Statistical Methods for Enhanced Metrology in Semiconductor/Photovoltaic Manufacturing

Dekong Zeng

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-237
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-237.html

December 12, 2012

Statistical Methods for Enhanced Metrology in Semiconductor/Photovoltaic Manufacturing

by

Dekong Zeng

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Costas J. Spanos, Chair

Prof.Seth R Sanders

Prof. Ramamoorthy Ramesh

Fall 2012

ABSTRACT

Statistical Methods for Enhanced Metrology in Semiconductor/Photovoltaic Manufacturing

by

Dekong Zeng

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor Costas J. Spanos, Chair

As semiconductor technology is aggressively scaling to finer feature sizes, manufacturing complexity increases dramatically. This drives the need for extensive control on processing equipment and on the efficiency of the associated metrology. Similarly, in the field of photovoltaic (PV) manufacturing, processing technology is driven by cost reduction while increasing output power per cell. In either case, the variability impact on the final performance is critical.

In this thesis, we focus on the application of statistical methods for enhanced metrology in both semiconductor and PV manufacturing. The work falls into three main topics: Wafer-to-Wafer (W2W) Virtual Metrology (VM) via predictive modeling, Site-to-Site (S2S) metrology modeling for Fault Detection and Classification (FDC), and predictive variability modeling for solar PV.

The first topic is on creating predictive VM models for W2W control in plasma etching, one of the bottlenecked processes for technology node scaling. The idea is to utilize equipment sensor data to predict the wafer processing results, so that actual wafer measurements can be reduced or eliminated. VM comprises four main steps: data extraction, outlier removal, variable selection, and model creation. They aim to deal with the special characteristics of equipment sensor data which are high dimensional, collinear and non stationary. VM models are trained and tested with approximately one production year worth of wafer data collected from a single plasma etching tool. The best model result is obtained by a hybrid model that utilizes step-wsie parameter selection and Neural Network (NN) based prediction, which achieved a testing $R^2 \approx 0.75$.

The second topic aims to develop FDC schemes for wafer-level S2S metrology. We first focus on utilizing spatial and multivariate statistics for detecting outlier wafers. Spatial and multivariate methods are preferred given the temporal and spatial varying nature of wafer level metrology data. We then focus on selecting the optimal measurement sites for process monitoring. Various site selection schemes are evaluated within the FDC application, showing than more than 70% metrology savings with no discernable reduction in performance is possible.

The third topic addresses modeling the variability of solar cells. The impact of environmental and manufacturing variability is simulated and discussed. A predictive model for manufacturing variability-induced mismatch power loss is proposed and evaluated with various PV array configurations. Finally, spatial statistics are used to model the non-uniformities of solar cell properties. A SPICE-based distributive solar cell simulator is constructed to estimate electrical performance for various defect distribution patterns. Finally, a statistical model is created in order to correlate the spatial characteristics of defect patterns with the corresponding electrical performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like thank my research advisor Professor Costas.J.Spanos for his excellent guidance on my research, his continuous support, and patience. His insights in the semiconductor manufacturing industry have led me to the interesting research topics.

Secondly I would like to thank my committee members: Professor Tsu-jae King, Professor Ramamoorthy Ramesh and Professor Seth Sanders for their kind support and precious time to serve on my thesis committee.

I would also like to thank Henry Lin from TSMC and Broc Stir from Global Foundries for providing process data on various semiconductor equipments and useful feedback on my Master research work.

I would also like to thank my supervisor Thuy Tran Quinn from Philips Semiconductor and NXP semiconductor for offering two summer internships to work in the etching group in the summer of 2005 and 2008.

I would also like to thank Lam Research Corporation for providing the Lam fellowship to support my research and offering me the internship opportunity to work with the sensor group at Lam in the summer of 2010. Special thanks also go out to all members of the sensor group at Lam for their support and enlightened discussions on plasma etching equipment: Vijay Venugopal, Eric Pape, and Sathyendra Ghantasla.

Special thanks go out to all my team members of BCAM group for their support and useful discussions on all subjects: Jing Xue, Kun Qian, Yu Ben, Qianying Tang , Claire Baike and Qiao Ying.

I would also like to thank all my family members especially my mom, my sister and my dear wife Vivian Chen for their love and support throughout my life. It is their encouragement that drives me to overcome all the challenges.

Chapter 1

Introduction

1.1.Motivation

Semiconductor processes and equipment have gone through tremendous amount of changeover time in order to meet the scaling required by the new technology nodes. The semiconductor industry has been driven to double the number of transistors on a wafer every two years. Thus, the ability to obtain good yield per wafer is critical for higher profit. In order to achieve good yield, numerous metrology steps along with advanced process control schemes have been implemented to ensure that the process stays in control. We show the typical process flow for CMOS along with key metrology measurements in figure 1.1.

The cost of metrology required to meet control specs is increased over time. There is also increasing demand of statistical methodology for process control. Measurements must be sampled statistically to accommodate process dynamics and correlation among measured parameters.

In current practice, it is difficult to measure wafer process results after each step for all the wafers since each wafer can go through hundreds of steps while equipment are operating in high volume manufacturing mode. Therefore, 1 to 3 wafers are typically sampled from each lot of 24 or so in order to perform measurement. The sampled wafers might be different from step to step, thus it is rare to have exactly the same wafer measured across all the steps. The measurements results of these wafers then used as statistical inputs to construct statistical process control charts (SPC) for product quality and process capability monitoring.

Run-to-run (R2R) control schemes are based on these measurements to adjust process recipes for each lot; however such lot-to-lot (L2L) control is not sufficient to reduce process excursions in a production environment. Therefore wafer-to-wafer (W2W) control is often necessary. The goal of W2W control is to adjust the incoming wafer recipe based on process results of previous wafers.

In order to provide inference on process results, various types of sensors have been implemented on tool to extract information on equipment and wafer. The information carried by these sensors can infer the process chamber condition as well as wafer processing results. One can utilize the information provided by these sensors to predict the wafer metrology results immediately after processing; these predicted results can then be fed into W2W control to adjust the recipe for next wafer. We define the prediction of metrology variables with the process and chamber state information as W2W Virtual Metrology (VM). W2W VM can be used for reducing metrology measurement as well as fault detection and classification (FDC).

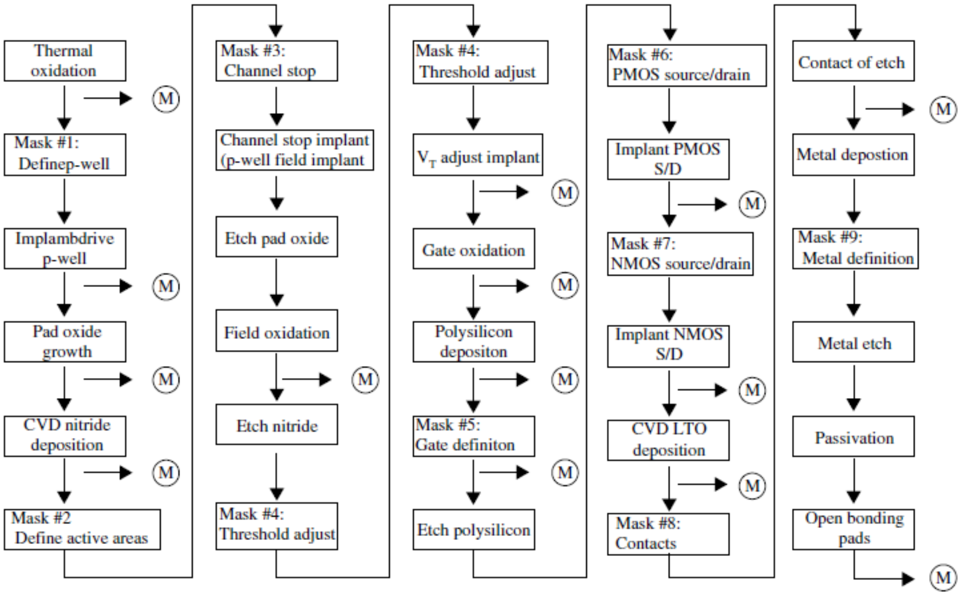


Figure 1.1 CMOS process flow showing key measurement points (denoted by M)[1.1]

However, VM does not mean that actual measurements can be completely eliminated. Due to the time varying nature of process dynamics, malfunctioning of sensors and frequent equipment maintenance, one would need to constantly update the VM models with measured wafers.

Metrology requirements and opportunities for different process steps vary. In table 1.1, we provide typical metrology operations for four main process steps. For all the measurements, several sites across wafer are usually measured. These locations are usually selected based on engineering knowledge. Thus, optimality of the selection is not guaranteed, and a portion of the measurements may be redundant in terms of the information they collect about the status of the process.

Measurement redundancy is especially likely for process steps that tend to result in specific spatial patterns across wafer. Such patterns make the process deviations site-dependent. Thus, it is possible to sample only the most informative sites for measurement across wafer and reduce measurement cost.

In plasma etching operation, for example, Scanning Electron Microscopy (SEM) has been used to measure the critical dimensions (CDs) of etched features. Typically, CDs of 9 to 14 sites across wafer are measured. It has been shown that these measured CDs are highly correlated with each other. Thus, one can really reduce the number of sites measured by selecting the ones with the highest predictive power. We define the process of selecting measurement set as Site-to-Site (S2S) VM. S2S VM can also be applied for electrical testing site selection, where there are hundreds of dies needed to be measured per wafer.

Process step	Measurement
Lithography	• Linewidth • Overlay

	• Print bias • Resist profiles
Etch	• Etch rate • Selectivity • Uniformity • Anisotropy • Etch bias
Deposition	• Sheet resistance • Film thickness • Surface concentration • Dielectric constant • Refractive index
Diffusion and Implantation	• Sheet resistance • Junction depth • Surface concentration

Table 1.1 Metrology for main CMOS process steps [1.1]

Photovoltaic (PV) manufacturing, while simpler, shares similar processing technology with IC manufacturing. Unlike IC manufacturing, however, where yield and performance dominate, in the PV industry low production cost is paramount. Indeed, in order to have competitive cost for each KWh produced by a PV system, one must tend to manufacturing cost, installation cost, and solar conversion efficiency over the lifetime of the system. To achieve low manufacturing cost, various materials have been used for PV cell fabrication along with different system design. We present the up-to-date reported efficiency for different PV cells in figure 1.2, and the cost of different components in figure 1.3.

The cost of PV components can vary depending on the market supply and demand circumstances. Additional variability comes from variations in manufacturing, in the balance of the system components, and in the environment in which the OPOV system will operate. Therefore, it is important to understand how these variability components will impact the final solar PV system. Once the relationship between these variability components and PV performance is understood, one can optimize the PV system to compensate the variability and maximize the power output.

There have been many attempts to mitigate the impact of variability in PV systems. In the case of manufacturing variability, binning on cells has been applied to group cells into a class with more conformal performance. Thus, mismatch induced power loss can be reduced. Mismatch induced power loss happens when cells with different performance capability are combined, where the maximum total power output is less than the sum of the maximum power of individual cells. Accurate estimation of the mismatch induced power loss under different conditions allows a PV plant to forecast power output.

Comprehending environmental variability is also needed in order to forecast the PV system performance. In current practice, uniform irradiance and temperature profiles are generally assumed across entire PV plant or array when calculating energy output. The assumption of uniform profiles can generally lead to over-estimation of energy output.

Besides the need of understanding variability impact at a macro-scale level, the introduction of thin-film PV cells is also driving more research towards understanding variability impact at a micro-scale level, such as non-uniform material properties across cell. These nonuniformities, including defect density variation, deposited layer thickness variation, and other material properties tend to diminish final cell performance. Therefore, it is important to establish the relationship between these nonuniformities and final cell performance.

Multiple measurements have also been employed at different process steps to monitor quality of cells. In table 1.2, we summarize the standard set of metrology measurements for fabrication of Si-wafer PV cells [1.4].

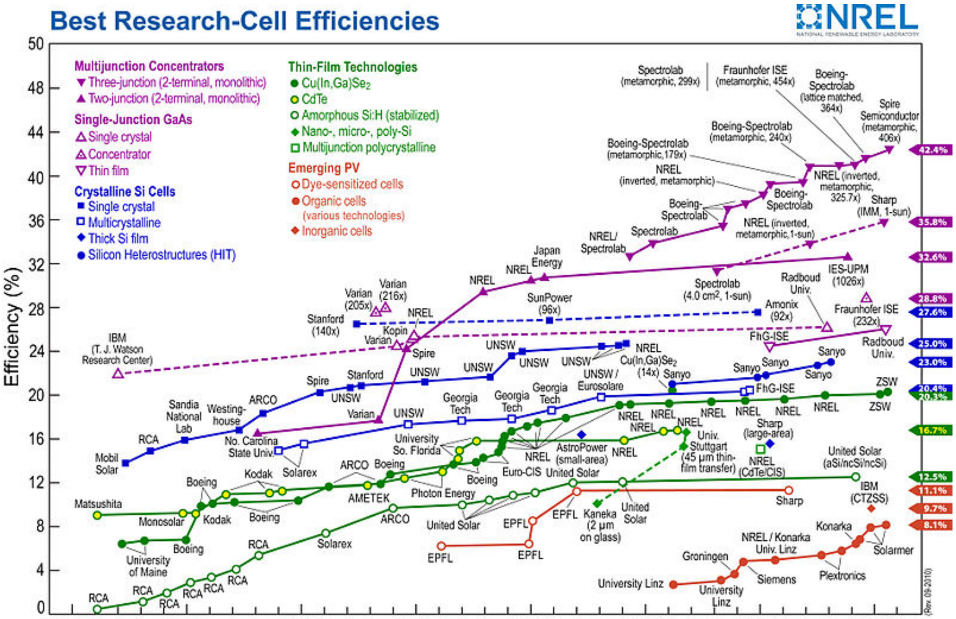


Figure 1.2 NREL compilations of best research solar cell efficiencies [1.2].

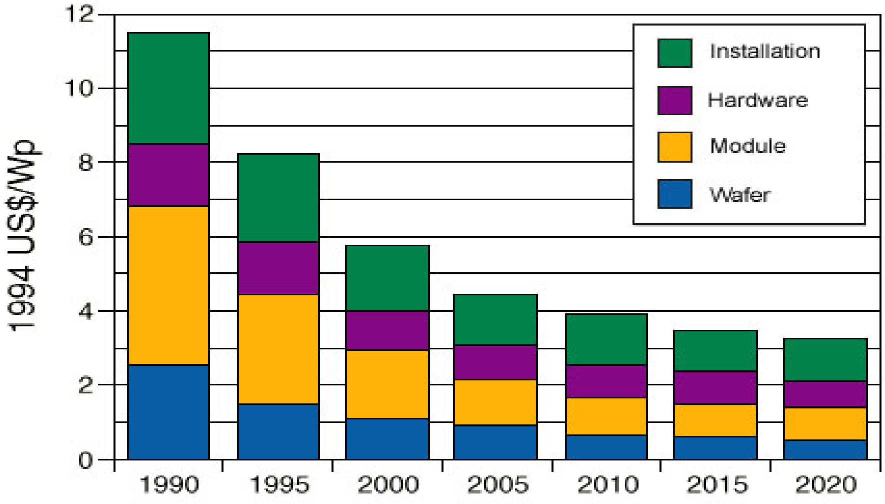


Figure 1.3 Component cost of PV system and Price reduction over time [1.3]

Process or Monitoring step	Parameters	Technique
Crystal growth (Ingot quality)	τ=minority carrier lifetime	PCD
Wafer quality and cleaning	Surface Roughness/residue	NA
Texturing	Texture Height	SEM/Optical microscopy/Reflectance
Electronic quality of wafers	$L/_{\tau}$ =diffusion length/minority carrier lifetime	PCD
Junction depth	Sheet resistance	4 Point Probe
Defect density	Dislocation density	TEM
Impurity concentration		SPV
AR-coating	Thickness and refraction index	Ellipsometer
Metallization	Line width	SEM/Optical
IV-of Cell	Open circuit voltage and short circuit current	Standard I-V measurement

Table 1.2 Metrology for main PV process steps [1.4]

1.2.Thesis organization

This thesis starts with the overview of enhanced metrology in the semiconductor manufacturing industry in Chapter 2. Multivariate statistical methods for FDC and VM model prediction are discussed qualitatively in Chapter 2. Chapter 3 develops a VM model for semiconductor process and its application on plasma etching. We first present the model creation flow and discuss each of components in detail.

Semiconductor manufacturing data are usually highly dimensional and collinear. Especially in the case of plasma etching process, each wafer generally goes through multiple recipe steps and each recipe step can have more than hundreds of sensor monitoring chamber and process conditions. Prior to applying a predictive method, we first describe how we compress these sensor data into statistical summaries that can capture most information at each recipe step. Given the collinear nature of sensor data, variable selection plays a critical role in creating a robust model with good performance. We present several variable selection schemes and evaluate their performances with different prediction methods.

In order to accommodate process dynamics over longer production period, we also present several ways of constructing the prediction model. Chapter 4 focuses on developing FDC schemes and optimal site selection schemes for S2S metrology. We demonstrate the results with the application on a set of wafers with frequency of pattern sensitive ring oscillator measured at each die. In order to gain insights into data with such hierarchical and spatial nature, we have utilized conventional spatial statistics and multivariate methods for detecting abnormal wafers. Another aspect of Chapter 4 is to discuss how we can select the sites with most information for process monitoring, since many sites are highly correlated with each other.

In Chapter 5, we utilize statistical techniques with SPICE simulation to model variability components in a PV system. Monte Carlo (MC) simulation is first used to simulate cells with different physical and electrical characteristics to reflect manufacturing variability. Then we construct a mismatch estimation model for different PV interconnect configuration with these generated samples. To mitigate environmental and manufacturing variability, we then propose clustering techniques to bin cells rather than using a single metric. The binning performances are then evaluated under different PV interconnect configuration and degradation conditions.

Nonuniformities of material properties across a cell also result in fabricated cells with different performance. These nonuniformities are introduced by manufacturing as well inherited from raw material. Especially in the case of thin film cell with large surface area, nonuniformities play s a critical role in final cell performance. Understanding the impact of different patterns of nonuniformities is critical for tuning process. We first generate different defect patterns to represent nonuniformities, then these patterns are simulated in SPICE as a distributive diode array. Statistical methods are then applied to correlate these defect patterns with the corresponding simulated performance. Finally, we conclude the thesis and propose future work in Chapter 6.

Reference

[1.1] Gary S. May and Costas J.Spanos, *Fundamentals of Semiconductor Manufacturing and Process Control*, John Wiley and Sons, New York, 2005

[1.2] Lawrence Kazmerski, National Renewable Energy Laboratory (NREL).

[1.3] IEA Photovoltaics Power Systems Programme (IEA PVPS) Annual Report April 2010

[1.4] B. Sopori, “Process Monitoring in Solar Cell Manufacturing”, 9th Workshop on Crystalline Silicon Solar Cell Materials and Processes, Breckenridge, Colorado, August 9-11, 1999

Chapter 2

Background on IC, PV Manufacturing and Relevant Statistical Methods

In this chapter, we will briefly review the semiconductor and photovoltaic (PV) manufacturing process and provide an overview on the application of statistical methods in manufacturing control. In Section 2.1, we will provide a brief overview of the conventional CMOS fabrication process with a focus on the bottlenecked steps. In Section 2.2, we will discuss the process flow for crystalline Si solar cells and thin film solar cells and their underlying differences. Statistical techniques and their previous applications in the field of manufacturing control are then discussed in Section 2.3.

2.1.Review of Semiconductor Manufacturing

There are typically 6 main groups of steps for CMOS manufacturing [2.1] including: diffusion, photolithography, etch, ion implant, thin films deposition, and polishing. A model for the typical wafer flow in Sub-Micron IC Fab is shown in figure 2.1. A blank silicon wafer generally goes through approximately 400 steps in 6 to 8 weeks to get patterned and then is sent for test and assembly, where electrical tests are performed for each die. A cross section view of a complete CMOS structure along with the main manufacturing steps is presented in figure 2.2.

Diffusion is generally applied to introduce impurities into pure silicon. Photolithography is a process that transfers the circuit pattern to the wafer with uses of photoresist and light exposure. Ion implantation is the process where dopants are implanted into silicon substrate to change its electrical characteristics.

Etching transfers patterns printed in photoresist into the wafer. Plasma etching is typically used in modern IC Fab. This is done by placing wafers into a vacuum chamber which is filled with chemical gas, and then a strong Radio Frequency (RF) electromagnetic field is applied to the wafer. This RF field converts the chemical gas into chemically reactive ions. The charged ions are accelerated toward the wafer surface by the electromagnetic field and where they chemically and physically remove the exposed material. For IC manufacturing, etching processes can be classified by the etched material: conductor etch and dielectric etch. Conductors include poly-silicon (p-Si), Silicon (Si) and metals. In both cases of Si and metal etching, high density plasmas operated at low pressure in the order of 10 mTorr in single wafer tools are employed. With low pressure, a long mean free path of reactive ions assures a high degree of ion sputtering and highly anisotropic etching. Metal and dielectric etchings are both for patterning, interconnect as well as active device features.

Thin film processes are used to interconnect the transistors on a chip and make them into a functional device. Chemical Vapor Deposition (CVD) is a class of thin film processing using controlled chemical reactions to create layers on wafers. There are many variants of CVD processes, such low pressure CVD (LPCVD), physical vapor deposition (PVD) and plasma enhanced CVD (PECVD). LPCVD is usually preferred for poly-Si deposition. PVD is also known as sputtering and is typically used for metal deposition. It uses a cathode to create Argon plasma which bombards the source metal. The “knocked off” metal molecules are then focused by a collimator, and deposited on the surface of the wafer. PECVD uses gas plasma to lower the temperature required to obtain a chemical reaction and achieve film deposition. It is the preferred method for depositing inter-metal dielectric since it delivers the required high aspect ratio between metal lines.

Polishing, also known as planarization, is a combination of chemical and mechanical polishing (CMP). This process is used repeatedly in the fabrication flow for oxide and metal planarization. It provides planarity for oxide dielectric and tungsten plug used at the metallization level. The success of CMP depends on several factors, including the chemistry of slurry, the nature of the pad, and mechanics of the tool.

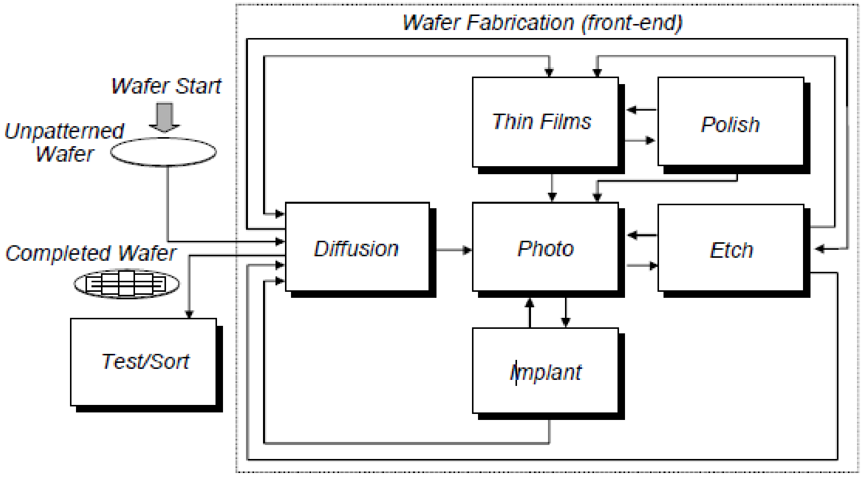


Figure 2.1 Model for typical wafer flow in Sub-Micron CMOS IC Fab [2.2]

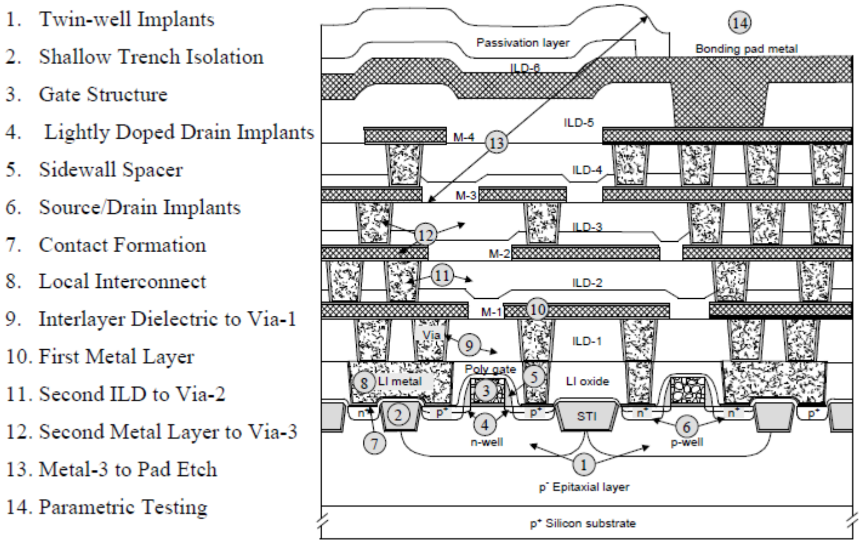


Figure 2.2 CMOS manufacturing steps and cross section view of complete CMOS [2.3]

Each finished wafer may contain several hundred individual devices called "die." Semiconductor device manufacturers use automated methods to test each device on the wafer before it is broken into "chips". A probe tester uses needle shaped "probes" to contact the bonding pads (the circuit connection points) on each device and check its operation. Devices that fail the test are marked with a colored dye spot so they will not be carried further into the production process. The efficiency of a given Fab is determined by its yield: the ratio of functional die to total die. State-of-the-art Fab typically operates at more than 90% yield within 18 months of startup [2.4]. After the electrical test, the wafer is sawed with a special diamond saw and broken into individual die. The marked (non-functional) die are discarded and functional die are passed on into the wire bonding process. In figure 2.3 (A) and (B), we show the die testing result and chip packaging for illustration.

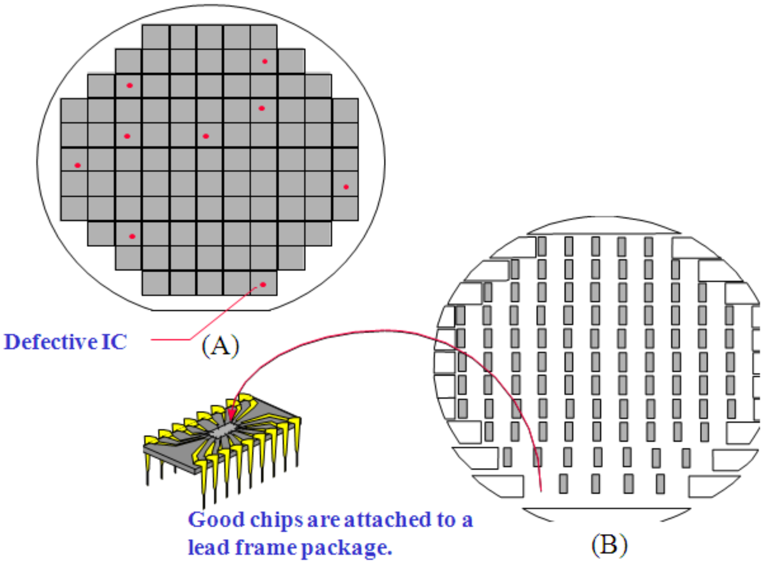


Figure 2.3 (A) Die testing (B) Chip packaging [2.5]

2.2.Review of PV Manufacturing

The fabrication process of PV cells is relatively simple in comparison to IC manufacturing since the PV cell does not have a complicated structure. However, there are many variants in terms of process due to different types of PV cells. PV cells can be classified into two categories broadly based on the cell material: crystalline silicon (C-Si) based cells and thin film cells. The chosen materials need to meet two criteria: high efficiency and low cost. The main focus of this section is to discuss manufacturing flow for PV cells, and we will discuss solar cell operation and electrical characteristics in detail in Section 5.

Solar cell plants take the wafer through a high technology semiconductor processing sequence to create working solar cells. In C-Si, wafers typically undergo a process sequence of etching, diffusion, and screen-printing steps before they are tested and graded for incorporation into modules. For thin films, glass or stainless steel substrates are processed through steps of transparent conducting oxide deposition, semiconductor layer growth, laser scribing, and metallization. The sequence is dependent on the substrate being used. Today, thin film plants are designed to handle large substrates in sheet or roll form. Therefore, the process equipment is much larger than for the wafer-based C-Si plants.

2.2.1. Silicon Solar PV cell fabrication

Most large-scale commercial solar cell factories today make screen-printed poly-crystalline silicon (Poly-Si) solar cells. Single crystalline silicon (C-Si) wafers which are used in the semiconductor industry can be made in to excellent high efficiency solar cells, but they are generally considered to be too expensive for large-scale mass production. The fabrication of Si wafer based PV cells can be summarized into the following steps [2.6, 2.7]:

1. Poly-Si or C-Si wafers are made by wire-sawing block-cast silicon ingots into very thin (250 to 350 micrometer) slices or wafers. The wafers are usually lightly p-type doped.
2. To make a PV cell from the wafer, n-type diffusion is performed on the front side of the wafer, forming a p-n junction a few hundred nanometers (nm) below the surface. Some cells can also have a textured surface to couple more light into the cell. Antireflection coatings (ARC) are typically applied on the front surface to increase light coupled into the solar cell. Silicon nitride is the typical ARC material because of its excellent surface passivity qualities, so that it can prevent carrier recombination at the surface of the solar cell. ARC is typically several hundred nm thick and applied by PECVD.
3. The wafer is then metalized, whereby a full area metal contact is made on the back surface, and a grid-like metal contact made up of fine "fingers" and larger "bus bars" is screen-printed onto the front surface using a silver paste. The rear contact is also formed by screen-printing a metal paste, typically aluminum. This contact usually covers the entire rear side of the cell, although in some cases this rear contact can also be printed in a grid-pattern. The metal electrodes will then require some kind of heat treatment or "sintering" to make Ohmic contact with the silicon.
4. After the metal contacts are made, the solar cells are interconnected in series (and/or parallel) by flat wires or metal ribbons, and assembled into modules or "solar panels". Solar panels have a sheet of tempered glass on the front, and a polymer encapsulation on the back.

In figure 2.4, we show the typical fabrication flow for C-Si wafer based solar cells.

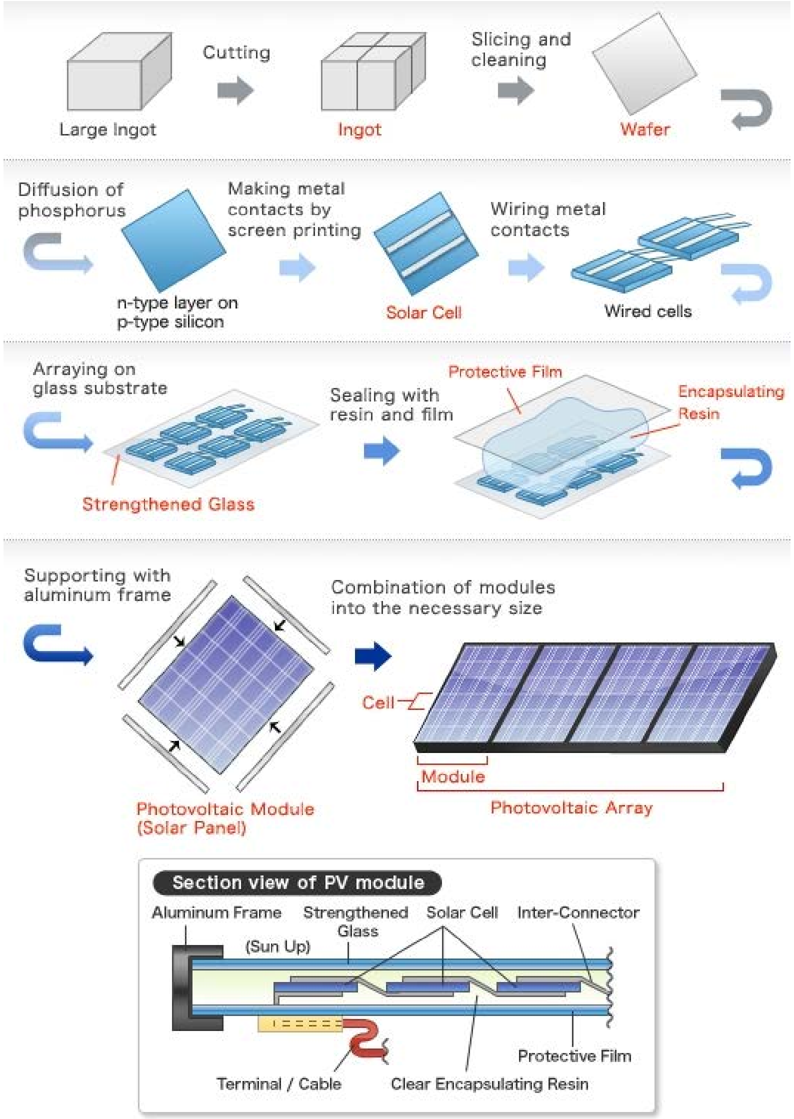


Figure 2.4 Typical fabrication flow for C-Si wafer based solar cell
[Courtesy of SUMCO CORPORATION]

2.2.2. Thin Film Solar PV Cell fabrication

Thin-film PV cells are the alternative for PV cell cost reduction. Thin-film PV cells use less than 1% of the raw material (silicon) compared to wafer-based PV cells, leading to a significant price drop per kWh. The reported thin-film cells with relative good efficiency and reasonable cost include: Cadmium Telluride (CdTe), Copper Indium Gallium Selenide (CIGS), Gallium Arsenide (GaAs) multi-junction, light-absorbing dyes, organic/polymer, and polycrystalline Si thin-film.

