Feature Selection to Address High Dimensionality in Industry 4.0 Multi-Emitter Laser Modules Assembly Lines

Nikolaos Grigorios Markatos, *Imperial College London, London SW7 2BX, U.K.*

Alireza Mousavi, *Brunel University London, Greater London UB8 3PH, U.K.*

Evina Katsou, *Imperial College London, London SW7 2BX, U.K.*

Giulia Pippione and Roberto Paoletti, *Convergent Photonics Italia, Torino 10148, Italy*

Industry 4.0 has increased data depth and breadth in high-tech manufacturing, but high dimensionality and sparsity persist. High-dimensional space's sparsity makes classical learning and knowledge extraction algorithms ineffective and error-prone. Dimension reduction methods like feature selection (FS) seem to address this problem. This study addresses these challenges by conducting a comparative analysis on a real laser assembly industrial case of high dimensions. We explore five stand-alone methods, NCFS, RReliefF, minimum redundancy maximum relevance (MRMR), recursive feature elimination (RFE), and least absolute shrinkage and selection operator (LASSO) [applied to datasets from two laser modules (d-serie and s-serie)]. Additionally, two hybrid methods, RReliefF-RFE and MRMR-RFE, are evaluated, broadening the scope of FS strategies. Time efficiency prioritizes RReliefF, NCFS, and LASSO, while RReliefF-RFE, NCFS, and LASSO excel in interpretability, achieving significant predictor reduction without compromising accuracy. The study thus provides insights into the selection of FS methods in a challenging industrial laser assembly setting.

Automation in the industry has led to the accumulation of high-dimensional datasets, demanding increased computational power and storage capacity. Handling high-dimensional data introduces a challenge known as the *curse of dimensionality*. This problem refers to the sparsity of the data in high-dimensional space and the negative effect it has on algorithms that are designed for low-dimensional space. Dimension reduction techniques like feature selection (FS) prove effective in mitigating these challenge.[1]

In multistep laser assemblies, the datasets that are available are usually high dimensional. This makes it difficult for predictive algorithms to be applied in real time as the irrelevant or inconsistent data may pollute the state space and reduce the accuracy of the predictive models. Thus, efficient FS is essential for rationalizing and controlling knowledge extraction from the assembly process.

This article focuses on studying various FS methods to decide which of those perform better in multistep laser assembly processes. Two filter methods [minimum redundancy maximum relevance (MRMR) and RReliefF], one wrapper method [recursive feature elimination (RFE)], and two embedded methods [neighborhood component FS (NCFS) and least absolute shrinkage and selection operator (LASSO)] are tested for FS. Furthermore, combinations of filters and wrappers (MRMR-RFE and RReliefF-RFE) are tested. Each FS method creates a subset of predictors. These subsets are then fed to two regression methods [artificial neural networks (ANNs) and XGBoost] with the aim to

Digital Object Identifier 10.1109/MIS.2024.3416678
Date of publication 21 June 2024; date of current version 12 June 2025.

predict the output power of the module. Based on the performance of the regression methods, we can decide which FS methods created optimal subsets of predictors, and thus performed better.

These regression methods were chosen as they are considered among the most suitable algorithms in the field of machine learning, lending themselves to the problem statement.

The FS methods are applied on two laser modules' assembly datasets (d-serie and s-serie) from a laser manufacturing. The datasets (datasets of two different laser models) contain predictors from all the stages of the assembly processes of the laser modules. In both datasets, the "output" variable that needs to be predicted is the final power of the laser module.

The results reveal that RReliefF, RFE, NCFS, RReliefF-RFE, and LASSO are the five methods that create subsets of predictors that perform well in both datasets, and their performance is not dependent on the choice of the regression model that will be used for predicting the output power of the module. The researcher and the practitioner can choose the most appropriate method based on the industrial setting. If the computational time vis-à-vis the product lifecycle is important, then RReliefF, NCFS, and LASSO are the best options. If interpretability (as discussed in the "Results" section) is the main focus, then RReliefF-RFE, NCFS, and LASSO are the best options.

RELEVANT WORK

FS methods are being used in many areas as dimension reduction techniques. Although the work around FS is vast, in this section, we present some representative works where FS has been successfully applied to enhance interpretability, improve the accuracy of later-applied prediction models, and improve operation speed. In St. John et al.,[2] RFE with a random forest (RF) classifier is used to analyze machining audio collected during down milling operations using a single microphone. The incorporation of RFE facilitates straightforward feature elimination, producing a concise and easily interpretable set of analyzed dimensions (interpretability). In Wu et al.,[3] the authors introduce a model-based approach for predicting air quality index (AQI). To optimize the model's efficacy, the RF–mRMR technique is employed for the selection of influential variables that influence AQI (accuracy improvement).

In Houndekindo and Ouarda,[4] the study assessed six FS methods to estimate wind speed quantiles in Canada. LASSO and MRMR emerged as the most efficient algorithms, requiring fewer parameters with good generalization performance, and identified specific predictors as more important for distinct exceedance probabilities (accuracy improvement).

In Wen and Xu,[5] the authors introduce a hybrid fault diagnosis method aimed at addressing challenges regarding the large amount of data generated during the operation of wind turbines, and inaccurate and untimely fault diagnosis for wind turbines. The proposed approach leverages RReliefF to efficiently extract fault-sensitive features (accuracy improvement/operation speed).

