A MACHINE LEARNING FRAMEWORK FOR DATA-DRIVEN DEFECT DETECTION IN MULTISTAGE MANUFACTURING SYSTEMS

A. Naudé[1]* & J.H. van Vuuren[1]

ARTICLE INFO

Article details

Submitted by authors	16 Feb 2024
Accepted for publication	7 Jun 2024
Available online	30 Aug 2024

Contact details

* Corresponding author
ansunenaude@gmail.com

Author affiliations

1 Department of Industrial Engineering, Stellenbosch University, Stellenbosch, South Africa

ORCID® identifiers

A. Naudé
https://orcid.org/0000-0002-6645-4916

J.H. van Vuuren
https://orcid.org/0000-0003-4757-5832

DOI
http://dx.doi.org//10.7166/35-2-3008

ABSTRACT

Economic transformation and escalating market competitiveness have prompted manufacturers to adopt zero-defect manufacturing principles to lower production costs and maximise product quality. The key enabler of zero-defect manufacturing is the adoption of data-driven techniques that harness the wealth of information offered by digitalised manufacturing systems in order to predict errors. Multi-stage manufacturing systems, however, introduce additional complexity owing to the cascade effects associated with stage interactions. A generic modular framework is proposed for facilitating the tasks associated with preparing data emanating from multi-stage manufacturing systems, building predictive models, and interpreting these models' results. In particular, cascade quality prediction methods are employed to harness the benefit of invoking a stage-wise modelling approach. The working of the framework is demonstrated in a practical case study involving data from a multistage semiconductor production process.

OPSOMMING

Ekonomiese transformasie en toenemende markmededingendheid het vervaardigers aangespoor om zero-defek-vervaardigingsbeginsels toe te pas met die oog om produksiekoste te verlaag en produkkwaliteit te maksimeer. Die sleutelbemiddelaar van zero-defek-vervaardiging is die gebruik van data-gedrewe tegnieke om die magdom inligting wat deur gedigitaliseerde vervaardigingstelsels beskikbaar gemaak word, vir die voorspelling van foute te benut. Multi-stadium vervaardigingstelsels veroorsaak egter bykomende kompleksiteit as gevolg van kaskade-effekte wat verband hou met stadiuminteraksies. 'n Generiese modulêre raamwerk word voorgestel vir die fasilitering van take wat verband hou met die voorbereiding van data wat uit multi-stadium vervaardigingstelsels voortspruit, die bou van voorspellende modelle en die interpretasie van hierdie modelresultate. In die besonder word metodes vir die voorspelling van kaskadekwaliteit toegepas om die voordeel van 'n stadium-gewyse modelleringsbenadering uit te buit. Die werking van die raamwerk word in 'n praktiese gevallestudie gedemonstreer wat betrekking het op 'n multi-stadium halfgeleier produksie proses.

1. INTRODUCTION

Economic transformation, a heightened emphasis on sustainability, and escalating market competitiveness attributed to globalisation are placing increased pressure on manufacturers to reduce costs, maximise efficiency, and deliver products of superior quality [1]. This phenomenon has prompted manufacturers to direct their attention to improving product quality, with a view to minimising the expenses associated with scrapping, reworking, or repairing defective products, as well as using their manufacturing equipment and materials effectively. One quality management ideology that has gained popularity in recent years is *zero-defect manufacturing*, which pertains to the continual pursuit of attaining the highest product quality achievable by eliminating defects and defect inducers from manufacturing systems [2]. The strategy encourages a 'first-time-right' approach, thereby promoting the avoidance of time and resource wastage.

The key enablers of zero-defect manufacturing are the emergence of cutting-edge technologies and *artificial intelligence* (AI) systems, which have paved the way for so-called *smart manufacturing* [3]. The digitalisation of manufacturing systems owing to the proliferation of sensor technologies and information systems being employed has led to a plethora of available data, offering a wealth of information. By harnessing the intrinsic potential of the data related to these manufacturing systems, data-driven approaches such as machine learning may be employed to retrieve actionable insight from the data [4]. The exhaustive eradication of product defects in manufacturing may thus be pursued by employing machine learning models that are aimed at predicting the likelihood of defects with a view to preventing their occurrence altogether.

As a result of the increasing complexity and multi-faceted nature of manufactured products, modern-day manufacturing systems are typically multi-staged, and characterised by sequential interconnected production stages [5]. Multi-stage manufacturing significantly increases the complexity of defect prediction owing to stage interactions and cascade effects (i.e., the propagation of defects induced during one stage to subsequent stages) [6]. An ability to analyse data emanating from multi-stage manufacturing systems, however, raises the possibility of detecting defects during the early stages of production.

A generic modular framework is proposed in this paper for facilitating the tasks associated with pursuing zero-defect manufacturing in multi-stage manufacturing systems by preparing the data from such systems, building error prediction models, and interpreting the results returned by these models. The working of a proof-of-concept computerised instantiation of the framework is illustrated in the form of a case study involving a multi-stage semiconductor manufacturing process.

The remainder of this paper is structured as follows. In Section 2, the literature related to the study is briefly reviewed. In Section 3, a *multi-stage manufacturing defect analysis* (MSM-DA) framework is proposed. This is followed in Section 4 by a proof-of-concept demonstration of the framework in the form of a real-world case study application. The paper closes in Section 5 with a summary of its contributions and some recommendations for future follow-up work.

2. LITERATURE REVIEW

This section contains a discussion of the literature pertinent to the topic of this paper, namely predictive quality, cascade quality prediction modelling, and framework development. A brief discussion of similar studies conducted in the field of multi-stage predictive modelling is also presented.

2.1. Machine learning in predictive quality

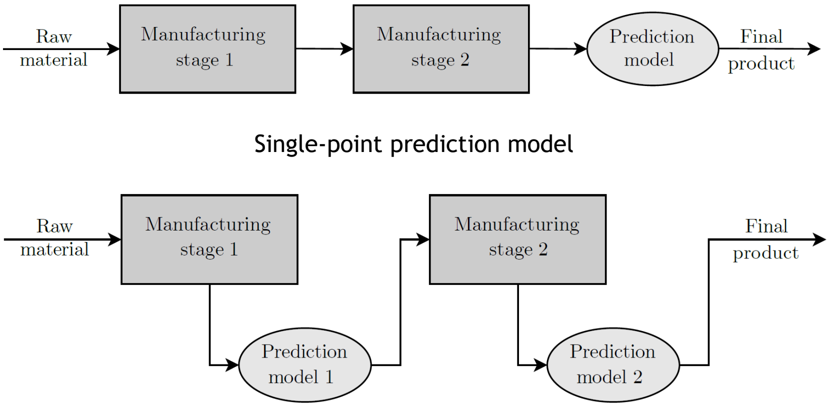
The prediction of the quality of manufactured products that is aimed at enabling proactive quality management is a concept referred to as *predictive quality* [7]. Predictive quality is achieved by extracting trends inherent in the product-related data emanating from manufacturing systems and relating these insights to quality measurements [8]. According to Tercan *et al.* [8], the application of predictive quality consistently incorporates the following three strategies: (1) the selection, collection, consolidation, and preparation of product-related manufacturing data; (2) the development of predictive machine learning models that take the aforementioned data as their input; and (3) the use of model outputs to facilitate decision-making in order to enhance product quality. The final point may be accomplished by invoking model interpretation techniques, including methods such as Shapley Additive exPlanations (SHAP) value analyses and feature importance analyses. A SHAP value analysis is a game theoretic approach to determining the contributions of features to existing coalitions or groups of features in a model's outputs

[9]. A feature importance analysis, on the other hand, involves determining the contribution of each feature to a model's predictions by evaluating the effect of shuffling the values of the feature [10].