The fabrication of thin-film PV cells is slightly different from the Si-wafer based one; thin-film cells are generally fabricated with monolithic integration where the entire PV module is fabricated. In figure 2.5, we present the fabrication flow of CIGS thin-film PV cell; this process is also similar to the one for CdTe cell. However, it can vary significantly for other thin-film ones. Fabrication process for thin-film cells involves the deposition of different thin films in sequence on a large area substrate. A typical manufacturing process flow begins with cleaning the glass substrate, and then followed by transparent conducting oxide (TCO) deposition and window layer deposition, sequentially. Then, laser scribing is used to define, interconnect and isolate the cells. This manufacturing flow allows monolithic integration of cells in the module with minimum area losses. The production yield for thin-film cells depends on important factors such as uniformity of deposition and the laser scribing process for interconnect Since thin-film PV cells collect current laterally with spaced metal fingers, variability in material properties over a large area will lead to severe degradation in overall device performance parameters. Open circuit voltage (Voc) is known to be the most impacted parameter due to local material variation in layer thickness and bandgap fluctuations. Therefore, it is important to understand the impact of non-uniformities in thin-film PV cells.

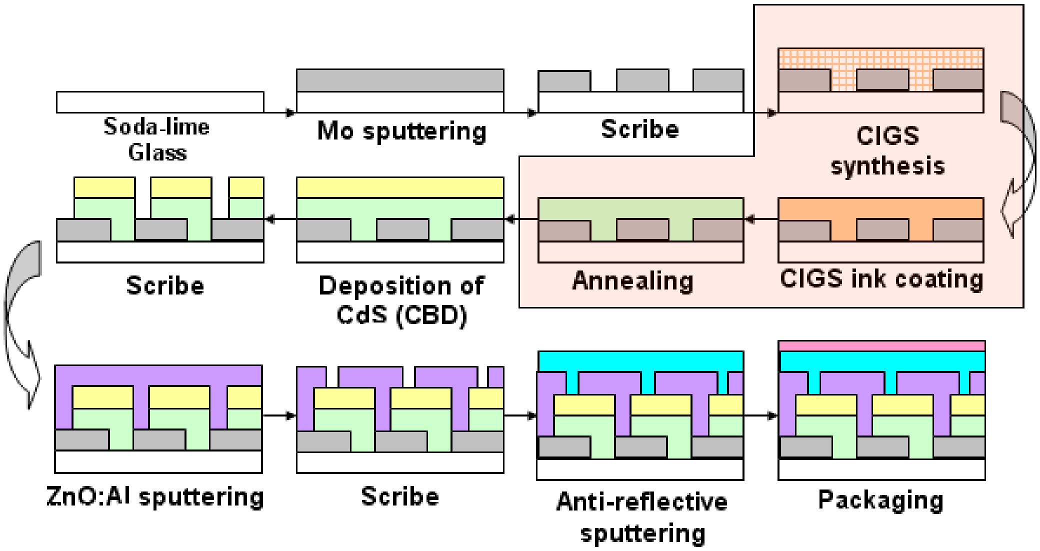


Figure 2.5 CIGS based solar cell fabrication flow [2.8]

2.2.3. Module Assembly and System Installation

The assembly of Si-wafer based PV modules is most commonly carried out in the cell plant, but can be done in smaller plants closer to the end market. Smaller plants can be preferable because, while solar cells are relatively inexpensive to transport, modules with a glass front sheet and an aluminum frame are heavy and bulky. In general, thin-film PV modules must be assembled in the cell plant because the cells are too susceptible to mechanical damage during transport, unless they are packaged within a module. PV module assembly usually involves soldering cells together to produce a string of cells, and then laminating it between toughened glass on top and a polymeric backing sheet at the bottom. Frames are usually created to allow for mounting in the field. The laminates may also be separately integrated into a mounting system for a specific application, such as building integration.

For system installation, an array structure is first chosen for the mechanical integration of the solar module. This array structure will depend on the final location of the system, which could involve retrofitting onto a roof, integrating into building materials for roofs or vertical walls, or pole-mounting, ground-mounting, or attaching to an industrial structure.

Second, the electrical components are integrated with other parts of the solar energy system. This will include the connection of elements such as inverters, batteries, wiring, and regulators (charge controllers). This process also requires matching the module array with the electrical load as required by the customer. Computer software, known as a sizing program, is typically used to make this calculation.

2.3.Statistical Techniques for Semiconductor Manufacturing Control

Two statistical techniques in the area of semiconductor manufacturing are of interest in this work: fault detection and classification (FDC) and predictive modeling.

The goals of FDC are to detect process excursion and identify its cause. Typical procedures for implementing a FDC model involve extensive off-line training of wafers with known specific fault signatures or wafers with abnormalities, and on-line testing of trained model and improvement of model by compensating for process dynamics. The performance metric of any FDC model should minimize the type I (true alarm missing rate) and type II (false alarm rate) errors.

Off-line FDC model training wafers can be obtained from two sources: using design of experiment (DOE) to get wafer with specific faults such as gas flow or RF power for plasma etching operation, and utilizing historical wafers from production. DOEs can provide much cleaner data for model training, but they are very costly and the number of training wafers is usually small. Sampling from production wafers often produces noisy data, but it can cover larger process space and a trained model can be more robust by capturing real process dynamics such as tool aging and process drifting. Equipment aging can cause a monitoring sensor to provide false readings and process drifting can either be introduced by changing of chamber condition due to aging or needed adjustments to compensate for process dynamics.

Online implementation of an FDC model is usually done with either univariate or multivariate statistical process control (SPC) charts. The concept is to define upper and lower control specs on monitoring parameters. The trained FDC model is used to define the control specs. Monitoring parameters can be the ones with physical meaning such as pressure sensor readings, electrical metrology results, or statistics computed by utilizing available information from each wafer. We will discuss FDC methods in detail in Section 2.3.1.

The goals of predictive modeling in semiconductor manufacturing can be inferring equipment condition based on equipment sensor data log, predicting typical process results such as etch bias after plasma etching or source/drain current after ion implantation, or sampling electrical measurement metrology. Different prediction methods are discussed in Section 2.3.2.

2.3.1. Fault Detection and Classification

FDC (fault detection and classification) methods can be classified into two categories based on the number of inputs the model requires.

2.3.1.1. Univariate Methods

Univariate SPC for semiconductor manufacturing applications generally assumes the monitored parameters follow Normal distribution, and 3-sigma control charts with +/- 3 sigma for upper and lower limits are used to track if a monitored parameter stays within specs during production. The outliers are assumed to be a small number of observations randomly sampled from distributions differing from the original Normal distribution [2.9]. The probability density function (PDF) of the univariate Normal distribution is:

$$f(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{[(x-\mu)/\sigma]^2/2} \tag{2.1}$$

where $f(x) > 0$ and $\int_{-\infty}^{+\infty} f(x)dx = 1$.
To identify the regions where outliers are located in, confidence coefficient α , $0 < \alpha < 1$ is used to define the α-outlier region:

$$out(\alpha, \mu, \sigma^2) = \left\{x: |x-\mu| > z_{1-\alpha/_2}\sigma\right\} \tag{2.2}$$

$z_{1-\alpha/_2}$ is the $1 - {}^{\alpha}/_2$ quintile of the $N(0,1)$. An observation x is an α-outlier outlier if $x \in out(\alpha, \mu, \sigma^2)$. The area under the Normal distribution PDF between $\mu + k\sigma$ and $\mu - k\sigma$ are 0.6827, 0.9545, and 0.9974. $k = 3$ is often used as the threshold for outlier detection. That means the probability that a point lies outside 3 sigma away from the mean is just 0.0027.

The definition in Equation 2.2 is for normal distribution, but it can be extended to any unimodal symmetric distribution. In the case of non-normal distribution, Kernel density estimation is a popular empirical non-parametric method for density estimation [2.10]. The kernel estimator takes the form:

$$\hat{f}(t) = \frac{1}{n}\sum_{j=1}^{n} K\left(H^{-1/2}\left(t - T_j\right)\right) \tag{2.3}$$

where $\hat{f}$ is the probability density for a new observation, and is estimated by taking a weighted summation of n kernels, each centered at an observation, T_j. The kernel function, K, provides the shapes of the underlying distribution are summed in order to give the overall density estimate. Most commonly, the kernel function is selected to be multivariate normal, yielding

$$\hat{f}(t) = \frac{1}{n(2\pi)^{l/2}} |H|^{-1/2} \sum_{j=1}^{n} e^{-1/2\left(t-T_j\right)^T H^{-1}\left(t-T_j\right)} \tag{2.4}$$

The bandwidth, H, defines the smoothness of the underlying distributions, which has a direct impact on the breadth of the final density estimate. In this thesis, we have swept the H to obtain optimal value. Martin [2.11] applied kernel estimator to estimate the distribution of PCA scores where the underlying distribution is not normal.

2.3.1.2. Multivariate Methods

Multivariate SPC charts utilize high dimensional data to compute a single statistical index (statistic) for each wafer, and monitor to ensure the statistic is within defined limits. The limits for a multivariate SPC chart can be defined with parametric assumption or empirical estimation.

2.3.1.2.1. Principal Component Analysis (PCA)

PCA [2.12] has been discussed in detail in literature as an effective technique for data compression and feature extraction, and it has been widely used in semiconductor manufacturing for process monitoring purposes. The methods focus on finding the main directions of data variations, reducing the analysis into a smaller dimensional space that is more suitable for real-time application such as process monitoring where fewer variables are monitored.

The common approach for applying PCA is to perform Eigen-decomposition on the empirical covariance matrix of the dataset. In order to ensure that the extracted principal components maximizing the variance of the entire dataset, mean centering and unit variance scaling is needed prior to compute covariance matrix in the following form:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_n)\,(x_i - \bar{x}_n)^T = \left(\frac{1}{n}\right)X^TX \tag{2.5}$$

X is the *n*-by-*p* data matrix having each wafer as an observation. PCA decomposes the data matrix X as a sum of the outer product of the score vectors $\boldsymbol{t_i}$ and loading vectors $\boldsymbol{p_i}$ plus residual matrix E.

$$X = \boldsymbol{t_1 p_1^T} + \boldsymbol{t_2 p_2^T} + \cdots \boldsymbol{t_k p_k^T} + E = TP^T + E \tag{2.6}$$

The loading matrix P contains all the loading vectors as column vectors, and the loading vectors are the eigenvectors of the covariance matrix $\sum$. The eigenvalues explain the variance of each direction defined by the corresponding loading vectors. The loading vector with the largest eigenvalues is the first principal component (PC) and the second largest as the second PC. The first PC aligns with the greatest variation in the data while the second PC aligns with the greatest amount of variation that is orthogonal to the first PC. The score is obtained by projecting the X into the new space defined by P.

$$T = XP \tag{2.7}$$

The score vector is a linear combination of *p* variables. It contains the information where the data points are located in the new spaces. A formal criterion used to select the number of PCs is the total variation explained by the retained loadings. It is typically required that it is greater than 90%.

Conventional PCA is sensitive to outliers due to its dependence on the covariance matrix. A robust version of PCA has been discussed in literature [2.13]. A group of robust PCA methods replaces the classical covariance matrix with a robust covariance estimated, such as the reweighted minimum covariance determinant (MCD) [2.14] estimator. Some methods also scale the data with depending on robust parameters such as the median and the quartile, rather than mean and variance. Unfortunately, the use of these affine equivariant covariance estimators is limited to small to moderate dimensions. When the dimension is larger than the sample size, the MCD estimator is not defined. A second problem is the computation of these robust estimators in high dimensions. A second approach to robust PCA uses projection pursuit techniques. These methods maximize a robust measure of variance to obtain consecutive directions on which the data points are projected.

PCA model based Hotelling's T^2 and sum of squared residuals also known as SPE statistics [2.15] are the two most common metrics applied in semiconductor manufacturing for multivariate process monitoring. Hotelling's T^2 measures the variation of each sample within the PCA model. T^2 is the sum of normalized squared scores defined as:

$$T_i^2 = \boldsymbol{t_i} \wedge^{-1} \boldsymbol{t_i^T} = x_i P \wedge^{-1} P^T x_i^T \tag{2.8}$$

where t_i is the i_{th} column of T and $\wedge = diag\{\lambda_1, \lambda_2, \lambda_3 \dots, \lambda_k\}$ with λ_i being the eigenvalues of the covariance matrix. The SPE statistic measures the projection of the sample vector on the residual space, indicating the amount of variation not captured by the PCA model.

$$SPE_i = \|e_i\|^2 = \left\|x_i(I - PP^T)x_i^T\right\| \tag{2.9}$$

where e_i is the i_{th} row of E, and I is the identity matrix.

For both T^2 and SPE statistics, an excursion is detected if $T_i^2 \geq \tau^2$ or $SPE_i \geq \delta^2$, where $\tau^2$and δ^2 are the control limits given a $1-\alpha$ confidence level. T_i^2 follows a χ^2distribution with k degrees of freedom , and SPE was derived by Jackson[2.11] assuming that $\boldsymbol{x}$ follows a multivariate normal distribution. In practice, both control limits are set with confidence level $= 95\%$:

$$T_{lim}^2 = \frac{k(n-1)}{n-k} F(k, n-k, \alpha) \tag{2.10}$$

where $F(k, n-k, \alpha)$ corresponds to the probability point on the F-distribution with $(k, n-k)$ degrees of freedom and confidence level α, and n is the number of observations.
The limit for SPE at confidence level $\alpha = 95\%$ is :

$$SPE_\alpha = \Theta_1 \left[\frac{c_\alpha h_0 \sqrt{2\Theta_2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1}\right]^{\frac{1}{h_0}} \tag{2.11}$$

where $\Theta_1 = \sum_{j=k+1}^{p} \lambda_j^i$ and $h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2}$. Θ_1 is the second largest eigenvalue, and Θ_2 and Θ_3, respectively, are the 2nd and 3rd power of the second eigenvalue. Kernel density estimation has previously been applied to deal with data that are non-normal. Because the major correlations among variables are captured in the first few PCs, deviation from multivariate normality will strongly impact the frequency of type I and II errors when monitoring Hotelling's T^2. The impact on the SPE index and its limits will be less pronounced because the residual space, by definition, will mostly be made up of process noise and random fluctuations. We illustrate the PCA analysis results in figure 2.6 and 2.7.

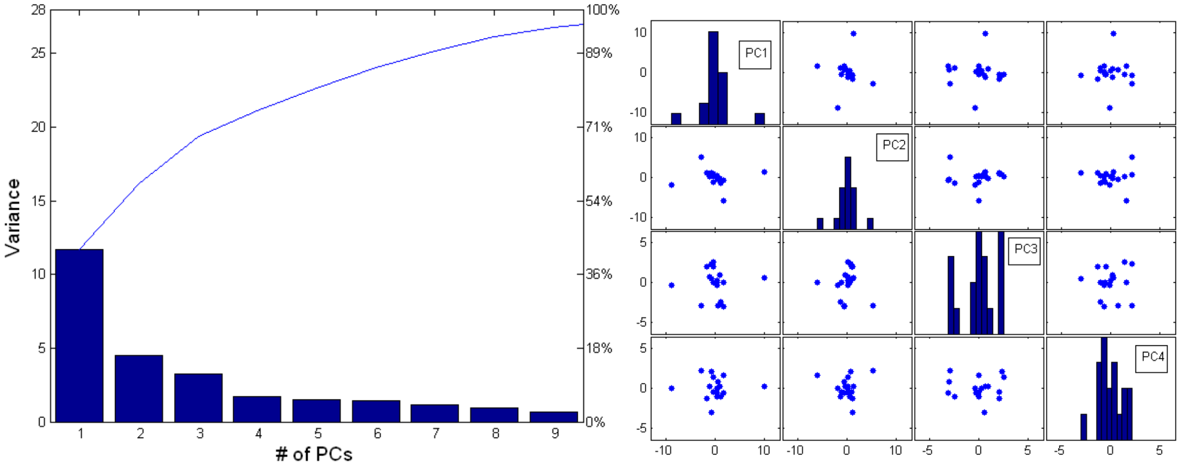


Figure 2.6 Scree plot for Number of PCs retained and Scatter of Scores

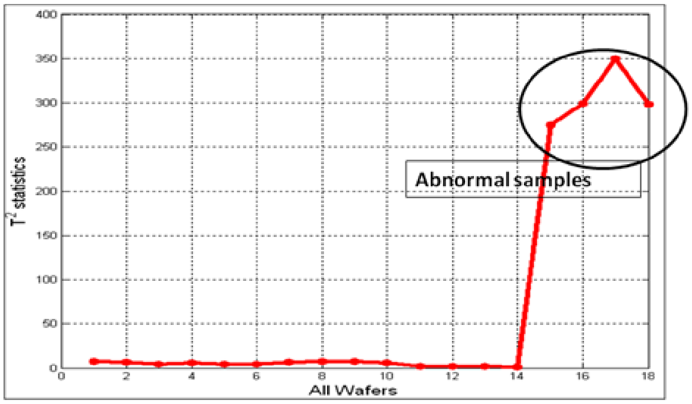


Figure 2.7 Illustration of outlier detection with Hotelling's T^2

Given that most manufacturing processes drift over time, monitoring statistic using a static model will lead to higher type I and II error rates. Therefore, recursive models have been proposed to update the inline model. Recursive PCA [2.16] was proposed to update the mean and covariance of PCA model with an adaptive weighting scheme on the new observations. To implement the RPCA algorithm it is necessary to first recursively calculate the correlation matrix. Given a new vector of unscaled measurements x_{k+1}^{0}, the updating equation for the correlation matrix is given by:

$$R_{k+1} = \mu\sum_{k+1}^{-1}(\sum_k R_k \sum_k + \Delta b_{k+1}\Delta b_{k+1}^T)\sum_{k+1}^{-1} + (1-\mu)x_{k+1}x_{k+1}^T \tag{2.12}$$

Where $\boldsymbol{x_{k+1}^T}$ is the scaled vector of measurements, b is a vector of means of data and $\sum$ is the diagonal matrix with element I being the standard deviation of the variable i. The forgetting factor μ is used to weight recent data more than older data. A smaller μ discounts old data faster. After the correlation matrix is updated, calculation of the loading matrices is performed in the same manner as ordinary PCA.

2.3.1.2.2. Independent Component Analysis

The goal of independent component analysis (ICA) [2.17] is to find a linear representation of non-Gaussian data so that the components are statistically independent, or as independent as possible. ICA assumes the independent components s_i are statistically independent, and must have non-Gaussian distributions. In the basic ICA, each measured variable can be represented as a linear combination of underlying independent components:

$$x(t) = \sum_{j=1}^{n} a_j s_j(t) + e(t) \tag{2.13a}$$

$$X = AS + E \tag{2.13b}$$

where the X is assumed to be a full rank matrix. Thus, the ICA model aims at finding a de-mixing matrix W such that:

$$s(t) = \sum_{j=1}^{n} w_j x_j(t) \tag{2.14a}$$

$$S = WX \tag{2.14b}$$

where $s(t)$ is the IC vector. The ICs are then used to estimate the latent source components $s_j(t)$. Process monitoring statistics with ICA can be extracted in a similar way as the PCA case. In the normal operating condition, we first obtain W and S from training data with the conventional ICA algorithm. Then the W dimension is reduced by selecting a few rows of W based upon the assumption that the i_{th} row of W, w_i^T with the largest sum of squares coefficient have the greatest effect on the variation of the corresponding element of the ICs. For a new data sample, the new ICs can be obtained by:

$$s_{new}(t) = W_k x_{new}(t) \tag{2.15}$$

W_k is the retained k rows of demixing matrix. Two monitoring statistics can then be extracted:

$$I^2(t) = s_{new}(t)^T s_{new}(t) \tag{2.16}$$

$$SPE(t) = e(k)^T e(k) = \left(x_{new}(t) - \hat{x}_{new}(t)\right)\ ^T\left(x_{new}(t) - \hat{x}_{new}(t)\right) \tag{2.17}$$

$\hat{x}_{new}(t) = A_k s_{new}(t)$ is the estimated data from computed ICs and W_k .

In ICA monitoring, the ICs do not necessarily conform to a multivariate Gaussian distribution, therefore, the confidence limits of the I^2 and SPE cannot be computed directly. Kernel density estimation can be used to obtain the control specs for these two monitoring statistics. Lee et al. (2003) [2.18] investigated the utilization of kernel density estimation to define the control limits of ICs that do not follow a Gaussian distribution. In order to monitor the batch processes which combine ICA and kernel estimation, Lee at al. extended their original method to multi-way PCA and multi-way ICA. Xia (2003) [2.19] developed a spectral ICA approach to transform the process measurements from time domain to the frequency domain and identify major oscillations. Shannon et al. (2003) [2.20] used ICA in monitoring a semiconductor manufacturing process.

2.3.1.2.3. Clustering

Clustering is another class of methods that is popular for fault detection in manufacturing control. It aims to group similar patterns or observations in the same groups so that within-group variation is minimized and between-group variation is maximized. The process used in our work is “unsupervised learning” since the nature of data is unknown. Clustering methods can be categorized into five groups: hierarchical clustering, pattern partitioning, density-based methods, grid-based methods, and model-based methods. We briefly discuss each group of methods in this section.

Hierarchical clustering [2.21] methods work by grouping data samples into a tree of clusters. There are generally two types of hierarchical clustering methods; agglomerative and divisive. Agglomerative methods start by placing each object in its own cluster, and then merge clusters into larger and larger clusters, until all objects are in a single cluster or until certain stopping criterion is met. Divisive methods do just the opposite. Hierarchical clustering method generally suffers from its inability to perform adjustment once a merge or split decision has been made.

Partitioning based clustering [2.22] starts with k random seed values, and then assigns observations into the k mutually exclusive clusters based on specified distance metric. K-means clustering [2.23] is the most popular portioning based method for its low computational cost and good classification ability for large datasets (sample size>1000) .The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (k) fixed a priori. The main idea is to define k centroids first; one for each cluster .These centroids should be placed far away from each other. The next step is to assign each point in the dataset to the nearest centroid. When no point is pending, the first step is completed and an early partitioning is done. At this point we need to re-calculate k new centroids of the clusters resulting from the previous step. After we have these k new centroids, a new assignment has to be done between the same data set points and the nearest new centroid. As a result of this iteration, the k centroids change their location step by step until no more changes are needed, and the centroids become fixed. The algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is:

$$Obj = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^j - c^j\right\|^2 \tag{2.18}$$

where $\left\|x_i^j - c^j\right\|$ is a chosen distance measure between a data point x_i^j and the cluster center c^j, as a measure of the distance of the *n* data points from their respective cluster centers. The result of k-means depends on the distance metric, so it is always better to normalize the variable to unit variance. There are also other distance metrics such as Manhattan distance or cosine distance. Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. There are other partitioning-based clustering methods with soft threshold such as fuzzy c-means.

The idea of density-based clustering methods such as DBSCAN [2.24] is to continue growing a cluster as long as the density or number of data points in the "neighborhood" exceeds some pre-defined threshold. Grid-based methods attempt to quantify the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The popular grid-based method is STING [2.25], which uses several levels of rectangular cells corresponding to different levels of spatial resolution and store statistical information regarding each variable in each cell. Model-based clustering methods generally assume a model for each of the clusters and attempt to fit the data with the assumed model. There are two major approaches for model-based methods: the statistical approach and neural network approach. An example of statistical approach is Gaussian mixture modeling, which assumes each cluster follows a Gaussian distribution, and then the Expectation Maximization (EM) algorithm is used to estimate the cluster parameters. The popular neural network approaches are artificial resonance network (ART) and self-organizing maps (SOM).

2.3.1.2.4. Mahalanobis Distance (Conventional and Robust version)

To detect if an observation is located far from the center of data distribution in terms of distance, Mahalanobis Distance (MD) is an appropriate metric when the data is generated from multivariate distribution [2.26]. MD outperforms Euclidean distance by taking the variance of data into account. Given n observations from a *p*-dimensional space, the MD for each observation is defined as:

$$MD_i = \left(\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{x_i}-\mathrm{\bar{x}_n})^{\mathrm{T}}\textstyle\sum_{\mathrm{n}}^{-1}(\mathrm{x_i}-\mathrm{\bar{x}_n})\right)^{1/2} = \left((\mathrm{X}-\mathrm{\bar{X}})^T\textstyle\sum^{-1}(\mathrm{X}-\mathrm{\bar{X}})\right)^{1/2} \tag{2.19}$$

where $\bar{X}$ is the center and $\sum^{-1}$ is inverse of the covariance matrix of the targeting distribution. Observations with large MD are indicated as outliers. However, masking and swamping effects of outliers can affect the empirical estimates of the mean vector and the covariance matrix. The masking effect decreases the MD of an outlier while the swamping effect might increase the MD of normal observation. Therefore, it is necessary to use robust estimates of mean and covariance in order to compute MD in the presence of outliers.

In this thesis, MCD estimate [2.27] is used to compute the covariance matrix and the mean. The MCD looks for h observations in the dataset whose empirical covariance matrix has the lowest possible determinant. The MCD estimate of mean $\bar{\mathrm{x}}_\mathrm{n}$ is the average of these h points, where the MCD estimate of covariance $\sum$ is their covariance matrix multiplied with a reweighted factor. In this thesis, we have assigned the weights w_i for each observation $\mathrm{x_i}$ in the way by giving $w_i = 1$ if $(\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{n})^\mathrm{T} \sum_\mathrm{n}^{-1} (\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{n}) \leq \chi^2_{\mathrm{p},0.975}$ and $w_i = 0$ otherwise. The resulting center and covariance matrix are then defined as:

$$\bar{\mathrm{x}}_\mathrm{n} = \left(\sum_{i=1}^{n} w_i\, x_i\right) \Big/ \left(\sum_{i=1}^{n} w_i\right) \tag{2.20}$$

$$\sum\nolimits_\mathrm{n} = \left(\sum_{\mathrm{i}=1}^{\mathrm{n}} w_i (\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{n})\, (\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{n})^\mathrm{T}\right) \Big/ \left(\sum_{i=1}^{n} w_i - 1\right) \tag{2.21}$$

It is clear that MCD results are determined by h, a value of $0.75n$ is recommended for h, whereas $0.5n$ is preferred when a large number of outliers is expected. In this thesis, we have used $0.75n$ consistently with the same weighting schemes for all MD computation.

For a set of p-dimensional data points from Normal distribution, the square of the MD follows the χ_p^2 distribution with p degrees of freedom. However, MD also suffers from the curse of dimensionality, i.e., as distances between data points tend to become close as the dimensional increases. Outlier detection with MD can also be defined by isolating the sample points with MD greater than $\sqrt{\chi_p^2(\alpha)}$, where α is usually set to 0.025.

2.3.1.2.5. Discriminant Analysis

In linear discriminant analysis (LDA), each class is assumed to have a common covariance matrix [2.28]. One can use the maximum likelihood rules to classify an observation x into k , it is equivalent to maximizing the determinant scores $d_k(x_i)$:

$$d_k(x_i) = x^T \sum\nolimits_k^{-1} \mu_k - \frac{1}{2} \mu_k^T \sum\nolimits_k^{-1} \mu_k + \ln p_k \tag{2.22}$$

In practice, we have to estimate the μ_k, $\sum_k$ and p_k empirically. μ_k and $\sum_k$ can be estimated using all samples within the class. The class probabilities are estimated by the relative frequencies of the samples in each class.

$$\mu_k = \sum_{g_i = k} {}^{x_i}\!/_{n_k} \tag{2.23}$$

$$\sum_k = \frac{1}{n-k}\sum_{k=1}^{K}\sum_{g_i=k}(x_i-\mu_k)(x_i-\mu_k)^T \tag{2.24}$$

$$\pi_k = \frac{n_k}{n} \tag{2.25}$$

where $N_k$is the number of class k samples and N is the total number of samples.

In the case of classes not sharing a common covariance matrix, quadratic discriminant analysis (QDA) [2.29] can be applied.

$$d_k(x_i) = \frac{1}{2}\ln|\sum_k| - \frac{1}{2}(x-\mu_k)^T\sum_k^{-1}(x-\mu_k) + \ln p_k \tag{2.26}$$

2.3.2. Prediction Methods

In this section, we will review several statistical prediction methods including principal component regression (PCR), partial least square (PLSR), neural network (NN), LASSOLASSO, classification and regression tree (CART), and k-nearest neighbor (KNN) regression.

2.3.2.1. PCR and PLS

Principal component regression (PCR) [2.30] and Partial least squares regression (PLSR) [2.31] are two common methods used to build regression in high dimensions. PCR is an extension of PCA for regression, where the predictor variables are reduced to PCs. PCR addresses the singularity and dimensionality problems for predictor variables, but does not leverage any dependencies among the response variables, nor does it optimize the scores with respect to response prediction.

PLSR as another projection technique goes one step further to utilize information from both predictor and response variables. Projection to latent structures is a regression method based upon projecting the information from high dimensional space (X, Y) down onto a lower dimensional space. PLSR decomposes the variables in following forms:

$$X = \boldsymbol{t_1p_1^T} + \boldsymbol{t_2p_2^T} + \cdots + \boldsymbol{t_kp_k^T} + E \tag{2.27}$$

$$Y = \boldsymbol{u_1q_1^T} + \boldsymbol{u_2q_2^T} + \cdots + \boldsymbol{u_kq_k^T} + F \tag{2.28}$$

where t_i and u_i are the latent score vectors for X and Y, and $\boldsymbol{p_i}$ and $\boldsymbol{q_i}$ are the corresponding loading vectors. $\boldsymbol{t_1}$ and $\boldsymbol{u_1}$ are evaluated as below:

$$\boldsymbol{t_1} = X\boldsymbol{w_1} \tag{2.29}$$

$$\boldsymbol{u_1} = Y\boldsymbol{q_1} \tag{2.30}$$

Equations 2.25 and 2.26 are referred to as the outer relations for the X and Y blocks. The vectors w_i and q_i are the factor weights. PLSR finds the factor weights in such a way that maximize the linear correlation between $\boldsymbol{t_1}$ and $\boldsymbol{u_1}$. A linear regression is then performed between the first pairs of factors t_1 and u_1 as inner model:

$$\boldsymbol{u_1} = \boldsymbol{b_1 t_1} + \boldsymbol{r_1} \tag{2.31}$$

Equation 2.22 links the X and Y blocks together through latent variables $\boldsymbol{t_1}$ and $\boldsymbol{u_1}$. The second pair of latent variables is calculated by decomposing the residuals:

$$E_1 = X - \boldsymbol{t_1 p_1^T} \tag{2.32}$$
$$F_1 = Y - \boldsymbol{u_1 q_1^T} \tag{2.33}$$

The coefficients p_1 and q_1 are found with least square estimation:

$$\boldsymbol{p_1^T} = \left(\boldsymbol{t_1^T t_1}\right)^{-1} \boldsymbol{t_1^T} X \qquad \boldsymbol{q_1^T} = \left(\boldsymbol{u_1^T u}\right)^{-1} \boldsymbol{u_1^T} Y \tag{2.34}$$

Similar to PCA, we can compute up to *p* pairs of latent factors, but only the first *k* pairs contain the relevant information. The commonly used PLS method is the nonlinear iterative partial least squares (NIPALS) algorithm [2.32], which construct the weights vectors W such that :

$$[cov(t_i, u_i)]^2 = \max_{|\mathrm{w_i}|=1} [cov(Xw_i, Y)]^2 \tag{2.35}$$

The difference between PLS and PCA is that the former creates orthogonal weight vectors by maximizing the covariance between elements in **X** and **Y**. Thus, PLS considers the variance of the input and output.

A polynomial expansion of the inner relationship, which is linear in the linear PLS is another way to model nonlinear relationships between X and Y. Quadratic PLS [2.33] is where the polynomial terms are of the second order. Instead of the linear relationship between the X and Y score matrices U and T, the polynomial expression is used:

$$\boldsymbol{u_i} = \boldsymbol{b_{i1} t_i} + \boldsymbol{b_{i2} t_i}^{\boldsymbol{2}} + \boldsymbol{r_i} \tag{2.36}$$

The coefficients are also determined by least square estimation (LSE) in an iterative manner.

2.3.2.2. Neural Network

A Neural Network [2.34] is a data mapping tool that links input information to output. It has been applied in the field of classification and regression. NNs consist of neurons distributed across layers with defined structure. Each of the links between neurons is characterized by a weight value. A *neuron* is a processing unit that takes a number of inputs and gives a distinct output. Apart from the number of its inputs it is characterized by a function *f* known as *transfer function*. The most commonly used transfer functions are listed in table 2.1: *pure linear, sigmoid* and *tansigmoid* functions.

There are three types of layers: the *input layer*, the *hidden layers*, and the *output layer*. Each network has exactly one input layer and one output layer. The number of hidden layers can vary from 0 to any number. The input layer is the only layer that does not contain transfer functions. Once the architecture and the transfer function of each neuron have been defined for a network the values of its weights should be defined. The training procedure 'fits' the network to a set of samples (*training set*). A standard NN structure is presented in figure 2.3.

Pure linear	Sigmoid	Tansigmoid
$f(x) = x\ ;f(x)\epsilon(-\infty, +\infty)$	$f(x) = \frac{1}{1+e^{-x}}\ ;f(x)\epsilon[0,1]$	$f(x)\frac{2}{1+e^{-2n}} - 1\ ;f(x)\epsilon[-1,1]$

Table 2.1. The activation functions for Neural Network.

