# Feature Selection for Virtual Metrology Modeling: An application to Chemical Mechanical Polishing

Oussama Djedidi*, Rebecca Clain†, Valeria Borodin‡, Agnès Roussy§
*Mines Saint-Etienne, Univ. Clermont Auvergne,*
*CNRS, UMR 6158 LIMOS,*
F-42023 Saint-Etienne, France
Emails: *oussama.djedidi@emse.fr, †rebecca.clain@emse.fr, ‡valeria.borodin@emse.fr, §roussy@emse.fr

2022 33rd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC) | 978-1-6654-9487-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASMC54647.2022.9792527

***Abstract*—This paper focuses on the feature selection problem in a virtual metrology task applied to a chemical mechanical polishing process. One of the main challenges specific to virtual metrology modeling is the relatively wide availability of measurements and traces (features) versus the scarcity of samples (entries), as they are usually costly to obtain. To overcome these challenges, we propose a hybrid feature selection algorithm, called *Enhanced Hybrid Feature Selection* (EHFS), that combines a filter approach and a genetic algorithm embedding a machine learning model. The filter starts by eliminating noisy and uninformative features. Then, in the wrapper stage, the genetic algorithm is augmented by a solution archive to favor exploration. This added feature avoids the reevaluation of duplicate candidate solutions and consequently decreases the computational time of EHFS.**

**Numerical experiments, conducted on industrial and benchmark datasets, show that the proposed solution approach performs competitively in terms of both solution quality and computational time compared with two existing approaches: the general-purpose Forward Feature Selection (FFS) and virtual metrology-specific Evolutionary Repetitive Backward Elimination (ERBE).**

***Index Terms*—Chemical Mechanical Polishing, Feature selection, Machine Learning, Genetic Algorithm, Virtual Metrology**

## I. Context, motivations, and problem statement

In the current era of Industry 4.0 and Big Data, multiple paradigms are actively promoted to make excelling the manufacturing sector. For instance, *Smart manufacturing* is oriented towards the integration of systems at different decision levels (from real-time to strategical levels, passing by operational and tactical levels), by combining physical, cyber, and big data capabilities [1]. *Zero defect manufacturing* aims to decrease and mitigate failures in manufacturing processes through the product and/or process prisms [2]. The major enablers towards a smart factory are: data, technology and analytics [3]. In this sense, the integration of a predictive capability in the current process monitoring practices is identified as a key lever to enhance industrial competitiveness [1, 3, 4].

In this context, the concept of Virtual Metrology (VM), which aims to predict an attribute of the product quality directly from production process data without a physical measurement, falls within the global scope of the aforementioned goals. VM plays the role of a complementary (soft) tool to traditional costly and time-consuming metrology (physical) tools [5].

VM systems have historically evolved from linear regression models to more complex machine learning-based, such as support vector regressions and artificial neural networks [3, 5]. In the high-precision and sensor-rich environments specific to semiconductor facilities, the over-abundance of emergent features makes the training of metrology models computationally intractable and industrially infeasible [5]. To circumvent the curse of dimensionality, the Feature Selection (FS) is applied as a preprocessing step.

Acting as a dimensionality reduction technique, FS aims to reduce the set of extracted features to only the relevant ones for modeling [6]. The main FS approaches are filters, wrappers and embedded approaches [4]. Interested readers are referred to [4] for a detailed literature review on the major tasks to develop a VM system, including the dimensionality reduction in data-intensive environments.

The current paper extends the hybrid filter-wrapper FS algorithm proposed by Korabi et al. [6]. At the algorithmic level, the FS approach improvements are twofold.

- *Filter stage:* By applying a correlation filter, that deletes low correlated features with the target variable, while maintaining the rank of the input dataset, and thus reducing the risk of deleting low correlated informative features.
- *Optimization stage:* The performance of the genetic algorithm provided by Korabi et al. [6] is enhanced, by implementing a solution archive to store the already evaluated candidate solutions to avoid their time-consuming reevaluation [7, 8].

At the validation level, additional numerical experiments are conducted on industrial and benchmark datasets. The proposed approach is compared against two existing approaches, namely: Forward Feature Selection (FFS), and VM-specific Evolutionary Repetitive Backward Elimination (ERBE) proposed by Lenz [5].

The remainder of this paper is organized as follows. The extended version of the approach provided by Korabi et al. [6], called *Enhanced Hybrid Feature Selection* (EHFS), is presented in Section II. Section III provides and analyzes the numerical results obtained on industrial and benchmark datasets. Section IV concludes this paper and provides several perspectives.