In Nawandhar et al.,[6] NCFS was utilized to reduce the dimensionality of the feature space in microscopic images by selecting discriminative feature sets from individual as well as combinations of features (accuracy improvement).

In Kim,[7] the author sought to establish a novel machine-learning-assisted hybrid-input model for forecasting automobile demand, addressing research gaps related to input data, the methodology, and the scope of demand forecast. The identification and verification of significant input features were conducted using the RReliefF algorithm (interpretability).

As it can be concluded, FS methods play a vital role in various domains where dimension reduction is needed.

FS METHODS

There are three categories of FS methods: filter, wrapper, and embedded.

Filter methods employ diverse ranking techniques, assigning scores to variables based on specific criteria, and subsequently applying a threshold to exclude less-relevant variables.[8] This approach offers notable advantages: it demands minimal computational resources, has a negligible impact on predictive error rates,[9] and helps to prevent overfitting.[8] However, a key limitation lies in the potential exclusion of important variables that may not individually be informative but contribute to interrelationships and enhancement of general knowledge (state space) of the system.

Wrapper methods necessitate a classifier (or regressor), helping to specify feature subsets. Although these methods often surpass filter methods, their drawbacks include heightened computational demands as the classifier (or regressor) undergoes training for each feature subset. Applying wrapper methods in high-dimensionality problems can be challenging and time consuming.[10]

Embedded methods streamline the quest for an optimal feature subset as a part of the learning process.[11] Distinct from wrapper methods, these techniques seamlessly integrate FS and modeling concurrently. This unique approach translates to lower computational requirements compared to wrapper methods, presenting a more efficient alternative.[12]

RReliefF

RReliefF[13] is a nonparametric method that selects the relevant features in regression based on weight vectors. RReliefF first iterates over each training instance. For each instance, it randomly selects another instance as a reference point and identifies its *k*-nearest neighbors based on a specified distance metric. During the iteration over nearest neighbors, the algorithm updates the weights based on the differences between instances. After accumulating the weights, RReliefF computes the final estimation for each attribute. The algorithmic form of the method can be seen in Algorithm 1 (as presented in Robnik-Sikonja and Kononenko).[13]

Algorithm 1. RReliefF

Input: for each training instance a vector of attribute values x and the predicted value $\tau(\mathrm{x})$
Output: the vector W of estimations of the qualities of attributes

1. set all $N_{dC},\ N_{dA}[A], N_{dC\ \text{and}\ dA}[A], W[A]$ to 0
where N_{dC}: weights for different prediction (class), $N_{dA}[A]$: weights for different attribute, $N_{dC\ \text{and}\ dA}[A]$: weights for different prediction and different attribute and $W[A]$: final estimation of each attribute
2. **for** $\mathrm{i} := 1$ to m **do begin**
3. randomly select instance R_i;
4. select k instances I_j nearest to R_i;
5. **for** $\mathrm{j} := 1$ to k do begin
6. $N_{dC} := N_{dC} + |f(R_i) - f(I_j)| \cdot d(i,j)$;
7. **for** $\mathrm{A} := 1$ to number all_attributes **do begin**
8. $N_{dA}[A] := N_{dA}[A] + \mathrm{diff}(A, R_i, I_j) \cdot d(i,j)$;
9. $N_{dC\text{and}dA}[A] := N_{dC\text{and}dA}[A] + |f(R_i) - f(I_j)| \cdot \mathrm{diff}(A, R_i, I_j) \cdot d(i,j)$;
10. **end;**
11. **end;**
12. **end;**
13. **for** $\mathrm{A} := 1$ to #all_attributes **do**

$$W[A] := \frac{N_{dC\ \text{and}\ dA}[A]}{N_{dC}} - \frac{(N_{dA}[A] - N_{dC\ \text{and}\ dA}[A])}{(m - N_{dC})};$$

MRMR

MRMR, introduced in Ding and Peng,[14] is an algorithm that finds the optimal set of features that is mutually and maximally dissimilar by minimizing the redundancy and maximizing the relevance of a feature set to the response variable. MRMR iteratively selects features based on their relevance and redundancy. It starts by selecting the feature with the highest relevance and adds it to the set of selected features. Then, it iteratively adds features that have high relevance and low redundancy until the redundancy becomes nonzero for all features. Finally, it incorporates features with zero relevance into the selected set. This process aims to create a feature set, with the features being ordered by importance. For more information on the method, refer to Ding and Peng.[14] The algorithmic form of the method can be seen in the following (as presented in MathWorks)[15]:

Algorithm 2. MRMR

1. Select the feature with the largest relevance, $\max_{x \in \Omega} \ V_x$, where $V_x = I(x,y)$ is the mutual information. Add the selected feature to an empty set S.
2. Find the features with nonzero relevance and zero redundancy in the complement of S, S^c.
   a. If S^c does not include a feature with nonzero relevance and zero redundancy, go to step 4.
   b. Otherwise, select the feature with the largest relevance, $\max_{x \in S^c, W_x = 0} V_x$. Add the selected feature to the set S.
3. Repeat step 2 until the redundancy is not zero for all features in S^c.
4. Select the feature that has the largest mutual information quotient value with nonzero relevance and nonzero redundancy in S^c, and add the selected feature to the set S.

$$\max_{x \in S^c} \mathrm{MIQ}_x = \max_{x \in S^c} \frac{V_x}{W_x} = \max_{x \in S^c} \ \frac{I(x,y)}{\frac{1}{|S|}\sum_{z \in S} I(x,z)}.$$

5. Repeat step 4 until the relevance is zero for all features in S^c.
6. Add the features with zero relevance to S in random order.