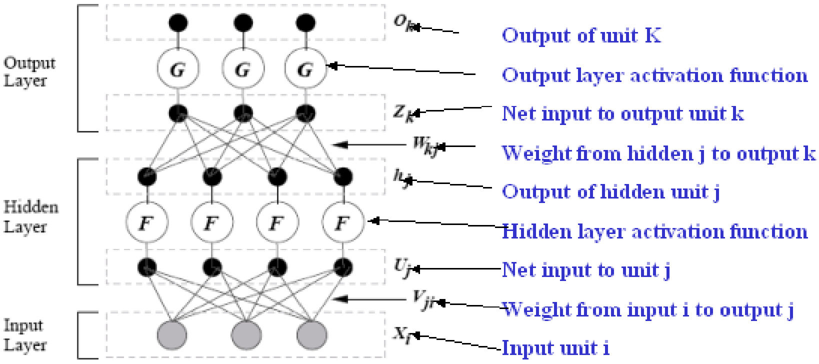

Output Layer
Hidden Layer
Input Layer
G
F
O_k
Z_k
W_{kj}
h_j
U_j
V_{ji}
X_i
Output of unit K
Output layer activation function
Net input to output unit k
Weight from hidden j to output k
Output of hidden unit j
Hidden layer activation function
Net input to unit j
Weight from input i to output j
Input unit i

Figure 2.8 Neural Network structure

We have chosen the back propagation Neural Network for the modeling method in this thesis due to its fast computation speed. Supervised Feed-forward BPNN (Back propagation Neural Network) [2.35] is more like a black box--even less transparent than PCA and PLS. The outputs of a neural network are weighted non-linear transformations of the inputs. There are no preset rules for training a neural network—that is, deciding the number of hidden nodes and number of layers, the training criteria etc. Each node in an FNN performs a non-linear transformation with a sigmoid function. At every node, weighted inputs are transformed to give an output whose weighted transformation is carried out at the nodes in the subsequent layer. Three layers (input, hidden, output) are sufficient for most tasks. For number of nodes in the hidden layer, too few will lower its fitting capability, and too many will over-specialize the net to the training data, limiting its ability to predict other data points.

In many Chemometrics cases, PCA is used in conjunction with NN for prediction. This takes advantage of dimensionality reduction via PCA and the inherent nonlinearity of NN model. Therefore, large number of predictor variables can be reduced to several PCs and reduce the complexity and computational cost of fitting and using a NN model.

2.3.3. LASSO

LASSO [2.36] is a shrinkage and selection method for linear regression. If one is looking for variable selection and stable regression coefficients, LASSO is recommended. It minimizes the sum of squared errors with a bound on the sum of the absolute value of the coefficients. The LASSO estimate can be defined as:

$$\beta = \mathrm{argmin}_{\beta}\left\{\sum_{i=1}^{N}\left[y_i - \beta_0 - \sum_{j=1}^{p}(x_{ij})\,\beta_j\right]^2\right\} \tag{2.37}$$

$$\text{subject to } \sum_{i=1}^{P}\left|\beta_j\right| \le t \tag{2.38}$$

where β_j are the coefficients of predictor variables and β_0 is the intercept term. If a larger threshold *t* is chosen, the constraint has no effect and the solution is the same as that of the OLS. When *t* is smaller, the solutions are shrunken versions of the least squares estimates. Some of the coefficients are set to zero. Choosing *t* is similar as choosing the number of predictors to use in a regression model, and cross-validation is used to choose the optimal *t*. Thus, LASSO regression penalizes the 1-norm.

In this thesis we use the least angle regression (LAR) [2.36] to solve the LASSO problem. The LAR procedure follows a similar strategy as forward stepwise regression. At the first step, it identifies the variable most correlated with the response. Rather than fit this variable completely, LAR moves the coefficient of this variable continuously, toward its least squares value (causing its correlation with the evolving residual to decrease in absolute value). As soon as another variable catches up in terms of correlation with the residual, the process is paused. The second variable then joins the active set, and their coefficients are moved together in way that keeps their correlations tied and decreasing. This process is continued until all the variables are in the model, and ends at the full least squares fit. In figure 2.9, we illustrate how to choose the λ from the coefficient stability plot and MSE of CV sets for LASSO regression.

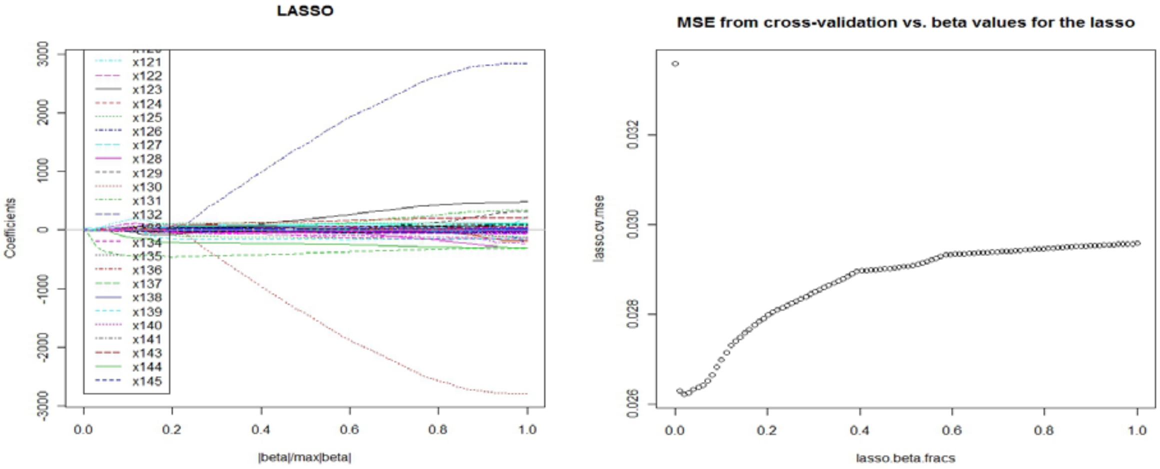


Figure 2.9 Coefficient stability and MSE of CV

Based on the plots in figure 2.9, we choose $\lambda = \frac{|\beta|}{max|\beta|} = 0.1$ for the LASSO.

2.3.4. CART

The Classification and Regression Tree (CART) [2.37] is a distribution-free method without any assumption about the distributional properties of the data. Predictor variables are used to partition the data points into regions with similar response. This partitioning allows one to approximate more general response surface than standard regression methods. Typically a binary partitioning is used. At each partitioning step, trying to maximize the average purity of two child nodes during a partitioning process, the CART algorithm looks for the best predictor variable and the corresponding decision criterion by using a brute force method. For the selected predictor variable, a threshold is determined such that data points with the predictor below the threshold are placed in one subgroup, while the other data points form a second subgroup. The method continues by partitioning each of these subgroups by the same procedure.

If the response variable is continuous, then a regression tree is chosen to minimize the sum of square errors of prediction. The initial training stops when the tree cannot grow any more resulting in decreasing R^2 of the testing data. In the second training, child nodes are pruned away, which increases the error of the training data. Finally, the mean of the training samples, which are grouped under a specific terminal node, is assigned to the corresponding terminal node. The prediction of a new sample is performed by assigning the sample through the tree and assigning the value of the corresponding node as the predicted response value.

If the response variable is categorical, then a classification tree is used to minimize the residual likelihood-ration chi-square, which is the change in total entropy of all samples. CART is favorable in categorical data application given its capability of handling nonlinearity and distribution free nature. We also adopt it for variable selection since CART proves a sparse set of predictor variable as solution.

2.3.5. KNN

The *k* nearest-neighbor (KNN) [2.38] method is a nonparametric algorithm to predict the class of a test sample, and has been applied extensively in the field of fault classification for semiconductor manufacturing. In KNN, a data point is classified by majority votes of its k neighbors. These k neighbors are the data points with correct class labels. This makes KNN to be a supervised learning algorithm, where one cannot try to define classes based on results of grouping as in unsupervised clustering k-means.

In KNN, each training point consists of independent variables and dependent variables. The independent and dependent variables can be either continuous or categorical. For continuous dependent variables, the task is regression; otherwise it is a classification. Thus, KNN can handle both regression and classification tasks. The training phase is trivial: simply store every training sample with its corresponding class label. To make a prediction for a test sample, we first compute its distance to every training sample. The most used metric is the Euclidean distance as shown in Equation 2.35.

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2} \tag{2.39}$$

where p is the dimension of samples, x_i and y_i are samples.

We then keep the k nearest training examples, where $k \geq 1$ is a fixed integer. The nearest neighbors are determined by the distance between points. Other distance metrics can also be explored depending on the nature of dataset.

Similar as k-means, the choice of k is essential in building the KNN model. For any given problem, a small value of k will lead to a large variance in modeling results. Alternatively, setting k to a large value may lead to a large model bias. Thus, k should be set to a value large enough to minimize the probability of misclassification and small enough for the training set result, so that the k nearest points are close enough to the classified point. Cross-validation in often used to find the optimal k. Majority voting can be used to determine the class label or numerical result of the response variable of the test samples based on the most common class label among its k neighbors. For regression problems, KNN predictions are based on averaging the outcomes of the k nearest neighbors; for classification problems, a majority of voting is used. We show the model results for both regression and classification cases as follow:

Regression case:

$$\hat{y}_{predicted} = \frac{1}{K} \sum_{k=1}^{K} y_k \tag{2.40}$$

Classification case:

$$\hat{y}_{predicted} = \text{most common class in set } \{y_1, \dots, y_k\} \tag{2.41}$$

For binary classification, the most common choice of k is a small odd integer, such as $k = 3$. The obvious disadvantage of the KNN algorithm is that relatively high computation time is needed. Similar to all the methods using distance metric, KNN also suffers from the curse of dimensionality.

Reference

[2.1] CMOS Process Flow in Wafer Fab, Semiconductor Manufacturing Technology, DRAFT, Austin Community College, January 2, 1997.

[2.2] Gwozdz, Peter. “Semiconductor Processing Technology” SEMI, 1997.

[2.3] Fullman Company website. “Fullman Company - The Semiconductor Manufacturing Process,” http://www.fullman.com/semiconductors/index.html, 1997.

[2.4] Barrett, Craig R. “From Sand to Silicon: Manufacturing an Integrated Circuit,” Scientific American Special Issue: The Solid State Century, January 22, 1998.

[2.6] Solar Cells: Operating Principles, Technology and System Applications by Martin A.Green

[2.7] J. Zhao, A. Wang and M. A. Green, “24% efficiency silicon solar cell” *IEEE Trans. Electron Devices*, 41, (1994), 1592.

[2.8] Kapur, V. K., Bansal, A., Le, P., “NON-VACUUM PROCESSING OF CIGS SOLAR CELLS ON FLEXIBLE POLYMERIC SUBSTRATES.” International Solar Electric Technology, Inc. (ISET), Inglewood, California, 2003

[2.9] Wing-Kam Fung. Outlier diagnostics in several multivariate samples. *The Statistician*, 48(1):73{84, 1999.

[2.10] Cao, R.; Cuevas, A.; Manteiga, W. G. (1994). "A comparative study of several smoothing methods in density estimation". *Computational Statistics and Data Analysis* 17: 153–176

[2.11] J. E. Jackson, *A User’s Guide to Principal Components*, John Wiley and Sons, New York, 1991.

[2.12] E.B. Martin and A.J. Morris. Non-parametric confidence bounds for process performance monitoring charts. J. Proc. Cont., 6(6):349–358,1996.

[2.13] Croux, C. and Haesbroeck, G. (2000). Principal components analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika* 87 603–618.

[2.14] Hubert, M. and Debruyne, M. (2009). Minimum Covariance Determinant. *Wiley Interdisciplinary Reviews: Computational Statistics* in press.

[2.15] C. A. Lowry and D. C. Montgomery, "A review of multivariate control charts," IIE Transactions, vol. 27, pp. 800-810, 1995.

[2.16] Li W, Yue H, Valle S, Qin J. Recursive PCA for adaptive process monitoring. *J. Process Control* 2000; 10:471–486.

[2.17] A. Hyvärinen, J. Karhunen, E. Oja (2001): *Independent Component Analysis*, New York: Wiley

[2.18] Lee JM, Qin SJ, Lee IB (2006) Fault detection and diagnosis based on modified independent component analysis. AIChEJ 52:3501–3514

[2.19] Xia C (2003) Control loop measurement based isolation of faults and disturbances in process plants. PhD thesis. University of Glasgow, UK

[2.20] Shannon, T.T., D. Abercrombie, J. McNames, "Process monitoring via independent components," *Proc. of the IEEE International Conference on Systems, Man and Cybernetics* Washington DC, 2003, pp. 3496-3500.

[2.21] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). "14.3.12 Hierarchical clustering" (PDF). *The Elements of Statistical Learning* (2nd ed.). New York: Springer. pp. 520–528.

[2.22] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297

[2.23] J. Hartigan and M. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100-108, 1979.

[2.24] Martin.Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD’96, Portland, OR, pp. 226~231, 1996

[2.25] Wang W., Yang J., Muntz R., STING: A Statistical Information Grid Approach to Spatial Data Mining, Proc. of Int. Conf. Very Large Data Bases (VLDB’97), pp. 186-195, 1997.

[2.26] Gnanadesikan, R., and J.R. Kettenring (1972). Robust estimates, residuals, and outlier detection with multiresponse data. Biometrics 28:81-124.

[2.27] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection* (New York: Wiley,1987).

[2.28] Martinez & Kak (2004)*PCA versus LDA* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2): 228–233

[2.29] Friedman, J. H. (1989). "Regularized Discriminant Analysis". *Journal of the American Statistical Association* (American Statistical Association) 84 (405): 165–175.

[2.30] I. T. Jolliffe. Principal Component Analysis. Springer Series in Statistics, Aberdeen, UK, 2nd edition, 2004.

[2.31] S. de Jong. SIMPLS: An alternative approach to partial least squares regression. Chemom. Intell. Lab. Syst., 18(3):251-263, Mar. 1993.

[2.32] P. Geladi and B. R. Kowalski. Partial least-squares regression: A tutorial. Anal. Chim. Acta, 185:1{17, 1986.

[2.33] R.l D. Tobias. An introduction to partial least squares regression. Technical report, SAS Institute Inc., 1997.

[2.34] Bishop, C.M (1995). *Neural Network for pattern recognition*. New York, NY,USA: Oxford Press

[2.35] S.F. Lee and C.J. Spanos. Prediction of wafer state after plasma processing using real-time tool data. IEEE T. Semiconduct. M., 8(3):252{261, August 1995.

[2.36] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. The
Annals of Statistics, 32(2):407-499, Apr. 2004.

[2.37] Breiman, Leo; Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA

[2.38] Dasarathy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. McGraw-Hill Computer Science Series. IEE Computer Society Press, Las Alamitos, California, 1991.

Chapter 3

Virtual Metrology for Plasma Etching

3.1 Introduction and Previous Work

Traditional lot-to-lot process control with time-delayed external metrology is being replaced with wafer-to-wafer control due to the need to provide quick feedback onto process tools. Wafer-to-wafer control requires timely, no-lag metrology, and this is unavailable for several critical manufacturing steps. On the other hand, modern processing tools generate large amounts of real-time data streams and, historically, these data sets have been used for fault detection and classification purposes. In addition, assuming that the real-time tool data reflect actual processing conditions, they can be used to establish models that could predict wafer properties. This type of modeling has been referred to as Virtual Metrology (VM) [3.1]. An accurate VM model that describes the relationship between tool state variables and metrology results can contribute towards enhancing or even replacing direct metrology operations, thereby increasing manufacturing efficiency.

VM deployment needs a model that links equipment and process data with wafer outcomes. This model has to be built using available sensor traces, and this involves several steps, such as data scaling, outlier removal, de-trending, dimensionality reduction, handling collinearities, and, finally, selecting the most useful subset of the available parameters. After that, a model is fitted to the training data and properly validated. This sequence is depicted in figure3.1. Once the appropriate model is created, is can then be used to “predict” wafer outcomes in lieu of performing actual, time-consuming measurements. In practice, VM is deployed in tandem with actual metrology, the latter being used at reduced frequency and needed for periodic model validation and update.

The concept of VM has attracted great interest in recent years. Chen et al. (2005)[3.1] constructed a VM system for a 90nm chemical vapor deposition (CVD) process with experimental data, and achieved R^2 over 0.98. Besnard and Toprac (2006) [3.2] built a VM system based on context information, raw FDC data, and metrology measurements. Their system was designed to produce two types of outputs: model quality value (MQV) measured the confidence of the whole prediction and predicted quality value (PQV), a local estimator, measured the confidence of each prediction. Chen et al. tried to reduce the number of variables by removing non-normally distributed and highly correlated variables based on the statistical features. Then, a set of relevant variables were selected based on shared information between inputs and target metrology measurements. Using a regression tree-based model, their VM system was applied to predict the thickness in the SiGe epitaxial growth process. Yung-Cheng and Cheng (2005) [3.3] built a VM system based on 4-layer feed forward neural networks to predict three different metrology measurements (thickness mean, range, and uniformity) at once. The neural network consists of 2345, 11, 20 and 3 neurons in the input, first hidden, second hidden, and output layer. Under TSMC’s advanced 300 mm FAB environment, wafers collected over a one-month period were used for training the neural networks while wafers collected over approximately two and half months were used for testing. Their VM system achieved 1.7% of MAE and 0.39% of maximum projection error (MAPE). Chen (2006)[3.4] addressed both high dimensionality and the lack of available wafers in their VM system with simulated wafer data and concluded that abnormal wafers could be detected very well by depicting the predicted values of the generated wafers. Lin et al. (2006) [3.5] built a VM system based on radial basis function networks (RBFN). They also adopted the PCA in order to reduce the input dimensionality. Among a total of 20 process steps, five important steps were selected by process engineers. Then, 28 sensor parameters from each of the five steps were collected. Therefore, a total of 140 parameters from production equipment were used as model input, and the thickness means, range and uniformity were used as outputs.

In this thesis, our studies of VM focus on developing a methodology to create a robust VM model and discuss its most challenging aspects: selecting the important predictor variables from existing high-dimensional sensor parameters and accommodating process dynamics. The following sections present the methodology that is proposed for creating the virtual metrology model and discuss the VM application results for the plasma etching operation with different approaches under the same framework.

The organization of this chapter is as follows. Section 3.1 describes the hierarchical flow for creating a predictive metrology model for a plasma etching operation as shown in figure 3.1.It includes detailed description of data extraction, missing value imputation, visualization, detrending, and variable selection for plasma etching operations. Section 3.2 presents several variable selection methods and their pros and cons. In section 3.3, we present the study of building predictive metrology modeling for a plasma etching operation with the proposed model building flow. The plasma etching operation is studied since it is considered as a bottleneck process in IC manufacturing. Linear regression methods including PCR, PLS, Lasso, and nonlinear methods including neural networks were compared with a combination of different variable selection schemes. In section 3.4, we conclude our results of VM models and propose a novel approach to building a virtual inference model with a classification approach. The concept is based on creating a classification model to infer whether an incoming wafer is needed for metrology measurement or not. This takes advantage of classification models that are generally easily to train and validate given the high-dimensional nature of our data set.

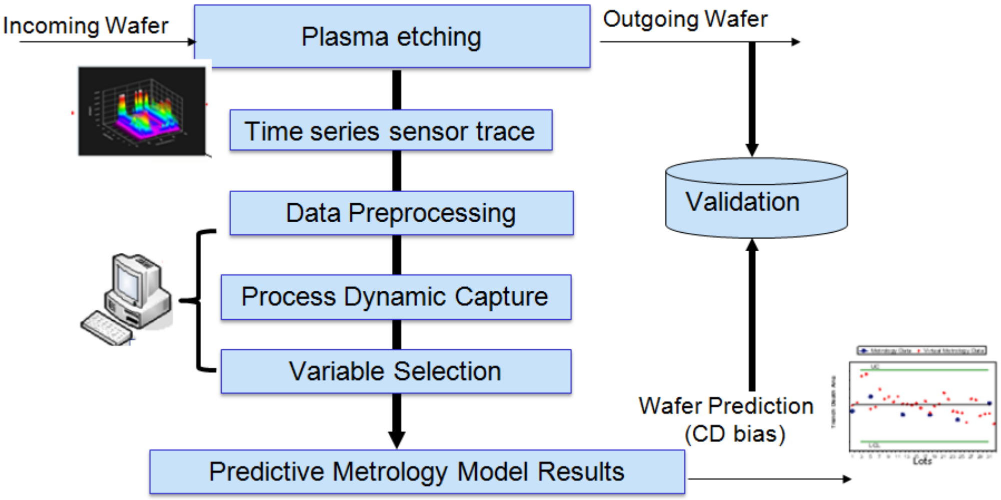


Figure 3.1 VM model building flow

3.2 Data Preprocessing and Visualization

Before attempting to build a VM model, and depending on the specific application, tool and process data needs to be preprocessed carefully to ensure that the built model will be robust. More specifically, and as it will be described next, long time-series need to be reduced to simple statistical descriptors, and overall, data might need to be de-tended and have occasional outlier identified and removed. In this section, we describe ways to preprocess tool sensor data and visualize it to gain insights from its high dimensional nature. The application for study is the plasma etching operation. The data preprocessing discussed here is most suited to plasma etching operation consisting of multiple recipe steps with non-linear behavior.

3.2.1. Plasma Etching Data

We will illustrate data pre-processing by focusing on a process that is eminently suitable for VM deployment, namely plasma etching. Plasma etching is a complex operation with numerous chemical reactions enhanced by the application of an RF field. Etching is usually carried out with preset recipes that optimize process parameters, including etch rate, selectivity, residue, micro-loading effects, and profile control. The plasma etch operation we studied in this case is ion energy-driven etching, where ion bombardment is combined with chemical etching. Ion bombardment can increase the chemical reaction rate by damaging the surface. The plasma etching operation is typically used in IC manufacturing to etch three types of material: polysilicon (poly-Si), dielectrics, and metal. Poly-Si etching is usually used to define the transistor gate profile, dielectrics etching is used for ILD oxide removal, and metal etching is used for interconnect patterning. Among these, poly-Si etching operations require the finest control since they directly impact transistor performance. That is why in this work we focus on a poly-Si etching, even though the methodology can be also be applied to other etching tools. The common plasma etchers can be classified into two areas based on their plasma sources: capacitive planar etchers and inductive or transformer coupled plasma etchers, the later being high density plasma etchers. Some details of plasma etcher and process can be found in reference [3.6]. We show a simplified schematic view of a plasma etcher in figure 3.2.

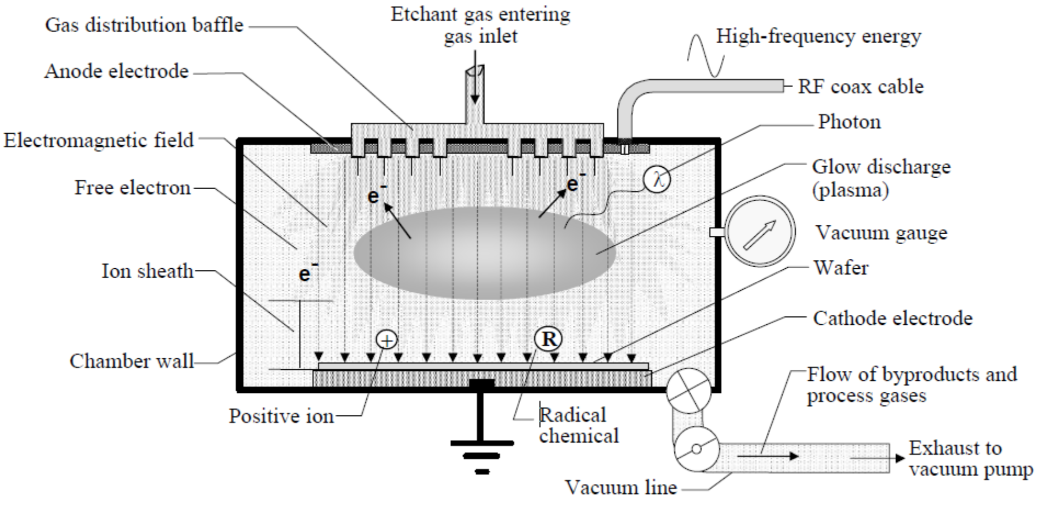


Figure 3.2.A simplified schematic view of a plasma etcher (reproduced from [3.6]).

The resulting etching profiles with the same recipe can be quite different as the equipment conditions change with time. Changes of equipment condition can be due to aging and maintenance events. Equipment aging results from impurities from process gases and contaminants left in the reaction chamber, and natural wear of key mechanical components can be adjusted back to the normal state with a scheduled preventive maintenance (PM) event. The key objective for a manufacturing engineer is to keep delivering consistently etched profiles, especially in terms of maintaining precision in the so-called Critical Dimensions (CD) of the pattern of interest. Also, since the role of plasma etching is to transfer an existing pattern from photoresist (PR) into a pattern in a permanent layer, the quantity of interest is frequently the difference between the CDs of the two patterns, known as etch bias (EB). Unfortunately, EB is likely to drift, so there are many sensors installed to track equipment conditions and chamber states over time to ensure EB changes can be detected through identification of changes in equipment condition or chamber states. That is the concept of the VM model, according to which one can infer or predict the process results on the wafer with information provided by sensors monitoring the machine and process state, as defined next.

In general, sensors fall into three categories: machine state sensors such as the ones monitoring the RF power source, gas flow, and pressure; process state sensors for plasma optical emissions, density and delivered power; and wafer state sensors for etch rate and film thickness.

Machine state sensors are usually coupled with a feedback controller to adjust recipe-setting variables. Process state sensors are in-situ, real time, and non-intrusive, such as the optical emission spectroscopy (OES) sensors shown in figure 3.3. OES sensors are used widely in end-point detection for etching operations. OES signals are generated by detecting emitted light when gas-phase chemical species collide with energetic electrons in the plasma [3.7].

Wafer state sensors (also known as direct metrology) are generally used for quality control and run-to-run control. This type of sensing is often deployed by means of large metrology tools that are separate from the processing tools. In general, wafer state sensors cannot be used in-situ, introducing considerable lag in data availability. Due to cost and throughput considerations, it is also common that only a small subset of the produced wafers is tested in this fashion. As mentioned before, one of the key objectives of VM is to reduce or eliminate reliance to direct metrology.

Table 3.1shows the existing sensors available on plasma etching equipment [3.8].

Monitoring Objectives	Sensors
Machine state	
Chamber temperature	Thermocouple
Pressure	Ionization gauge
Gas flow	Mass flow controller
RF power	Capacitance sensor, RF sensor
Bias voltage	Voltage sensor
Process state	
Chemical concentration	OES
Endpoint detection	OES, residual gas analyzer (RGA)
Ion density	Langmuir probe

RF plasma parameters	Voltage, current and phase sensors
Wafer state	
Etch rate	Ellipsometer
CD	Scanning electron microscopy (SEM) or scatterometer

Table 3.1. Sensors in a plasma etching tool [3.8]

As we can see in table 3.1, there can be multiple sensors monitoring similar mechanisms; this introduces collinearities into the collected sensor readings. Thus, many of these sensors are highly correlated to each other and carry significant levels of noise. For the purpose of VM modeling, one has to screen many of these sensors so as to not be included in the predictive VM model.

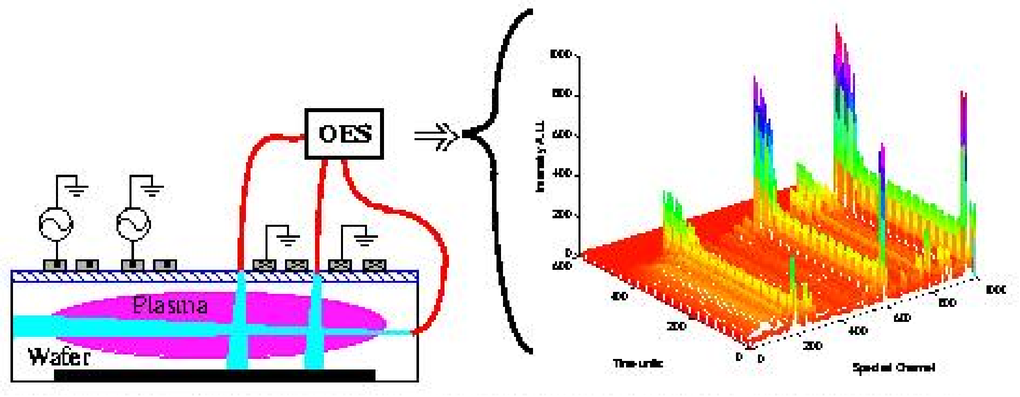


Figure 3.3. OES sensors in a plasma chamber [3.9]

Machine state data is usually collected per wafer in time trace format. Each wafer can have up to 400 recording points given the sampling frequency of the sensors. Most equipment can be equipped with hundreds of sensors, and multiple sensors are used to monitor similar equipment conditions. Therefore, the data we have is highly dimensional and collinear. A graphical representation of collinear sensor data is shown in figure 3.4. Each wafer is also given a corresponding metrology result, which is the response variable in modeling. For plasma etching operations, CD bias, or etch bias, is the response variable. The CD/etch bias for each wafer is computed as the average of all the measured sites on the same wafer. Data preprocessing requires extraction of a statistical summary from a time trace, removing trends in data, and eliminating variables based on engineering knowledge and statistical selection criteria. The high dimensional nature of sensor data makes it difficult to visualize its overall structure. Dimensionality reduction prior to visualizing the overall data structure assumes that the reduction techniques capture the data structure well, while in many cases, techniques such as PCA underperform in the case of nonlinearity in the data. In this work, we have utilized a spectral heat-map plot to visualize the data in a high number of dimensions.

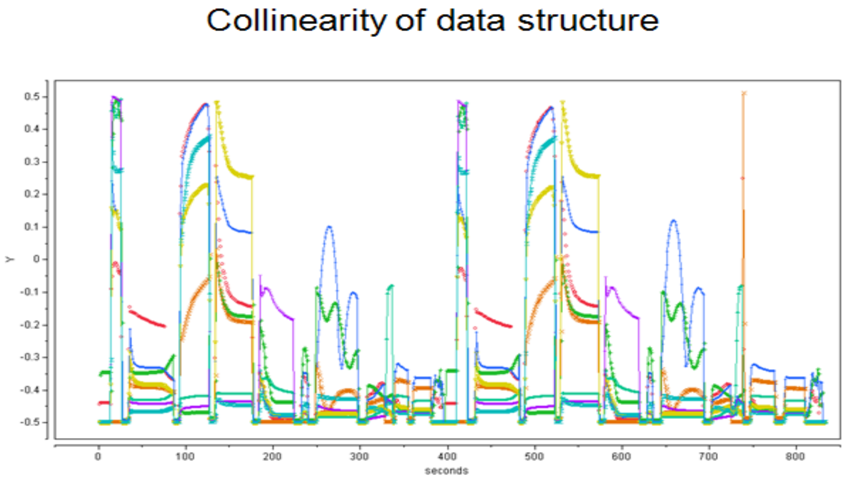


Figure 3.4.A collinear sensor parameter varying in similar patterns across different recipe steps for the same wafer

For a plasma etching operation, each recipe contains approximately 25 steps, and recipe parameters are adjusted at each step to meet process yields. For each recipe step, there will be transient time and stable time for each sensor reading, and computing statistical summaries with entire time window would lead to noisy results. Figure 3.5 shows a time trace plot for four sensor parameters across the entire process internal to a single wafer. On top of each plot, the corresponding recipe step numbers are shown. The transient portion can be clearly seen for each step.

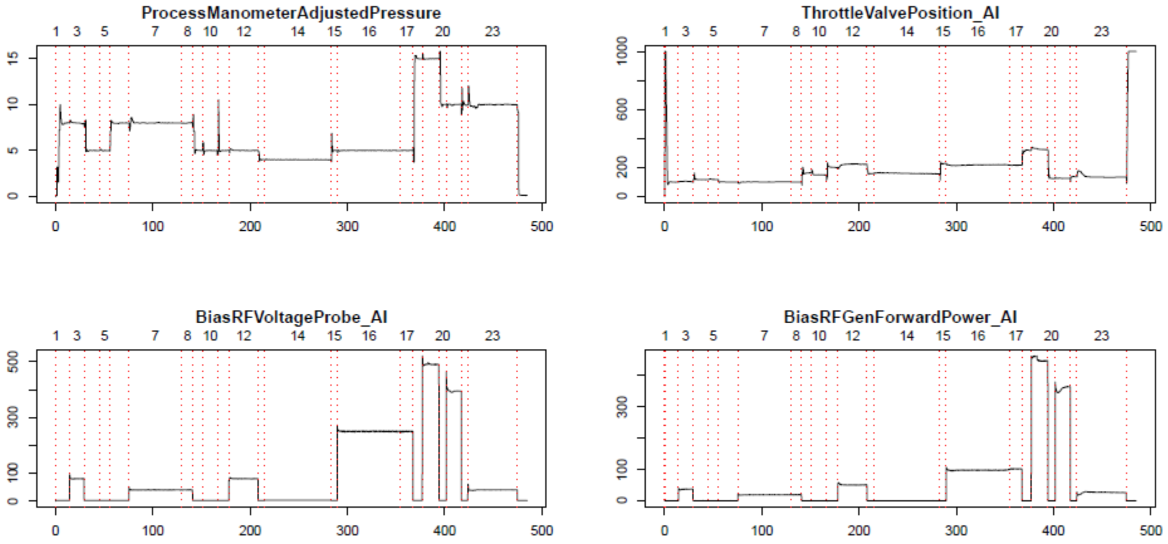


Figure 3.5.Time trace plots of sensor parameters for a single wafer. The dotted lines designate transition points between recipe steps.