## II. Feature selection approach

The proposed Enhanced Hybrid Feature Selection (EHFS) approach is designed to conciliate the following three specifications:

- *From low to high variate data:* To handle both low-dimensional and high-dimensional datasets.
- *Effectiveness:* To reduce the set of features to the right size and composition, while fostering the prediction capabilities of learning models.
- *Efficiency:* To be computationally affordable.

To meet the aforementioned specifications, the EHFS approach is composed of two stages as indicated by Korabi et al. [6]: a *filter stage*, and an *optimization stage*, wrapping a given machine learning model.

### A. Filter stage

Let $X^{n \times m}$ be the initial dataset with $n$ samples of $m$ vectors (features). Let $y = (y_1, y_2, \ldots, y_n)$ be the target variable. The filter stage relies on three parameters: the minimum correlation threshold ($corrMin$), the maximum correlation threshold ($corrMax$), and a step to increase the correlation threshold ($corrStep$).

**Algorithm 1** : Filter stage of the EHFS algorithm $(X, y)$

**Input:**
- $X^{n \times m}$: The matrix of features
- $y^{n \times 1}$: The target vector
- $corrMin$: The minimum correlation threshold
- $corrMax$: The maximum correlation threshold
- $corrStep$: The step to increase the correlation threshold

**Output:** $F^{n \times k}$ The filtered set of $k$ features ($k \leq m$)

$r_X \leftarrow rank(X)$ ▷ Compute the rank of the input matrix
$C \leftarrow correlation(X, y)$ ▷ Compute the correlation vector between the input and target variables
$F = []$ ▷ Create the set of filtered features
$D = []$ ▷ Create the set of discarded features
**while** $corrMin < corrMax$ **do**
  $F \leftarrow X[C \geq corrMin]$
  $D \leftarrow X \setminus F$
  $r_F \leftarrow rank(F)$ ▷ Compute the rank of the filtered matrix
  $i \leftarrow lengh(D)$
  **while** $r_F < r_X$ and $i \geq 0$ **do**
    $F.append(D[i])$
    $D.drop(D[i])$
    $r_F \leftarrow rank(F)$
    $i \leftarrow i - 1$
    $corrMin \leftarrow corrMin + corrStep$
    **if** $r_X < r_F$ **then** $r_X \leftarrow r_X$
    **end if**
  **end while**
**end while**
**return** $F$

Algorithm 1 describes the filter stage, that identifies and deletes the noisy and redundant features. This algorithm aims to only keep the independent features, that are well-correlated with the target variable.

The algorithm starts by computing the correlation between $X$ and $y$, and the rank of matrix $F$ ($r_F$). Then, it isolates the features of which correlation value is below the minimum threshold ($corrMin$) in *discarded* set $D$, and keeps the correlated features in *filtered* set $F$.

If $r_F$ is greater or equal than the rank of input matrix $X$ ($r_X$), Algorithm 1 continues to iterate, by increasing $corrMin$ by $corrStep$, until $corrMax$ is reached. Otherwise, while $r_F$ is lower than $r_X$, Algorithm 1 reintegrates the discarded features from $D$ one by one until $r_F$ is restored to $r_X$. Thus, the proposed filter stage allows for a fast elimination of uninformative features, while also keeping independent features as a potential source of variability.

### B. Optimization stage

The second stage of the EHFS algorithm focuses on minimizing the remaining feature set $F$ to the most valuable ones for a given Machine Learning (ML) model. To do so, an ML model is wrapped in a Genetic Algorithm (GA) instance.

Following the genetic algorithm scheme introduced by Reeves and Rowe [9], the adopted GA instance extends the optimization core provided by Korabi et al. [6].

*Solution representation and search space:* Each individual is a bit string with a length equal to the number of features. The bits in the bit string indicate whether the corresponding feature is activated or not, allowing for the construction of a candidate solution set.

*Evaluation of individuals:* The set of candidate solutions is then used to train and test the ML model wrapped in the GA instance. Once the training is finished, the model is tested, generating a so-called fitness score. This score constitutes the evaluation function, that the GA uses to evaluate and rank the individuals. In the current paper, the Root Mean Squared Error (RMSE) is used.