NCFS

In Yang et al.,[10] the authors introduced the NCFS algorithm, a nearest-neighbor-based FS method. NCFS was inspired by the neighborhood components analysis algorithm, proposed by Goldberger et al.[16] With a regularization term, NCFS maximizes the expected leave-one-out classification accuracy using the gradient ascent technique. NCFS aims to iteratively update a weight vector to minimize an error function. It does so by computing probabilities based on the current weights, updating the weights based on these probabilities, and adjusting the step length dynamically to optimize convergence. The algorithm continues iterating until convergence, at which point it returns the final weight vector. The algorithmic form of the method can be seen below (please refer to Yang et al.[10] for more detailed information about NCFS).

Algorithm 3. NCFS

1. T: training set; α: initial step length; σ: kernel width; λ: regularization parameter; η: small positive constant;
2. Initialization: $\mathbf{w}^{(0)} = (1, 1, ..., 1), \epsilon^{(0)} = -\infty, t = 0$
3. **repeat**
4. **for** $i = 1, \cdots, N$ **do**
5. Compute p_{ij} and p_i (probabilities) using $\mathbf{w}^{(t)}$ according to the following equations

$$D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{d} w_l^2 |x_{il} - x_{jl}|$$

$$p_{ij} = \begin{cases} \dfrac{\kappa\left(D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j)\right)}{\sum_{k \neq i} \kappa\left(D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_k)\right)}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

$$p_i = \sum_j y_{ij} p_{ij}$$

where $\kappa(z) = \exp(-z/\sigma)$ is a kernel function, and the kernel width σ is an input parameter that influences the probability of each point being selected as the reference point.

6. **for** $l = 1, \cdots, d$ **do**
7. $\Delta_l = 2\left(\frac{1}{\sigma}\sum_i (p_i \sum_{j \neq i} p_{ij}|x_{il} - x_{jl}| - \sum_j y_{ij} p_{ij}|x_{il} - x_{jl}|) - \lambda\right) w_l^{(t)}$
8. $t = t + 1$
9. $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \alpha\Delta$
10. $\epsilon^{(t)} = \xi\left(\mathbf{w}^{(t-1)}\right)$
11. **if** $\epsilon^{(t)} > \epsilon^{(t-1)}$ **then**
12. $\alpha = \text{constant}1 * \alpha$ (constant1 selected by the user)
13. **else**
14. $\alpha = \text{constant}2 * \alpha$ (constant2 selected by the user)
15. **until** $|\epsilon^{(t)} - \epsilon^{(t-1)}| < \eta$
16. $\mathbf{w} = \mathbf{w}^{(t)}$
17. return $\mathbf{w}$

LASSO

In Tibshirani,[17] the author introduced LASSO as a method for both shrinking and selecting variables in regression and generalized regression problems. The LASSO algorithm does not prioritize subsets but instead implements a continuous shrinking process that can generate coefficients that are equal to zero. For a given value of λ (positive), the aim of the algorithm is to minimize the following:

$$\left(\frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \beta_0 - x_i^T\beta\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right)$$

where N is the number of observations, y_i is the response at x_i, x_i is a vector of length p, and λ is a regularization parameter. The parameters β_0 and β are a scalar and a vector of length p, respectively.

RFE

RFE is a greedy algorithm that searches for the optimal subset. The primary concept is to repeatedly create a model and then choose the worst variable. The selected variable is then eliminated, and the procedure is repeated until all the remaining variables have been explored. To develop the regressor and assess the importance of predictors, the RFE algorithm requires a machine learning algorithm (e.g., a support vector machine, an RF, logistic regression, or naive Bayes).[18] The RF regressor (RFR) technique was used as the RFE model's regressor in this investigation. The stopping criteria is either a number of required features (specified by the user) or performance metrics (e.g., when the specified criterion value does not decrease anymore). The algorithmic form is presented here.[19]

Algorithm 4. RFE

Inputs: Training set $\mathbf{Tr}$, set of features $\mathbf{F} = \{\boldsymbol{f}_1 \ldots\ldots \boldsymbol{f}_{\boldsymbol{x}}\}$, ranking method $\mathbf{RFR}$
Outputs: Ranked list of features
for i =1:a **do**:
Rank set **F** based on RFR
$\boldsymbol{f}^* \leftarrow$ last ranked feature in **F**
$\mathbf{F} \leftarrow \mathbf{F} - \boldsymbol{f}^*$
end
Return ranked list of features

USE CASE, DATASETS, AND PREPROCESSING

Use Case

The objective function in the currently studied laser assemblies is to utilize the gathered information throughout a multiple-step assembly operation and predict whether a product will be defective or otherwise.

Information was gathered through sensors, actuators, and a human–machine interface for the assembly of two laser modules, d-serie and s-serie, which consist of emitters, mirrors, and lenses that combine the laser beams of the emitters into one unified beam (output beam). For the d-serie laser module, there are 62 assembly steps [fast-axis collimating (FAC) placements, slow-axis collimating (SAC) placements, mirrors placements, turning mirror placement, polarization, and fiber lens placement] in the assembly process.