We often truncate the time window of each recipe step by removing a certain percentage of data points at the beginning and the end, and then we compute the mean, median and standard deviation of each step for each sensor. With *p* as the number of recipe steps, and *k* as the number of equipment sensors, a wafer can be represented by a 1-by-*pxk* dimensional vector. The dimension of variables can easily exceed 10,000, while the number of available wafers is relatively small. This can lead to a curse of dimensionality when statistical techniques are applied. We discuss extensively how to perform variable selection in Section 3.3

3.2.2. Data Scaling and Outlier Removal

Sensors often malfunction in production, and can lead to missing values in our data set. There have been extensive studies on imputing missing values in the area of DNA array data, and imputing missing values is not the focus of this thesis. Besides missing values, equipment sensors tend to give readings at different levels due to the chamber physics they are monitoring. Thus, it is necessary to scale the sensor data prior to any statistical analysis. In all the analysis performed in this chapter, we have applied Z-score standardization [3.10] to scale the data as shown:

$$x_{new} = \frac{x - \bar{x}}{\sigma} \tag{3.1}$$

where $\bar{x}$ is the empirical mean of x and σ and is the empirical sigma.

There are several approaches to avoid skewing due to outlier effects when scaling the data, which involve using the robust summary statistics median and median absolute deviation (MAD) rather than the mean and standard deviation.

Once data is scaled, it is also necessary to remove outliers form the training samples. Including outliers in the training set can affect the robustness of the resulting model since outlier data samples can be bad leverage points that skew the regression result, as shown in figure 3.6. In this thesis, we have focused on using PCA results to isolate outlier wafers and exclude them from model building. A scatter plot of the PCA score is an effective way to visualize outliers and isolate them, as shown in figure 3.7.

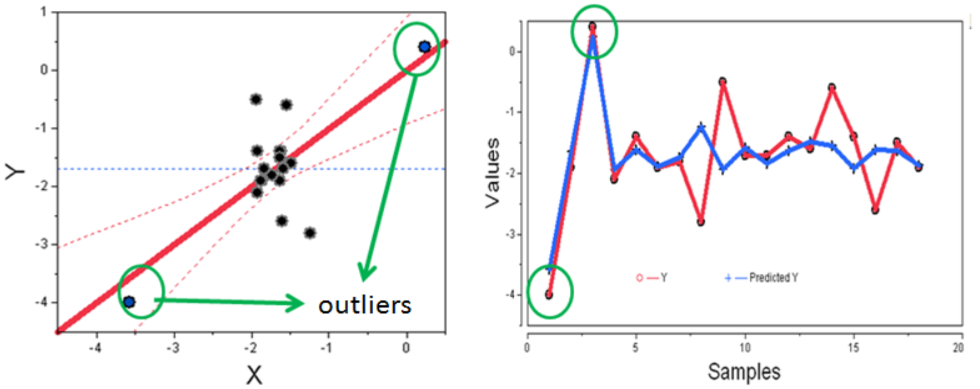


Figure 3.6. Plot for bad leverage points affecting the regression line

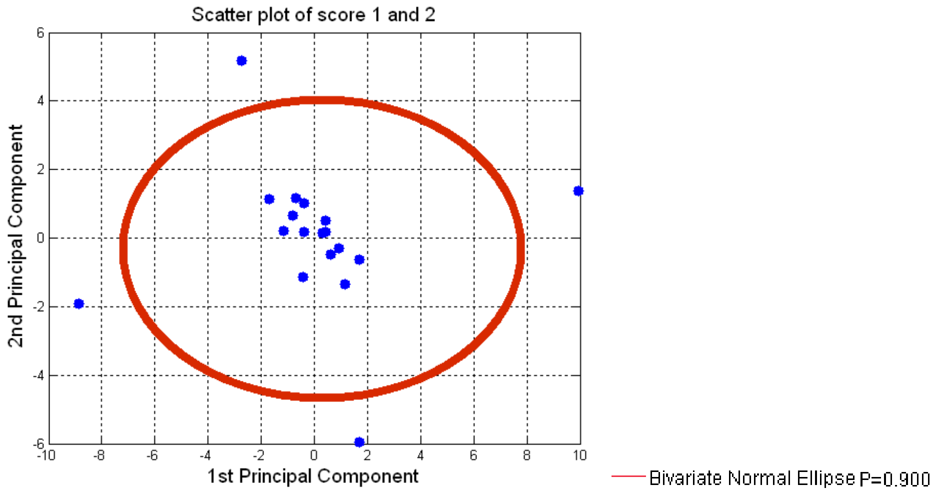


Figure 3.7. Isolating outliers with PCA score scatter plot

3.2.3. Data Detrending and Visualization

Process tools are known to age over time, and aging is reflected in the drifting of sensor reading data. Sensors monitoring different chamber physics tend to drift in different patterns, but the patterns are mostly correlated to the timing sequence of the wafers. PM events are performed on chambers regularly, which can set the sensor reading back to the normal range. Therefore, sensors can display various types of wafer to wafer trending depending on their physical meaning. In figure 3.8, we illustrate parameter trending with OES sensors reading from plasma etching tools. We can observe the resetting of OES sensor reading after PM events. Trends need to be corrected before parameters can be incorporated into models.

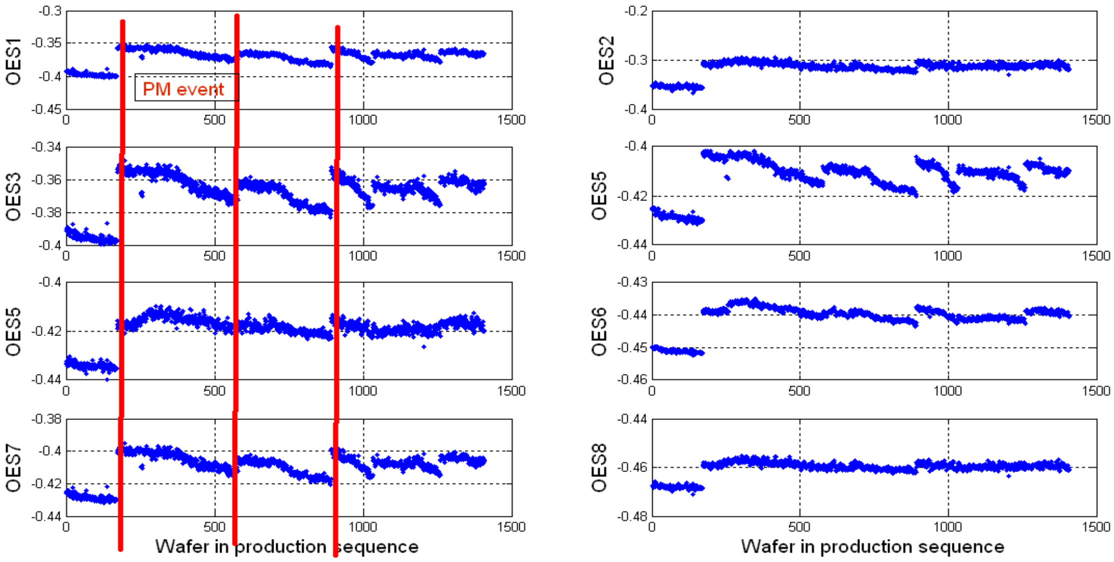


Figure 3.8. OES sensors plot for wafers covers 1 production year.

There are two approaches to relieving the trending effect from modeling. The first approach is to assign weights to wafers based on their corresponding positions within production cycles. The production cycle can time the window between PM events. Then, wafers at the end of the time window can be weighted less since their sensor readings tend to be corrupted more due to aging chamber conditions. A second approach is to remove the trends from sensors by applying a time series model to the parameters. This approach can be illustrated in this form: $x_i = f(t) + e_i$. We first assign index variable t as the timing indicator of a specific wafer, then we model the trend pattern as a function of t. The residuals of the model will then be used as inputs into either FDC or predictive models. The function $f(t)$ can either be linear or nonlinear.

In figure 3.9, we show an example of applying Gaussian Process Regression (GPR) [3.11] as a nonlinear function to remove the long term wafer to wafer trend; we then compare the results against PC-based trend removal in figure 3.10. The PCA-based trend removal approach assumes there is a main trend across all the parameters that contribute most to the variance of the dataset, while the time series function approach assumes that every parameter has its own trend that is dependent on the position indicator. Equation shows how PCA decomposes the data matrix X into corresponding scores matrix T and loadings matrix P. For trend removal, we can assume that the main trends can be captured by the first k principal components. Then we retain the E as an input f further modeling.

$$X = TP^T + E \tag{3.2}$$

Where X is nxp matrix, T is nxk matrix, P^T is kxp matrix and E is nxp matrix. n is number of samples, p is number of dimension, and k is number of principal components.

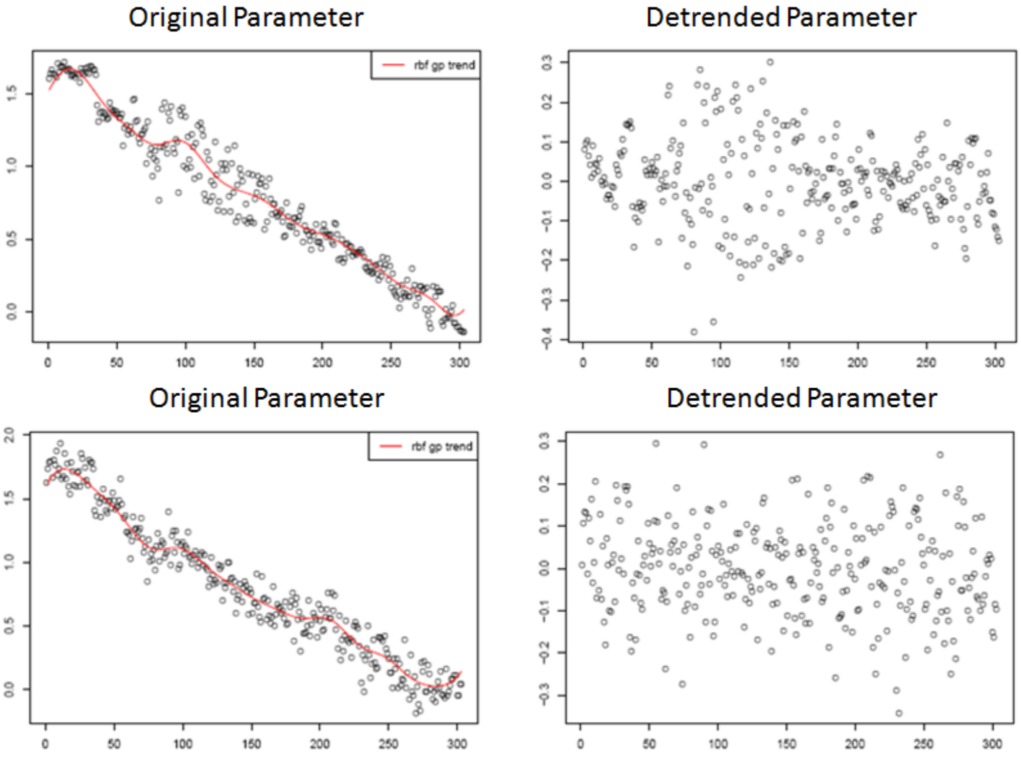


Figure 3.9. GPR based trend removal

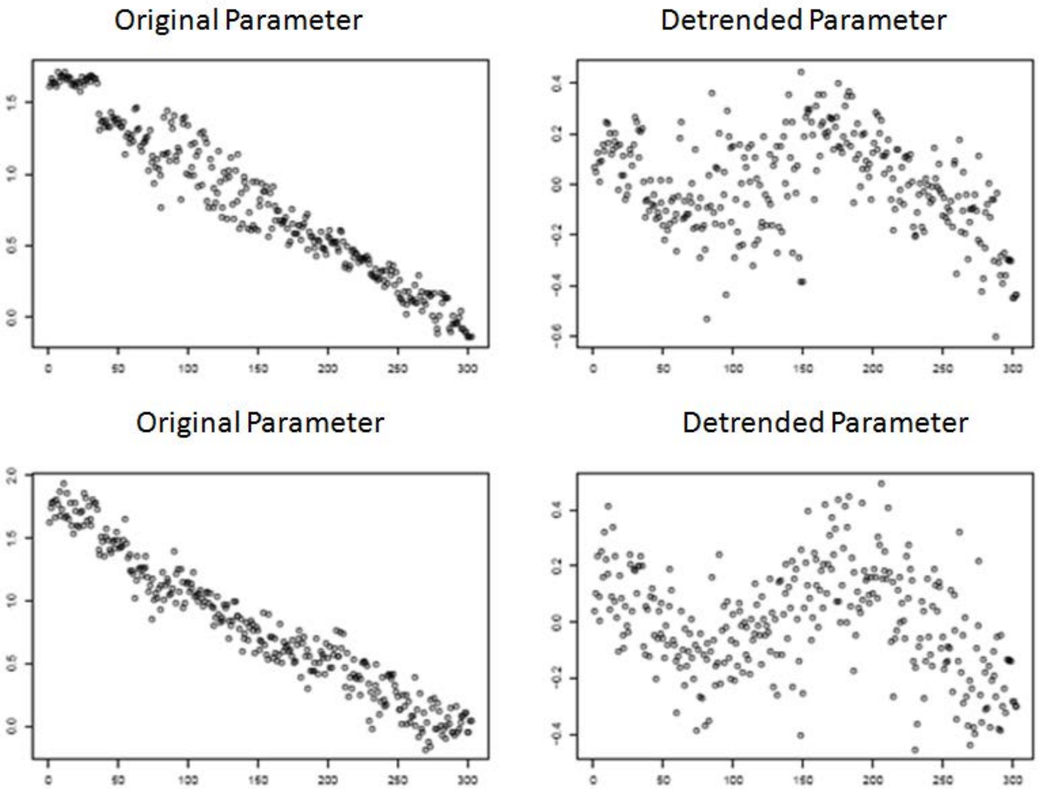


Figure 3.10. PCA-based trend removal after the first 3 PCs have been removed

Given that there are multiple collinear structures among sensor parameters, it is highly recommended to individually de-trend the sensor parameters based on the pattern.

After the time trace process data is compressed into one wafer per observation, with each observation being *p*-dimensional, it is difficult to visualize how each parameter varies temporally over all the wafers. Temporal parameters variation provides us with insights on how sensitive the equipment sensors are with respect to process excursions. In this thesis we have utilized the spectral heat map [3.12] for high-dimensional visualization. We find it extremely useful for wafer data sets from DOEs. In figure 3.11, we show a heatmap of a sampled set of the production wafer. The data includes wafers from same process and equipment. Each wafer has 520 sensor parameters retained. The heat map allows an engineer to quickly visualize the sensor parameters corresponding to any fault. In this case, we can visualize the difference of OES sensor parameters for different production periods.

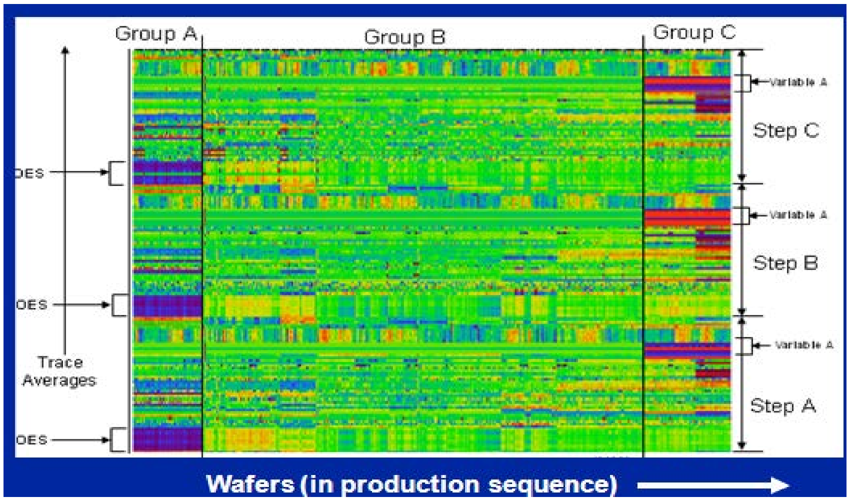


Figure 3.11. Heat map for high-dimensional visualization

3.3 Variable Selection

Once parameters have been pre-treated as described above, we can use them to build the predictive VM model. However, there are typically too many parameters available, with large subsets exhibiting co-linearity or lacking predictive meaning. The selection of which variables should go into the prediction model is a hard problem. In the case of semiconductor manufacturing, the highly collinear nature of equipment sensors results in many redundant variables for modeling. In figure 3.12, we show the correlation heat map of variables extracted from the plasma etching tool sensor data. We can observe several blocks of variables having a Pearson correlation coefficient of nearly 1. Many approaches have been suggested in different applications, but there is still no general solution. Working solutions have been found only for a very specific subset of the problem space. The aim in this section is to generalize the variable selection schemes into two categories, and we discuss several methods based on applications for each category. The performance of variable selection is then evaluated with a plasma etching operation dataset.

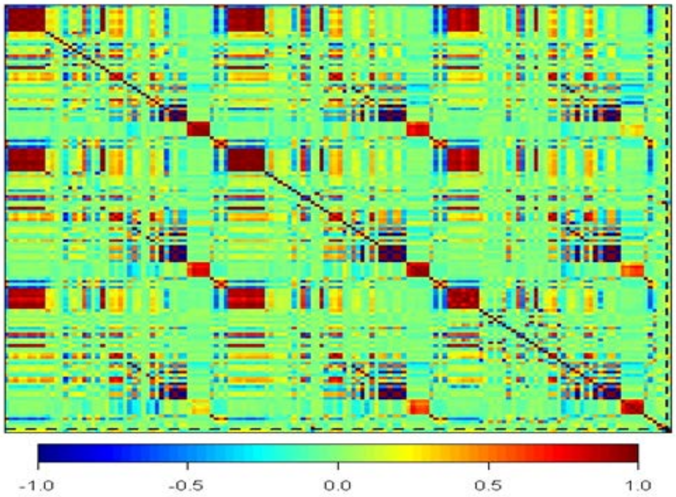


Figure 3.12. Correlation heat map for extracted variables set from the plasma etching tool sensodata

Variable selection in the field of semiconductor manufacturing can be classified into two categories: the domain knowledge-based approach and the statistical selection criterion-based approach.

3.3.1. Domain Knowledge-based Approach

Variable filtering with domain knowledge can improve the final results of models significantly, especially for the FDC model case. In both the FDC and predictive metrology cases, one should remove the control variables and noisy variables, and exclude variables from non-process steps. Thus, variable removal for this step is recipe-dependent. For most recipes, control variables are either constant or discrete values. Thus, variables carry near-zero variance, which does not explain any variation in the response variable. Noisy variables come from either miscalibration of equipment sensors or tool aging, and domain experience is often used in order to identify them.

In this thesis, the study was carried out on the plasma-etching operation. We have filtered out (i.e. removed from further consideration) three types of variables:

1) variables from non-critical process steps that are unlikely to relate to product outcomes
2) variables that are largely constant (defined here as having more than 70% of their samples having the same value)
3) control variables that simply reflect the intended process settings.

Attempting to include those in the analysis would complicate the dimensionality reduction step and may also lead to misleading model results.

For statistical selection criterion-based approaches, the concept is to select the variables based on a statistically-defined threshold. The approaches can be categorized as filter and wrapper approaches [3.13].

Filter approaches rely on selecting the variables that can explain the greatest variance in the dataset. The selected variables do not necessarily explain short range variation in response variables, but they often contribute to explaining any long range clustering of the data points. Therefore, filter approaches are more appropriate for FDC applications. Qin (2004) [3.14] used information-theoretic ranking criteria to select a subset of variables as input to the support vector machine (SVM) model for fault classification of RTA wafers.

Wrapper approaches are common for predictive models. The concept is to select a subset of variables, and then evaluate how well the variable set can predict the response variable. Thus, wrapper approaches are biased by the applied evaluation methods, and nonlinear and linear methods can influence the final selected variables. We illustrate the difference between filter and wrapper methods in figure 3.13.

In this thesis, we have pursued both approaches. Among filtering approaches, we have applied sequential PCA to find sensors traces with significant trends, since they can provide insight into the process. We have utilized wrapper approached such as bootstrapping [3.15] and stepwise frameworks in conjunction with multivariate methods to select predictive variables for the predictive metrology model. Since our objective in this study is to create a robust VM model with high predictive power, we focus more on wrapper approaches in the following sections.

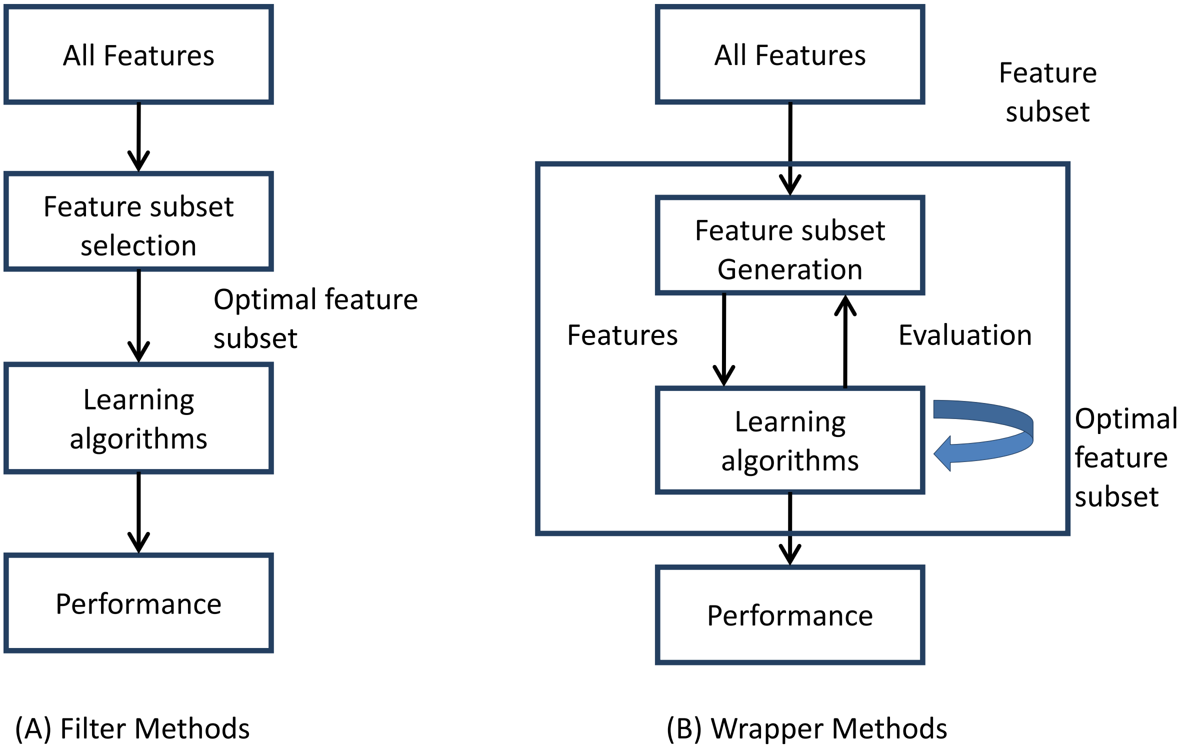


Figure 3.13. Filter and wrapper methods

3.3.2. Filter Methods

Filter methods can be considered as preprocessing steps since they are independent of the response variable. The advantages of filter methods are that they make it possible to avoid over-fitting and usually have a lower computational cost, since they do not require training. They are appropriate for FDC application, since the variables contain most of the information of the dataset and can distinguish well between “good” and “bad” sample clusters. Duda (2000) [3.16] used Fisher’s criterion to rank variables in a classification problem where the covariance matrix is diagonal, which is optimum for Fisher’s linear discriminant classifier. Statistically, filter methods are robust against wrapper methods in terms of over-fitting since they can introduce bias with less variance.

In this study, we have used PCA heuristically for variable selection. The Pearson correlation coefficient $r(i)$[3.16] is the selection statistic we used along with PCA:

$$r_i = \frac{cov(x_i, y_i)}{\sqrt{var(x_i)var(y_i)}} \tag{3.3}$$

The coefficient is also the cosine of the angle between the two vectors ***x*** and ***y*** after they have been mean-centered. Its value ranges from 1.0 for perfect positive correlation to -1.0 for perfect negative correlation and is near 0 for uncorrelated variables.

3.3.2.1. PCA-based Variable Selection

Conventional PCA has also been used for variable selection by comparing the loading coefficients for variables. The variables with higher loading coefficients are assumed to have more weights in the main direction of variances, and contain more information about the dataset. This is similar to the method proposed by Jollife [3.17], who uses the loading vectors of the first k PCs for variable selection. The total number of variables selected is equal to the number of retained PCs. The algorithm begins by finding the variable that has highest absolute loading coefficient on the first PC. That variable is then placed in the selected set. The algorithm continues into loadings of the second PC, and finds the variable with the highest absolute loading coefficient. The algorithm stops until the k most important PCs are checked. The method assumes that the i_{th} loading coefficient of one PC indicates that the i_{th} element is the dominant one in that PC. By choosing the variables corresponding to the highest coefficients of each of the first k PCs, the majority of the information for the dataset can be captured. The shortcoming of this method is that it considers each PC independently, whereas variables with similar information content might be chosen. Sparse PCA [3.18] tries to find a sparse set of variables that explain the PCs. Lu [3.19] used the loadings of the first several PCs as inputs for k-means clustering, selecting the features that are closest to the cluster's centroid.

In this section, we utilize a sequential scheme to perform PCA and use extracted scores to identify the set of sensors have different trends. The method can be described below:

1) Let us suppose we have n wafers with p sensor trace parameters extracted. Thus the input matrix X^k has $n \times p$ dimensions. We first perform PCA on X^t:

$$X_{re}^{k} = \boldsymbol{t_1}\boldsymbol{p_1^T} + E^k \tag{3.4}$$

   where $\boldsymbol{t_1}$ is the score computed by retaining the first loading vector, X_{re}^{k} is the reconstructed matrix with the first PC, and E^k is the residual after reconstruction.
2) $\boldsymbol{t_1}$ is a column vector and is used as the basis to select sensor parameters. We compute the Pearson correlation coefficient for t_1 vs X^k. Let us denote the sensor parameter with the highest correlation coefficient as f_j^k.
3) Repeat Step 1 and 2 with input matrix as E^k. Find the sensor parameter with the highest correlation coefficient with respect to the retained score vector $\boldsymbol{t_1}$.

We can repeat the above steps until the variance explained by the first PC is less than 10% of total variance. This means the data is becoming sparse and there is not a dominant variation direction for the samples. We demonstrate the application of this method for finding sensors with trends.

The dataset used in this case is collected from 1043 wafers processed in Poly-Si etchers. After preprocessing a total of 143 sensor parameters remain. In figure 3.14a, the scree plot of the PCA result is plotted after every iteration (run). For up to the eighth iteration, there is still a single component that dominates the variation. We highlighted the sensors with the highest correlation coefficients with respect to the first score in figure 3.14b, and the sensor parameters exhibiting trends are identified and shown in figure 3.14c.

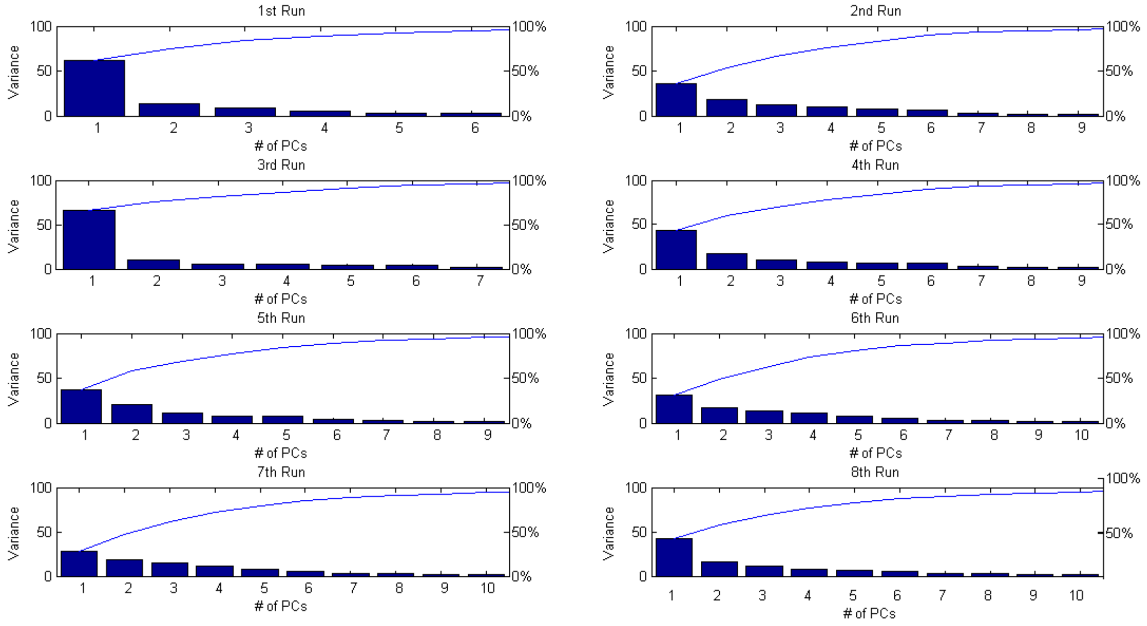


Figure 3.14a. Scree plot of PCA for first 8th iterations.

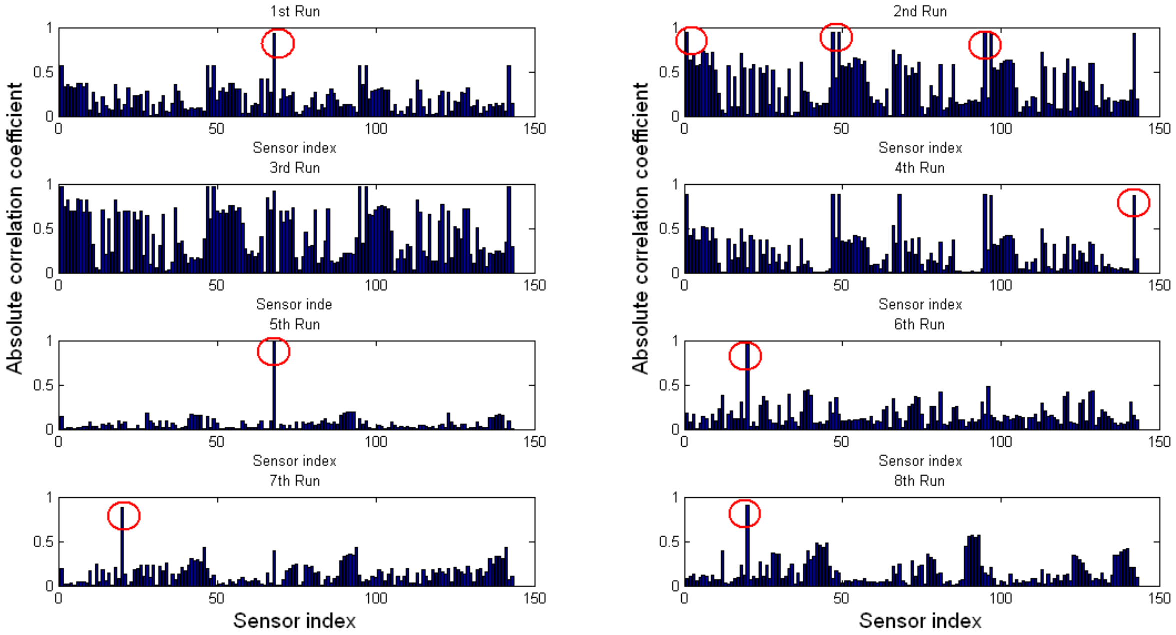


Figure 3.14b. Identifying sensors with highest correlation coefficient to 1st PCA score

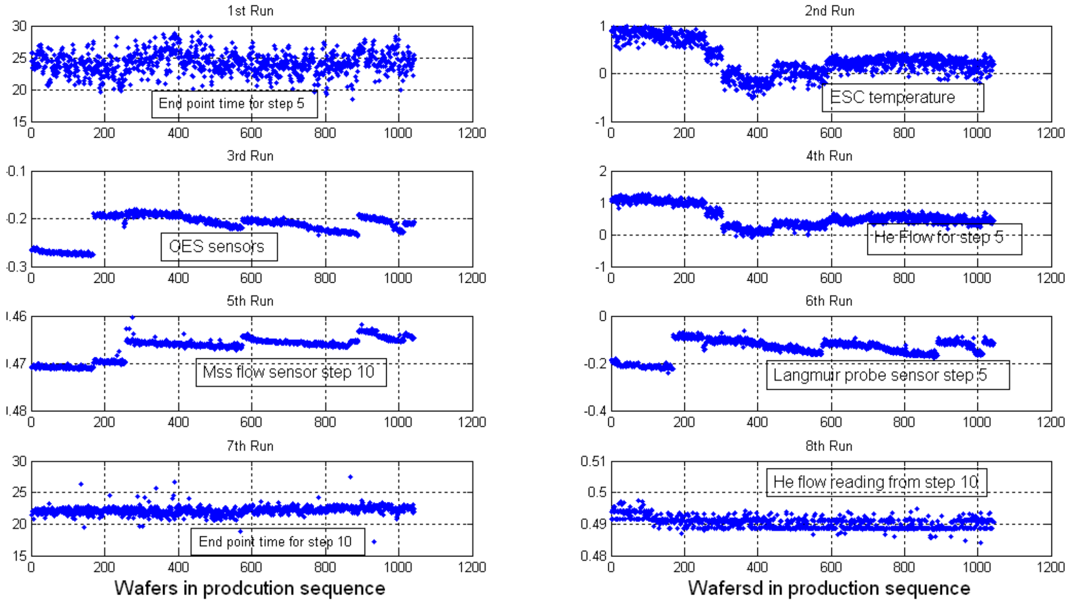


Figure 3.14c. Sensors have highest absolute correlation with respect to 1st PCA score

3.3.3. Wrapper Methods

Wrapper methods generally utilize a framework to select a subset of variables based on their predictive power. For these methods, one needs to define: (1) how to search all the possible subsets; (2) how to evaluate the prediction performance of a variable subset; (3) which predictor to use. The problem is NP-hard. The popular methods for performance evaluation include tree-based regression or classification and the least square regression method. In this thesis, we combine greedy search algorithms such as a stepwise, heuristic genetic algorithm and bootstrapping along with multivariate regression methods such as Lasso, PLS, and CART for predictive variable selection.

