average values shown in the Table 1. As expected, computation time decreased with the reduction in dataset size.

Dataset	Selected features	Time (ms)
Whole	23, 17, 30, 100, 59	6
Training	23, 17, 30, 100, 59	4
Test	23, 17, 30, 59, 99	2
Test-Predicted outputs	0, 19, 30, 100, 20	2

Table 1: Comparison of the first 5 selected features and computation time across different datasets.

Stability of feature selection with respect to Gaussian kernel parameters Eleven different σ values (0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 10, 20, 50, 100) were tested, resulting in 11 subsets of selected features. These features predicted 19 target variables, yielding a stability index of 0.82 and a stability correlation of 0.27. We used three tree-based models (Clark and Pregibon 2017) for prediction: DTR, RandomForestRegressor (RFR), and XGBoost (XGB). Fig. 4c shows the R^2 scores for prediction model as function of σ on a logarithmic scale. All models performed best with features selected at $\sigma = 0.5$, achieving R^2 scores of 0.49 (DTR), 0.75 (RFR), and 0.71 (XGB). The R^2 scores exhibited oscillatory behavior across different σ values. For comparison, when using all features, the R^2 scores were 0.46 (DTR), 0.73 (RFR), and 0.75 (XGB).

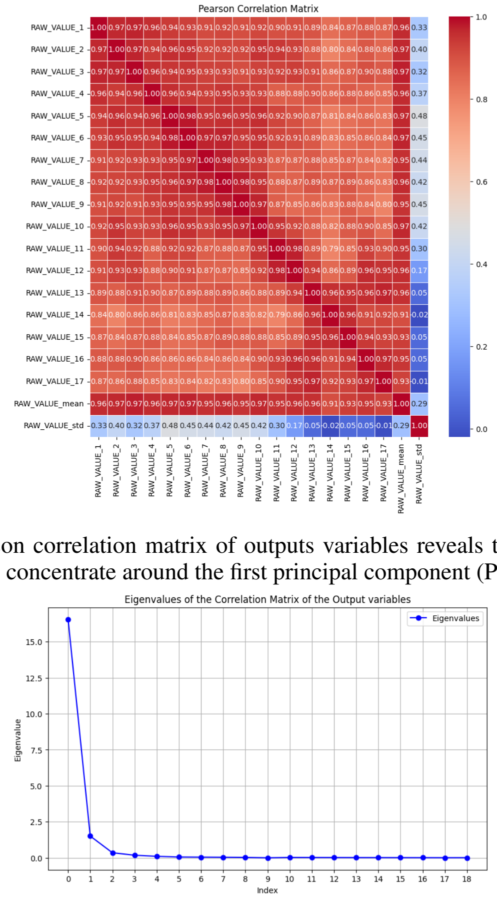
Fig. 5a shows the Pearson correlation of 19 output variables: 17 raw values, their mean value, and standard deviation. We expected uniform resistivity on the wafer, so we consider standard deviation as an indicator of uniformity. The raw values are mutually highly correlated and also correlated with the mean values, while the correlation with the standard deviation is significantly lower. Pearson correlation matrix of outputs variables reveals that those variables concentrate around the first principal component (PC) which might be interpreted as the general quality of the wafer, and the other, significantly weaker PCs relating to the location and geometry of the measurement points.

Comparison of ProjSe with tree-based feature selection methods We compared ProjSe with feature selection methods based on tree-based models for multi-output prediction. A 5-fold cross-validation was conducted to assess the stability of these feature selection methods. Table 2 presents the stability indices and correlations calculated across the 5 subsamples for each technique. While the stability indices are similar across the methods, ProjSe shows the lowest stability correlations. ProjSe also demonstrates superior computational efficiency, with average computation times significantly lower than tree-based methods.

Method	Index	Correlations	Time (s)
ProjSe	0.83	0.64	0.0036
DecisionTree	0.88	0.85	0.3120
RandomForest	0.95	0.95	18.2079
XGBoost	0.94	0.87	12.5884

Table 2: Feature selection method comparison.