Predictive quality in the manufacturing domain today often involves the use of machine learning models because they can uncover intricate patterns and insight from large amounts of data [11]. Machine learning models offer the potential to predict the likelihood of defects occurring in manufacturing systems, thereby encouraging a proactive approach to minimising defects, with the ultimate goal of attaining zero-defect manufacturing. In particular, supervised learning algorithms are primarily invoked for this purpose, with product quality serving as the label to be predicted. Some of the successful applications of predictive quality facilitated by machine learning are predicting the prevalence of cracks in a deep drawing manufacturing process [12], the prediction of porosity defects occurring during additive manufacturing [13], and the prediction of roughness in laser cutting [14], to name but a few.

2.2. Cascade quality prediction modelling

Defect prediction models for multi-stage manufacturing systems have traditionally taken the form of single-point prediction models, as illustrated in Figure 1(a) [15]. Predicting the possibility of defects post-production offers value by enabling the flagging of potentially defective products before they are released from production. This approach, however, offers limited assistance, since the product has already been manufactured at the point of prediction and requires reworking or scrapping if found to be defective. Multiple defect predictive models should instead be embedded within the manufacturing system to harness the potential of in-production quality monitoring, facilitated by multi-stage manufacturing systems, by predicting the product quality throughout the production process, as illustrated in Figure 1(b). The outcome of each stage's prediction model is a binary classification value that indicates whether a product is likely to fail or pass that stage. This approach involves training each prediction model on the data emanating from the output of only a single production stage on the assumption that each stage is subjected to unique operating conditions and behaviours [16].



Single-point prediction model

Multi-point prediction models

Figure 1: Single-point versus multi-point prediction models, adapted from [15]

Although multi-point prediction models offer the advantage of each focusing all its attention on one stage of production, the approach fails to model the interactions between stages. As a result, *cascade quality prediction models* (CQPMs) have emerged to incorporate the cumulative effects that preceding (or upstream) production stages may have on the quality of any specific downstream stage of production, as illustrated graphically in Figure 2. By effectively modelling the stage interactions and incorporating the data from successive stages, the performances of CQPMs typically surpass those of multi-point prediction methods [15].

According to Kao *et al.* [17], the data from a multi-stage manufacturing system may be described in respect of three primary relationships, as shown in Figure 3, each of which warrants consideration when predicting product quality. The relationship R_1 is indicative of an association which exists between the manufacturing variables within each manufacturing stage. The association between the subsequent stages of production, denoted by R_2, is indicative of the interaction between stages. Finally, the relationship denoted by R_3 portrays the relationship between the manufacturing variables and the final product quality.

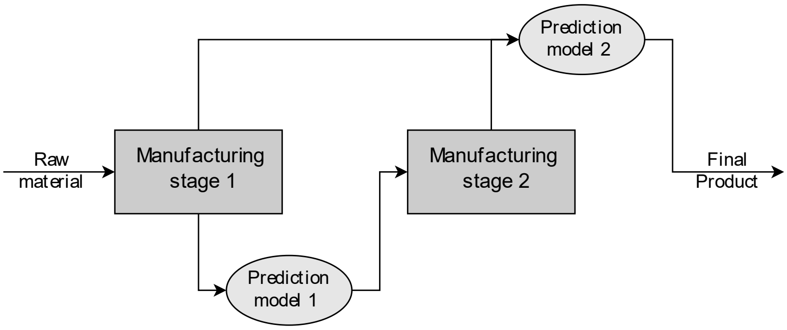


Figure 2: A cascade quality prediction model

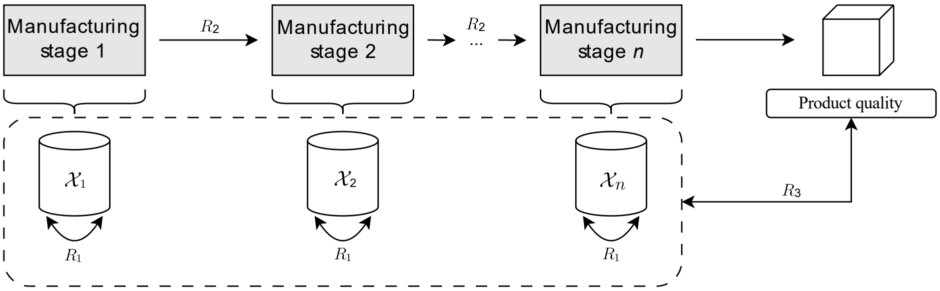


Figure 3: The relationships occurring in multi-stage manufacturing data

The most rudimentary CQPM (referred to here as 'the basic CQPM') incorporates only the relationship R_3 between the stage-wise manufacturing variables and the total quality. This relationship may be learnt by employing machine learning models, since the objective of a machine learning model is to learn about the relationship between the input features (manufacturing variables in this case) and the target feature (the product quality). The prediction model for any stage j is trained on an accumulation of all the manufacturing variables of the data sets $\boldsymbol{\mathcal{X}_1}, \boldsymbol{\mathcal{X}_2}, \dots, \boldsymbol{\mathcal{X}_j}$ coming from the preceding manufacturing stages, thereby considering the effects that preceding stages of production may have on succeeding stages.

Arif *et al.* [15] proposed the adoption of a CQPM that uses so-called latent variables (i.e., observations describing the partial quality) to account for the effect of the relationships between the manufacturing stages R_2 and the relationships between the variables of a particular stage R_1 on the final product quality. According to this approach, the partial quality q_j at a specific stage j is considered to be a function of the manufacturing variables emanating from that particular stage as well as the quality characteristic q_{j-1} of the previous stage, expressed mathematically as

$$\boldsymbol{q_j} = \boldsymbol{f}(\boldsymbol{q_{j-1}}, \boldsymbol{\mathcal{X}_j})$$

The total quality Q may be derived from the quality characteristics of the final stage. That is, $Q = f(\boldsymbol{q_n}, \boldsymbol{\mathcal{X}_n})$, as $\boldsymbol{q_n}$ encapsulates an accumulation of all the previous stages' manufacturing variables and partial qualities.

According to Arif *et al.* [6], the process of uncovering the relationships R_1 between inter-correlated manufacturing variables may be achieved in the same manner as mapping those variables to a new set of dimensions. This may be achieved by performing a *principal component analysis* (PCA) on the manufacturing variables so as to attain a set of uncorrelated variables that represent the quality characteristics associated with those manufacturing variables. For each stage j, $k \in \{1, 2, \cdots, r_j\}$ PCs may be derived. The k^{th} partial quality characteristic of stage j may then be determined as

$$q_{j,k} = \sum_{i=1}^{m_j} a_{k,i} x_{j,i} + \sum_{i=1}^{r_{j-1}} a_{k,(m_j+1)} q_{j-1,i}$$

where $x_{j,i}$ represents the i^{th} value of the manufacturing variables related to stage j, and m_j represents the number of manufacturing variables related to stage j. Moreover, $a_{k,i}$ denotes the degree of contribution of $x_{j,i}$ to the partial quality characteristic $q_{j,k}$ (i.e., the weights returned from the PCA analysis). This approach is referred to here as the CQPM-PCA method.