Stepwise selection [3.20] can be either through forward selection, backward elimination, or a combination of the two. Forward selection methods start with one or a few features selected according to a method-specific selection criteria. More features are iteratively added until a stopping criterion is met. Backward elimination methods start with all features and iteratively remove one feature or several features at a time until stopping criterion is met. For mixed selection methods, features can be added as well as removed from the data during iterations. Since stepwise approaches are heuristic in nature, they are often incapable in handling collinearity, and the resulted R^2 values highly depend on the various often arbitrary choices made when constructing the selection algorithm. In the stepwise variable selection case, we have utilized a bootstrapping framework on a sample space to reduce the shortcomings of the stepwise approach.

Bootstrap is a simulation approach to estimating the distribution of test statistics. The method is to create "bootstrap" samples by resampling the data randomly, and then estimate the distribution of an estimator or test statistic. In this section, we explore bootstrapping regression models to select important variables. Given $X = [x_1, x_2, \dots, x_N] \in R^p$ and $= [y_1, y_2, \dots, y_N] \in R^1$, we simply select k bootstrap samples of size n from the x_{is} and corresponding y_{is} , fit the model and save the coefficients from each bootstrap sample. We can then obtain distributions for regression coefficients of each predictor variable. Critical variables can then be selected based on distribution of the regression coefficient.

3.3.3.1. Bootstrapping with Stepwise Regression

1) For a given X as an *n*-by-(*p*+1) sample matrix with the last column vector as the response variable, we select b independent bootstrapped samples, each consisting of n data samples drawn with replacement from X.
2) Constructing stepwise linear regression with each bootstrapped samples, we obtain *b* stepwise regression models.
3) For each of the stepwise regression models, we select the top k_s predictor variables out from p predictor variables that result into an R^2 value of at least 0.9 for the fitted model. k_s denotes the number of predictor variables retained for a specific trained stepwise model.
4) A "1" is assigned to the predictor variable if it is selected; otherwise, a "0" is assigned.
5) Frequency counts are obtained for all predictor variables from the *b* bootstrapped models.
6) We retain the top k predictor variables, where k is the smallest of all the k_s values.

3.3.3.2. Bootstrapping with Lasso

1) For a given X as an *n*-by-(*p*+1) sample matrix with the last column vector as the response variable, we select b independent bootstrap samples, each consisting of n data samples drawn with replacement from X.
2) Constructing Lasso model with each bootstrapped sample, we obtain *b* Lasso regression models.
3) For each Lasso regression model, a sparse set of predictor variable is obtained. k_l denotes the number of predictor variables for a specific sparse solution set.
4) A "1" is assigned to the predictor variable if it is selected; otherwise, a "0" is assigned.
5) Frequency counts are obtained for all predictor variables from the b Lasso regression models.
6) We retain the top k predictor variables, where k is the smallest of all the k_s values.

3.3.3.3. Bootstrapping with PLS (VIP)

PLS-based variable selection is done in two ways: variable importance upon projection (VIP) [3.21] and the interval of each coefficient. VIP provides a score for each feature, so that it is possible to rank the features according to their predictive power in the PLS model (the higher the score the more important a variable is). VIP for *j*-th variable is defined as :

$$VIP_j = \sqrt{p \sum_{j=1}^{k} b_j^2 w_{kj}^2 / \sum_{j=1}^{k} b_j^2} \tag{3.5}$$

where p is the number of variables, w_{kj} is the j-th element of vector of w_k , and b_j is the regression weight for the k-th latent vector, $b_j = u_j^T t_j$. We adopt VIP with bootstrapping in this thesis as one of variable selection schemes.

1) For a given X as an n-by-(p+1) sample matrix with the last column vector as the response variable, we select b independent bootstrap samples, each consisting of n data samples drawn with replacement from X.
2) Constructing a PLS model with each bootstrapped sample, we obtain b PLS regression models.
3) For each of the PLS regression models, we compute the VIP for each predictor variable.
4) From all bootstrapped PLS models, we obtain distributions of VIPs for each predictor variable.
5) The median for each distribution of values VIPs is computed and used as the selection statistic.
6) Predictor variables with median statistics above 2 are retained as important predictor variables.

3.3.3.4. Bootstrapping with CART

(1) For a given X as an n-by-(p+1) sample matrix with the last column vector as the response variable, we select b independent bootstrap samples, each consisting of n data samples drawn with replacement from X.
(2) Constructing CART with each bootstrapped sample, we obtain b CART models.
(3) For each of the CART regression models, a sparse set of predictor variables is obtained. We denote the number of predictor variables for each sparse set as c_i.
(4) If a predictor variable is selected for one model, we then compute its contribution to the model.
(5) From all bootstrapped CART models, we obtain distributions of the variable contribution for each predictor variable.
(6) The median for each distribution of the variable contribution is computed and used as the selection statistic.
(7) Each variable is then sorted in descending order based on the extracted median statistics. The number of predictor variables retaining is the $\max_{i \in B}[c_i]$.

3.3.3.5. Genetic Algorithm with PLS

Genetic Algorithm (GA) [3.22] is a popular optimization method that imitates the natural selection process in biological evolution with selection, mating reproduction, and mutation. The algorithm flow is shown in figure 3.15. In GA, the parameters to be optimized are represented by a chromosome where each parameter is encoded in a binary string called a gene. Thus, a chromosome consists of as many genes as parameters to be optimized. A population consists of a certain number of chromosomes, and is initially created by randomly assigning “1” or “0” to all genes. The best chromosomes are the ones with the highest probability to survive evaluation by the fitness function. The next generation is obtained by selecting the best chromosomes, mating the chromosomes to produce an offspring population, and by a probabilistic mutation. The evaluation and reproduction steps are repeated for a certain number of generations, or until a convergence criterion of the population is met. The theory and benefits of GA in variable selection have been described in the literature [3.23] and will not be discussed here. In this thesis, we describe the implementation of the GA framework with PLS as the fitness function. The evaluation of fitness is defined by the R^2 of the testing dataset. Random noise is an important issue that affects the result of a GAPLS model. If the predictors are very noisy and sample size is small (predictor #/sample #), the GA will model the noise rather than information.

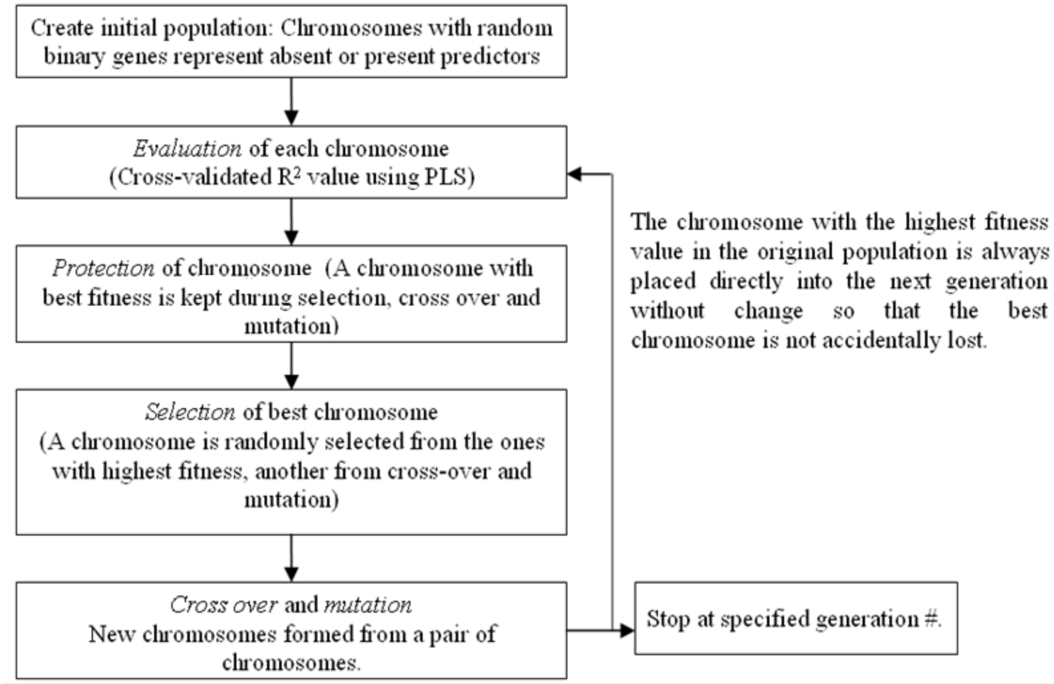

Create initial population: Chromosomes with random binary genes represent absent or present predictors
Evaluation of each chromosome (Cross-validated R² value using PLS)
Protection of chromosome (A chromosome with best fitness is kept during selection, cross over and mutation)
Selection of best chromosome (A chromosome is randomly selected from the ones with highest fitness, another from cross-over and mutation)
Cross over and *mutation* New chromosomes formed from a pair of chromosomes.
The chromosome with the highest fitness value in the original population is always placed directly into the next generation without change so that the best chromosome is not accidentally lost.
Stop at specified generation #.

Figure 3.15. GAPLS

Another result of GAPLS is that only a few variables are present in the final model; this might not utilize the full power of PLS. Some randomly correlated variables might be selected in the final resulting model and carry less information on the dataset. One would have to run GA many times to obtain clear information on the dataset; we often use 100 as the optimal run number. The critical variables defined are based on the frequency with which each variable appears in the top chromosome of each run. A stepwise selection can also be performed on the ranked variables; the results of this stepwise procedure can provide the optimum variable selection. This stepwise selection step is highly recommended if the predictor size or sample size is large, and if the random correlation is higher in the dataset.

The variable screening is done on all the wafers rather than on the chambers separately. Artificial “predictor” variables are added into the model; they are randomly generated numbers from Gaussian distribution. Predictor variables with a selection frequency less than the maximum frequency of noise factors can be considered irrelevant for further modeling.

3.4 Model Creation

After critical predictor variables are selected based on training data, we can then apply prediction methods to train a VM model. There are multiple ways to construct a VM model in terms of splitting the training and testing dataset. In this section, we compare four approaches for constructing prediction models: (1) prediction with serial correlation; (2) the operation state-driven approach; (3) the moving window approach; (4) chronological splitting, with 70% training and 30% testing in temporal order.

In the first approach, a new data matrix X that combines consecutive samples is constructed to consider the time series correlation among samples. In the second approach, we consider the fact that samples are collected over a long production period which contains multiple operation states. The third approach employs frequent updates and short-term prediction to create a robust model. The fourth approach is the conventional approach, where one trains a model on a historical dataset, and uses this model to forecast the remaining dataset. The robustness of the model can be observed in the variation of the temporal residuals. A combination of these model creation approaches with variable selection schemes are evaluated with different statistical modeling methods.

3.4.1. Dynamic Prediction with Serial Correlation Consideration

Directly applying regression methods on the data matrix X constructs a static linear model while the data might be generated from a dynamic process, which takes into account any autocorrelation in the participating data series. The static model assumes that the observations are not correlated serially; i.e., current observation is independent of previous ones. However, this might not be true for real world processes. In this section, we transform the data matrix X to contain previous observations. We indicate this matrix as lagged X. For a *n* x *p* data matrix X, the lagged-*k* symbolized by X_L (k) is defined as:

$$\mathrm{X_L(k) = [X(t)\,X(t-1)\ldots X(t-k)]} \tag{3.6}$$

where X(t - k) is an *(n-k)* x *p* matrix, obtained by removing the last k samples from the original n x p data matrix X. The lagged data matrix $\mathrm{X_L(k)}$ is then used with the regression method we mentioned above. In this study, we used $\mathrm{X_L(1)}$ as the input data matrix since most industrial processes data exhibit 1st order autocorrelation. That is, we modeled the Y-response variables as a function of $\mathrm{X_t}$ and $\mathrm{X_{t-1}}$, and for a single prediction at one time point:

$$y_t = f(x, x_{t-1}) + e_t \tag{3.7}$$

It is clear that this approach will suffer from the curse of dimensionality given that the number of the predictor variable is doubled. In the section 3.5, we have utilized wrapper methods to select an optimal set of predictor variables.

3.4.2. State-driven Prediction Approach

Given that the data are collected over a long production period, it is expected that there are multiple operation modes. Each operation mode tends to have a different data structure; thus different prediction models are needed for different modes. A predictor variable might not have a high correlation with response variables across the entire dataset, but it can have a higher correlation within a certain operations mode. The multi-mode nature of the data can be reflected in a multimodal distribution of predictor variables and clustering in the projected subspace.

In this study, the wafer data we collected covers approximately one production year. We show distribution plots of several predictor variables in figure 3.16, and clustering behavior of data structure in the PCA subspace is shown in figure 3.17. Both figures validate the fact that there are at least two operation modes in the data. Once we identified those states with unsupervised clustering methods such as k-means clustering, we then create prediction models corresponding to specific tool states. And once we obtained the tool state labels with clustering, we then create classification models of the training dataset with KNN methods. In this way, each incoming wafer is classified into the corresponding operation state before the corresponding metrology results are predicted.

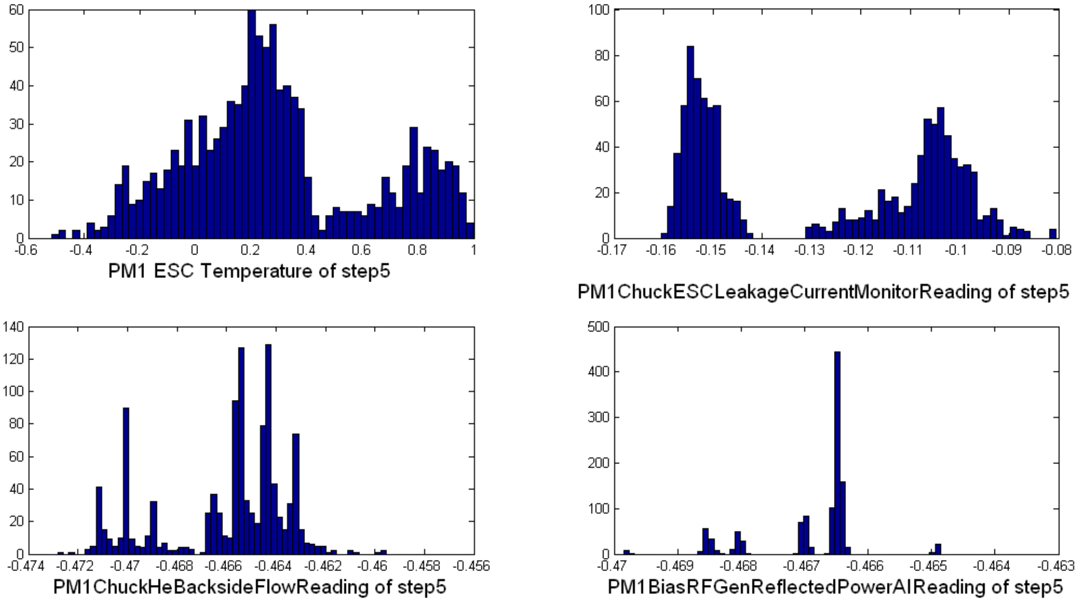


Figure 3.16.Multimodal distribution of predictor variables

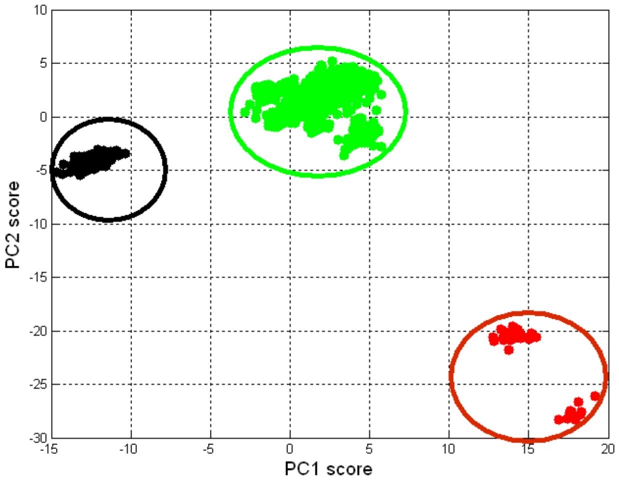
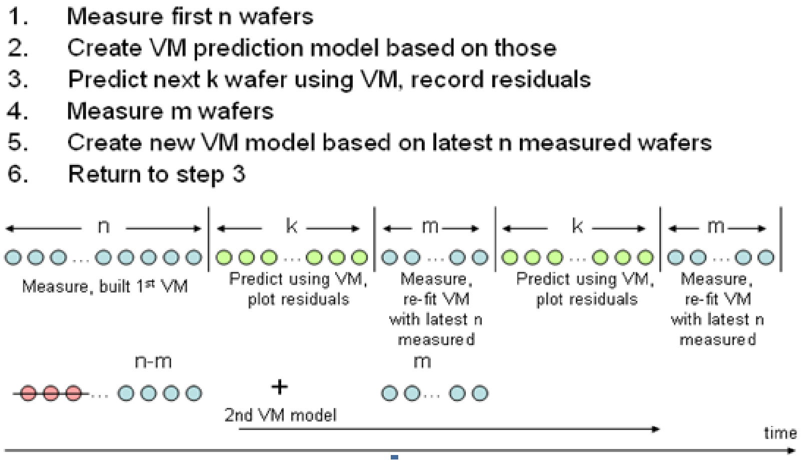


Figure 3.17.Clustering in PCA subspace

In figure 3.17, the “red” cluster at the lower right of the graph contains approximately 31 points that are actually corrupted data points. They are removed in the preprocessing phase of all modeling analysis.

3.4.3. Moving Window Approach

The moving window approach is depicted in figure 3.18. The idea is to constantly update the trained model with newly-measured wafer metrology to capture process dynamics. There are three critical parameters to define: n as the number of training samples, k as the number of predicted samples, and m as the number of measured samples for the model update.



1. Measure first n wafers
2. Create VM prediction model based on those
3. Predict next k wafer using VM, record residuals
4. Measure m wafers
5. Create new VM model based on latest n measured wafers
6. Return to step 3

n
k
m
k
m
Measure, built 1st VM
Predict using VM, plot residuals
Measure, re-fit VM with latest n measured
Predict using VM, plot residuals
Measure, re-fit VM with latest n measured
n-m
+
m
2nd VM model
time

Figure 3.18.The moving window approach

This moving window approach can be combined with several prediction methods to obtain the best appropriate VM model. The optimal (n, k, m) is generally determined in multiple trials with consideration of the number of predictor parameters and available data samples.

3.5. Prediction Results

In this section we summarize model prediction results for a combination of different variable selection schemes with different prediction methods. The dataset used here is collected from a single poly-Si plasma etching tool running with the same recipe which includes 1115 wafers that cover approximately one production year. The recipe contains 25 process steps with 47 sensor parameters monitored for each step. Sensor parameters from three critical steps, corresponding to pre-etch, main etch, and post-processing have been retained to extract the initial set of predictor variables. Median statistics for the sensor parameter for each step are extracted as initial set predictor variables. The single response variable is the difference in Critical Dimension from the photoresist pattern to the finally etched pattern, also known as Etch Bias (EB. Therefore, after preprocessing and removing of corrupted samples, we are left with 1080 wafers and 145 possible explanatory variables, which consist of the 3x47=141 parameter traces plus the process time[1] for each of the three steps plus the hard mask trim time. So the input matrix X

[1] In these etch steps the duration is determined for each wafer through automatic, Optical Emission based endpointing.

1080-by-145, and the output matrix consists of the mean values of the EB on each wafer so that Y is 1080-by-1. The performance measure for VM model is R^2 defined as follow:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{pred} - y_{actual}\right)^2}{\sum_{i=1}^{n}(y_{actual} - y_{mean})^2} \tag{3.8}$$

where y_{pred} is the predicted EB, y_{actual} is the actual EB, and the y_{mean} is the mean of metrology for all the actual ones. Detrending in this study was only done on OES sensor parameters given that their trending patterns are strongly correlated to PM event cycles.

In table 3.2, we summarize the prediction results building VM with dynamic prediction. For the dynamic prediction case, where the lag-1 autocorrelation is taken into account, the input data matrix doubles in size by including both present-time and lag-1 past values of the input parameters, so that $X_{dynamic}$ is 1080-by-290 and the output matrix is $Y_{dynamic}$ is 1080-by-1. Outlier removal is done with PCA-based Hotelling T^2 statistics. The PCA model in this case is trained using 60% of the produced wafers $X_{dynamic}$. Hotelling T^2 is then computed for each training and testing wafer, and wafers with T^2 statistics above the upper bound of the 97.5% confidence interval are excluded. The Hotelling T^2 is computed by retaining the number of PCs explaining more than 95% of the total variance of the training set. The remaining 40% of the wafers serve as testing data points.

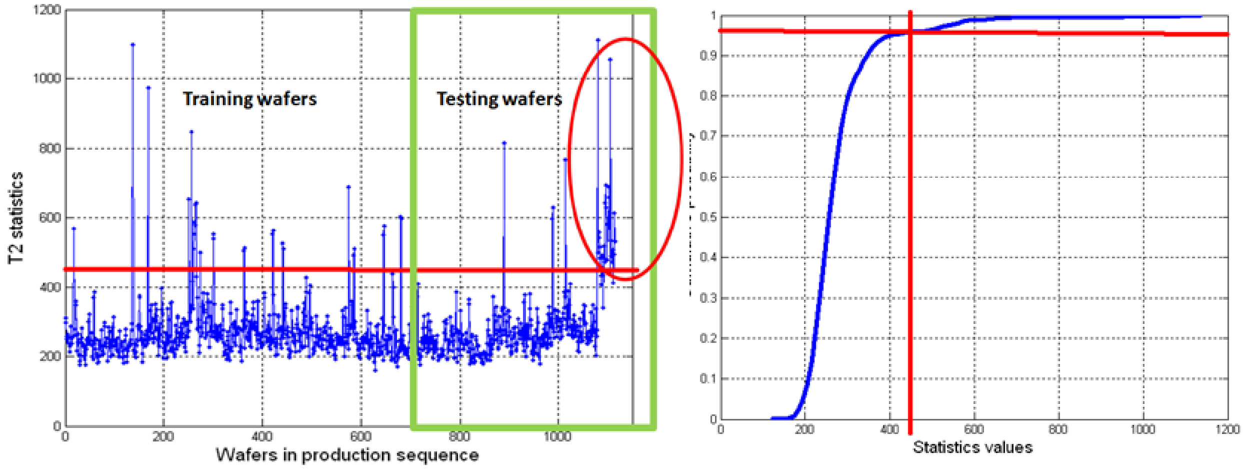


Figure 3.19. Outlier detection with PCA

Based on the T^2 statistics, 30 wafers from the training set and 36 wafers from the testing set are excluded from further modeling. After outlier removal, the variable selection based on wrapper methods that we discussed in Section 3.3.2 is applied to the training dataset since our objective is to predict the response variable-EB.

	Dynamic prediction with serial correlation consideration					
Variable selection	# of predictor variables	PCR Adjusted R^2	PLS Adjusted R^2	BPNN Adjusted R^2	KNN Adjusted R^2	CART Adjusted R^2
Bootstrapped	88	0.54	0.66	0.68	0.49	0.55

stepwise						
Bootstrapped Lasso	65	0.42	0.50	0.63	0.42	0.54
Bootstrapped PLS	80	0.53	0.65	0.69	0.45	0.55
Bootstrapped CART	58	0.54	0.55	0.58	0.41	0.54
GA PLS	78	0.55	0.63	**0.73**	0.45	0.53

Table 3.2. VM prediction results for the testing dataset with dynamic prediction

In table 3.3, we summarize the prediction results building VM with the operation state-driven approach. For the operation state-driven prediction case, the input data matrix X_{os} is 1080-by-145 and the output matrix Y_{os} is 1080-by-1. The first step is to see whether the wafers needed to be treated all together or separated into clusters and then k-means clustering was applied to group the wafers into clusters. Optimal value for k is determined to be 2 based on the distribution of pairwise distances for all the wafers. The distribution of pairwise distances computed with the Euclidean distance metric is shown in figure 3.20. The bimodal shape suggests a higher possibility of clustering among the samples.

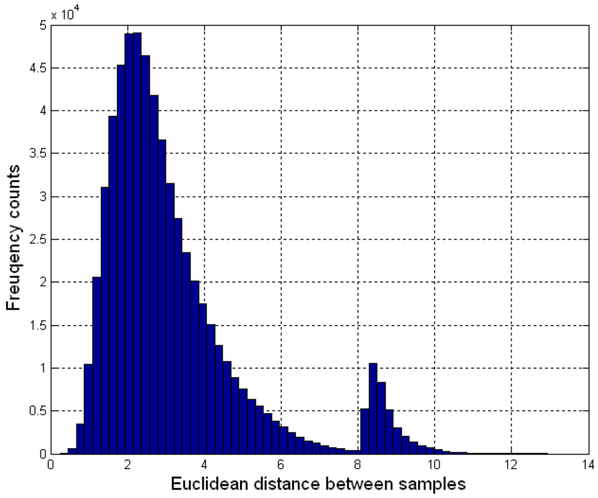


Figure 3.20. Distribution plot of pairwise distance (Euclidean)

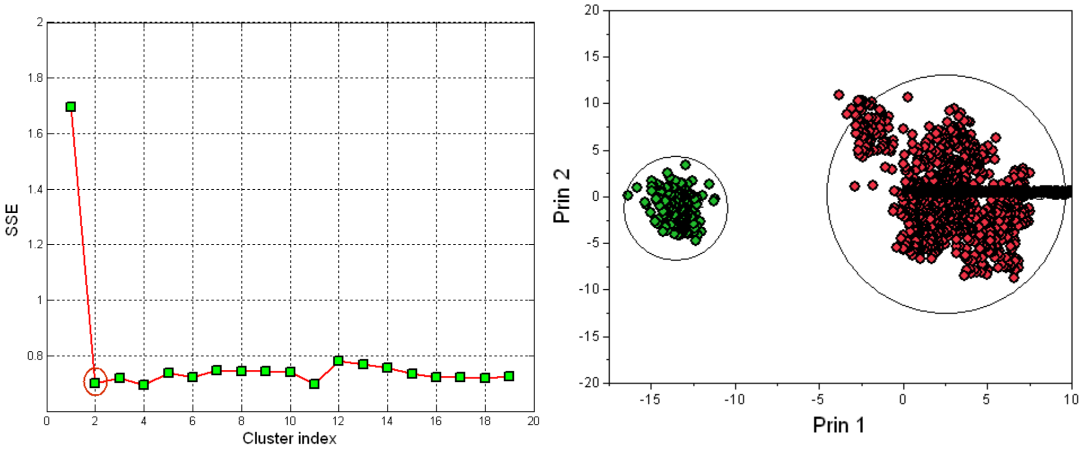


Figure 3.21. Clustering index vs. clustering visualization in PCA a subspace

Based on the clustering results, we labeled 169 wafers as "state 1" wafers, and 911 as "state 2" wafers. Outlier removal was then done with PCA-based HotellingT^2 statistics within each cluster. Wafers with T^2 statistics above the upper bound of the 97.5% confidence interval are excluded. The Hotelling T^2 is computed by retaining the number of PCs explaining more than 95% of the total variance of the training set. The remaining 40% of wafers serve as testing data points. The final testing R^2 is computed over all testing wafers from both clusters and presented in table 2.

The outlier results are shown in figures 3.22 and 3.23. For "state 1" wafers, there are only 4 outlier wafers. For "state 2" wafers, there are 19 outlier wafers. They are all excluded from model building and testing. After outlier removal, the variable selection based on wrapper methods that we discussed in Section 3.2.2 is applied to the training dataset within each cluster.

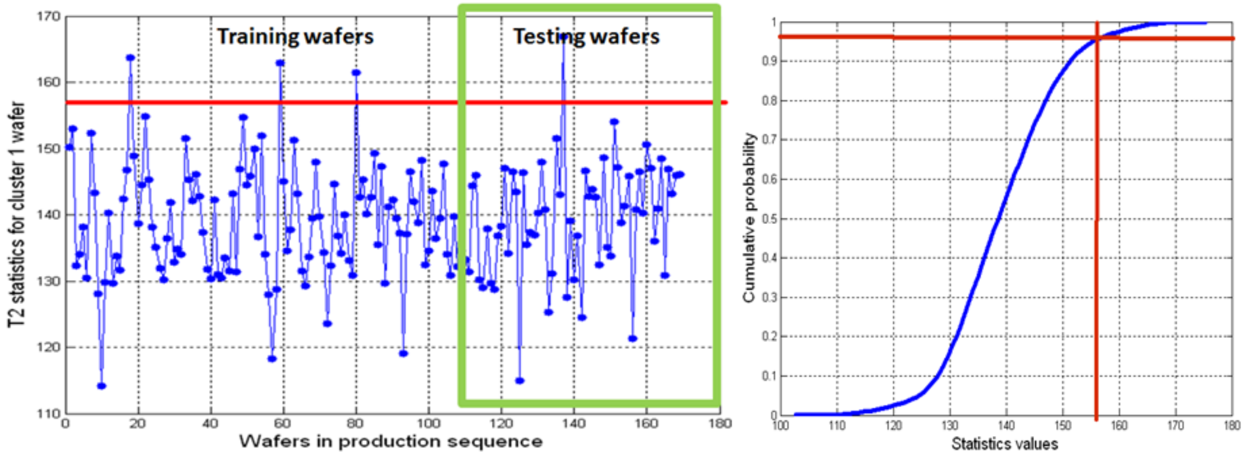


Figure 3.22. Outlier isolating for "state 1"wafers

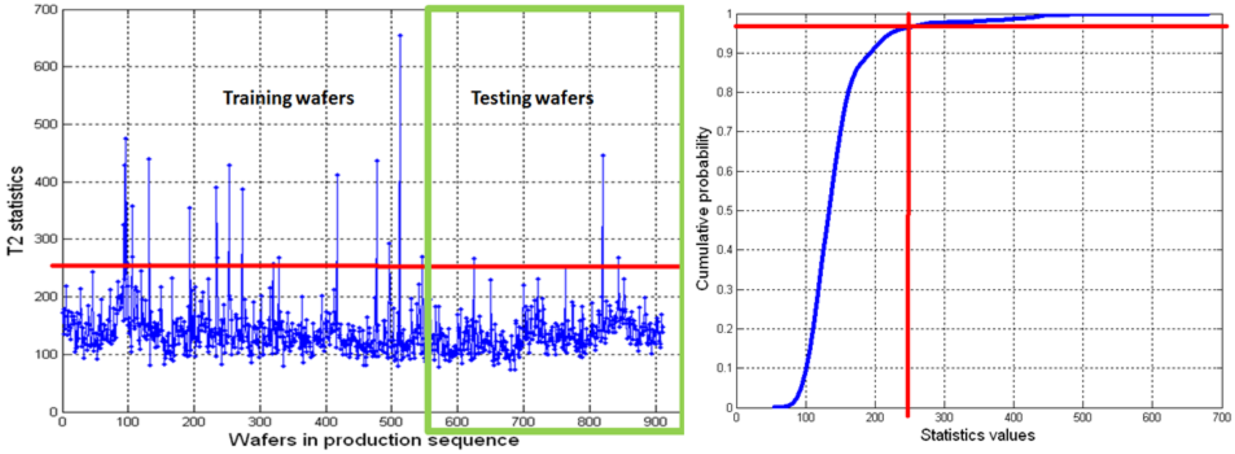


Figure 3.23. Outlier isolating for “state 2” wafers

	Operation state driven approach					
Variable selection	# of predictor variables	PCR Adjusted R^2	PLS Adjusted R^2	BPNN Adjusted R^2	KNN Adjusted R^2	CART Adjusted R^2
Bootstrapped stepwise	58	0.53	0.69	**0.73**	0.46	0.54
Bootstrapped Lasso	45	0.51	0.55	0.49	0.51	0.45
Bootstrapped PLS	50	0.54	0.69	0.69	0.42	0.53
Bootstrapped CART	45	0.47	0.58	0.65	0.54	0.58
GA PLS	46	0.51	0.65	0.70	0.39	0.48

Table 3.3. VM prediction results for testing wafers with operation state approach

In table 3.4, we summarize the prediction results of different VM models with the moving window approach. For the moving window case, the input data matrix X_{mw} is 1080-by-145 and the output matrix is Y_{mw} is 1080-by-1. Given that the number of predictor variables in the initial set is 145, we start with window size $n = 290$ to ensure that we have enough degrees of freedom for model training. Variable selection with wrapper methods is then applied on the first window. Selected predictor variables are then retained for the rest of the windows. A combination of (n, k, m) was tested to construct the best VM. The Adjusted R^2shown in table 3 is averaged over the predicted results of all windows. The selected (n, k, m) in this case is (290, 30, and 15).

	Moving window approach					
Variable selection	# of predictor variables	PCR Adjusted R^2	PLS Adjusted R^2	BPNN Adjusted R^2	KNN Adjusted R^2	CART Adjusted R^2
Bootstrapped stepwise	48	0.54	0.69	**0.73**	0.59	0.61
Bootstrapped Lasso	35	0.52	0.60	0.61	0.42	0.54
Bootstrapped PLS	45	0.60	0.66	0.71	0.52	0.61
Bootstrapped CART	41	0.53	0.55	0.58	0.44	0.59
GA PLS	39	0.59	0.65	0.72	0.55	0.60

Table 3.4. VM prediction results for testing wafers with moving window approach

In table 3.5, we summarize the prediction results building VM in chronological order. For the moving window case, the input data matrix X_{CO} is 1080-by-145 and the output matrix Y_{CO} is 1080-by-1. In this case, a static model is built with the first 60% of wafer data, and then the remaining 40% of wafers provide testing data. Outlier removal is done with PCA-based HotellingT^2 statistics. The PCA model in this case is trained with 60% of X_{CO}. HotellingT^2 is then computed for each of the training and testing wafers, and wafers with T^2 statistics above the upper bound of 97.5% confidence interval are excluded. Hotelling T^2 is computed by retaining the number of PCs explaining more than 95% of the total variance of the training set. Variable selection with wrapper methods is applied on the training dataset only.