*Selection and crossover:* In the following generations (iterations), the process is repeated after diversifying the population through specific genetic operators (mating, mutating, etc.) applied to selected individuals using selection strategies (tournaments, in our case). To further enhance the selection process, we also implement a solution archive, as proposed by Biesinger et al. [8]. The implemented complete solution archive allows the algorithm to explore only new candidates and out of local optima. More precisely, if a solution is found in the archive, its cost is not evaluated again. However, the individual will be mutated until a new solution is obtained. This mutation ensures that the algorithm continuously explores new solutions, and thus avoids early convergence into local minima. All the solutions are stored together with their fitness scores allowing for a trie-based exploration [7].

To implement the GA instance, we extend the DEAP framework [10], by integrating the concept of solution archive. By virtue of its hybrid nature, the proposed approach allows us to select the best features using GA consistently with the requirements of the wrapped ML model. This advantage is notably valuable in our case, where the sample count is low versus the number of features. From an interpretability perspective, it also makes possible the investigation of the impact of one particular feature, when the wrapped model contains an embedded FS algorithm, such as Random Forest and Lasso.

## III. Computational experiments

### A. Description of the datasets

Numerical experiments are conducted on two industrial datasets. The first industrial dataset is provided by ST-Microelectronics Rousset (STM-R), an 8-inch semiconductor manufacturing facility. STM-R dataset is also used in the works by Yang et al. [11] and Korabi et al. [6]. The second dataset is a benchmark dataset from the PHM Data Challenge 2016 [12].

Both datasets contain measurements from Chemical-Mechanical Polishing (CMP) processes, which are polishing processes that remove undesired surface materials via chemical reactions [11]. The STM-R dataset contains 130 time series obtained via two measurement strategies. Features extracted from these time series aim to predict the thickness of the layer after the polishing process of 545 wafers. The PHM challenge contains the average removal rate from 2,406 wafers and 25 associated raw measurements. Table I provides the main characteristics of the studied datasets.

TABLE I: Description of the studied datasets.

| **STM-R** [11] | **PHM 2016-Data Challenge** [12] |
|---|---|
| Predicts layer thickness | Predicts average removal rate |
| Two measurement strategies | Two stages to predict per wafer |
| 130 original features | 25 original features |
| No outliers | 3 outliers present |
| 545 metrology values | 2,406 metrology values |

These two datasets are quite representative in terms of the issues related to VM modeling. They both contain a relatively low number of samples (545 and 2,406) to be handled by machine learning models. Furthermore, each metrology value is associated with multiple time series: 130 for the STM-R dataset and 25 for the PHM dataset. In the case of the STM-R dataset, the time series also vary in size and number of samples, further accentuating the issue of feature extraction and selection.

### B. Feature extraction and cleaning

To test the proposed approach, two strategies are used to extract features from original datasets STM-R et PHM: Simplified Extraction (SE) and *Tsfresh* library. The first consists of extracting common statistical attributes (10 features per time series), whereas the second is an extensive extraction using the *Tsfresh* library (more than 70 features per time series).

These two extraction strategies generate two sets of features of very different sizes, thus allowing us to test and compare the performance of the approaches under evaluation, in high and low dimensionality settings. Table II shows the feature count after the extraction phase for both datasets. As expected, the simplified extraction results in datasets, having a number of features lower than the number of samples, whereas the *Tsfresh*-based extraction generates datasets having a number of features much larger than the number of samples.

Table II also shows the number of rejected features. These are features that are unusable with the applied machine learning model: features with constant values, features containing *Not a Number* (NaN) entries, and unscalable features (containing infinities, for instance). As the datasets already contain enough useful features for both scenarios, we have opted for their deletion rather than amputation.

TABLE II: Feature count before and after extraction of the studied datasets.

| **Dataset** | **Count** | **Original set** | **Simplified extraction** | **Tsfresh extraction** |
|---|---|---|---|---|
| **STM-R** | Features count | 130 | 404 | 27,266 |
| | Rejected features | 53 | 45 | 15,284 |
| | Remaining features | 77 | **359** | **11,982** |
| | Time to extract (s) | 304 | 904 | 2,706 |
| **PHM** | Features count | 25 | 223 | 15,580 |
| | Rejected features | 3 | 24 | 2,237 |
| | Remaining features | 22 | **199** | **13,434** |
| | Time to extract (s) | 37 | 568 | 8,560 |

### C. Experimental framework

For the experimentation phase, the GA instance in the second stage of the EHFS algorithm uses Random forest as an ML-wrapped model. Additionally, during the experimental runs, both the simplified and *Tsfresh*-based extracted datasets are split into training (60%), validation (20%), and test (20%) subsets. The training and validation sets are used for the selection process, whereas the test set is used as a holdout set to evaluate the selected features.