2.3. Decision support frameworks

The growing demand in the business domain for software solutions has raised the need to reduce development time in the software life-cycle [18]. The focus has shifted from developing models on a case-by-case basis to using reusable generic frameworks with a view to minimising development time and improving the overall efficiency of software solutions. These frameworks encapsulate the general workflows and tools required for solving similar problems in a specific domain. The MSM-DA framework proposed in this paper, for instance, is applicable to any defect detection classification problems found in the domain of multi-stage manufacturing systems. The framework should be modularised into a collection of constituent building blocks to achieve the flexibility and reusability of a framework in a particular domain.

Shim *et al.* [19] proposed that a typical decision support framework or tool has three main elements, namely a database for the storage of data sets required and generated by the analysis, a *graphical user interface* (GUI) for interaction with the user, and a central processing element comprising the modular executable components. Kazmaier [20] further suggested that a central processing element applicable to the data-mining or machine-learning frameworks should have three primary components, namely a processing component (to prepare the data), a modelling component (to build, apply, evaluate, and compare models), and an analysis component (to elicit insight and valuable information).

2.4. Related work

Various approaches to and frameworks for predictive quality in manufacturing systems have been proposed in the literature. Arif *et al.* [6] initially proposed the use of the CQPM-PCA stage modelling approach, and concluded empirically that CQPMs in conjunction with decision trees outperformed single-point models. Building on this, Kao *et al.* [15] proposed a methodology for the offline and online monitoring of manufacturing defects. In their approach, Kao *et al.* [15] utilised a combination of the CQPM-PCA method and association rule mining for the prediction of product quality and the analysis of root causes respectively. Ismail *et al.* [5] proposed a smart real-time quality monitoring and inspection framework, also incorporating the CQPM-PCA method. They, however, focused on building machine learning models and identifying the most influential manufacturing stages. In this paper, the research is focused on exploring the best preprocessing and modelling approaches with a view to developing machine learning models that achieve superior prediction outcomes.

3. A NOVEL MULTI-STAGE MANUFACTURING DEFECT ANALYSIS FRAMEWORK

The framework proposed in this paper is aimed at providing guidance to a user for the tasks associated with pre-processing the data obtained from manufacturing facilities, constructing machine learning models for defect prediction, and interpreting the outputs returned by these models. The overarching objective of the framework is to facilitate a flexible exploratory analysis of the best techniques for preparing, modelling, and analysing the available data with a view to extracting actionable insights. A high-level process model of the proposed MSM-DA framework is illustrated graphically in Figure 4. The remainer of the section is devoted to a discussion of the three main framework components, namely its preprocessing, classification, and model interpretation components. For the sake of brevity, the database and GUI elements are not discussed.

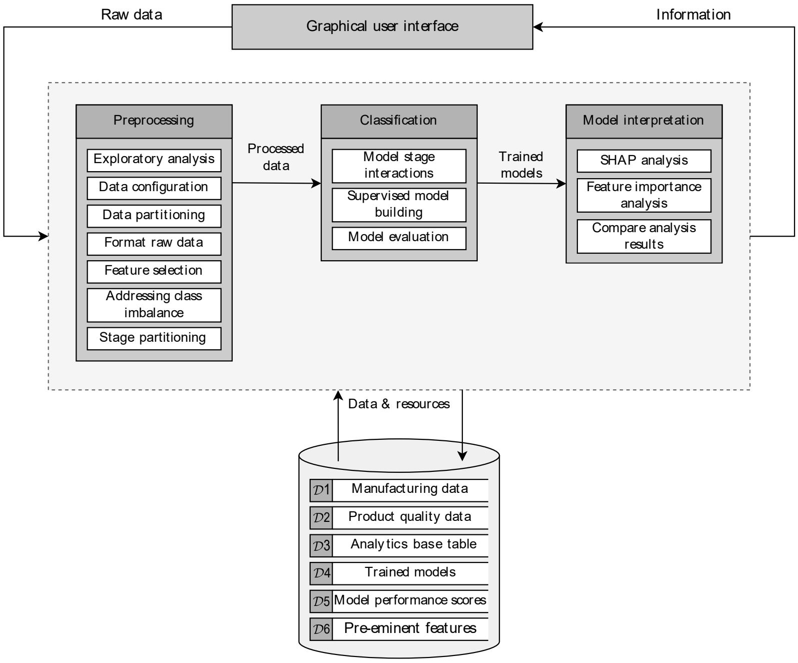


Figure 4: A high-level process model of the proposed MSM-DA framework

3.1. Preprocessing component

The preprocessing component is aimed at preparing raw input data with a view to ensuring that the data are suitable for the various analysis tasks to be performed in the subsequent components. The component takes raw manufacturing data (i.e., observations related to the product under consideration at the different manufacturing stages) as input, and the product quality data supplied by the user. The transformation process of these raw data into prepared data is done by invoking seven constituent sub-processes, referred to as *modules*. These embody standard procedures in the machine learning process, differing only in the final module, which partitions the data for stage-wise modelling purposes.

The first module is responsible for the exploratory analysis conducted on the raw data. This analysis facilitates understanding of the data by examining the available data sets with a view to reporting on the data characteristics (typically by considering appropriate statistical measures). Moreover, the information gathered throughout the exploratory analysis is stored in the database for later use during the preprocessing processes. The resulting model is concerned with configuring the data into a suitable features-label pair table for each product instance. This ensures compatibility with subsequent machine learning algorithms.

The data are then partitioned into training and testing data subsets. The training data serve as input to machine learning models with a view to approximating an underlying function between the data features and their corresponding labels. The testing data, on the other hand, are not included in the training process, and serve the purpose of an unseen data set that is kept aside for the evaluation of the overall performance and generalisability of the machine learning models. The next module is concerned with formatting the entries of the raw data. The primary function of the data formatting module is to process the raw data so as to obtain a data set that is complete (i.e., free of missing values), correct (i.e., error-free), relevant (i.e., has prediction power), and scaled (i.e., with features having similar ranges or means).

Having undergone all the necessary formatting, the training data are passed to the feature selection module. One of the difficulties associated with manufacturing data is the presence of noise (i.e., irrelevant or redundant data) in the data set. Furthermore, if the feature set is large, the analysis may suffer from the so-called *curse of dimensionality*. Feature selection techniques take as input only the training data from which to select the relevant features, with a view to avoiding information leakage and bias during the later performance assessment of machine learning models. The testing set is also reduced to have the same features as the training set in order to evaluate the models' performance during subsequent model-building stages.

In manufacturing, it is typically expected that the number of non-defective products will outnumber the number of defective products. Thus a significant imbalance may exist, leading to a class-imbalanced classification problem. Class imbalance should be addressed (typically by resampling), since class imbalance may jeopardise the results of a machine learning model.

The final preprocessing module involves partitioning the data into stages. The framework is specifically designed for multi-stage manufacturing systems in which products are manufactured over a series of sequential stages. The number of stages depends on the nature of the manufacturing process, and should be either apparent from the data or known by the user as a result of domain knowledge.