For the s-serie laser module, there are 31 assembly steps (FAC, SAC, Mirrors and Fiber Lens placement). The assembly processes follow the same pathway with a number of differences. The process is automated using a special assembly robot.

Note that from this section on, the details of the assembly process and the associated components are proprietary and commercially sensitive. The case is explained in a wider context, avoiding the revelation of sensitive information from the manufacture.

Datasets and Preprocessing

Prior to the assembly process, the emitters' power is measured to ensure that they are working. Measuring the geometry of the beams, their power, and their convergence to the centroid is done at each stage of the assembly process. Each assembly stage (FAC, SAC, and so on) introduces new variables as inputs. The width of the fast-axis beam, width of the slow-axis beam, the centroid of the beam, beam pointing, and power of the beam are important inputs that are measured. The response variable that needs to be predicted is the output power of the laser module.

The data accumulated for the d-serie module reached 1411 observations (completed assembled laser modules), and the input features (predictors) for the d-serie module are 1181. The data accumulated for the s-serie module reached 1628 observations, with 530 input features. The datasets were shuffled to reduce bias.

IMPLEMENTATION

The FS methods applied on the provided datasets are RReliefF, MRMR, RFE, RReliefF-RFE, MRMR-RFE, NCFS, and LASSO. The methodology steps are presented in Figure 1.

To create the combined methods, a filter method is applied to the original dataset, and then a wrapper method is applied to the subset that was created by the filter. The filter is applied first, as it does not need high computational power, it is not as time consuming as the wrapper in high-dimensional datasets, and it avoids overfitting. The wrapper is applied second, on the subset that was created by the filter, to reduce the size of the subset even more but at the same time maintain predictive power.

To create the hybrid methods, RFE was applied on the subsets generated by RReliefF and MRMR. However, for the d-serie dataset, RFE was not applied to the subset formed by MRMR due to MRMR identifying only five features as significant, which is considered a limited number.

For NCFS, variables that have weight less than 0.1 are considered as irrelevant and are not included in the subset.

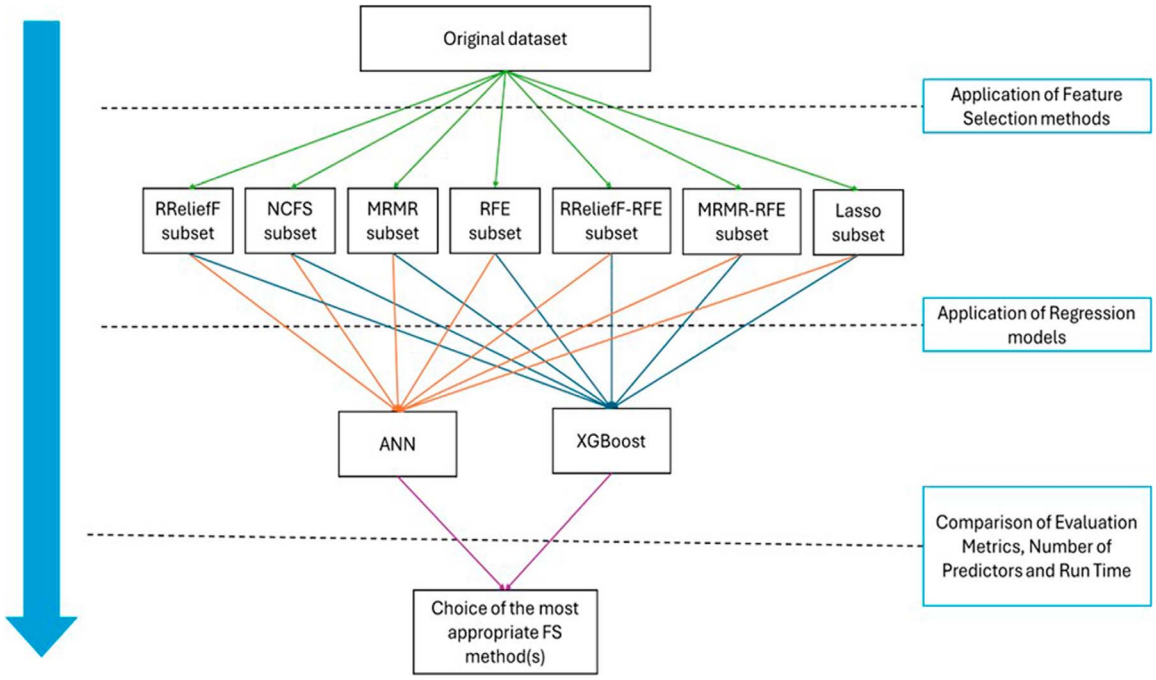

FIGURE 1. The methodology of the current study.

To execute RReliefF, k neighbors were set to 10. The predictors that scored a positive importance weight were included in the RReliefF subset.

For MRMR, the features that had an importance weight higher than 0.1 were considered important.

To deploy the RFE method, RFR was chosen as the regressor. Based on RFR's performance, the importance of the predictors was assessed. Due to the high dimensionality of both datasets, the step of the RFE was set to 10. That way, in each iteration, instead of one predictor being removed, 10 were removed. This measure was taken to minimize the computational time because of the high complexity of the algorithm. Fivefold cross validation was also applied to avoid misleading results.

Regarding LASSO, all the features that had an importance that was not zero were included in the subset.

The size of the created subsets, along with the computational time needed to create them, can be seen in Tables 1 and 2 in the supplementary information, available at https://doi.org/10.1109/MIS.2024.3416678.