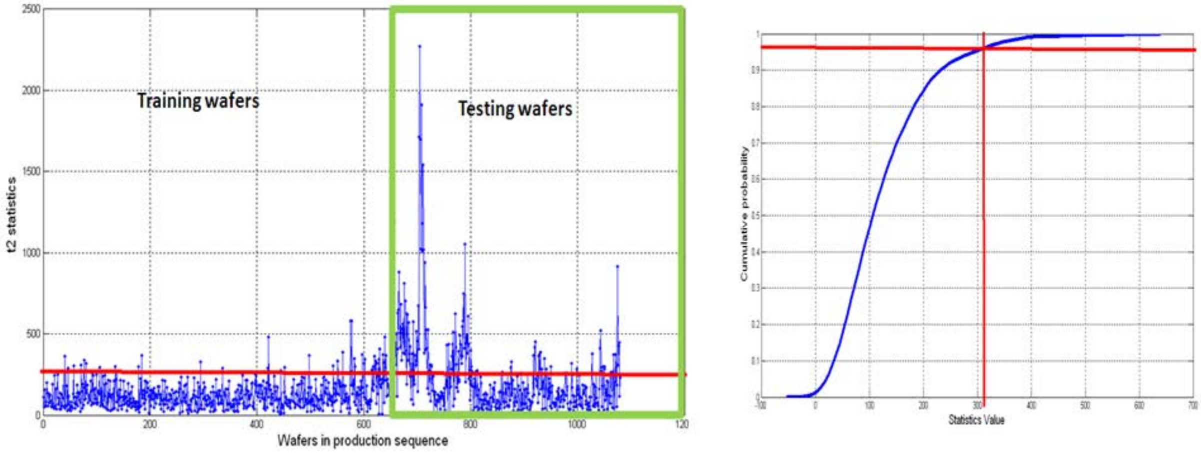


Figure 3.24. Outlier isolation with PCA for the chronological order model

	Chronological order					
Variable selection	# of predictor variables	PCR Adjusted R^2	PLS Adjusted R^2	BPNN Adjusted R^2	KNN Adjusted R^2	CART Adjusted R^2
Bootstrapped stepwise	58	0.53	0.71	**0.75**	0.53	0.55
Bootstrapped Lasso	40	0.45	0.59	0.59	0.42	0.54
Bootstrapped PLS	52	0.50	0.65	0.72	0.45	0.55
Bootstrapped CART	49	0.52	0.53	0.59	0.41	0.54
GA PLS	45	0.51	0.64	0.75	0.45	0.53

Table 3.5. VM prediction results for testing wafers with chronological order

3.6 Conclusion and Future Work

The best VM model was the one created with the operation state approach. The best model, with testing $R^2 = 0.75$, was achieved with BPNN with the following setting in combination with a bootstrapped stepwise variable selection scheme.

Number of nodes in the input layer	**58**
Number of nodes in the hidden layer	**3**
Number of nodes in the output layer	**1**
Over-fit penalty	**0.005**
Number of tours	**20**
Max. iterations	**100**
Converge criterion	**0.00001**

Table 3.6. Parameter setting for NN

The success of virtual metrology prediction models relies on careful selection, outlier isolation, clustering and de-trending of sensor parameters. In practice, engineers can reduce the cost of metrology by utilizing statistical indicators to infer which wafers need metrology measurements rather than selecting the measured wafers based on experience. It takes advantage of classification methods to create a robust virtual inference system. The concept of virtual inference to classify a wafer is based on corresponding tool sensor data into different classes, where in each class we have upper and lower performance bounds. By this classification, an engineer can have an indicator for possible metrology results of the wafers or an indicator can be generated to signal if a wafer needs metrology measurement or not. This can be shown as follows:

1) We first construct the distribution function of wafer metrology from historical wafers.

2) We then bin the wafers into different performance classes based on metrology.

    1) Hard control specs defined by engineers as: good wafers within specs (+/- 3 sigma), bad wafers outside of +/- 3 sigma.

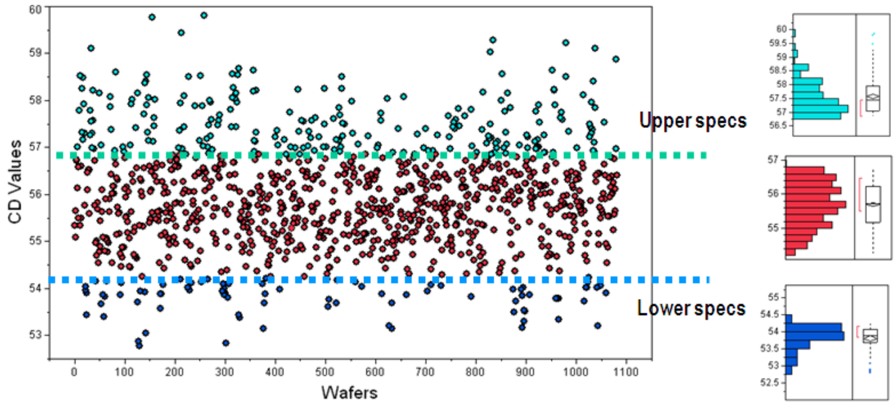


Figure 3.25. Hard specs defined by engineer

    2) Statistical clustering approach: Gaussian mixture modeling is used for demonstration in this case.

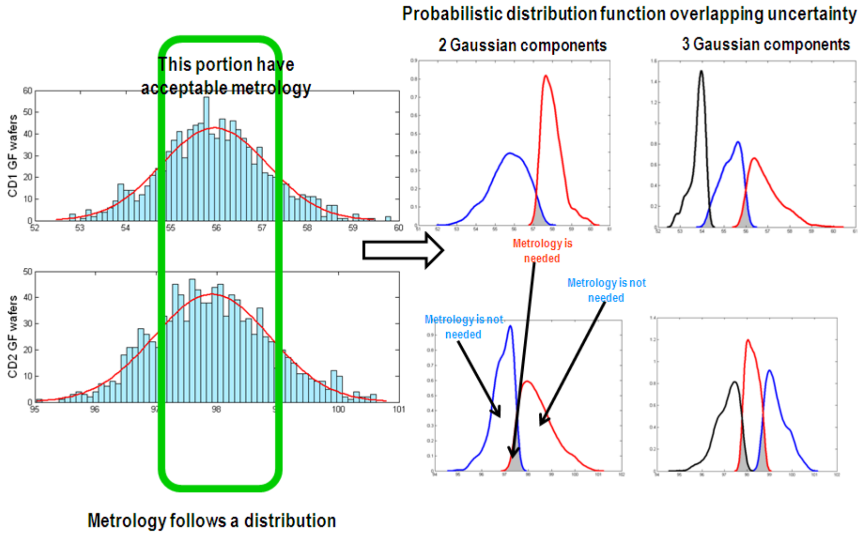


Figure 3.26. Statistical clustering

3) We then train a classification model with the wafer class labels as response and equipment sensor as inputs.

4) We classify the new wafer into different bins (metrology is measured for a wafer only if it is classified in way that makes it ambiguous as to whether the wafers meets or fails the specifications.). We will call those categories "ambiguous bins". VM will prove to be beneficial if only a small subset of wafer will fall into a ambiguous bin and will therefore need to be subjected to actual metrology.

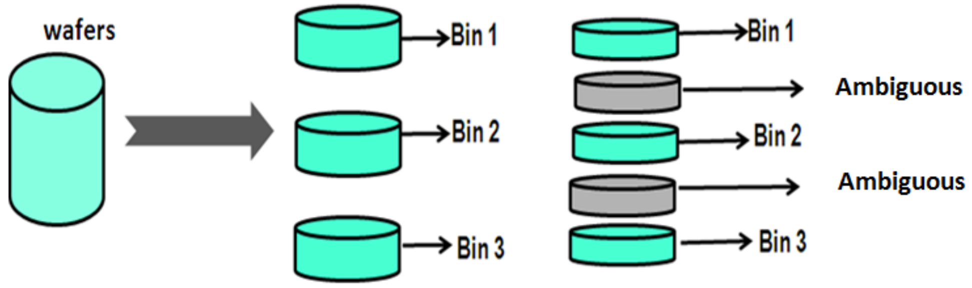


Figure 3.27. Virtual Inference. While wafers actually fit into one of the three bins illustrated above (middle), due to noisy inference, some of them maybe misclassified into the in-between ambiguous bins (right) and therefore require actual metrology.

References

[3.1] Chen, P., Wu, S., Lin, J., Ko, F., Lo, H., Wang, J., et al. (2005). Virtual metrology: A solution for wafer to wafer advanced process control. In *Proceedings of IEEE International Symposium on Semiconductor Manufacturing* (ISSM 2005). San Jose, CA, USA.

[3.2] Besnard, J., &Toprac, A. (2006). Wafer-to-wafer virtual metrology applied to run-to-run control. In *Proceedings of the Third ISMI Symposium on Manufacturing Effectiveness*. Austin, TX, USA.

[3.3] Yung-Cheng, J. C., & Cheng, F.-T. (2005). Application development of virtualmetrology in the semiconductor industry. In *Proceedings of the 32nd AnnualConference of the IEEE Industrial Electronics Society* (IECON 2005). Los Alamitos, CA,USA.

[3.4] Chen, Y.-T., Yang, H.-C., &Cheng, F.-T. (2006). Multivariate simulation assessment for virtual metrology. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (ICRA 2006). Orlando, FL, USA.

[3.5] Lin, T.-H., Hung, M.-H., Lin, R.-C., & Cheng, F.-T. (2006). A virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (ICRA 2006).Orlando, FL, USA.

[3.6] Lieberman, M. A.,and Lichtenberg,A. J. (2005).*Principles of Plasma Discharges and Materials Processing*, 2nd ed., Wiley Interscience, Hoboken, NJ, USA: Wiley.

[3.7] Selwyn,G. S. (1993). *Optical Diagnostic Techniques for Plasma Processing*, New York, NY, USA: AVS Press.

[3.8] Chen,R. (1997). “OES-based Sensing for Plasma Processing in IC Manufacturing” PhD dissertation University of California, Berkeley.

[3.10] Ross, S. M. (2004). *Introduction to Probability and Statistics for Engineers and Scientists*. San Diego, CA, USA: Elsevier Academic Press.

[3.11] Williams, C. K. I.,and Rasmussen, C. E. (1996). Gaussian Processes for regression, chapter8, pp. 514-20. Cambridge, MA, USA: MIT Press.

[3.12] Wouters L., Goehlmann H. W., Bijnens L., Kass S.U., Molenberghs G., and Lewi P.J. (2003). Graphical exploration of gene expression data: a comparative study of three multivariate methods. *Biometrics* 59 (4): 1131-40.

[3.13] Guyon, I. (2003). An introduction to variable and feature selection.*Journal of Machine Learning Research* 3:1157-82

[3.14] Chu, Y.-H., Qin, S. . (2004).Fault detection and operation mode identification based on pattern classification with variable selection. *Ind. Eng. Chem. Res.43* (7): 1701-10.

[3.15] HastiemT., Tibshirani,R., Friedman, J. (2009)..*The Elements of Statistical Learning*,2nd ed. Springer.

[3.16] Duda, R. O., Hart P. E., and Stork, D. G. (2001). *Pattern Classification*. 2nd edition.John Wiley & Sons.

[3.17] Jolliffe, I.T. (1986). *Principal Component Analysis*, New York, NY, USA: Springer-Verlag.

[3.18] Zou,H., HastiemT., and Tibshirani, R. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, in press.

[3.19] Lu, Y., Cohen, I., Zhou, X. S., &Tian, Q. (2007). Feature selection using principal feature analysis. *Proceedings ofthe 15th International Conference on Multimedia*.

[3.20] Montgomery, D. C., Peck, E. A.,&Vining, G. G.. (2001). *Introduction to Linear Regression Analysis.* John Wiley & Sons, Inc.

[3.21] Wold, S. (1994). PLS for Multivariate Linear Modeling QSAR*: Chemometric Methods in Molecular Design*. Ed. H. van de Waterbeemd. Series “Methods and Principles in Medicinal Chemistry.”Verlag-Chemie.

[3.22] Holland, J. H. (1975).*Adaptation in Natural and Artificial Systems,* Ann Arbor, MI: University of Michigan Press.

[3.23] Schmitt, L. M. (2001).Theory of genetic algorithms. *Theoretical Computer Science* 259: 1–61

Chapter 4

Enhanced Metrology for Process Monitoring with site-to-site Metrology Data

While Virtual Metrology, as introduced in the previous chapter performs model-based extrapolation across production time, there is also an interest to perform extrapolation across production space (i.e. across wafers and wafer lots).

In this chapter, we present a statistical framework for creating an FDC model for wafer and die level electrical metrology data. Our work goes beyond tradition SPC methods that focus primarily on change point detection for process monitoring, as we also present methods for optimal measurement site selection for process monitoring and wafer map interpolation. In this manner, engineers can reduce the cost of metrology measurements by measuring selected sites only.

In process monitoring, selecting the most informative sites can provide enough information for process control monitoring, since there are redundant sites due to the existence of highly collinear structures among sites. The goal of process monitoring in this work is to detect wafers with disrupted spatial patterns. By this we mean a wafer containing a large portion of dies having significantly different electrical metrology results from other wafers. Such is the definition of outlier wafer in this work. For wafer map interpolation, it is desirable to measure a few sites and interpolate the entire wafer map for further yield analysis. Tyrone and Kameshwar [4.1] performed a selection of measurement sites for CD metrology after a litho-etch process. In their work, they were able to reduce the number of measurement sites down to 2 of 14, while maintaining acceptable prediction error rates.

In this thesis, we have focused on developing FDC schemes for electrical metrology data and an optimal selection of sites for such data structures. The performance of site selection is evaluated with developed FDC schemes. The electrical metrology data is spatially distributed across all the dies within the same wafer and temporally distributed across wafers. The spatial and temporal variation in the data can provide us insights into the impact of process variation on transistors.

In Section 4.1, we first discuss the nature of the process variation induced by different semiconductor process steps. In section 4.2, we offer a brief discussion of how spatial statistics are used to characterize the spatial variation of wafers. In section 4.3, we explore the data structure of electrical metrology results. In section 4.4, we review the application of formal and graphical statistical methods to detect global and local outliers. Global outliers are measurements deviate that significantly from the rest of measurements across all the wafers, whereas local outliers are ones that have values that deviate significantly from their neighbors within the same wafer. In Section 4.5, we construct FDC models that can capture entire wafer information using transistors’ performances varying spatially and temporally from wafer to wafer. SPC charts can then be produce based on computed statistics and alarm thresholds. Kernel density estimation was applied to estimate the empirical PDF and CDF of similarity factors, after which alarm thresholds were obtained based on defined confidence levels. Section 4.6 then discusses methods for optimal selection of measurement sites for multivariate process control monitoring. The selection sets from different methods are then evaluated using the constructed FDC models discussed in Section 4.5.

4.1. Introduction on Source of Spatial Variation in Semiconductor Process

It has been shown in the literature that metrology results such as physical CDs, electrical performance such as threshold voltage, or the driven current of transistors tend to have spatial structures across wafer. Such spatial structures can be described using statistical quantities such spatial correlation [4.2]. In a sense, transistors that are located close to each other on the same wafers will have more similar performances than ones far away. Consequently, one can fit a deterministic function to describe how the transistor performance depends on its *x* and *y* locations on a wafer [4.3].

The spatial variation of transistor performance is introduced by the nature of the fabrication processes, along with pattern dependent effects. Most semiconductor processes exhibit spatially revolved variation, where the edges of wafers tend to result in different conditions after processes. For example, the lithography step can introduce overlay error, which includes errors in the position and rotation of the wafer stage during exposure, wafer stage vibration and the distortion of the wafer with respect to the exposure pattern. Magnification and rotation components of overlay error increase from the center of the wafer outwards [4.4]. For the CVD step, species depletion and temperature non-uniformity on the wafer at lower temperatures may cause thickness non-uniformity. In plasma etching operations, the center peak shape of RF electric field distribution also leads to a center peak shape of etch rate, while chamber wall conditions also cause etch rate non-uniformity. Etch rate also varies radially across the wafer; it is higher at the center and decreases toward the edges. In figure 4.1, we show two wafer CD maps after plasma etching, where we can clearly see the radial shape for CD variation across wafers.

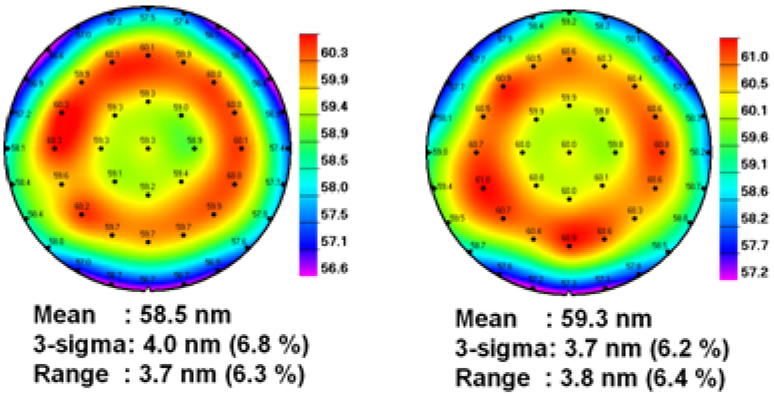


Figure 4.1 Illustration of CD map after plasma etching [Courtesy of Lam Research Corporation]

All these process effects are convoluted by the fabrication flow and impact the final transistor performance spatially along with the temporal variation introduced by process drifting. The effects of process variation can also be decomposed hierarchically: Lot-to-Lot, Wafer-to-Wafer, Die-to-Die and within-Die. Lot-to-lot refers to process variation existing in different lots.

Wafer-to-wafer process variations are those affecting different wafers within a lot. Die-to-die means variation in different dies within a wafer. Within-die denotes variation within an identical device or circuit within a die. This hierarchical decomposition is shown in figure 4.2.

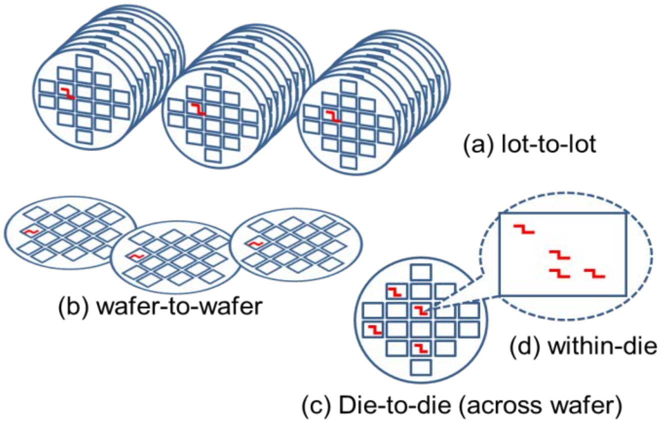


Figure 4.2 Hierarchical decomposition of process variation [4.5]

Given strong spatial correlation for metrology results observed across the wafer, it is desirable to just measure the most informative sites that can capture process variation information. The information contained by these sites can then be fed into a Fault Detection / Characterization (FDC) model for process monitoring [5.1]. Thus, the problem becomes that of selecting the right sites and interpolation methods.

In this section, we utilize variable selections schemes to perform optimal selection of metrology sites and evaluate the performance of different selections for FDC and interpolation purpose. The objective of optimal metrology selection is shown in figure 4.3. In this work, the selected sites from different methods are evaluated using the developed FDC schemes. The performance results are compared against results in which all the sites are measured.

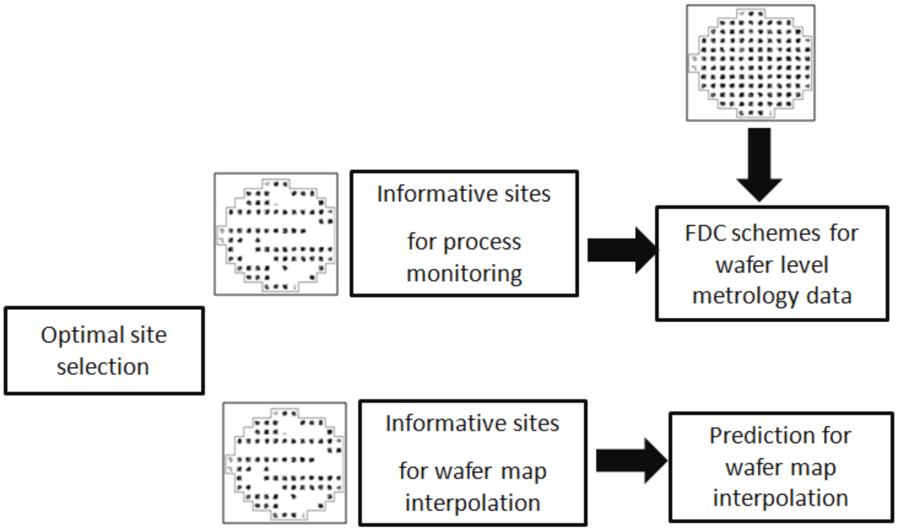


Figure 4.3 Objective of site selection

4.2. Spatial Statistics and Kriging

Given the nature of process variations, it is common to observe a radial spatial pattern across wafer. Given the assumption that electrical metrology data is normally distributed and stationary (mean and variance do not vary significantly in space), we can characterize each wafer using a spatial variogram and correlogram [4.6] based on the spatial variation across different performance parameters. Normality of data can be checked using a histogram, graphically with a normal probability plot, or with hypothesis testing.

A spatial variogram characterizes the spatial smoothness and pattern of a dataset; it is defined as the expected squared difference of attribute values between two locations [4.7]. Given the assumption of data is coming from a weak stationary process, the relationship between the values of the process at any two locations can be summarized by a covariance function $C(\mathrm{h})$, and this function depends only on the separation vector h. Then we can express the variogram in the form of equation 4.1:

$$\begin{aligned} E[y(u+h)-y(u)]^2 &= Var\big(y(u+h)-y(u)\big) = 2\gamma(h) \\ 2\gamma(h) &= Var\big(y(u+h)\big) + Var\big(y(u)\big) - 2Cov\big(y(u+h), y(u)\big) \\ &= C(0) + C(0) - 2C(h) \\ &= 2[C(0) - C(h)] \end{aligned} \tag{4.1}$$

The $2\gamma(h)$ is called the variogram and $\gamma(h)$ is called the semivariogram. u is the vector of spatial coordinates, and $y(u)$ and $y(u+h)$ are the associated attribute value separated by vector **h**. The empirical estimation of variogram is shown in Equation 4.2:

$$\hat{\gamma}(h) = \frac{1}{2|n(h)|}\sum_{\alpha=1}^{n(h)}[y(u_\alpha + h) - y(u_\alpha)]^2 \tag{4.2}$$

$n(h)$ is the set of pairs of sample points such that the distance between them is h, and $|n(h)|$ is the number of pairs in this set. An empirical semivariogram estimate can be plotted and an appropriately shaped theoretical variogram model can be fitted to the data.

Similarly, robust estimators of variograms were also introduced by Cressie and Hawkins (1983) [4.8]. They estimate the variogram in two alternative ways:

$$\hat{\gamma}(h) = \frac{\left\{\frac{1}{|N(h)|}\sum_{\alpha=1}^{N(h)}[Y(u_\alpha + h) - Y(u_\alpha)]^{1/2}\right\}^4}{2\left(0.457 + \frac{0.494}{|N(h)|}\right)} \tag{4.3}$$

and

$$\hat{\gamma}(h) = \frac{\left| med \left\{ |Y(u_\alpha + h) - Y(u_\alpha)|^{1/2} : \alpha \in N(h) \right\} \right|^4}{2\left(0.457 + \frac{0.494}{|N(h)|}\right)} \tag{4.4}$$

The spatial correlogram employs covariance between values at two different locations for empirical estimation. The covariance function $C(h)$ is empirically estimated as follows:

$$C(h) = \frac{1}{N(h)} \sum_{\alpha=1}^{N(h)} Y(u_\alpha) \cdot Y(u_\alpha + h) - m_0 \cdot m_{+h} \tag{4.5}$$

where m_0 and m_{+h} is the means estimated as:

$$m_0 = \frac{1}{N(h)} \sum_{\alpha=1}^{N(h)} Y(u_\alpha) \tag{4.6}$$

$$m_{+h} = \frac{1}{N(h)} \sum_{\alpha=1}^{N(h)} Y(u_\alpha + h) \tag{4.7}$$

The spatial correlogram is computed as:

$$\rho(h) = \frac{C(h)}{\sqrt{\sigma_0^2 \cdot \sigma_{+h}^2}} \tag{4.8}$$

where σ_0^2 and σ_{+h}^2 are the estimated corresponding variances:

$$\sigma_0^2 = \frac{1}{N(h)} \sum_{\alpha=1}^{N(h)} [Y(u_\alpha) - m_0]^2 \tag{4.9}$$

$$\sigma_{+h}^2 = \frac{1}{N(h)} \sum_{\alpha=1}^{N(h)} [Y(u_\alpha + h) - m_{+h}]^2 \tag{4.10}$$

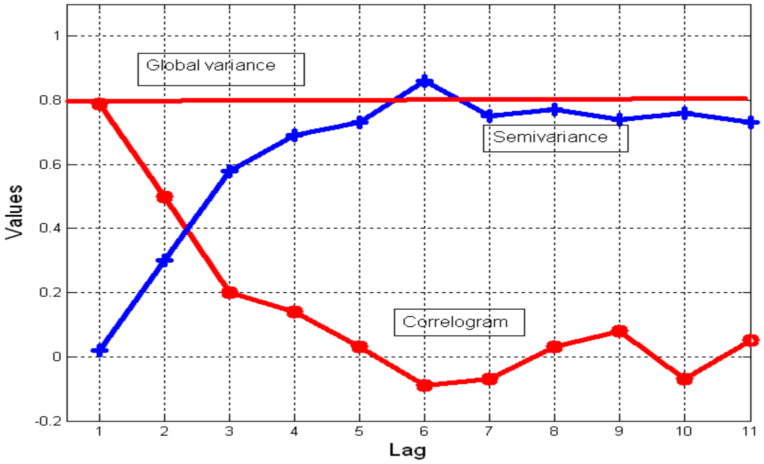


Figure 4.4 Spatial correlogram and semivariogram difference

Spatial correlograms measure the similarity at different locations while a spatial variogram is a measure of dissimilarity. Figure 4.4 shows the statistics of a correlogram and semivariogram for same dataset. For time series analysis, it is preferable to work with correlograms, while in spatial statistical analysis, variograms are to be preferred. This is due to the fact that a variogram averages the squared difference of the spatial attribute and tends to filter the influence of a spatially varying mean.

In our work, we have assumed that spatial correlation structure is isotropic, that is, the same in all directions. By that, we mean the correlogram and variogram depend on the magnitude of the lag vector and not the direction, and the empirical variogram or correlogram can be computed by pooling data pairs separated by the specific lag distances, regardless of direction. The applications of spatial variograms and correlograms to disrupted wafer detection are discussed in Sections 4.4 and 4.5.

4.3. Site-to-Site Metrology Data Description

The data we used consist of measurements from 348 wafers that span 23 lots as shown in figure 4.5, with an uneven number of wafers for each lot. Each wafer contains 117 dies, and the reticle field consists of only one die. Each die contains 14 values, which are frequency measurements of ring oscillators (PSROs) at different locations within the die. There are some missing dies for most of the wafers. It is necessary to impute such missing measurements prior to analysis.

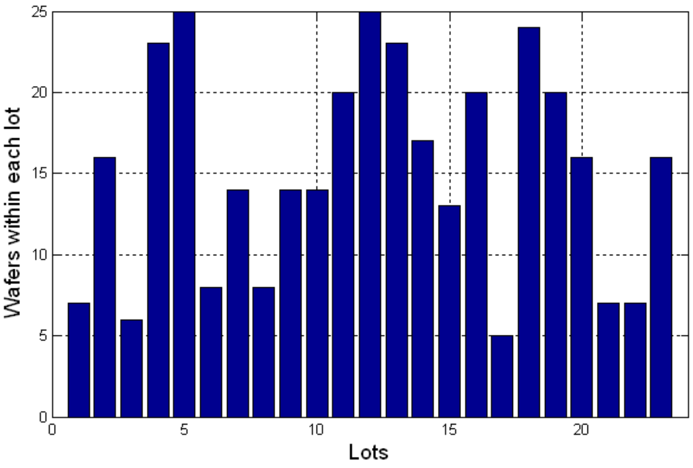


Figure 4.5 Wafer allocations for lot

There are many methods for imputing missing values in the literature. Singular value decomposition (SVD) and k-nearest neighbors (KNN) imputation methods are two popular ones. SVD imputation incorporates an EM algorithm and PCA concept, because SVD is identical to standard PCA when applied to a matrix normalized so that row-wise mean is zero [4.9]. For KNN imputation, the missing value is estimated as the average of the corresponding entries in the selected k expression vectors. The selected k vectors are obtained by computing similarity between two vectors with their non-missing values [4.9]. In this work, we have used Kriging to interpolate the missing values for each wafer prior to further analysis. Then a set of wafers with "complete" measurements are used for optimal metrology sampling.

In figure 4.6, we show mosaic plots of a wafer before and after wafer-to-wafer and lot-to-lot mean removals. The left plot shows raw data, while the right shows the data after wafer-level and lot-level mean removal for every measurement. We can observe the radial variation of PSROs across wafers.

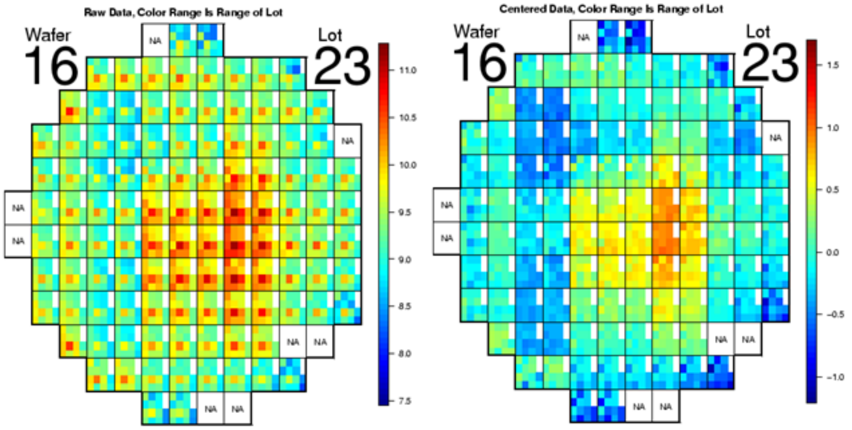


Figure 4.6 Mosaic plot of wafer#16 in lot#23, showing raw (left) and lot-to-lot centered data (right).

In figure 4.6, we can also see that dies share similar performances with their neighbors. Such spatial correlation has been discussed extensively in the literature. It is suggested that spatial correlation is largely contributed by the systematic variation from wafer-to-wafer and lot-to-lot. In figure 4.7, we show correlation of a specific die with the rest of dies across the wafers before and after wafer-to-wafer and lot-to-lot mean removals. The correlation statistics are computed as Pearson linear correlation coefficients. Rather than showing spatial correlation for all the PSROs, we only the show the correlation of PSRO_00 for better visualization purposes. PSRO_00 indicates the measured ring oscillator frequency at a specific location on a die.

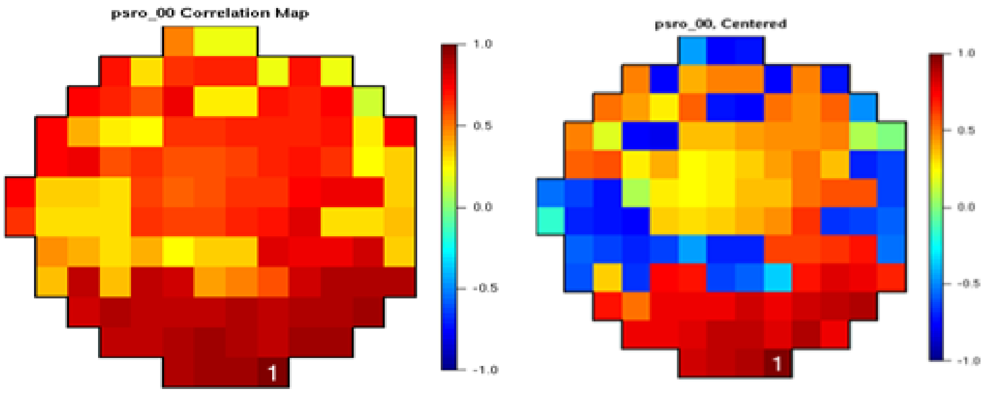


Figure 4.7 Spatial correlation heat map before and after mean removal

We can observe that spatial correlation is weakened after wafer-level and lot-level mean removal. In figure 4.8, we observe a trend in the value of PSRO_00 from lot-to-lot in the left plot, while the systematic variation disappears after the mean removal. We first performed K-means clustering with k=4 to cluster all the PSRO_00s into 4 groups. Then we visualized the positions of these PSRO_00 across wafers and lots with histogram plot. ChipX and ChipY are the x and y coordinates of the PSRO_00 on wafer. The PSRO_00 with higher values are coded with green color and also highlighted in the histogram plot. It tells us there is no spatial dependency for the higher values of PSRO_00.