3.2. Classification

The overarching purpose of the classification component is to construct a machine learning model that is capable of predicting product defects at various stages of production. The three constituent modules are responsible for modelling the stage interactions, building machine learning models, and finally evaluating the performances of these models.

Modelling the stage interactions is aimed at attempting to capture the complex relationships between manufacturing variables, the sequential manufacturing stages, and the final product quality. The cascade effect in manufacturing systems epitomises the interactive nature of manufacturing stages, as defects induced during earlier production stages are propagated downstream to subsequent stages. This may be achieved by invoking the previously discussed basic CQPM, the CQPM-PCA method, or any alternative stage modelling approach.

Thereafter, suitable models for the prediction of defective products are built for each stage in the manufacturing process in pursuit of a multi-model solution. This module takes the training data as input, and involves an iterative process during which the model's hyperparameters are tuned with a view to finding model configurations that maximise the performances of the models. Finally, the evaluation module involves evaluating the trained models on the testing data to ascertain their degree of generalisability.

In a manufacturing context, where new data continually emerge, it is important to reimplement the preprocessing steps outlined in Section 3.1 based on the new data. The models should then undergo retraining to ensure that they maintain optimal performance in detecting minority instances. By regularly retraining the models with relevant data, the models may better learn from the underlying patterns in the data.

3.3. Model interpretation

While the machine learning prediction models deployed at the various manufacturing stages may offer the capacity to detect and prevent the downstream propagation of defects early, an additional capacity to interpret these models holds the potential to contribute an additional dimension of value. The overarching motivation for model interpretation in this context is that explainable prediction results may aid manufacturers in discovering the root causes of defects. A SHAP analysis and a feature-importance analysis may serve as a means of model interpretation in this component. Their collective purpose is to shed light on the features that bear defect prediction power (i.e., have a high discrimination ability). The findings of the three model interpretation modules are consolidated in the final module, which is responsible for amalgamating the outcomes to form a unified list of pre-eminent features.

4. DEMONSTRATIVE CASE STUDY IMPLEMENTATION

A computerised proof-of-concept implementation of the framework is applied to a benchmark data set in this section to showcase the practical working of the proposed MSM-DA framework. This section is devoted to a walk-through of the application of each component in the framework, along with a presentation of the results obtained. In an attempt to demonstrate the efficacy and utility of the framework, the results are compared with those of Salem *et al.* [21] and McCann *et al.* [22].

The 'no free lunch' theorems suggest that the particular combination of techniques resulting in the best optimisation performance differ for each situation and are not known beforehand [23]. The framework modules were therefore implemented iteratively by invoking different variations of the preprocessing techniques to discover which techniques resulted in the best performances.

4.1. Case study background

The freely available SECOM data set [24] was selected as a benchmark data set for demonstrating the working of the MSM-DA framework proposed in the previous section, as it has been studied both as a single-stage and as a multi-stage manufacturing system by various authors [5, 17, 21, 22, 25, 26]. The SECOM data set is a collection of data entries emanating from a semiconductor wafer manufacturing process. A silicon wafer is a thin slice of crystal semiconductor produced in a circular form, and is an integral microelectronic component that is often used in integrated circuits [27]. The intricate process of producing semiconductor wafers is characterised by a series of highly specialised sequential processes, requiring a high level of precision. May *et al.* [28] described the modern semiconductor manufacturing process as "the most sophisticated and unforgiving volume production technology that has ever been practised successfully".

The wafer production processes may be described at different levels of abstraction. On the lowest level of abstraction, the process may be described as a series of more than 500 individual processes that have to be executed precisely so as to avoid defects. On a higher level of abstraction, the process may be defined as five sequential operations, namely oxidation, photolithography, etching, ion imputation, and metallisation, as illustrated graphically in Figure 5 [29].

Modern-day wafer manufacturing processes are characterised by constant surveillance as a result of the sensor technologies that are available to monitor the precise execution of the various production stages [22]. Sensor technologies are the key enabler for achieving a high level of precision during the production process, and so have the potential to improve production yield, minimise defects, and enhance the quality of the wafers produced.

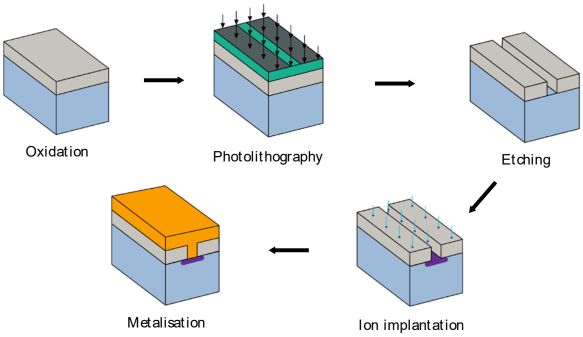


Figure 5: An overview of the semiconductor manufacturing process, adapted from [29]

The SECOM data set comprises two raw text files. The first file contains a collection of 590 distinct sensor readings measured for each of 1 576 product instances. The second file contains the time-stamped results of a quality inspection conducted post-production. The product quality is represented as either a −1 or a 1, respectively indicating a pass (i.e., no defect present) or a fail (i.e., defective).

4.2. Preprocessing

The preprocessing steps of the MSM-DA framework were applied to the SECOM data set with a view to preparing the data for the subsequent analysis tasks. During an initial exploratory analysis it was found that, of all possible 1 567 × 590 = 924 530 data entries, 41 951 entries were missing, corresponding to 4.52 per cent of the entire data set. Moreover, 116 features (corresponding to sensor readings) had a cardinality[1] of 1, suggesting that these features had no discriminatory power in a classification problem. There was also a large target class imbalance, with a ratio of failed tests to passed tests of 1:14, corresponding to 104 products having failed and 1 463 products having passed the quality test.

[1] In the context of a relational database, cardinality is the number of distinct values a feature can potentially assume. A cardinality of 1 signifies that all the feature values are identical, suggesting that a machine learning model cannot derive meaningful insights from the feature for predicting the target feature.

After having explored the data set, the data set was configured into a collection of feature-label pairs by amalgamating the two data files. In order to set apart data for the purpose of the training machine learning models and testing their performances on unseen data, the data set was partitioned, using stratified sampling, into training data and testing data, with a partitioning ratio of training-to-testing data set to 70%:30%. The partitioning process (and hence all the subsequent steps) was repeated for twenty different random splits to ascertain the degree of variability in the results returned by the machine learning models.

The data were formatted to ensure that all data entries were complete, relevant, error-free, and re-scaled. Since machine learning algorithms typically cannot be trained on incomplete data sets, the features containing missing entries were formatted. For this purpose, a threshold for deleting features from the training data was set at 40 per cent, as illustrated graphically in the scatter plot in Figure 6. This led to the deletion of 32 features (represented by the red dots in Figure 6). The remainder of the missing values were imputed by subjecting the data set to multi-variate k-NN imputation [30]. Outlier errors (i.e., meaningless aberrations introduced into the data set) were identified using the 3σ-rule and removed from the data set so as to improve the quality of the data [31]. These values were also imputed using k-NN imputation.