After all the subsets were created, ANNs and XGBoost were applied to all of them to predict the output power of the laser module. The metrics that were used to evaluate the performance of the ANNs and XGBoost are root-mean-square error, mean absolute error, and mean absolute percentage error (MAPE).

RESULTS

The evaluation metrics for the training and testing sets can be seen in Figures 2 and 3 and in Tables 1 and 2 in the supplementary information, available at https://doi.org/10.1109/MIS.2024.3416678.

It should be noted that overfitting has been observed in XGBoost for the s-serie module. Tuning the applied models would probably solve this problem, but it should be emphasized that enhancing the predictive power of the models lies outside the scope of this research.

D-Serie Module Dataset

Predictors

With the exception of RReliefF, all the methods reduced the original number of predictors (1181 predictors in the original dataset) by more than 80% (see Table 1 in the supplementary information, available at https://doi.org/10.1109/MIS.2024.3416678). RReliefF achieved a reduction of 17.4%.

Evaluation Metrics

In the following results, small differences between the metrics of different models will not be taken into consideration (e.g., an MAPE difference of 0.1%–0.2% will be considered insignificant).

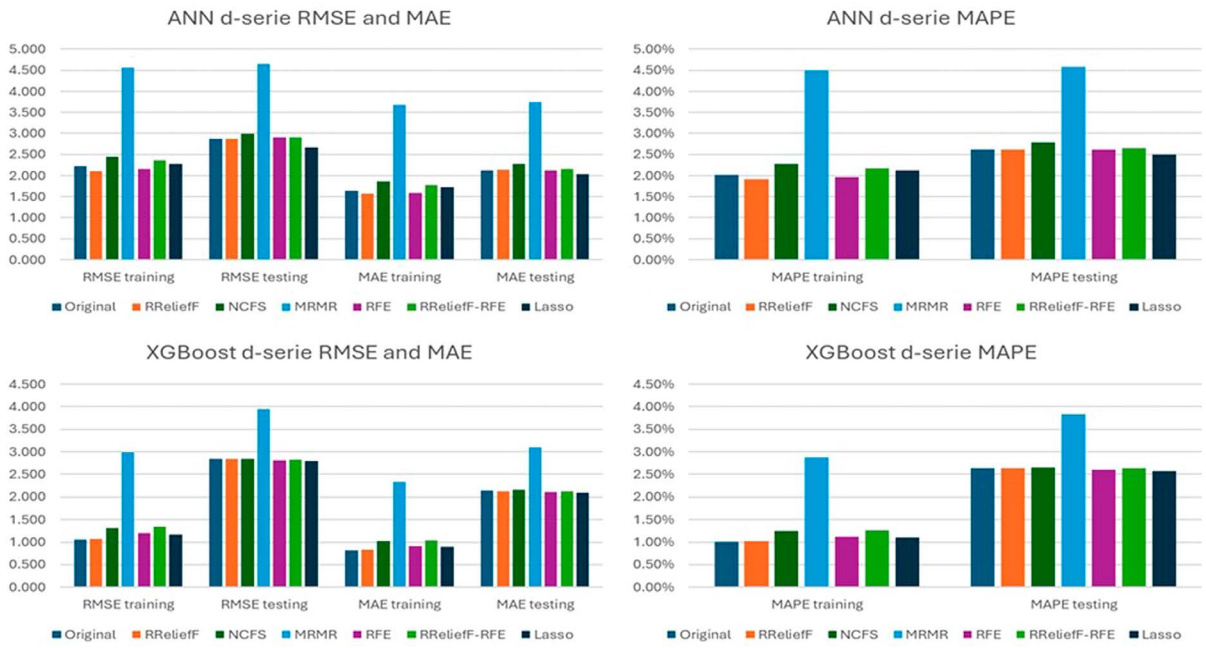

FIGURE 2. The train and test metrics for all methods for the d-serie module. (a) ANN. (b) XGBoost. RMSE: root-mean-square error; MAE: mean absolute error.