The performance of the proposed EHFS algorithm is evaluated and compared with two existing FS algorithms:

- *Forward Feature Selection (FFS)*: An algorithm that adds iteration by iteration a new feature and evaluates its effects using a wrapped ML model.
- *Evolutionary Repetitive Backward Elimination (ERBE)* [5]: A VM-specific multistage algorithm, that relies on a combination of Leave-One-Out selection, integration of artificial features, and genetic algorithm to iteratively remove uninformative features.

Table III provides a comparative summary of the main characteristics of the studied approaches.

All of the three algorithms ran on identical hardware and used identical software libraries, namely: Python 3.8.5, Scikit-Learn 1.0.2, and the GA framework DEAP [10].

### D. Results and comparative analysis

The detailed results achieved by the proposed EHFS algorithm, in both selection and test, are displayed in Table IV.

TABLE III: Main characteristics of the studied approaches.

| | EHFS | FFS | ERBE |
|---|---|---|---|
| **Number of stages** | 2 | 1 | 3 |
| **Approach** | Filtering and optimization | Iterative selection | Iterative elimination |
| **Wrapped ML model** | Random forests | Random forests | SVR |
| **Purpose** | Regression | General | Regression |

For the STM-Rousset dataset, EFHS reduces the number of features to 14 and 34, in the cases of simplified and *Tsfresh*-based extraction, respectively. Given the initial number of features in both sets, this represents a 96% and 99.71% reduction, respectively. The EHFS algorithm achieves these performances, while also improving the accuracy of the Random Forest model. Experiments conducted on the PHM dataset provide a similar conclusion with 85.23% and 98% reduction in the number of features for simplified and *Tsfresh*-based extraction strategies, respectively.

Training and testing the Random Forest model using the feature selected by the EHFS algorithm, yields results competitive with those found in the literature [13], even without tuning the ML model or separating the dataset by stage or group. Figure 1 displays these results.

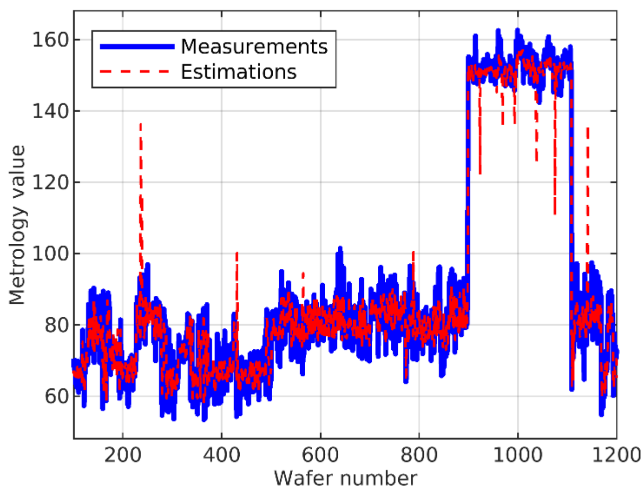



Fig. 1: Test predictions against targets from PHM 2016-Data challenge.

Table V displays the test results achieved by the three algorithms on both datasets and for both simplified and *Tsfresh*-based extraction strategies. For reference, we include the results obtained by a Random Forest model with no FS (NS) applied.

The test results show that the proposed EHFS algorithm outperforms both the ERBE and FFS algorithms, notably on the PHM dataset. It is worthwhile mentioning that the ERBE algorithm exceeds both FFS and NS algorithm in the simplified extraction, despite the performance gap with EHFS. This gap might be explained by the lack of flexibility of the ERBE algorithm, which can only wrap a Support Vector Regression (SVR) model. Under the *Tfresh*-based extraction, the use of the ERBE algorithm is intractable, as it does not provide a selection after more than $1.3 \times 10^6$s.

For the STM-R dataset, the EHFS algorithm still clearly provides better results. In terms of computational time, the iterative approaches adopted by FFS and ERBE are computationally greedy and time-consuming, compared to EHFS. The architecture of the latter, combining filtering and optimization, is proven suitable and effective, being the fastest and the best scoring.