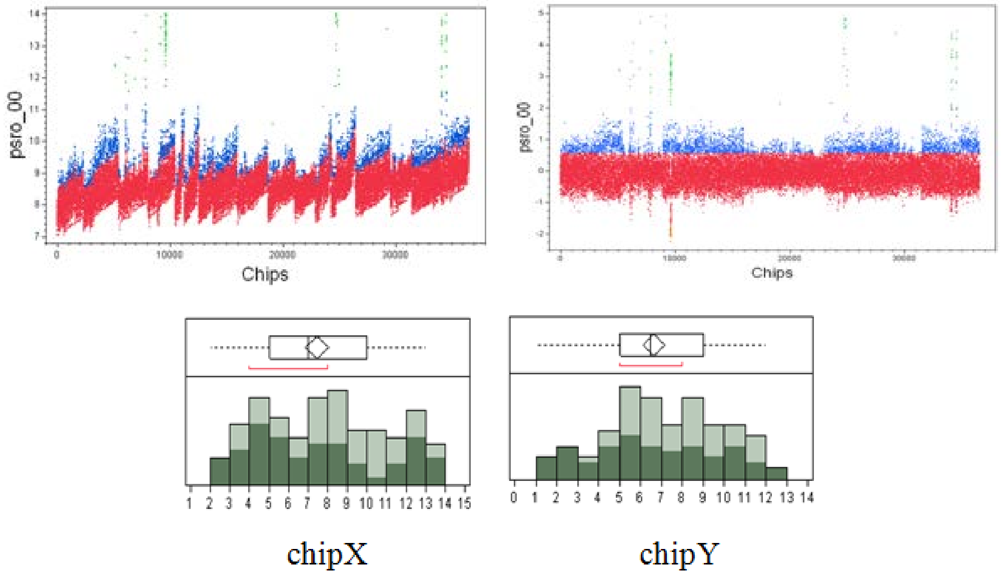


Figure 4.8 PSRO_00 values for each die across wafer and lot after and before mean removal.

Nassif [4.10] has also studied this dataset; They assumed that the underlying distribution of the data is a multivariate normal distribution (MVN), and then constructed a matrix of *n*-by-*(d*x*r*), where *n* is the total number of wafers, d is the number of dies per wafer and *r* is the number of measurements per die. The EM algorithm was then applied with objective of estimating the μ and $\sum$.

This first initializes the MLEs of distribution parameters μ and $\sum$ from complete data, and then repeats the following steps until convergence:

a) E-step: estimate the expected value of the missing measurements, given the current MLEs

b) M-step: given the expected estimates of the missing measurements, re-estimate the distribution parameters to maximize the likelihood of the data

This method makes use of the entire dataset for missing values imputation. The objective of their work was to estimate the missing measurement with adequate confidence level for the estimation. Our objective is to optimally select measurement sites for process monitoring purposes. Given that our data consists of electrical metrology results, each site refers to a single die in this work. Prior to further analysis, we first labeled the sites in a spiral order shape as shown in figure 4.9, such that edge sites are assigned a lower index while center sites are assigned a higher index. The spiral ordering also makes it easy to visualize the radial spatial pattern across the wafer.

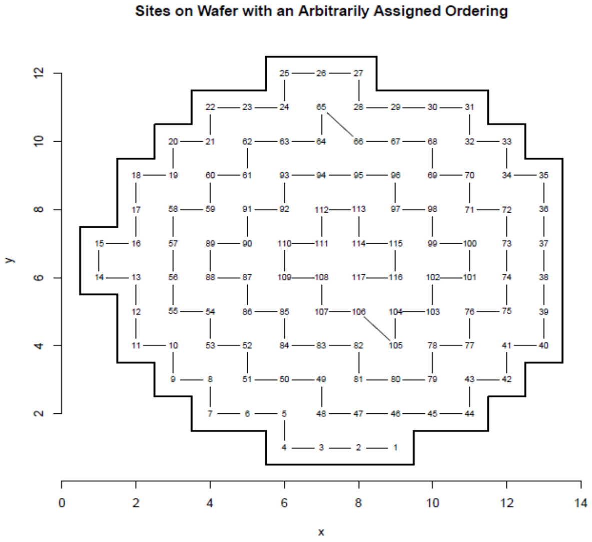


Figure 4.9 Ordering of site on wafer

4.4. Outlier Wafer Detection

Outlier detection based on multivariate space has been done in on CDs for wafers after plasma etching. However, the hierarchical structure of wafer level metrology data makes it harder to apply such multivariate detection methods where the spatial information can be lost. In this section, we first review the common spatial outlier detection methods, and then discuss the application of a neighborhood based outlier detection scheme and MCD Mahalanobis distance to detect outlier wafers heuristically.

4.4.1. Spatial outlier (local outlier)

Spatial outliers represent locations that are significantly different from their neighborhoods even though they may not be significantly different from the entire population. Spatial outlier detection in the literature of spatial statistics can be grouped into two categories, the graphic approach and quantitative tests.

Graphic approaches are based on visualization of spatial data that highlights spatial outliers. Variogram cloud is one example [4.11]. A variogram cloud is an extension of the spatial variogram model designed for spatial outlier visualization; it displays data points related by neighborhood relationships. For each pair of locations, the square root of the absolute difference between attribute values at the locations is plotted against the Euclidean distance between the locations. In data sets exhibiting strong spatial dependence, the variance in the attribute differences will increase with increasing distance between locations. Locations that are near to one another, but have large attribute differences, can indicate a spatial outlier, even though the values at both locations may appear to be reasonable when the dataset is examined spatially. In figure 4.10, we show the variogram cloud for a single wafer on the left, with the corresponding wafer-level heat map shown on the right.

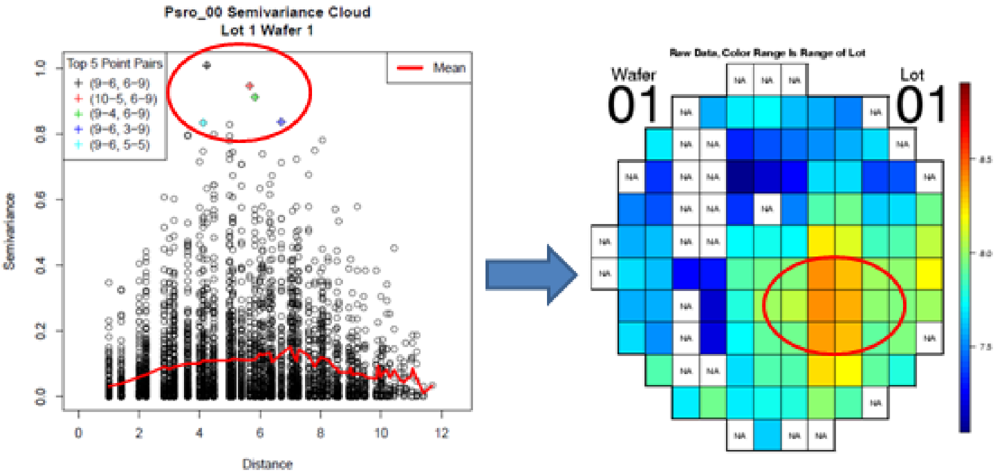
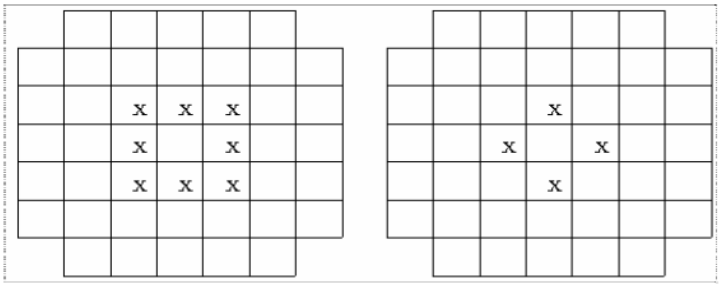


Figure 4.10 Variogram cloud for wafer 01 from lot 1, showing potential outliers.

The outliers based on variogram cloud are the pairs of locations that have higher semivariance. As we show in figure 4.10, we can see that the dies at the center tend to have higher PSRO values, significantly different from other dies. However, it is relatively hard to determine the threshold for outliers using the variogram cloud and it requires postprocessing of the results.

Quantitative methods provide tests to distinguish spatial outliers from the remainder of data. The Moran scatter plot is one example [4.12]. A Moran scatterplot is a plot of normalized attribute values against the neighborhood average of normalized attribute values. A scatter plot shows the attribute values on the X-axis and the average of the attribute values in the neighborhood on the Y-axis. A least square fit can be used to detect spatial outliers. A scatter sloping upward to the right indicates a positive spatial autocorrelation (adjacent values tend to be similar), and a scatter sloping upward to the left indicates a negative spatial autocorrelation. The residual is defined as the vertical distance from a point P with location (x, y) to the regression line. In this work, we have applied the King's move for neighborhood. King's move and Rook's move are the two common neighborhood definitions used in spatial statistics [4.13]. Both moves are shown in figure 4.11 below.



(a) King-move neighbors (b) Rook-move neighbors

Neighborhood construction rules

Figure 4.11 King's and Rook's moves.

We illustrate the concept of local outlier detection with Moran I scatterplot [4.14]. In the left plot of figure 4.12, we show the fit of neighbor average vs. attribute value at each die. The normalized residual for each wafer is shown in the right plot. Any die with a normalized residual value higher than 3 is considered an outlier, assuming that normalized residual follows a standard distribution.

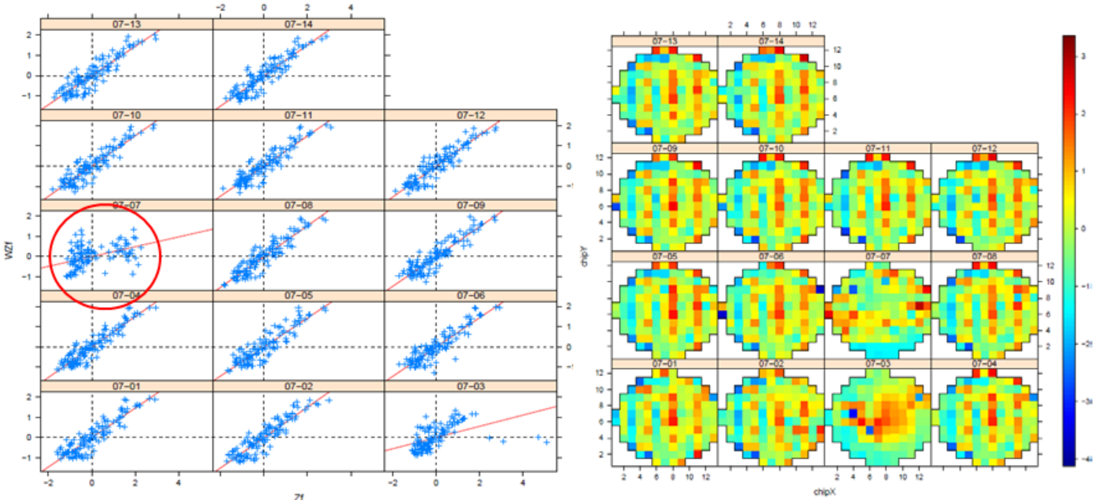


Figure 4.12 Wafers from Lot 7 using Moran scatter plot.

It is also clear that wafer 07 and wafer 03 do not have smooth spatial patterns across the wafer based on the scatter plot in figure 4.10. This is confirmed by the mosaic plot of the wafers we show in figure 4.13. Thus one can detect an outlier wafer with Moran's I scatter plot by capturing how good the least square fit is for each wafer. A fit resulting in lower R^2, like the circled one in figure 4.12 clearly indicates there is lack of strong spatial autocorrelation across the wafer, and thus a disrupted spatial pattern is expected.

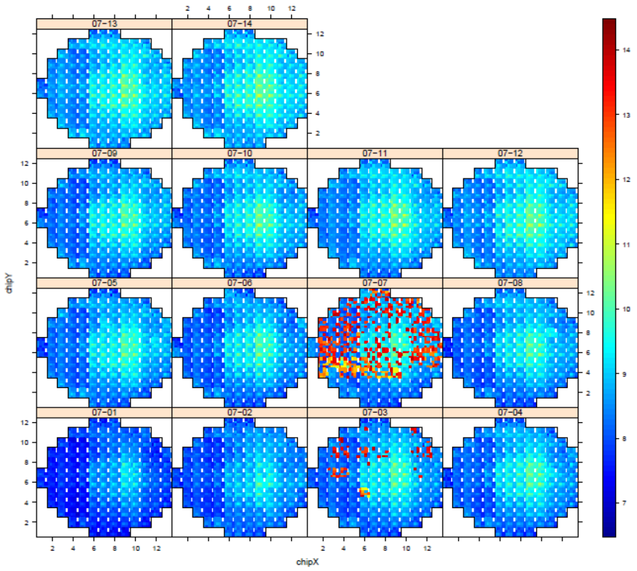


Figure 4.13 Mosaic plots of wafers from Lot 7

The major drawback of existing detection approaches is that their application requires extensive post processing and will lead to some true spatial outliers being ignored and some false outliers being identified. We utilized an algorithm that employs the median as the neighborhood function, thus reducing the negative impact caused by the presence of neighboring points with very high/low attribute values.

1. Given a set of spatial points $X = \{x_1, x_2, \ldots, x_n\}$ in a space with dimension $p \geq 1$, an attribute function F is defined as a mapping from X to R .
2. Let $NN_K(x_i)$ denote the k nearest neighbors of point x_i , where $g(x_i)$ returns summary statistics for attribute values of all the spatial points inside $NN_K(x_i)$.
3. To detect spatial outliers, we compare the attribute values of each point x_i with the attribute values of its $NN_K(x_i)$.
4. Comparison is then made in the form of a comparison of h, which is a function of f and g. h can be difference or ratios of f and g. In this work, h is defined as the ratio f/g.
5. g is median of defined neighbors and f is the median of 14 PSROs at X.

Once $h(x_i)$ is computed for every die within a specific wafer, we can then scale $h(x_i)$ within the same wafer so it follows a univariate normal distribution. PDF for univariate normal distribution is:

$$f(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{\frac{-[(x-\mu)/\sigma]^2}{2}} \tag{4.11}$$

$f(x)$>0, and its integral is equal to 1. The probability that a data point lies beyond 3 sigmas from the mean is just 0.0027. Thus, 3-sigma is often the cut-off threshold [4.15]. If a die within a wafer has $h(x_i)$ higher than 3, we can flag it as an outlier for the specific wafer. Both King's and

Rook’s move based neighborhood construction rules were studied in this work. They tend to provide similar results for this data set.

In figure 4.14, we again used Lot 7 wafers for illustration. Figure 4.12 shows the mosaic plot of $g(x_i)$ for all Lot 7 wafers. We can observe the smoothing effect introduced by this method. If the wafer contains many extreme outlier dies, that indicates the spatial pattern across wafer is not smooth and we can flag it as an outlier.

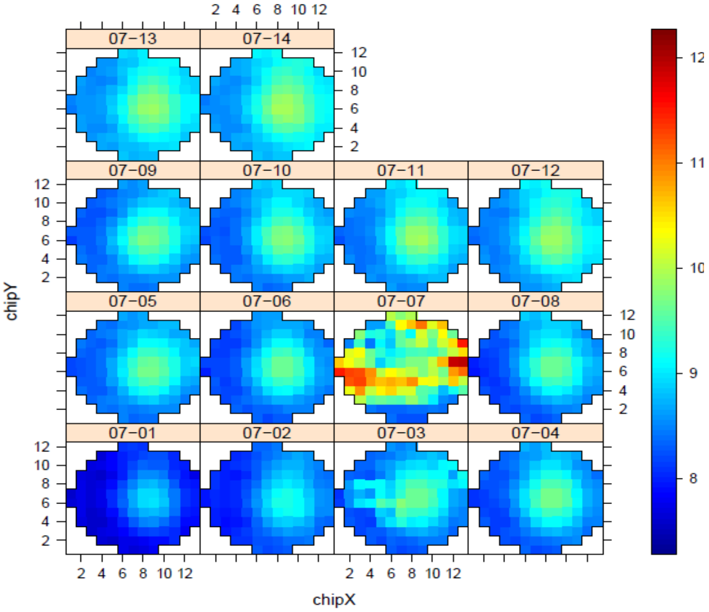


Figure 4.14 King’s move based G computed for Lot 7 wafers

In figure 4.15, we show the results of computing $h(x_i)$ using both neighborhood construction rules. It appears that they yield very similar results in terms of outlier detection where the number of dies with Z-score values outside of +/-3 boundary are very close: wafer 07 and wafer 03.

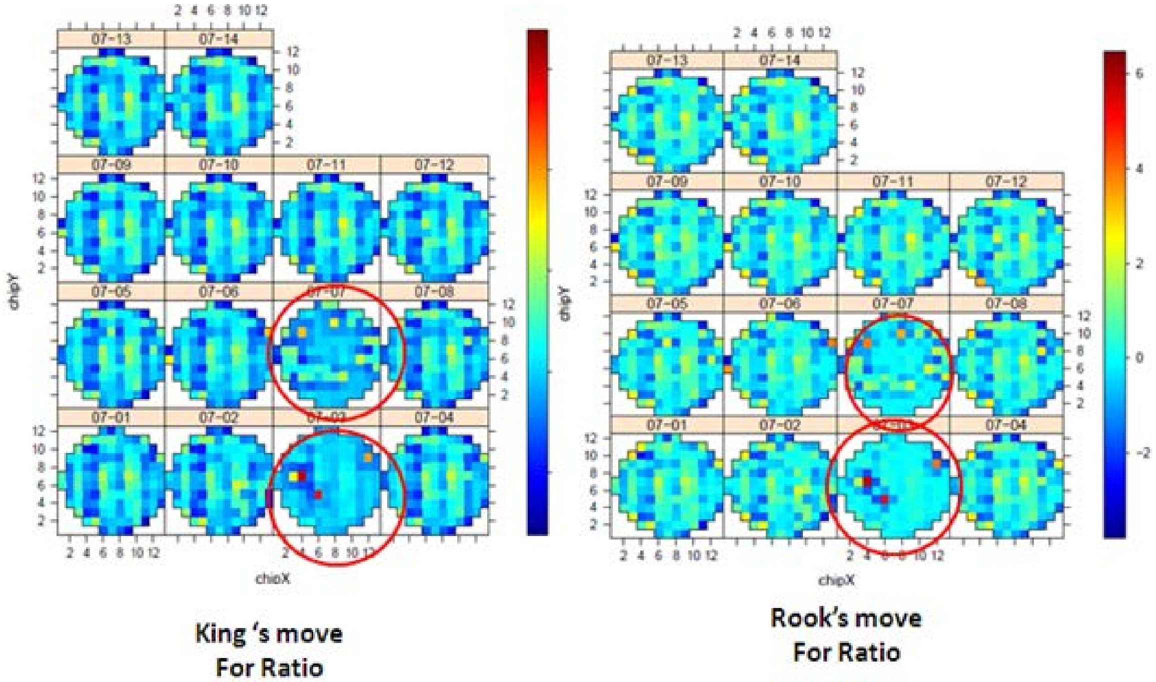


Figure 4.15 King's move and Rook's move for neighborhood based outlier detection

The neighborhood approach is affected by choice of the size of neighborhood as well as the selection of appropriate H and G functions. For application on this set of data, this approach is not robust enough, as shown in figure 4.12.

4.4.2. MCD based Multivariate Mahalanobis distance for Global outliers

The spatial outlier detection methods we have discussed do not take into account all 14 PSROs. They require first computing the statistical summary for each die, and then using these statistics for further analysis. To use all 14 PSROs' information, we adopted the MCD based Mahalanobis distance (MD) [4.16] for outlier wafer detection in this case. MD is a preferred distance metric when multivariate normal distributions are expected in the data. In this section, we illustrate the application of this concept for detecting outlier wafers and possible grouping of dies based on their similar performance. For outlier wafer detection, the idea is to observe which wafer contributes the most number of outlier dies, inferring that this indicates the wafer has disrupted spatial patterns. The steps are described below:

1) A specific die within a wafer is represented by a 1-by-14 dimension vector, since each die has 14 PSROs. Given that there are 348 wafers available, we have a 348-by-14 matrix for each specific die.
2) We compute the Mahalanobis distance for each wafer; each wafer is a 1x14 vector.
3) Statistical threshold is set to define outliers, given that the Mahalanobis distance follows χ^2distribution. The statistical threshold used in this work is the 0.99 quantile of $log\sqrt{\chi^2_{14}}$.
4) If a wafer has distance value above the threshold, we assign a "1" otherwise "0."
5) We then go through all the dies, and obtain a frequency count of being outlier or not for each wafer.

Results for the die at location (9, 6) are shown in figure 4.16. About 10 wafers of the 348 are considered outliers.

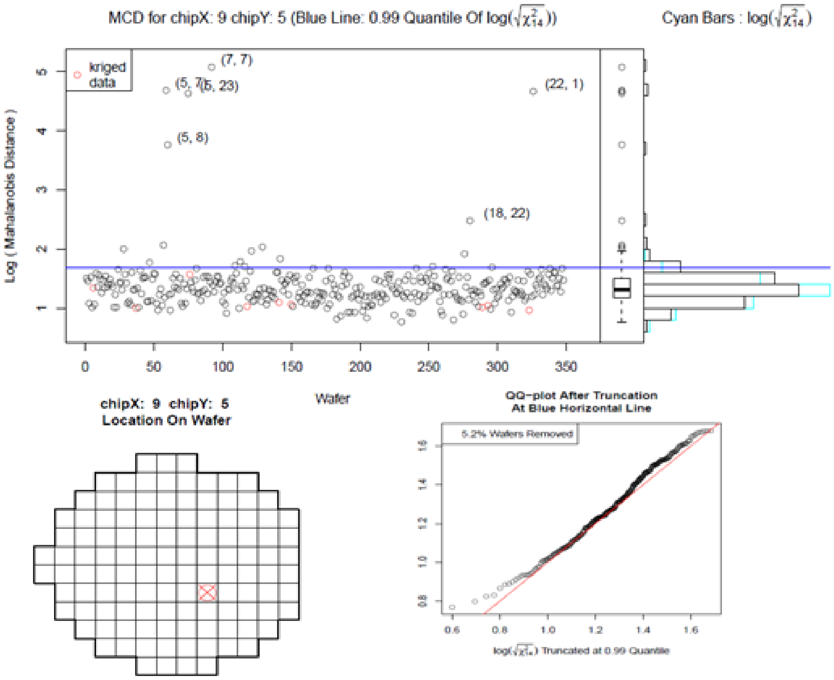


Figure 4.16 Mahalanobis distance for outlier wafer detection

We then show the frequency plot in figure 4.17, along with corresponding mosaic plots for wafers with high numbers of “1.” It is clear that the wafers with higher frequencies have disrupted spatial patterns. Therefore, this type of chart can be applied in practice for detecting abnormal wafers.

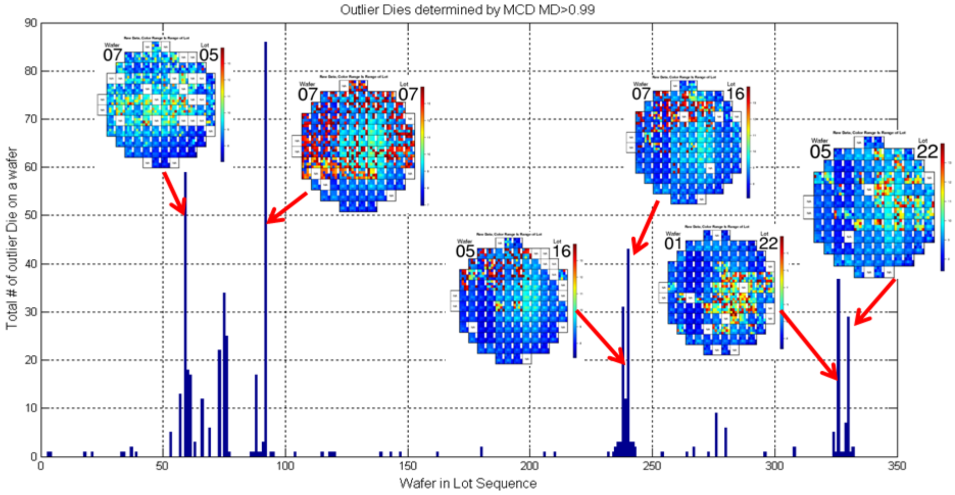


Figure 4.17 Frequency plot for disrupted wafers

There is also the possibility of die grouping, as described below:

1) For a specific die within a wafer, the die is represented by a 1-by-14 dimension vector, given each die has 14 PSROs. Therefore, each wafer is a matrix of 117-by-14, as there are 117 dies per wafer.
2) We compute the Mahalanobis distance for each die within the same wafer.
3) Statistical threshold is set to define outliers given that Mahalanobis distance follows χ^2distribution. The statistical threshold used in this work is the 0.99 quantile of $log\sqrt{\chi^2_{14}}$.
4) If a die is above the threshold, we assign a "1" otherwise "0."
5) We then go through all the wafers, and obtain a frequency count of outliers for each die.

We then show the frequency plot in figure 4.18. The frequency plot tells us the dies at the edge and center tend to have extreme values compared to the rest of dies across all wafers.

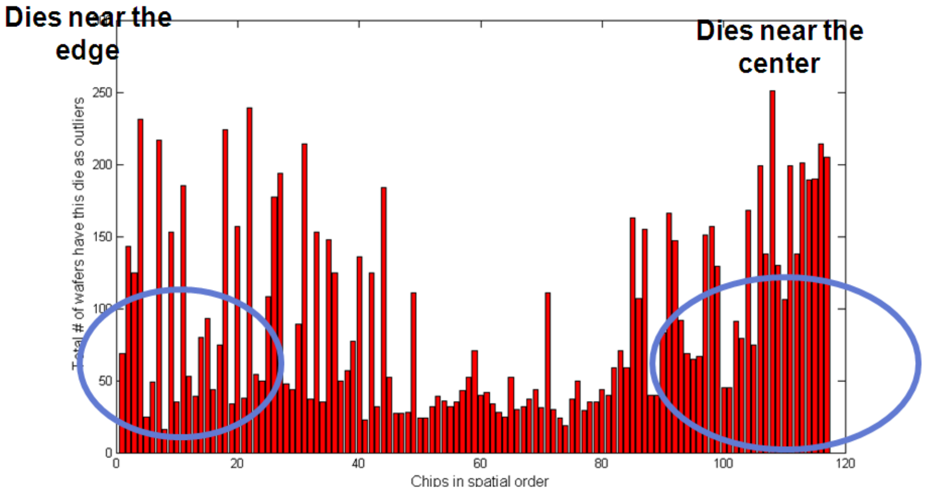


Figure 4.18 Frequency counts for dies.

The results in figure 4.18 suggest that there is a common main radial pattern across wafers. This can be validated by performing PCA for die-to-die decomposition. The concept is to observe die-to-die variation by representing each die with a 1-by-348 dimensional vector. In this work, we replace the 14 PSROs of the die with median PSROs. Then PCA is performed on a 117-by-348 matrix. The first 4 scores are shown in Figure 4.19 below, where one main PC dominates the variation, based on the scree plot. The cyclical pattern on PC2, PC3 and PC4 indicates the existence of a radial pattern for performance variation.

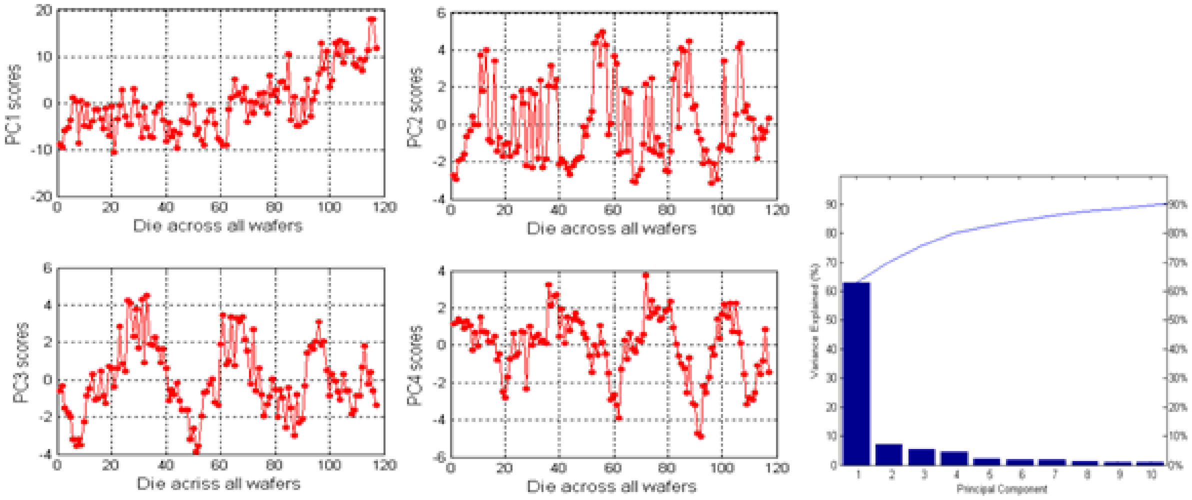


Figure 4.19 Die-to-Die PCA results.

In figure 4.20, we then overlay the scores onto a wafer map to observe how the scores vary radially. It is clear that the bulk of variation is contributed by dies at the center.

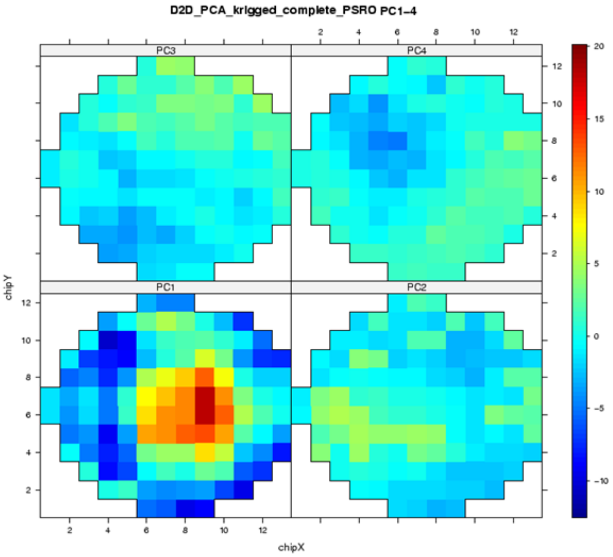


Figure 4.20 Scores for Die-to-Die PCA

One can also combine all the discussed methods together to provide robust outlier wafer detection, where the wafer contains the most number of outlier dies defined by all 4 methods: MCD MD by wafer, MCD MD by die, neighborhood based Z-score and Moran standardized residuals. We again use lot 7 wafers for illustration, as shown in figure 4.21. The dies denoted as outliers by each of the 4 methods are marked with different signs across the wafer in figure 4.21. The combined results easily identify the outlier wafers in this case. Wafer 07 and Wafer 03 from lot 7 are identified to be outlier wafers here.

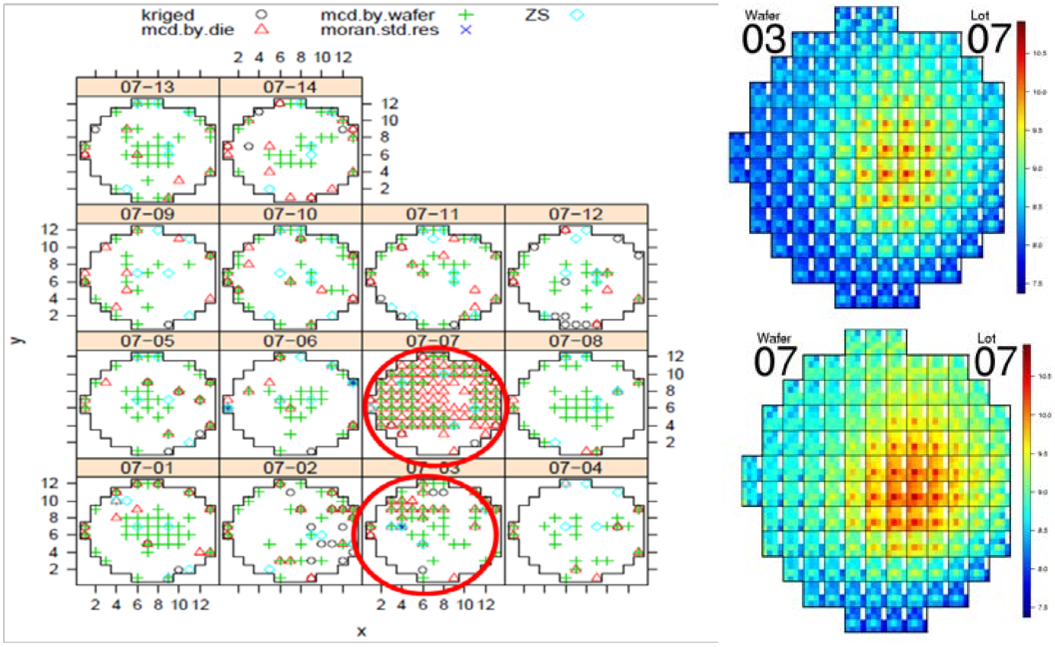


Figure 4.21 Combination of results from 4 methods

4.5. Multivariate Similarity Index based Wafer-level FDC

Given that there are usually multiple chips to test on the same die and hundreds of dies on the same wafer, multivariate statistics can provide great insight into wafer conditions. In the following sections, we present techniques utilizing PCA, ICA and MD to detect outlier wafers.