ProjEx multi-output prediction explanations We compared ProjSe with two widely used model explanation methods: SHAP and TreeInterpreter. Since these methods do not inherently support multi-output prediction, we adapted them by selecting the most influential features for each of the 19 output variables in a 5-fold cross-validation setting to ensure a fair comparison. The stability, computed independently for each output variable across the 5 folds, is generally high for all methods. However, when stability was assessed for each fold by considering the results across all output variables as samples within the folds, it was significantly lower due to the lack of consideration for interactions between the output variables. Table 3 presents the average stability indices



(a) Pearson correlation matrix of outputs variables reveals that those variables concentrate around the first principal component (PC).

(b) Eigenevalues of the Pearson correlation matrix

Figure 5: Pearson correlation of the output variables

and correlations across all folds. ProjSe achieved the highest stability index, while the stability correlation was equal for ProjSe and TreeInterpreter, and slightly lower for SHAP. For SHAP and TreeInterpreter, the summed average time across 5 folds to obtain explanations for each of the 19 variables is shown. ProjSe's explanation time is 3 orders of magnitude faster than TreeInterpreter and 4 orders than SHAP.

The concentration of the distribution of the PCs in Fig.5b can explain the significant difference between the values of the stability index and correlation in Tables 3 and 4. The weak secondary PCs allow only the selection of the unordered set of best-performing input variables, stability index, but do not yield sufficient information to accurately determine their order or stability correlation, except those that highly correlate with the first PC.

Similarity between explanations w.r.t. different prediction models The results are presented in Table 4. The stability of ProjEx varies slightly depending on the used model. The highest stability index was observed with RFR, while DTR yielded the highest correlation and the lowest execution time.

Physical consistency of feature selection In the PVD process, resistivity is primarily influenced by the current and voltage applied to the target, as well as the deposition time and temperature (Powell and Rossnagel 1999). The first seven variables selected by ProjSe represent aggregated values of current, voltage, and time, aligning with the physical principles of the PVD process.

Method	Index	Correlations	Time (s)
ProjEx	0.96	0.56	0.002
SHAP	0.88	0.54	54.64
TreeInterpreter	0.90	0.56	2.992

Table 3: Explanantion methods comparison.

Method	Index	Correlations	Time (ms)
DecisionTree	0.80	0.53	2.1
RandomForest	0.85	0.42	2.5
XGBoost	0.81	0.52	2.7

Table 4: Feature selection performance across models.

Conclusion

In this paper, we have introduced a novel, model-agnostic method based on the projective operation concept to provide explanations for multi-output prediction models. Our approach has demonstrated robustness and efficiency in the context of semiconductor production VM, showcasing its practical applicability in industrial settings where accuracy and interpretability are paramount. Additionally, we have examined a stability index to rigorously assess the reliability of the generated explanations, further enhancing the utility and trustworthiness of our method.

Acknowledgments

The work is supported by the IPCEI on ME/CT program of Infineon Technologies Austria AG.

References

Albini, E.; Long, J.; Dervovic, D.; and Magazzeni, D. 2022. Counterfactual shapley additive explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*.

Arrieta, A. B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*.

Arya, V.; Saha, D.; Hans, S.; Rajasekharan, A.; and Tang, T. 2023. Global Explanations for Multivariate time series models.

Borchani, H.; Varando, G.; Bielza, C.; and Larranaga, P. 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Boyd, N.; Hastie, T.; Boyd, S.; Recht, B.; and Jordan, M. I. 2018. Saturating splines and feature selection. *Journal of Machine Learning Research*.

Brouard, C.; Mariette, J.; Flamary, R.; and Vialaneix, N. 2022. Feature selection for kernel methods in systems biology. *NAR genomics and bioinformatics*.

Chen, P.; Wu, S.; Lin, J.; Ko, F.; Lo, H.; Wang, J.; Yu, C.; and Liang, M. 2005. Virtual metrology: A solution for wafer to wafer advanced process control. In *ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005.*

Choi, J.; Zhu, M.; Kang, J.; and Jeong, M. K. 2024. Convolutional neural network based multi-input multi-output model for multi-sensor multivariate virtual metrology in semiconductor manufacturing. *Annals of Operations Research*.

Clark, L. A.; and Pregibon, D. 2017. Tree-based models. In *Statistical models in S*.