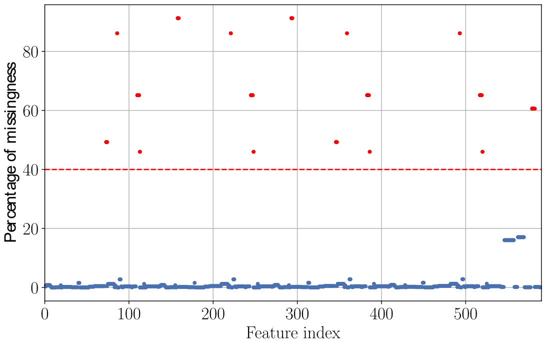


Figure 6: Deletion of features exceeding a deletion threshold of 40%

Features having a cardinality of 1 were removed from the data set so as to remove irrelevant features from the data set. This led to the deletion of 116 features, represented by the red dots in Figure 7. All the feature values were then re-scaled by performing z-score standardisation so as to obtain feature vectors of a similar range scale [32].

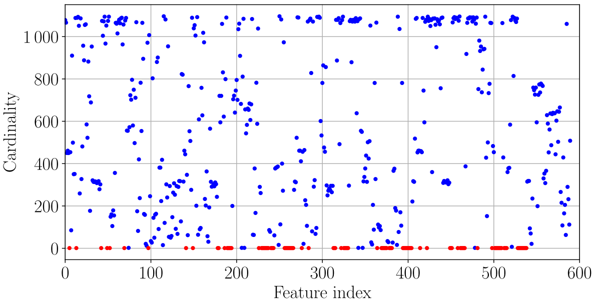


Figure 7: Deletion of the features having a cardinality of 1

The formatted data were passed on to the feature selection module with a view to filtering out noise and irrelevant and redundant features. It is typically not known beforehand which feature selection techniques would be most appropriate for eliminating redundant and irrelevant features from a particular data set. Thus seven feature selection techniques were employed and compared, namely an *analysis of variance* (ANOVA), *mutual information* (MI), the *select false positive rate* (SFPR), *recursive feature elimination* (RFE), the Pearson correlation, Kendall's Tau correlation, and *importance-based feature selection* (IBFS). Detailed information on the implementation of each technique may be found in Table A.1 in Appendix A.

The class imbalance problem was addressed by applying and comparing three different resampling techniques. These resampling techniques, namely random undersampling, random oversampling, and

SMOTE, were employed with a view to balancing the proportions of the minority and majority classes in order to attain an imbalance ratio of 1.

The manufacturing process may be viewed as five sequential stages on the highest level of abstraction, as illustrated in Figure 5. The SECOM data set gives no indication of which features are part of which stage. It was only known that the features represent sensor readings associated with sequentially executed tasks. The assumption was therefore made that the process comprises five stages, and that each of these stages corresponds to an equal number of sensor readings. Other authors who have analysed the data set as a multistage manufacturing system have made the same assumption [5, 17, 29, 31]. The feature vectors were thus grouped into the stages shown in Table 1.

Table 1: The features prevalent in each manufacturing stage

	Stage 1	*Stage 2*	*Stage 3*	*Stage 4*	*Stage 5*
Features	1 – 118	119 – 236	237 – 354	355 – 472	473 – 590

4.3. Classification component

The primary purpose of the classification component is to build and test machine learning defect prediction models for each stage in the manufacturing system. First, the stage interactions are modelled to capture the complex relationships in the stage-wise data. The two-stage modelling approaches considered during the analysis were the basic CQPM and the CQPM-PCA techniques. For each different combination of preprocessing techniques (i.e., the different feature selection and resampling techniques that were employed), models for the final stage were developed, evaluated, and compared so as to be in a position to select the best combination of preprocessing techniques for the data set. Having identified the best-performing combination of preprocessing techniques for the data set, these techniques were employed during the development of multi-point models for all the preceding manufacturing stages.

The supervised machine learning models considered during the analysis were a *decision tree*[2] (DT), a *random forest* (RF), a *support vector machine* (SVM), *logistic regression* (LR), and the *k-nearest neighbours* (kNN) algorithm. These models were selected for their interpretability (i.e., LR and DT) and their computational efficiency (which may facilitate the ease of implementation and deployment). In an attempt to maximise the performance of each machine learning model, a subset of the hyperparameters for each model was tuned in respect of the training data. The tuning process took the form of either a grid search or a random search. Details of the hyperparameter tuning processes may be found in Table A.2 in Appendix A.

Having determined the best combination of hyperparameter values for each machine learning model, the models were trained on the full set of training data, and then applied to the hold-out test set with a view to ascertaining the degree of generalisation of the models in respect of unseen data.

In the context of semiconductor manufacturing, the primary aim of defect prediction is to maximise the number of *true positives* (TPs) (where the positive class is synonymous with the defective target class) and to minimise the number of *false negatives* (FNs). Thus recall was selected as the main performance metric for the analysis. In pursuit of a more comprehensive performance evaluation, additional performance metrics that are robust with respect to class imbalance, such as the *area under the receiver operating characteristic curve* (AUROC) and the *true negative rate* (TNR), were also employed.

Each performance score represents the average performance that the model achieved for each of the twenty different random states selected during the train-test data partitioning. The results returned by the SVM machine learning model, which took as input the training data subject to different combinations of preprocessing techniques, may be found in Figure 8 for the basic CQPM stage modelling approach.

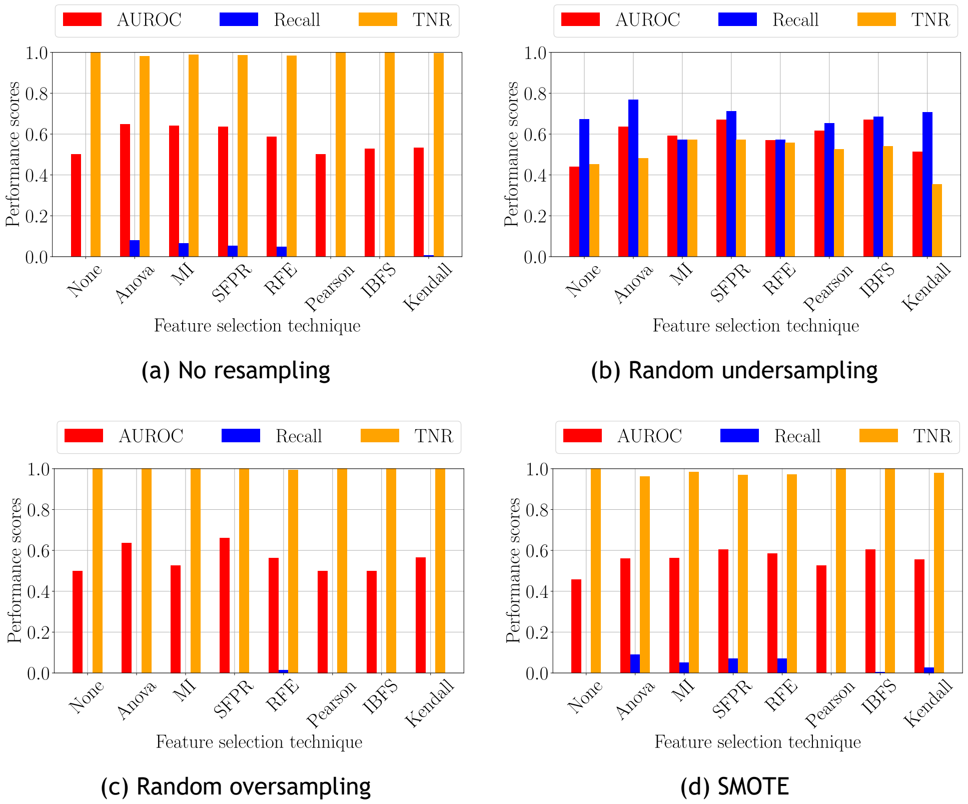
The results returned by all five models show a similar trend with respect to the application of the different resampling techniques. In all cases, performing no resampling or performing random oversampling led to