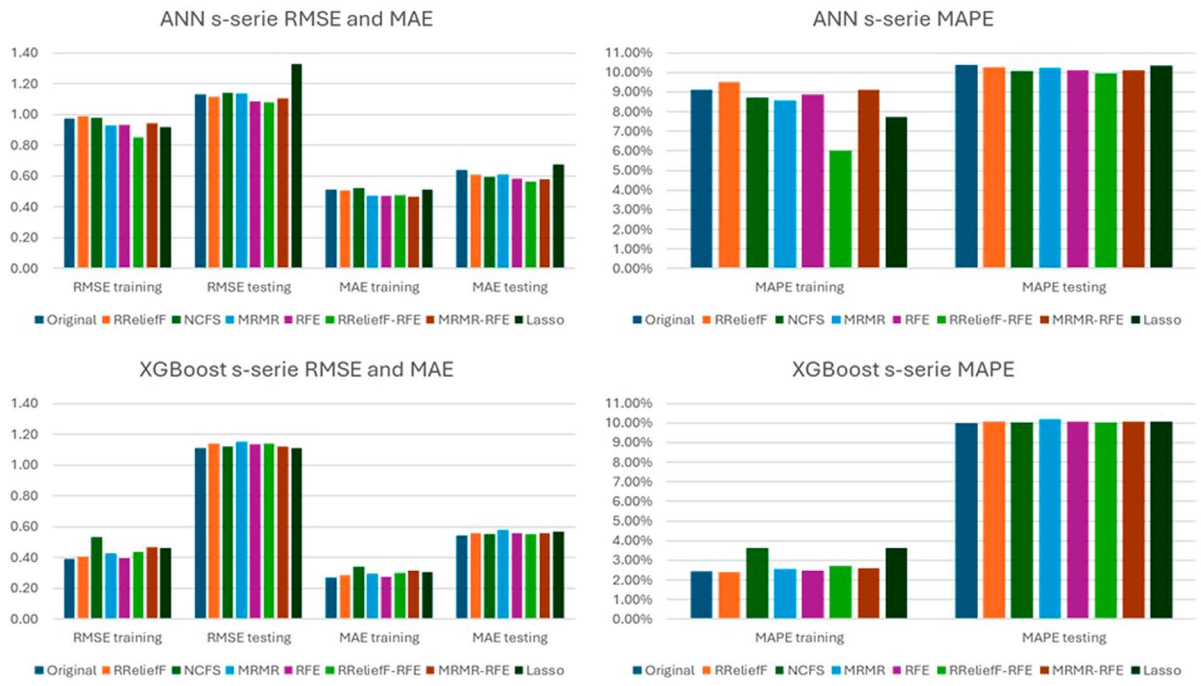

FIGURE 3. The train and test metrics for all methods for the s-serie module. (a) ANN. (b) XGBoost.

ANN

RReliefF and RFE performed similar to the original dataset on both the training and testing sets. NCFS and RReliefF-RFE demonstrated a performance that was slightly inferior to that of the original dataset on both the training and testing sets (but the differences are insignificant). MRMR performed poorly on both sets. LASSO demonstrated a performance that was similar to that of the original dataset on the training set and was slightly better on the testing set.

XGBoost

RReliefF, RFE, and LASSO demonstrated a performance that was similar to or slightly better than that of the original dataset on both the training and testing sets. NCFS and RReliefF-RFE exhibited a slightly inferior performance compared to that of the original dataset on the training set but a similar performance on the testing set. MRMR performed poorly on both sets.

It can be concluded that the FS methods that performed well, regardless of the applied regression model, are RReliefF, RFE, RReliefF-RFE, NCFS, and LASSO. MRMR performed poorly with both prediction methods, and for that reason will no longer be considered a possible candidate for FS for the d-serie dataset.

Computational Time

The model can be useful only if the prediction is achieved before the end of assembly-cycle time. Verification required us to compare the computational time of the FS methods by setting one method's computation time as the benchmark for our calculations. This allows for a more comprehensive representation of the time factor, irrespective of hardware settings. The computation time of RFE will be considered the benchmark computational time for our calculations. The saving percentage in computational time is given by

$$\text{Saving} = 100\,\frac{t_{\text{benchmark}} - t_{\text{solution}}}{t_{\text{benchmark}}}$$

With the exception of RReliefF-RFE, all the methods achieved saving percentages higher than 96%. RReliefF-RFE achieved a saving percentage of 30%. It should be noted here that the importance of computational time is defined by the nature of the problem. In cases like the one studied in this article, FS is not a repetitive process; it takes place at the beginning of the research to exclude redundant information and acts as a guide for researchers/experts on the field to understand which variables and processes have a higher impact on the system and the final outcome. Thus, the computation time in this specific case is not important. In cases that

require real-time updates of the models and repetitive FS, computational time is important and is a factor that the researchers should take into consideration to decide which FS method is the most appropriate to use.

Conclusions for the FS on the D-Serie Module Dataset

RReliefF, RFE, RReliefF-RFE, NCFS, and LASSO are the methods that perform well regardless of the applied regression model.

If computational time is the most important factor, the ranking (from best to worst) is as follows: RReliefF, LASSO, NCFS, RReliefF-RFE, and RFE.

If achieving the highest reduction in the number of predictors while maintaining predictive accuracy is the main goal, the ranking (from best to worst) is as follows: RReliefF-RFE, NCFS, LASSO, RFE, and RReliefF. Researchers may choose the method that produces the smallest subset for various reasons. A smaller subset of predictors is often easier to comprehend and communicate, making it more accessible to a broader audience, including stakeholders and decision makers. This enhanced interpretability not only facilitates a clearer understanding of the model's behavior but can also lead to more confident and informed decision making. Additionally, a concise subset may highlight the most influential predictors, shedding light on the key factors driving the prediction models.

S-Serie Module Dataset

Predictors

NCFS, RRelieF-RFE, MRMR-RFE, and LASSO reduced the original number of predictors (530 predictors in the original dataset) by more than 70% (see Table 2 in the supplementary information, available at https://doi.org/10.1109/MIS.2024.3416678). RReliefF, MRMR, and RFE did not reduce the original predictors by a high percentage compared to the other methods.

Evaluation Metrics

ANN

RReliefF and NCFS performed similar to or slightly worse than the original dataset on the training set and performed slightly better on the testing set. MRMR, RFE, RReliefF-RFE, and MRMR-RFE performed better than the original dataset on both the training and testing sets. LASSO performed slightly better than the original dataset on the training set and slightly worse on the testing set, with the exception of the testing MAPE, which was better (10.34% compare to the 10.40% of the original dataset).

The best performance was demonstrated by RReliefF-RFE and NCFS, which achieved a testing MAPE of 9.96% and 10.08%, respectively, compared to the original dataset, which achieved 10.4%.