Cui, L.; Xie, X.; Shen, Z.; Lu, R.; and Wang, H. 2018. Prediction of the healthcare resource utilization using multi-output regression models. *IISE Transactions on Healthcare Systems Engineering*.

Djedidi, O.; Clain, R.; Borodin, V.; and Roussy, A. 2022. Feature Selection for Virtual Metrology Modeling: An application to Chemical Mechanical Polishing. In *2022 33rd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*.

Dong, Q.; Zhu, X.; and Gong, S. 2019. Single-label multi-class image classification by deep logistic regression. In *Proceedings of the AAAI conference on artificial intelligence*.

Dreyfus, P.-A.; Psarommatis, F.; May, G.; and Kiritsis, D. 2022. Virtual metrology as an approach for product quality estimation in Industry 4.0: a systematic review and integrative conceptual framework. *International Journal of Production Research*.

Džeroski, S.; Demšar, D.; and Grbović, J. 2000. Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*.

Farebrother, R. W. 2018. *Linear least squares computations*.

Hamer, V.; and Dupont, P. 2021. An Importance Weighted Feature Selection Stability Measure. *Journal of Machine Learning Research*.

Hofmann, T.; Schölkopf, B.; and Smola, A. J. 2008. Kernel methods in machine learning. *The annals of statistics*, 1171–1220.

Jordan, M. I.; Liu, K.; and Ruan, F. 2021. On the self-penalization phenomenon in feature selection. *arXiv preprint arXiv:2110.05852*.

Kariyappa, S.; Tsepenekas, L.; Lécué, F.; and Magazzeni, D. 2024. SHAP@ k: Efficient and Probably Approximately Correct (PAC) Identification of Top-k Features. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kaur, D.; Uslu, S.; Rittichier, K. J.; and Durresi, A. 2022. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)*.

Kuncheva, L. I. 2007. A stability index for feature selection. In *Artificial intelligence and applications*.

Liang, Y.; Xia, Y.; Ke, S.; Wang, Y.; Wen, Q.; Zhang, J.; Zheng, Y.; and Zimmermann, R. 2023. Airformer: Predicting nationwide air quality in china with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Lutu, P. E. N.; et al. 2010. *Dataset selection for aggregate model implementation in predictive data mining*. Ph.D. thesis, University of Pretoria.

Mor, A. R.; Belinkov, Y.; and Kimelfeld, B. 2024. Accelerating the Global Aggregation of Local Explanations.

Nogueira, S.; Sechidis, K.; and Brown, G. 2018. On the Stability of Feature Selection Algorithms. *Journal of Machine Learning Research*.

Powell, R. A.; and Rossnagel, S. M. 1999. *PVD for microelectronics: sputter deposition applied to semiconductor manufacturing*.

Rai, A. 2020. Explainable AI: From black box to glass box. *Journal of the Academy of Marketing Science*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

Sanh, V.; Wolf, T.; and Ruder, S. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 6949–6956.

Shapley, L. S.; et al. 1953. A value for n-person games.

Sharma, P.; Mirzan, S. R.; Bhandari, A.; Pimpley, A.; Eswaran, A.; Srinivasan, S.; and Shao, L. 2020. Evaluating tree explanation methods for anomaly reasoning: A case study of SHAP TreeExplainer and TreeInterpreter. In *Advances in Conceptual Modeling: ER 2020 Workshops CMAI, CMLS, CMOMM4FAIR, CoMoNoS, EmpER, Vienna, Austria, November 3–6, 2020, Proceedings 39*.

Shawe-Taylor, J.; and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*.

Song, L.; Smola, A.; Gretton, A.; Bedo, J.; and Borgwardt, K. 2012. Feature Selection via Dependence Maximization. *Journal of Machine Learning Research*.

Szedmak, S.; Huusari, R.; Duong Le, T. H.; and Rousu, J. 2023. Scalable variable selection for two-view learning tasks with projection operators. *Machine Learning*, 1–20.

Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y.; and Singer, Y. 2005. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*.

Xu, D.; Shi, Y.; Tsang, I. W.; Ong, Y.-S.; Gong, C.; and Shen, X. 2019. Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*.

Yamaguchi, T.; and Yamashita, Y. 2024. Multi-target regression via target combinations using principal component analysis. *Computers & Chemical Engineering*.