[2] While typically considered inferior to random forests in their predictive performance, decision trees were selected because they have notable advantages in interpretability.

excellent TNRs but to poor recall results. Performing no resampling resulted in the models being biased towards the majority class (i.e., non-defective products). In the case of random oversampling, the introduction of multiple duplicates of the minority class instances led to overfitting in respect of the training data. Applying random undersampling, on the other hand, enabled the algorithms to learn effectively from the minority class — which is evident as a result of the increase in the recall performance. This, however, came at the cost of misclassifying non-defective products as defective — as is evident as a result of the decline in the TNR. The same observation generally holds for applying SMOTE to the training data (i.e., introducing synthetically generated samples into the data set), although the SMOTE technique resulted in smaller recall values and slightly larger TNRs.

When comparing the results returned by the algorithms after having applied different feature selection techniques, it is evident that the best feature selection technique in each case is affected by the resampling technique and by the machine learning algorithm selected. In general, the combination of random undersampling the data with ANOVA or the Pearson correlation feature selection performed well.

Based on the exploratory analysis in search of the best combinations of preprocessing techniques, the best-performing models (referred to as Models A1-A5 for the basic CQPM) were identified with respect to recall performance for each machine learning algorithm. The results of these best-performing models are presented in Table 2. The best-performing models were trained in respect of the entire data set, and were thus representative of the final manufacturing stage. The best combination of the identified preprocessing techniques was then applied to all five manufacturing stages, and models were built and evaluated for each stage. The results returned by the SVM model produced for each stage are illustrated graphically in Figure 9. The large variance in the results witnessed for the Stage-1 and Stage-5 models suggests that these models are sensitive to the noise still present in the data and are overfitting in respect of specific subsets of data. Adjusting the model’s complexity or applying regularisation techniques may yield more robust models.



(a) No resampling

(b) Random undersampling

(c) Random oversampling

(d) SMOTE

Figure 8: The basic CQPM results returned by the DT classifier with input data subjected to three resampling techniques (a)-(d), the inclusion of outliers, and the application of seven feature-selection techniques

Table 2: The best-performing models for each of the algorithms, achieved by applying the basic CQPM method

	Classifier	Feature selection	Sampling	Recall	AUROC	TNR
Model A1	DT	Pearson	Under	66.13	54.52	41.27
Model A2	RF	None	Under	65.48	66.98	60.51
Model A3	LR	Pearson	Under	72.58	67.13	54.20
Model A4	kNN	MI	Under	64.35	64.48	56.66
Model A5	SVM	Anova	Under	76.77	63.49	48.10

One would generally expect the prediction performances of the stage-wise models to increase per stage as more features are included in the data set and as one progresses through the production stages. This is because the basic CQPM takes an accumulation of all the data prior to a particular stage as input when training a particular stage's prediction model. The average stage-wise performance (indicated by the black triangles in the box plots of Figure 9) exhibits this general trend, with the final stage, Stage 5, achieving the largest average recall of 76.77 per cent.

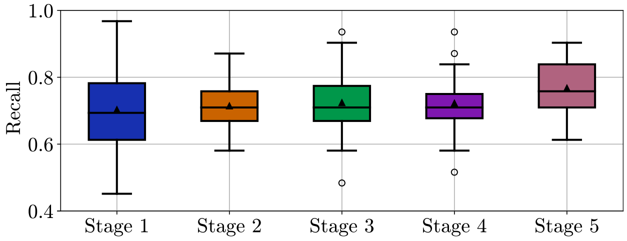


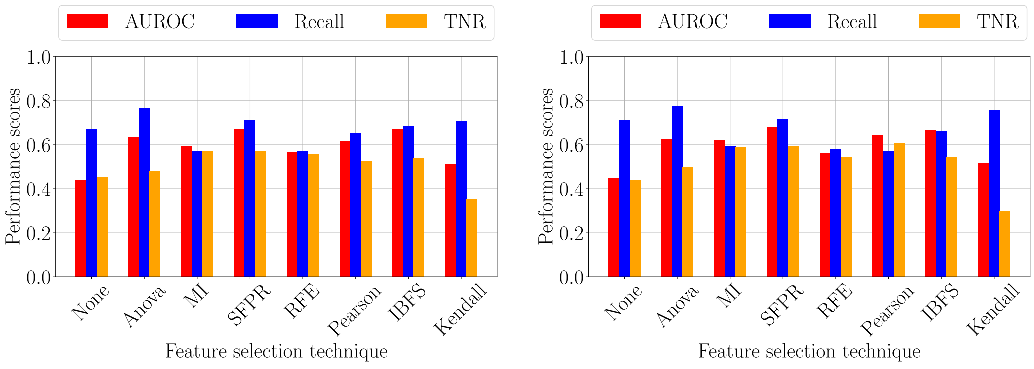
Figure 9: The average recall achieved by SVMs for each of the manufacturing stages

'By applying the alternative stage modelling technique, namely CQPM-PCA, a new set of best-performing models (referred to as Models B1–B5) was obtained. For these models, only the random undersampling technique was considered, since this technique outperformed the other resampling techniques. During the implementation of the CQPM-PCA method, the principal components were generated such that 95 per cent of the variance in the data was retained. The results returned by these best-performing CQPM-PCA models are presented in Table 3.

Table 3: The best-performing models for each of the algorithms, achieved by applying the CQPM-PCA method

	Classifier	**Feature selection**	**Recall**	**AUROC**	**TNR**
Model B1	DT	SFPR	64.87	62.21	56.84
Model B2	RF	Anova	66.90	71.29	67.66
Model B3	LR	Pearson	71.92	67.01	56.41
Model B4	kNN	None	68.04	67.47	61.43
Model B5	SVM	Anova	77.52	62.33	49.66

The performance results for the SVM algorithm according to the two distinct stage modelling techniques are compared in Figure 10. Overall, the application of the CQPM-PCA method to the final stage models outperformed the basic CQPM for all the models and for the majority of feature selection techniques. This may be attributed to the fact that the CQPM-PCA takes into account all three relationships R_1, R_2 and R_3 that are prevalent in a manufacturing data set, and therefore captures the stage interactions more effectively. This also alludes to the fact that the appropriate modelling of stage-wise interactions results in superior performances when compared with the results returned by single-point prediction models (the final-stage model of the basic CQPM is the same result as that returned by a single-point prediction model).



(a) SVM basic CQPM

(b) SVM CQPM-PCA

Figure 10: A comparison of the performances achieved by the SVM algorithm during application of the basic CQPM and the CQPM-PCA methods

4.4. Comparison with the work of other authors

In order to establish the credibility of the proposed framework in the domain for which it was designed, its performance should be validated [33]. This may be achieved by comparing the results returned by the models with those of other authors who have conducted analyses of the SECOM data set. The best results returned by applying both the basic CQPM and the CQPM-PCA methods were achieved by SVM algorithms. These results are presented in

Table 4, along with the results achieved by Salem *et al.* [21] and McCann *et al.* [22].