XGBoost

RReliefF, MRMR, and RFE performed similar to the original dataset on both the training and test sets. NCFS performed worse on the training set and similarly on the testing set compared to the original dataset. RReliefF-RFE and MRMR-RFE performed slightly worse on the training set and similarly on the testing set compared to the original dataset. LASSO performed worse on the training set and similarly on the testing set.

The best performance was demonstrated by NCFS and RReliefF-RFE, which achieved a testing MAPE of 10.01% and 10.04%, respectively, compared to the original dataset, which achieved 10%.

Computational Time

With the exception of RReliefF-RFE and MRMR-RFE, all the methods achieved saving percentages higher than 97%. RRelief-RFE and MRMR-RFE achieved a saving percentage of 74.26% and 38.23%, respectively.

Conclusions for the FS on the S-Serie Module Dataset

All the methods performed well regardless of the applied regression model.

If computational time is the most important factor, the ranking (from best to worst) is as follows: MRMR, RReliefF, NCFS, LASSO, RReliefF-RFE, MRMR-RFE, and RFE.

If achieving the highest reduction in the number of predictors while maintaining predictive accuracy is the main goal, the ranking (from best to worst) is as follows: NCFS, LASSO, RReliefF-RFE, MRMR-RFE, RReliefF, MRMR, and RFE.

CONCLUSION AND FUTURE WORK

The primary objective of this study was to provide actionable advice on selecting the most suitable FS methods for intricate industrial scenarios, particularly within manufacturing and assembly sectors, where the advent of Industry 4.0 has led to a proliferation of available data.

Overall, all the FS methods, with the exception of MRMR on the d-serie dataset, exhibited either similar or slightly better accuracy compared to the original datasets. This parity in performance metrics underscores the importance of tailoring the choice of FS method to the unique requirements of each individual case.

For instance, if the main concern is the time efficiency in the FS process, methods such as RReliefF,

NCFS, and LASSO emerge as particularly attractive options due to their ability to efficiently identify relevant features without sacrificing predictive accuracy. Conversely, if the primary goal is to enhance interpretability, thereby facilitating a deeper understanding of the manufacturing process and pinpointing critical areas for improvement or optimization, then methods like RReliefF-RFE, NCFS, and LASSO offer greater value. By reducing the number of predictors while maintaining predictive accuracy, these methods not only enhance interpretability but also provide actionable insights that can guide decision making in manufacturing processes.

It's worth noting that although MRMR demonstrated poor generalization, especially on the d-serie dataset, its performance on the s-serie dataset suggests potential utility in specific contexts. However, its inconsistent performance and the limited subset size on the d-serie dataset render it less favorable compared to other methods.

Throughout the study, evaluating the generalization of MRMR-RFE was rendered unfeasible due to the MRMR subset for the d-serie dataset containing only five predictors, rendering the application of RFE, after MRMR was applied to the dataset, inapplicable.

Despite the study's focus on laser module assembly lines, the findings are likely applicable to a broader range of complex, real-time assembly processes within manufacturing. This suggests that the insights obtained from this research have the potential to inform and improve manufacturing processes beyond the specific datasets examined, indicating the broader relevance and applicability of the selected FS methods.

The conclusions from this study also provide a validation for RReliefF-RFE's (which was briefly introduced in our previous work)[20] performance on manufacturing settings, which was proven in a laser assembly process. As future work, testing the proposed RReliefF-RFE on wider applications and datasets (e.g., water treatment, supply chains, and environmental impact) can further evaluate the generalization of the method in wider industrial applications.

DECLARATION OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

ACKNOWLEDGMENTS

The authors would like to express their sincere thanks to Kyriakos Kandris for his insight on the importance of FS in real-time scenarios and the suggestions on how to present the computational time. This work has been carried out in the framework of the innovative strategies, sensing and process chains for increased quality, reconfigurability, and recyclability of manufacturing optoelectronics (IQONIC) project, which received funding from the European Union's Horizon 2020 research and innovation program under Grant 820677.

This article has sdupplemental material available at https://doi.org/10.1109/MIS.2024.3416678.

REFERENCES

1. J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 9–15, Mar. 2017, doi: 10.1109/MIS.2017.38.
2. S. St. John et al., "Predicting chatter using machine learning and acoustic signals from low-cost microphones," *Int. J. Adv. Manuf. Technol.*, vol. 125, nos. 11–12, pp. 5503–5518, Apr. 2023, doi: 10.1007/s00170-023-10918-z.
3. H. Wu, T. Yang, H. Li, and Z. Zhou, "Air quality prediction model based on mRMR–RF feature selection and ISSA–LSTM," *Sci. Rep.*, vol. 13, no. 1, Dec. 2023, Art. no. 12825, doi: 10.1038/s41598-023-39838-4.
4. F. Houndekindo and T. B. M. J. Ouarda, "Comparative study of feature selection methods for wind speed estimation at ungauged locations," *Energy Convers. Manage.*, vol. 291, Sep. 2023, Art. no. 117324, doi: 10.1016/j.enconman.2023.117324.
5. X. Wen and Z. Xu, "Wind turbine fault diagnosis based on ReliefF-PCA and DNN," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115016, doi: 10.1016/j.eswa.2021.115016.
6. A. Nawandhar, N. Kumar, R. Veena, and L. Yamujala, "Stratified squamous epithelial biopsy image classifier using machine learning and neighborhood feature selection," *Biomed. Signal Process. Control*, vol. 55, Jan. 2020, Art. no. 101671, doi: 10.1016/j.bspc.2019.101671.
7. S. Kim, "Innovating knowledge and information for a firm-level automobile demand forecast system: A machine learning perspective," *J. Innov. Knowl.*, vol. 8, no. 2, Apr. 2023, Art. no. 100355, doi: 10.1016/j.jik.2023.100355.
8. G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.
9. L. Y. Chuang, H. W. Chang, C. J. Tu, and C. H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32,