The main purpose of FS is dimensionality reduction. Nevertheless, it can also give insights into which measurements are most impactful on the target metrology value. Figure 2 shows the number of common features between EHFS (simplified and *tsfresh*-based extractions) and ERBE algorithms. These features are aggregated from the solutions with the best fitness scores from each algorithm. The ERBE algorithm selects, in most part, features different from the EHFS algorithm, and only shares about a third of the features in both datasets. This difference might be the key to explaining some of the discrepancies in its scores. Further investigation reveals that most of the features present in the solutions provided by the EHFS algorithm and absent from those given by the ERBE algorithm have been lost because of the iterative deletion approach it uses.

Studying the aggregated features sets further, we found that the features common to the best solutions of the three instances come mainly from three variables: the rotation speed of the platen, the rotation speed of the head, and the inner pressure of the tube.

## IV. Conclusions

This paper addresses the problem of FS in VM, where dimensionality reduction is a crucial preprocessing step [4]. The proposed solution approach enhances the capabilities of the filter-wrapper method provided by Korabi et al. [6]. The performance of the improved approach is tested and validated on industrial and benchmark datasets from CMP processes. The obtained results clearly highlight the efficiency and the effectiveness of the *Enhanced Hybrid Feature Selection* algorithm with results outperforming competing algorithms.

Following these promising results, we intend to go further in the validation of the proposed approach, by evaluating its capabilities to maintain the same high performances for heterogeneous datasets from other semiconductor manufacturing areas or other related manufacturing sectors. In this paper, the Random Forest model is used to predict the metrology variable of interest. The use of other machine learning models as wrapped models in the framework of the proposed approach will be investigated for a twofold goal: on the one hand, to push the ML-models towards their predictive limits and, on the other hand, to challenge the proposed approach for its further improvement.

## Acknowledgments

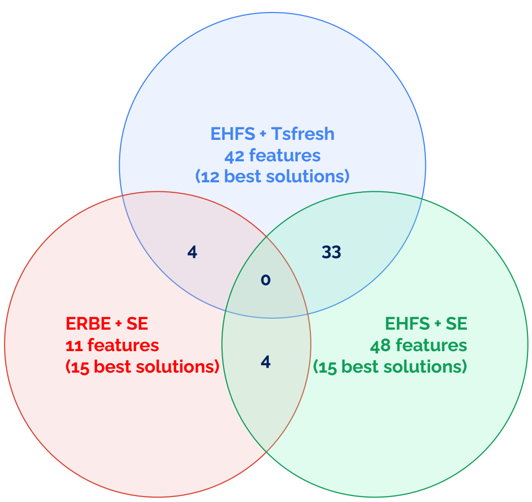
This paper is conducted in the framework of the project MADEin4, which has received funding from the ECSEL JU

TABLE IV: Results provided by the EHFS algorithm.

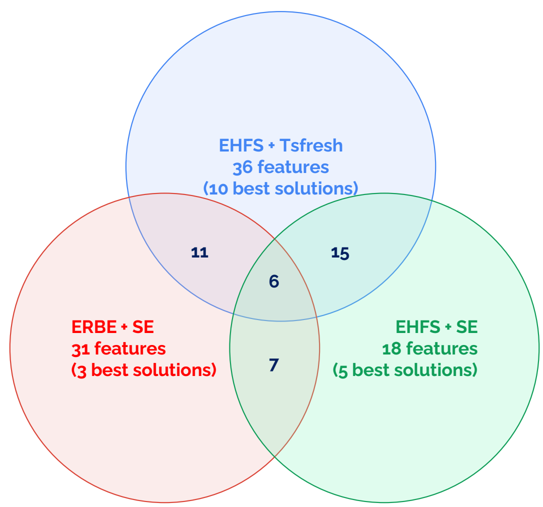
| | Approach | Simplified extraction | | | | *Tsfresh*-based extraction | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | # of features | # of deleted features | RMSE | Time (s) | # of features | # of deleted features | RMSE | Time (s) |
| **STM-R** | **Start** | 359 | 45 | - | 905 | 11,982 | 15,284 | - | 2,707 |
| | **Stage 1 (Filter)** | 29 | 230 | 38.9342 | 2 | 53 | 11,929 | 38.5426 | 3 |
| | **Stage 2 (GA)** | 14 | 15 | 38.3764 | 513 | 34 | 19 | 37.2992 | 2,028 |
| | **Test results** | **14** | **-** | **38.3764** | **1,420** | **34** | **-** | **37.2992** | **4,738** |
| **PHM** | **Start** | 298 | 100 | - | 2 | 13,434 | 15,284 | - | 244 |
| | **Stage 1 (Filter)** | 260 | 38 | 4.9371 | 2 | 81 | 13,353 | 5.392 | 152 |
| | **Stage 2 (GA)** | 89 | 171 | 3.1133 | 1,450 | 42 | 39 | 4.303 | 3,101 |
| | **Test results** | **44** | **-** | **2.72** | **1454** | **42** | **-** | **3.5960** | **3,497** |