By comparing the results of Model B5 with those returned by the model of Salem *et al.* [21], it is evident that there is only a small difference of 0.36 per cent between the recall performances obtained. Although the AUROC and TNR achieved by the model of Salem *et al.* [21] slightly outperformed those of Models A5 and B5, the results still compare well with those of Salem *et al.* [21].

Models A5 and B5 both outperformed the model of McCann *et al.* [22] with respect to the recall. The TNR of those authors, however, were significantly larger than any of those achieved by Model A5 and Model B5. The difference between the two sets of results may be accounted for by the fact that Models A5 and B5 were tuned and selected primarily based on recall, with a primary focus on maximising the TPs and minimising the FNs. This, however, comes at the cost of achieving a smaller TNR.

Table 4: A comparison of our analysis results with those of other authors

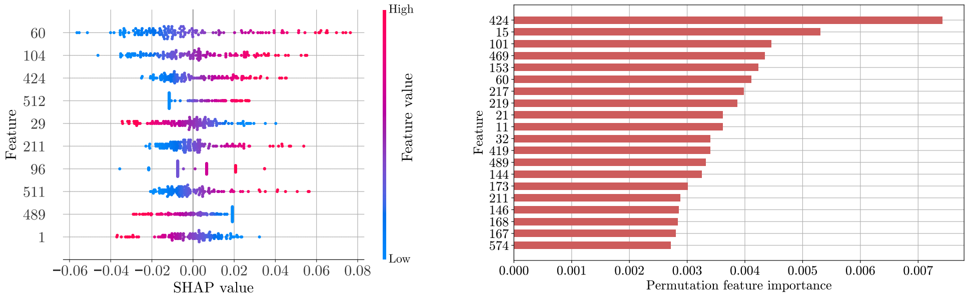
	Classifier	***Feature selection***	***Recall***	***AUROC***	***TNR***
Model A5	SVM	Anova	76.77	63.49	48.10
Model B5	SVM	Anova	77.52	62.33	49.66
Salem *et al.* [21]	kNN	SVM L1	77.88	65.96	50.07
McCann *et al.* [22]	Naïve Bayes	t-Test	59.60	—	73.00

A number of other authors also performed analyses on the SECOM data set, although in each of these cases data leakage occurred. Moldovan *et al.* [25], for instance, reported that data partitioning took place after data cleaning and feature selection. In order to evaluate fairly the generalisability of a model, the testing set should not be available for use during pre-processing decisions and model training so as to prevent the return of overly optimistic performances. Other authors applied resampling techniques to the entire data set [5, 17, 26]. Unfortunately, the results of the validation analysis cannot be compared with the results of these models, as they produce deceptively good results.

4.5. Model interpretation

In the context of the wafer manufacturing process, the interpretation of the machine learning defect prediction model results may offer significant insight to manufacturers. The model interpretation component was implemented with the aim of helping manufacturers to discover the root causes of defects by pinpointing manufacturing stages that are potentially inducing defects. Although Models B1–B5, resulting from the application of the CQPM-PCA method, yielded better classification results than Models A1–A5, the transformation of the feature space to principal components led to a loss of ability to interpret the pre-eminent features meaningfully. Thus Models A1–A5 were considered for interpretation purposes.

Interpretation of the results of the best-performing models was performed by two approaches, namely a permutation-based feature importance analysis and a SHAP value analysis. The results obtained when implementing these model interpretation techniques to the outputs returned by the LR model are shown in Figure 11.



(a) SHAP value analysis

(b) Feature importance analysis

Figure 11: The SHAP value scores and feature importance scores for the LR model

Model interpretation applied to different algorithms typically results in different sets of features being identified as significant, since these algorithms employ distinct methods for classifying data instances [34]. Furthermore, the various model interpretation techniques may also offer different results, because the manner in which they rate features as significant differs. Thus the interpretation of the results returned by the various models and model interpretation techniques was consolidated into a unified list of pre-eminent features. Figure 12 contains those features that were identified as important (i.e., ranked among the top ten) by at least two techniques applied to the five models. Feature 60 was the most frequently detected as influential, followed by features 104 and 424.

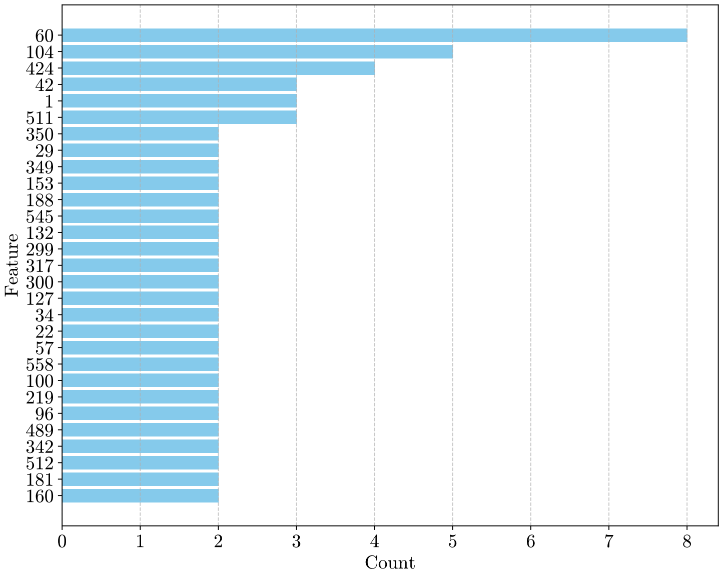


Figure 12: Top ten features ranked as important by at least two techniques applied to the five models

5. CONCLUSION

The primary research aim in this paper was to propose a generic modular framework for facilitating the tasks associated with preparing the data from multi-stage manufacturing systems, building machine learning models to predict the occurrence of product defects, and interpreting the results returned by these models. The efficacy and utility of the proposed framework was demonstrated by implementing a computerised proof-of-concept in respect of the SECOM data set. The implementation yielded satisfactory classification results that compared well with those achieved by other authors who conducted analyses of the data set. Moreover, the superior results returned by the final-stage CQPM-PCA modelling approach, as opposed to the basic CQPM (whose final stage model is equivalent to the single-point prediction method), suggested that there is an advantage in considering the data emanating stage-wise from a manufacturing system.

Potential avenues for future research include applying the framework to an additional data set and expanding the framework into a comprehensive decision support system that is capable of the real-time prediction of product defects. This would facilitate the deployment of trained models into a manufacturing system. Moreover, automation of the model retraining process based on data-driven techniques, such as detecting concept drift, may be pursued. Finally, further investigation of unstructured data, such as images, videos, or textual data, may hold promise for enriching defect detection capabilities.

REFERENCES

[1] Furman J & Seamans R, 2019, AI and the economy, *Innovation Policy and the Economy*, 19(1), pp. 161–191.

[2] Caiazzoa B, Nardob MD, Murinob T, Petrilloa A, Piccirillob G & Santini S, 2022, Towards zero defect manufacturing paradigm: A review of the state-of-the-art methods and open challenges, *Computers in Industry*, 134, pp. 1–15.

[3] Psarommatis F, Sousa J, Mendonca JP & Kiritsis D, 2022, Zero-defect manufacturing the approach for higher manufacturing sustainability in the era of Industry 4.0: A position paper, *International Journal of Production Research*, 60(1), pp. 73–91.