no. 1, pp. 29–38, 2008, doi: 10.1016/j.compbiolchem.2007.09.005.
10. W. Yang, K. Wang, and W. Zuo, "Neighborhood component feature selection for high-dimensional data," *J. Comput.*, vol. 7, no. 1, pp. 162–168, 2012, doi: 10.4304/jcp.7.1.161-168.
11. M. Fernandes, A. Canito, V. Bolón-Canedo, L. Conceição, I. Praça, and G. Marreiros, "Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry," *Int. J. Inf. Manage.*, vol. 46, pp. 252–262, Jun. 2019, doi: 10.1016/j.ijinfomgt.2018.10.006.
12. R. J. Urbanowicz, M. Meeker, W. la Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Informat.*, vol. 85, pp. 189–203, Sep. 2018, doi: 10.1016/j.jbi.2018.07.014.
13. M. Robnik-Sikonja and I. Kononenko, "An adaptation of relief for attribute estimation in regression," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 296–304.
14. C. Ding and H. Peng. "Minimum redundancy feature selection from microarray gene expression data." in *Proc. IEEE Bioinf. Conf. Comput. Syst. Bioinf. (CSB2003)*, Stanford, CA, USA, 2003, pp. 523–528
15. "Rank features for regression using minimum redundancy maximum relevance (MRMR) algorithm." MathWorks. Accessed: Apr. 22, 2024. [Online]. Available: https://www.mathworks.com/help/stats/fsrmrmr.html
16. J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 513–520.
17. R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. R. Stat. Soc., B: Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.
18. C. Wang, Y. Pan, J. Chen, Y. Ouyang, J. Rao, and Q. Jiang, "Indicator element selection and geochemical anomaly mapping using recursive feature elimination and random forest methods in the Jingdezhen region of Jiangxi Province, South China," *Appl. Geochem.*, vol. 122, Nov. 2020, Art. no. 104760, doi: 10.1016/j.apgeochem.2020.104760.
19. J. L. F. Chacón, T. F. de Barrena, A. García, M. S. de Buruaga, X. Badiola, and J. Vicente, "A novel machine learning-based methodology for tool wear prediction using acoustic emission signals," *Sensors*, vol. 21, no. 17, Sep. 2021, Art. no. 5984, doi: 10.3390/s21175984.
20. N. G. Markatos, A. Mousavi, G. Pippione, and R. Paoletti, "Industry 4.0-based framework for real-time prediction of output power of multi-emitter laser modules during the assembly process," *Electronics (Switzerland)*, vol. 12, no. 3, Feb. 2023, Art. no. 766, doi: 10.3390/electronics12030766.

NIKOLAOS GRIGORIOS MARKATOS was a research fellow with Brunel University London, Greater London, UB8 3PH, U.K., and is now a research fellow with the Civil and Environmental Engineering Department, Water Engineering, Imperial College London, London, SW7 2BX, U.K. His research interests include Industry 4.0 automation, real-time predictive models, and minimization of defects in manufacturing processes. Markatos received his Ph.D. degree in artificial intelligence and process optimization from Brunel University London. He is the corresponding author of this article. Contact him at n.markatos@imperial.ac.uk.

ALIREZA MOUSAVI is a professor with the College of Engineering, Design and Physical Sciences, Brunel University London, Greater London, UB8 3PH, U.K. His research interests include digital transformation and smartification of industrial systems, especially within the Industry 4.0 context covering sensor actuation, signal processing and feature extraction, machine learning, modeling, control, and optimization. Mousavi received his Ph.D. degree in system engineering from Brunel University. He is a Senior Member of IEEE. Contact him at alireza.mousavi@brunel.ac.uk.

EVINA KATSOU is a professor of water engineering in the Civil and Environmental Engineering Department, Water Engineering, Imperial College London, London, SW7 2BX, U.K. Her research interests include the development and implementation of frameworks for sustainability, circularity measurement and assessment, process optimization and control, and decision support systems. Katsou received her Ph.D. degree in chemical engineering from the National Technical University of Athens. Contact her at e.katsou@imperial.ac.uk.

GIULIA PIPPIONE is a laser diode packaging engineer with Convergent Photonics Italia, Torino, 10148, Italy. Her research interests include design for optical and mechanical packaging of semiconductor devices, optimization of manufacturing processes in a production line, and automation of micro-optics alignments for laser beams collimation. Pippione received her M.Sc. degree in advanced technologies in physics from the University of Turin. Contact her at giulia.pippione@convergent-photonics.com.

ROBERTO PAOLETTI is the director of Diode Laser Fab with Convergent Photonics Italia, Torino, 10148, Italy. Paoletti received his Ph.D. degree in electronics from Politecnico di Turin. His research interests include design for manufacturing, device engineering, and long-term reliability. Contact him at roberto.paoletti@convergent-photonics.com.