TABLE V: Comparative results between EHFS, FFS, and ERBE algorithms. N/A: Not applicable. NS: No Selection.

| | | Simplified extraction | | | *Tsfresh*-based extraction | | |
|---|---|---|---|---|---|---|---|
| | | **# of features** | **RMSE** | **Time (s)** | **# features** | **RMSE** | **Time (s)** |
| **PHM** | NS | 199 | 55.5410 | N/A | 13,434 | 56.6936 | N/A |
| | FFS | 50 | 52.8652 | 88,237 | 42 | 52.8652 | $6 \times 10^5$ |
| | ERBE | 11 | 18.4472 | 2,894 | - | - | Intractable |
| | **EHFS** | **44** | **2.7200** | **1,454** | **42** | **3.5960** | **3,497** |
| **STM-R** | FFS | 28 | 43.6876 | 63,4345 | 38 | 42.3745 | $4 \times 10^5$ |
| | ERBE | 31 | 42.0333 | 9,132 | 443 | 42.9061 | $> 10^6$ |
| | **EHFS** | **14** | **38.3764** | **1,420** | **34** | **37.2992** | **4,738** |




(a) PHM dataset




(b) STM-R dataset.

Fig. 2: Common feature selected by EHFS and ERBE algorithms aggregated from the set of best solutions each generated.

(Electronic Components and Systems for European Leadership Joint Undertaking) under grant agreement No 826589. The JU receives support from the European Union's Horizon 2020 research and innovation program and France, Germany, Austria, Italy, Sweden, Netherlands, Belgium, Hungary, Romania, and Israel.

## References

[1] J. Moyne and J. Iskandar, "Big data analytics for smart manufacturing: Case studies in semiconductor manufacturing," *Processes*, vol. 5, no. 3, p. 39, 2017.

[2] F. Psarommatis, G. May, P.-A. Dreyfus, and D. Kiritsis, "Zero defect manufacturing: state-of-the-art review,

shortcomings and future directions in research," *International journal of production research*, vol. 58, no. 1, pp. 1–17, 2020.

[3] M. S. Reis and G. Gins, "Industrial process monitoring in the big data/industry 4.0 era: From detection, to diagnosis, to prognosis," *Processes*, vol. 5, no. 3, p. 35, 2017.

[4] P.-A. Dreyfus, F. Psarommatis, G. May, and D. Kiritsis, "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*, vol. 0, no. 0, pp. 1–24, 2021.

[5] B. Lenz, "Smart Feature Selection to enable Advanced Virtual Metrology," Dissertation, Universität Tübingen, Sep. 2015.

[6] T. E. Korabi, V. Borodin, M. Juge, and A. Roussy, "A hybrid feature selection approach for virtual metrology: Application to CMP process," in *2021 32nd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2021, pp. 1–5.

[7] G. R. Raidl and B. Hu, "Enhancing genetic algorithms by a trie-based complete solution archive," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2010, pp. 239–251.

[8] B. Biesinger, B. Hu, and G. R. Raidl, "A genetic algorithm in combination with a solution archive for solving the generalized vehicle routing problem with stochastic demands," *Transportation Science*, vol. 52, no. 3, pp. 673–690, 2018.

[9] C. Reeves and J. E. Rowe, *Genetic algorithms: principles and perspectives: a guide to GA theory*. Springer Science & Business Media, 2002, vol. 20.

[10] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[11] W.-T. Yang, J. Blue, A. Roussy, M. S. Reis, and J. Pinaton, "Virtual metrology modeling based on gaussian bayesian network," in *2018 Winter Simulation Conference (WSC)*. IEEE, 2018, pp. 3574–3582.

[12] P. D. C. P. Society, "PHM data challenge," https://www.phmsociety.org/events/conference/phm/16/data-challenge, 2016.

[13] X. Jia, Y. Di, J. Feng, Q. Yang, H. Dai, and J. Lee, "Adaptive virtual metrology for semiconductor chemical mechanical planarization process using GMDH-type polynomial neural networks," *Journal of Process Control*, vol. 62, pp. 44–54, feb 2018.