[4] Auschitzky E, Hammer M & Rajagopaul A, 2014, How big data can improve manufacturing, [Online], [Retrieved August 2023], available from https://www.mckinsey.com/capabilities/operations/our-insights/how-big-data-can-improve-manufacturing

[5] Ismail M, Mostafa NA & El-Assa A, 2022, Quality monitoring in multistage manufacturing systems by using machine learning techniques, *Journal of Intelligent Manufacturing*, 33, pp. 2471–2486.

[6] Arif F, Suryana N & Hussin B, 2013, Cascade quality prediction method using multiple PCA+ID3 for multi-stage manufacturing systems, *Proceedings of the 4th International Conference on Electronic Engineering and Computer Science*, Beijing, pp. 201–207.

[7] Lianga J, Pelzerc L, Müllerd K, Cramera S, Gerb C, Hopmann C & Schmitt RH, 2021, Towards predictive quality in production by applying a flexible process independent meta-model, *Proceedings of the 54th International Academy for Production Engineering Conference on Manufacturing Systems*, Patras, pp. 1251–1256.

[8] Tercan H & Meisen T, 2022, Machine learning and deep learning based predictive quality in manufacturing: A systematic review, *Journal of Intelligent Manufacturing*, 33, pp. 1879–1905.

[9] Lundberg SM & Lee S, 2017, A unified approach to interpreting model predictions, *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach (CA), pp 1419-1432.

[10] Breiman L, 2001, Random forests, *Machine Learning*, 45, pp. 5–32.

[11] Wuest T, Weimer D, Irgens C & Thoben K, 2016, Machine learning in manufacturing: Advantages, challenges, and applications, *Production and Manufacturing Research*, 4(1), pp. 21–45.

[12] Meyes R, Donauer J, Schmeing A & Meisen T, 2019, A recurrent neural network architecture for failure prediction in deep drawing sensory time series data, *Proceedings of the 54th SME North American Manufacturing Research Conference*, Erie (PA), pp. 789–797.

[13] Zhang B, Liu S & Shin Y, 2020, In-process monitoring of porosity during laser additive manufacturing process, *Additive Manufacturing*, 28(4), pp. 497–505.

[14] Tercan H, Khawli TA, Eppelt U, Büscher C, Meisen T & Jeschke S, 2017, Improving the laser cutting process design by machine learning techniques, *Production Engineering Research and Development*, 11, pp. 195–203.

[15] Arif F, Suryana N & Hussin B, 2013, Framework of cascade quality prediction method using latent variables for multi-stage manufacturing, *International Journal of Management — Theory and Applications*, 1(1), pp. 13–22.

[16] Guo X, Wang F & Jia M, 2005, A stage-based quality prediction and control method for batch processes, *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, Guangzhou, pp. 18–21.

[17] Kao H, Hsieh Y & Lee J, 2017, Quality prediction modeling for multistage manufacturing based on classification and association rule mining, *Materials Science, Engineering, and Chemistry Web of Conferences*, 123, a00029.

[18] Valerio A, Succi G & Fenaroli M, 1997, Domain analysis and framework-based software development, *Applied Computing Review*, 5(2), pp. 4–15.

[19] Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda, R & Carlsson C, 2002, Past, present, and future of decision support technology, *Decision Support Systems*, 33(2), pp. 111–126.

[20] Kazmaier J, 2020, *A framework for evaluating unstructured text data using sentiment analysis*, PhD dissertation, Stellenbosch University, Stellenbosch.

[21] Salem M, Taheri S & Yuan J, 2018, An experimental evaluation for fault diagnosis from imbalanced and incomplete data for smart semiconductor manufacturing, *Big Data and Cognitive Computing,* 30(2), pp. 1–20.

[22] McCann M, Maguire LP, Li Y & Johnston AB, 2016, Causality challenge: Benchmarking relevant signal components for effective monitoring and process control, *Information Technology and Computer Science*, 133, pp. 79–84.

[23] Wolpert DH & Macready WG, 1997, No free lunch theorems for optimisation*, IEEE Transactions on Evolutionary Computation*, 1(1), pp. 67–82.

[24] McCann M & Johnston A, 2023, UCI SECOM dataset: Semiconductor manufacturing process dataset, [Online], [Retrieved February 2023], available from https://www.kaggle.com/datasets/paresh2047/uci-semcom

[25] Moldovan D, Cioara T, Anghel I & Salomie I, 2017, Machine learning for sensorbased manufacturing processes, *Proceedings of the 13th IEEE International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, pp. 147–154.

[26] Nuhu AA, Zeehan Q, Safaei B & Shahzad MA, 2023, Machine learning-based techniques for fault diagnosis in the semiconductor manufacturing process: A comparative study, *Journal of Supercomputing*, 79, pp. 2031–2081.

[27] Bhatia SC, 2014, Solar devices, in Bhatia SC (Ed), *Advanced renewable energy systems*, New Delhi, India: Woodhead Publishing, pp. 68–93.

[28] May GS & Spanos CJ, 2006, *Fundamentals of semiconductor manufacturing and process control*, Somerset (NJ): John Wiley & Sons.

[29] Bourget L, Brucker G, Feaver M, Hill G, Ichihashi Y, Koai K, Larson A, Tiec CL, Martinez L, Pokidov I, Radomski A, Rosenzweig G, Rotem E, Tai C, Tricard M & Spyk MV, 2023, *MKS instruments handbook: Semiconductor devices and process technologies*, 2nd edition, Andover (MA): MKS Instruments.

APPENDIX A

This section contains detailed information about the application of the MSM-DA framework to the SECOM data set.

Table A.1: The implementation details of the various feature selection techniques

Feature selection technique	*Implementation*	*Specifications*
Anova	*SelectKbest()* function of the SciKit-Learn library.	*K* = 46 scoring = Anova F-value
MI	*SelectKbest()* function of the SciKit-Learn library.	*K* = 68 scoring = MI

SFPR	*SelectFpr()* function of the SciKit-Learn library.	scoring = Anova F-value
		α = 0.1

RFE	*RFECV()* function of the SciKit-Learn library.	model = LR 5-fold cross validation
Pearson	*corr()* function of the Pandas library.	deletion threshold = 0.8 method = Pearson
Kendall's Tau	*corr()* function of the Pandas library.	selected features = 40 method = Kendall

IBFS	SelectFromModel() function of the SciKit-Learn library.	model = LR
		threshold = mean

Table A.2: The hyperparameters and their values, and the tuning method applied to each of the machine learning algorithms

Algorithm	*Hyperparameters*	*Hyperparameter values*	*Tuning method*
DT	*max_depth* *min_samples_split* *max_features* *min_samples_leaf*	[2:10, None] [2:10, None] [2:10, None] [2:10, None]	Random search (100 trials)
RF	*estimators* *max_depth* *min_samples_split*	[10:100] [2:6] [2:6]	Random search (100 trials)
SVM	*C* *gamma*	[0.001, 0.01, 0.1, 1, 10] [1, 0.1, 0.01]	Grid search
LR	*penalty* *C* *solver*	l1 or l2 [0.001, 0.01, 0.1, 1, 10] liblinear or lbfgs	Grid search
kNN	*n_neighbours* *weights* *p*	[0:20] uniform or distance [1, 2]	Random search (100 trials)