4.5.1. Extraction of Similarity Index

In section 4.4, we discussed heuristic ways to detect outliers based on visualizations. Most of these cannot be used as monitoring statistics to signal alarms. In this section, we develop a wafer-to-wafer similarity index based on multivariate statistics and empirically estimate its probability density distribution (PDF) using a kernel method. The similarity index must also be able to capture all 14 PSROs information rather than using a compressed statistical summary for each die. There are two statistics we want to employ for FDC: similarity for all wafers vs. reference wafer, and consecutive wafer-to-wafer similarity. In this work, we have computed a "Median" wafer as a reference wafer:

$$W_{i\in 117} = Median\left(w_{ij}, j \in 348\right) \tag{4.12}$$

Each die on the "Median" wafer is attributed the median statistics of that specific die across all 348 wafers. Kernel density estimator is then applied to estimate the empirical PDF and CDF of the computed similarity index, and then a statistical threshold is set based on corresponding probability extracted from a CDF plot. The threshold for alarm in this work is defined by the 95% quantile of the distribution of computed statistics.

The first similarity index we compute is based on PCA [4.17]. The concept is to represent each wafer with a 117-by-14 matrix, since each wafer has 117 dies and each die has 14PSROs.

An individual PCA model is created for each wafer. Then a monitoring statistic can be computed to track the how the PCA subspace for each wafer varies in temporal space. The concept is illustrated in figure 4.22 below where 2 PSROs are plotted as a scatterplot for two different wafers. We can observe that these two wafers have very similar spatial patterns, and thus that the spatial orientations of PSROs for each wafer are very similar. The spatial orientation can be extracted from PCA subspace, since the PCs indicate the main direction of variations. Thus, this similarity statistic can be applied to track spatial orientation of how the covariance structure among PSROs varies from wafer to wafer.

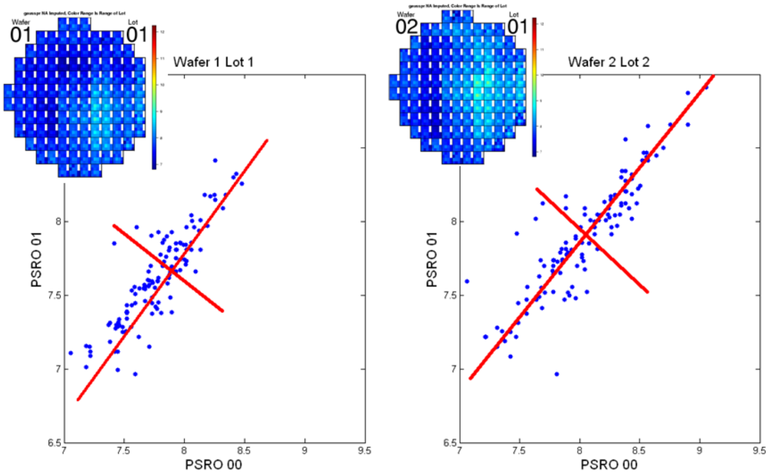


Figure 4.22 Scatterplot of PSR_00 vs. PSR_01 for 2 wafers.

The computation steps for the PCA based similarity index are described below:

1) Each wafer is represented by a 117-to-14 matrix and scaled by the reference wafer.
2) PCA is performed on each wafer matrix.
3) First 3 PCs retained (explained above 90% variance).
4) Compute the similarity matrix as shown in Equation 4.12:

$$S_{PCA}^{\lambda} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\left(\lambda_i^1\lambda_j^2\right)\cos^2\theta_{ij}}{\sum_{j=1}^{k}\left(\lambda_i^1\lambda_j^2\right)} \tag{4.12}$$

The θ_{ij} is the angle between PC_i from wafer $\boldsymbol{i}$ and PC_j from wafer $\boldsymbol{j}$. λ_i^1 is the corresponding eigen-value of PC_i and λ_j^2 is the corresponding eigen-value of PC_j. By including the eigen-value, we weight different subspace of the PCA model. The results of similarity for all wafers vs. reference wafer are shown in figure 4.23.

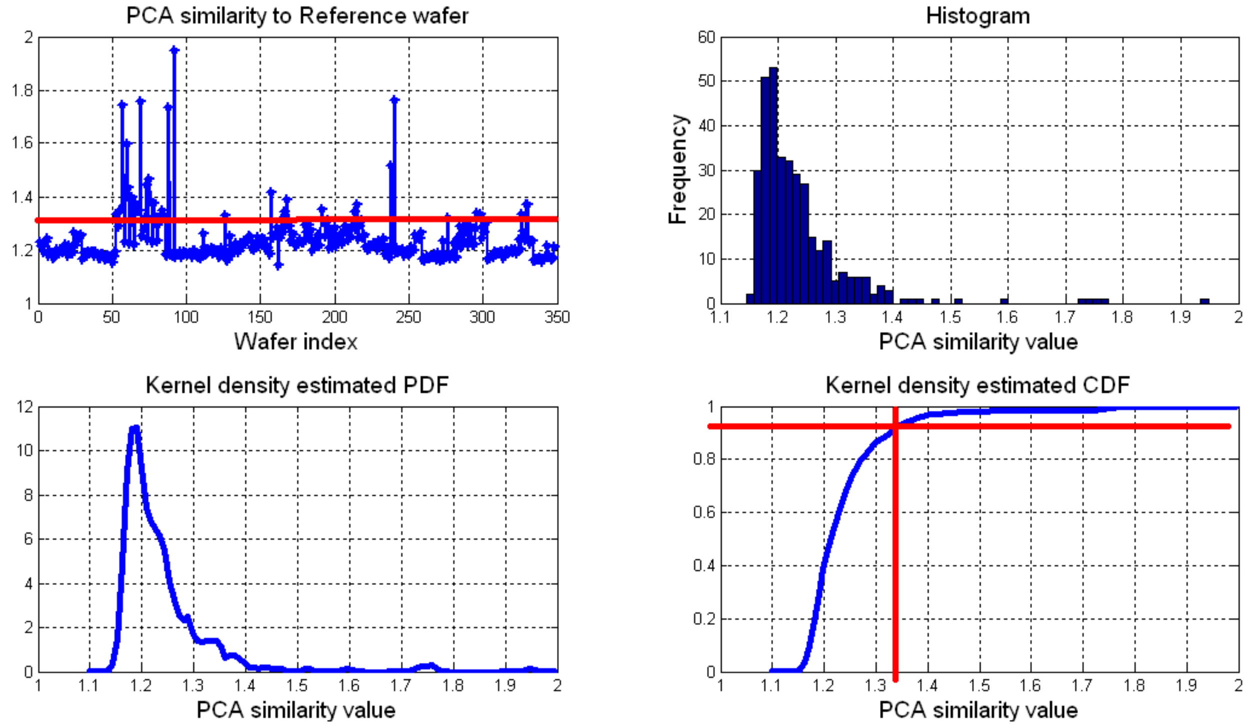


Figure 4.23 Similarity index from PCA

The distribution of the similarity index is estimated using all samples, and one can set a tighter alarm by using only a few training samples. Such training samples can be set in chronological order, or using random sampling.

The second similarity index we compute is based on ICA combined with PCA [4.18]. There are two variants to this method. The first is to perform PCA on the residual matrix after ICA decomposition, since ICA decomposition can remove the Non-Gaussian components.

$$X = A\hat{S} + E \tag{4.13}$$

$$E = TP^T + F \tag{4.14}$$

where $A\hat{S}$ is the reconstructed component of the original, X and E is the residual. The similarity statistics can then be computed in the same way shown in equation 4.12. We demonstrate this concept with data from a single wafer and show the results of PCA prior to and after ICA decomposition in figure 4.24, where we can observe that the data tends to be more Gaussian after ICA removal since the dominant 1st PC explains 20% more variance than the one prior to ICA removal.

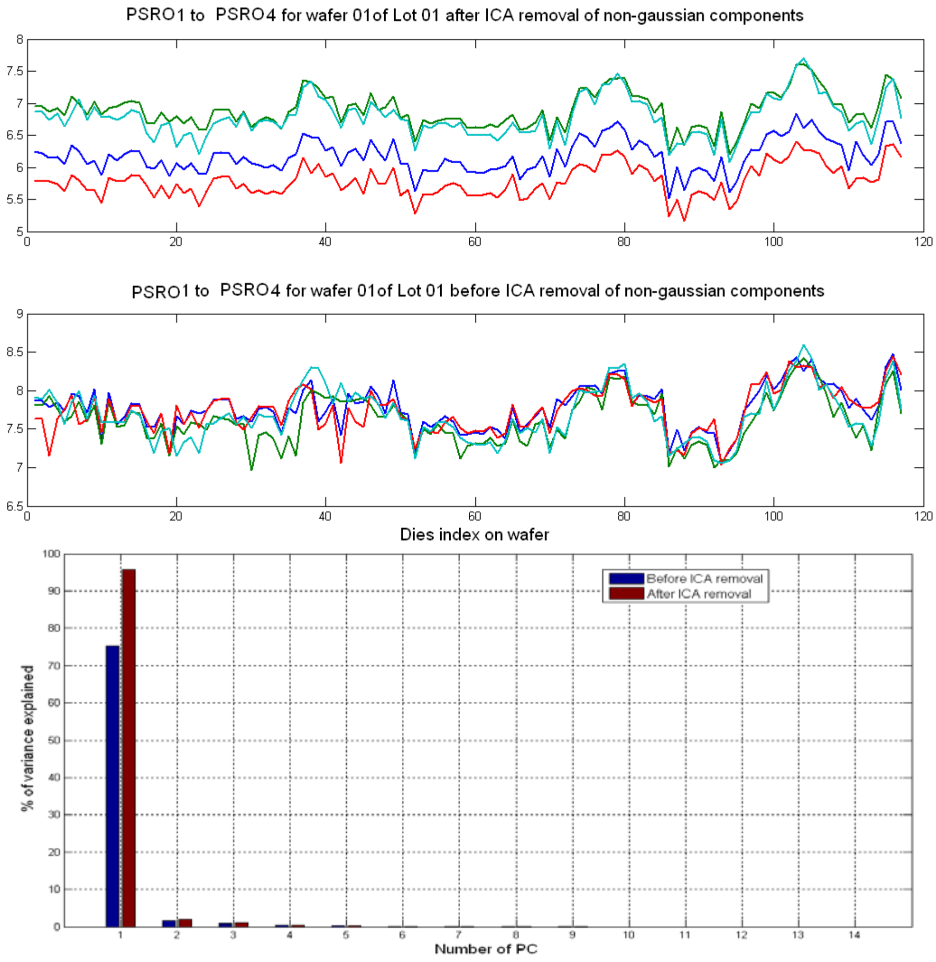


Figure 4.24 Die values and Scree plot of PCA prior and after ICA removal of non-Gaussian components

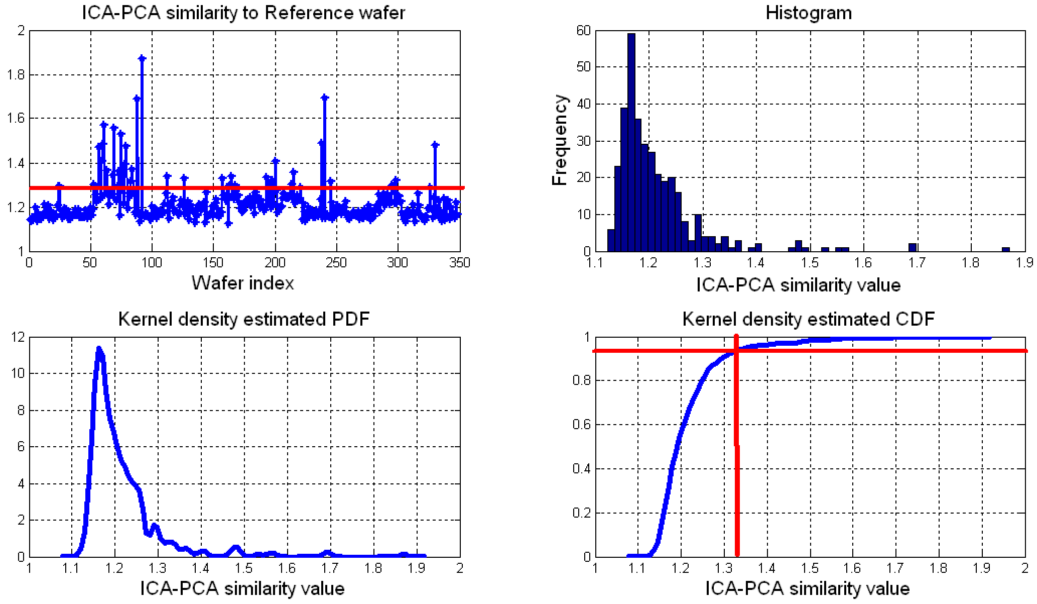


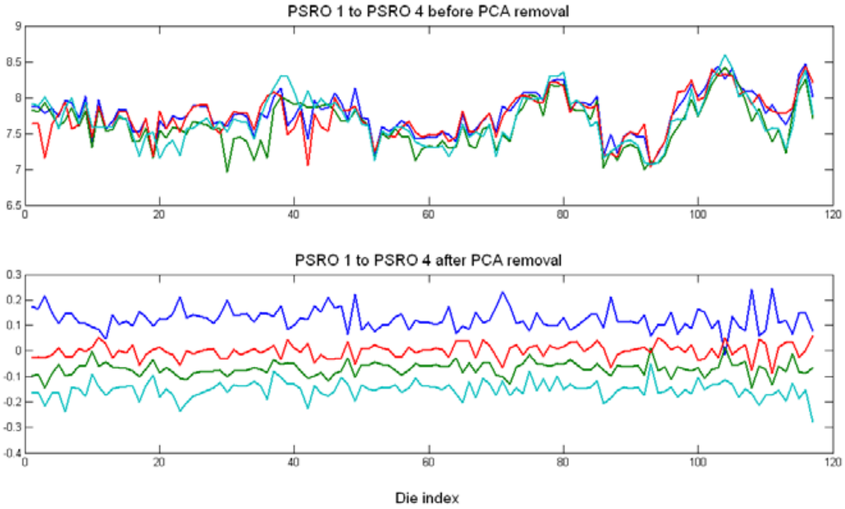
Figure 4.25 ICA-PCA similarity indexes

The second variant is to perform ICA on the residual matrix after PCA decomposition is done since PCA decomposition can remove the Gaussian components. Then the similarity index is computed in a slightly different way from the one in PCA case since ICs do not have corresponding variance associated with them. The computation of the similarity index for this variant is shown below:

1) Each wafer is represented by 117 by 14 matrix.
2) ICA is performed on each wafer matrix after PCA removal.
3) All ICs retained (ICs don't have associated variance).
4) Compute the similarity matrix according to Equation 4.15.
    a. We first search for the pair of ICs that have the smallest angle.
    b. Remove the extracted 1st pair of ICs.
    c. Search for 2nd pair again.
    d. Continue until all ICs are paired.

$$S_{ICA} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} cos^2\theta_{ij}}{k} \tag{4.15}$$

θ_{ij} is the angle between IC$_i$ from wafer $\boldsymbol{i}$ and IC$_j$ from wafer $\boldsymbol{j}$. The results of similarity for all wafers vs. the reference wafer are shown in figure 4.27. A Kernel density estimator is applied to estimate the empirical PDF and CDF, then a statistical threshold is set based on corresponding probability extracted from the CDF plot.



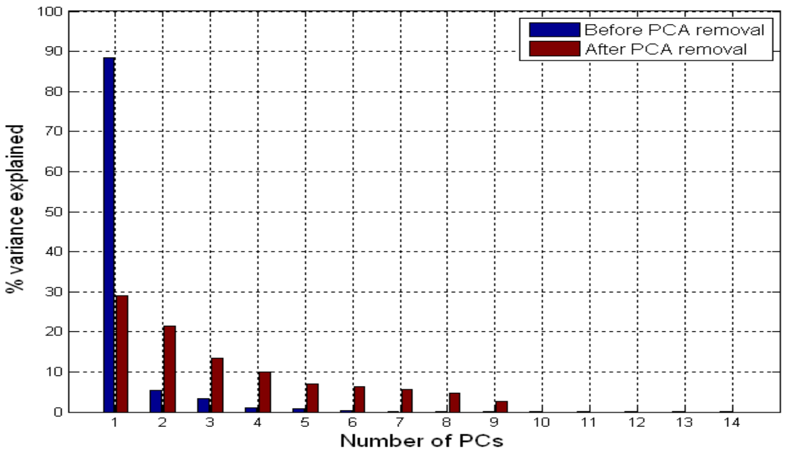


Figure 4.26 Die values and Scree plot of PCA prior and after PCA removal of Gaussian components

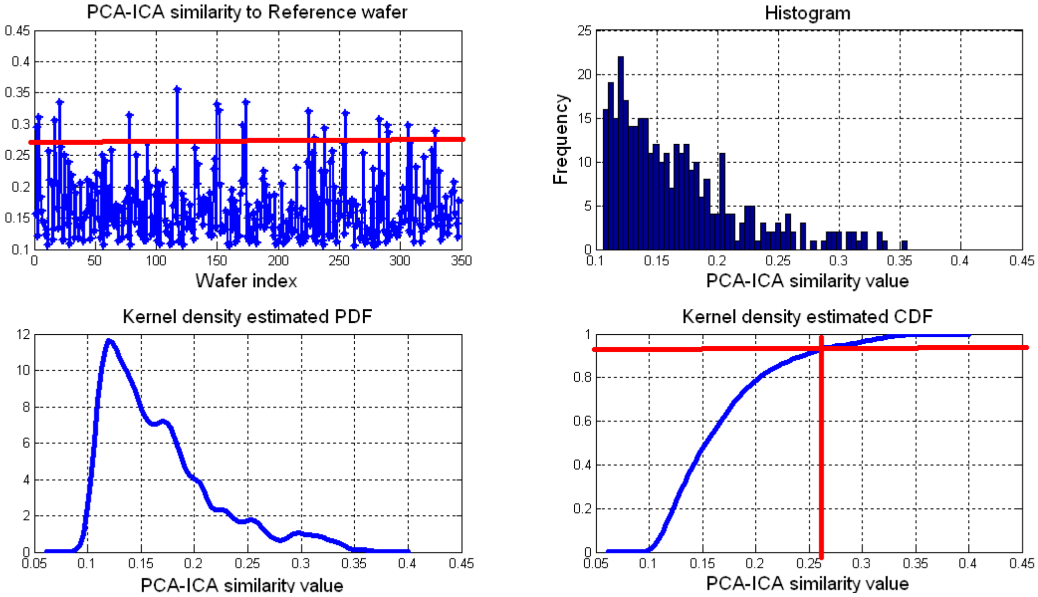


Figure 4.27 Similarity index from PCA-ICA

Wafers can have similar spatial orientations for their subspaces, but can have different magnitudes due to process drifting. Mahalanobis distance (MD) between the means of two classes is adopted in this section to compute the MD based similarity index [4.19]. The MCD version of MD is computed by the equation:

$$MD = \sqrt{(\bar{X}_2 - \bar{X}_1)^T \sum^{*-1} (\bar{X}_2 - \bar{X}_1)} \tag{4.16}$$

where $\bar{X}_2$ and $\bar{X}_1$ are the centers of wafer 2 and wafer 1, and MD is here the distance between wafer 1 and wafer 2. In this work, to ensure robust measure, a robust estimate of center and covariance is used as discussed in Chapter 2. $\sum^*$ is the common covariance matrix, the weighted mean of empirical covariance matrices of wafer 1 and wafer 2, such that:

$$\Sigma^{*} = \frac{n_1\Sigma_1^{*} + n_2\Sigma_2^{*}}{n-2} \tag{4.17}$$

where $= n_1 + n_2$, n_1 and n_2 are the sample sizes of wafer 1 and wafer 2. $\sum_1^{*}$ and $\sum_2^{*}$ are empirical covariance matrices of wafer 1 and wafer 2. Each wafer is represented by a 117-14 matrix. The similarity index for the reference wafer vs. all the wafers is shown in figure 4.28.

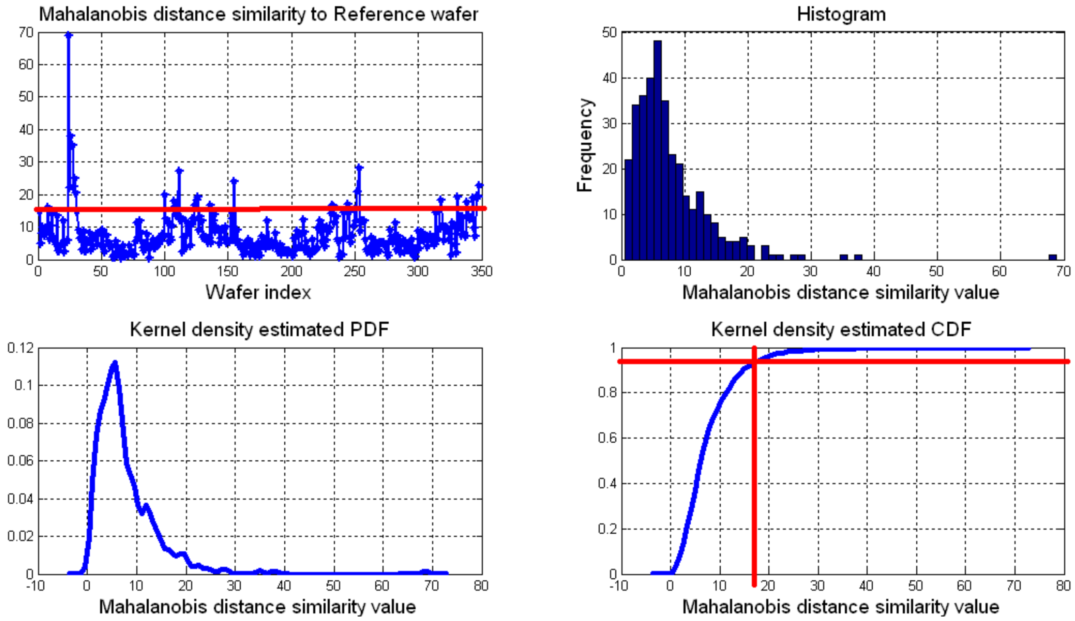


Figure 4.28 Similarity index from MD

The spatial variogram and correlogram discussed in section 4.2 are adopted here to compute a similarity index for wafer-to-wafer. The goal is to detect wafers with spatial patterns significantly different from the reference wafer. Each wafer is represented as a map with 117 entries; each entry is represented by (x, y) coordinates on the wafer. At each (x, y) coordinate, the median of 14 PSROs is used as the attribute value. The spatial variograms and correlograms of wafers from lot 1, 2, 3 and 4 are shown in figure 4.29. The left plots with blue curves are spatial variograms and the right plots are for spatial correlograms.

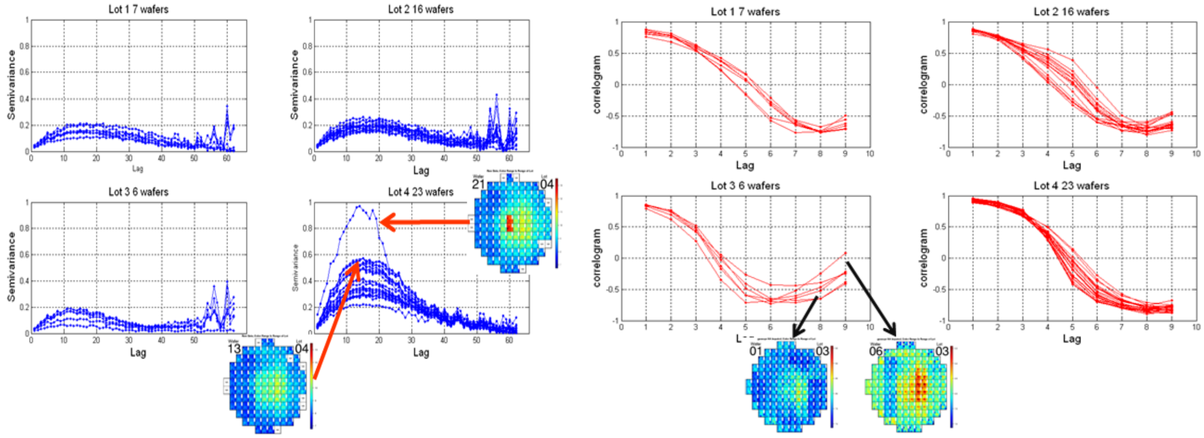


Figure 4.29 Spatial variogram(left) and correlogram (right)

It is possible to visually pinpoint the wafers with disrupted spatial patterns based on the variogram, while the correlogram is not very informative. In figure 4.30, we demonstrate detecting wafers with disrupted patterns with spatial variogram. The variograms of the first wafer from all 23 lots are shown in the left plot of figure 4.30; here we can see that wafer 01 from Lot 22 has significantly higher semivariance than other wafers. The disrupted spatial pattern for this wafer can be confirmed by a mosaic plot of all Lot 22 wafers shown in the right panel of figure 4.31.

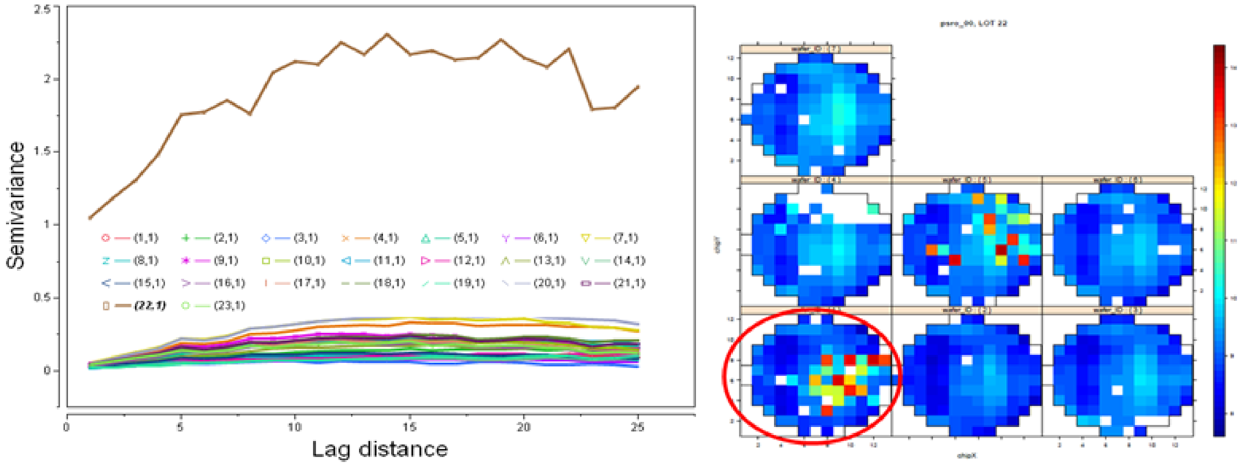


Figure 4.30 Left: Spatial variogram for wafer 1 from all Lots (Lot 1 to Lot 23); Right: Median of each die shown for all wafers within Lot 22.

In order to construct a FDC model using spatial variograms or correlograms, one needs to characterize the spatial variogram or correlogram as input vectors per wafer, and then train the FDC model to capture the similarity index between disrupted wafers and the good "reference wafer." If the similarity index extracted for a new wafer falls below certain threshold, an alarm will be raised.

Given the curved nature and high dimensionality of spatial variogram and correlograms, dynamic time warping (DTW) is adopted for similarity computing along with angle measures [4.20]. DTW is for comparing discrete sequences to sequences of continuous values. DTW in this paper aligns the two spatial variogram or correlogram curves so that the Euclidean distance between them is minimized. The result of the matching is analyzed through a warping path W that connects the mapping points of the two time series data in two-dimensional Euclidean space. The optimal warping path that minimizes the distance, expressed as W, consists of a total of K elements, while the measured distance $DTW(S,T)$ is the magnitude of dissimilarity between the elements of S and T, where S and T are the two curves for comparison. Final dissimilarity is defined as the mean distance determined by the curves:

$$DTW(S,T) = min_w \left\{ \sum_{k=1}^{K} \delta(i_k, j_k)/K \right\} \tag{4.18}$$

$$\delta(i_k, j_k) = \left(s_{i_k} - t_{jk}\right)^2 \tag{4.19}$$

Dynamic programming can be used to effectively find this path by evaluating the following recurrence, which defines the cumulative distance as the sum of the distance of the current element and the minimum cumulative distances of the adjacent elements. The recursive formula that searches an optimal warping path with the dynamic programming technique is expressed as:

$$y(i_k, j_k) = \delta(i_k, j_k) + min\{y(i_{k-1}, j_k), y(i_{k-1}, j_{k-1}), y(i_k, j_{k-1})\} \qquad (4.20)$$

The more linear the W, the higher the degree of similarity between the two patterns' shapes.

In figure 4.31 and figure 4.32, we show the results of computing a similarity index with variograms and correlograms along with DTW and angle approaches. To avoid the noisy nature of spatial variogram and correlograms, we have used the smoothed wafers as input data. By smoothed wafer, we mean that a King's move based neighborhood construction rule and median are used. The results from the DTW approach indicate that this distance-based approach does not work well with higher numbers of lags and is not suitable for outlier wafer detection or this specific dataset. The comparison of similarity using correlation between variogram and correlogram does show the drifting of similarity within each lot. However, detection of extreme outlier wafer using spatial variograms and correlograms does not work well for this specific dataset. This is due to the fact that spatial variogram and correlograms are both characterized by attribute difference at each lag and number of lags used. At larger lag distance, more pairs of coordinates are used for computation, thus the noise level is also increased. This makes the spatial variogram and correlogram not suitable for detecting outlier wafers for this specific dataset.

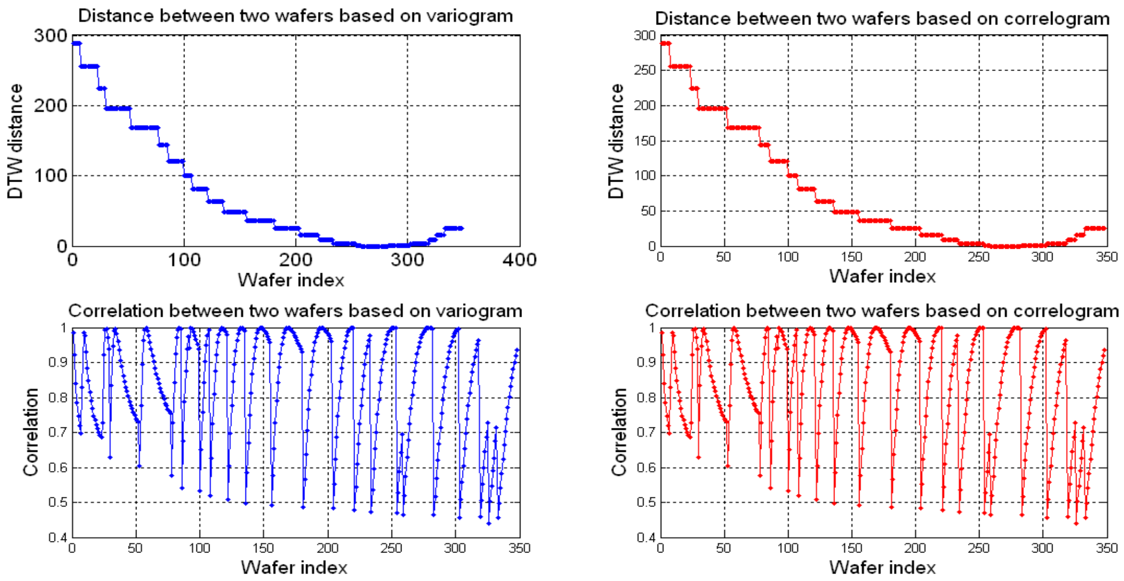


Figure 4.31 Similarity index for spatial variogram with smoothed wafers

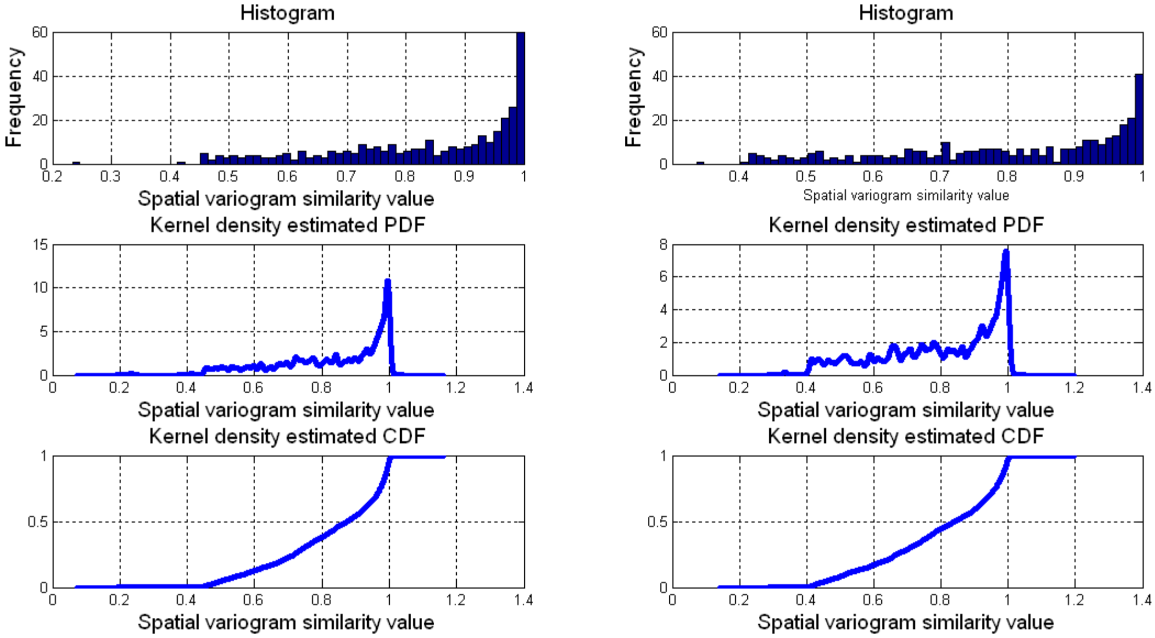
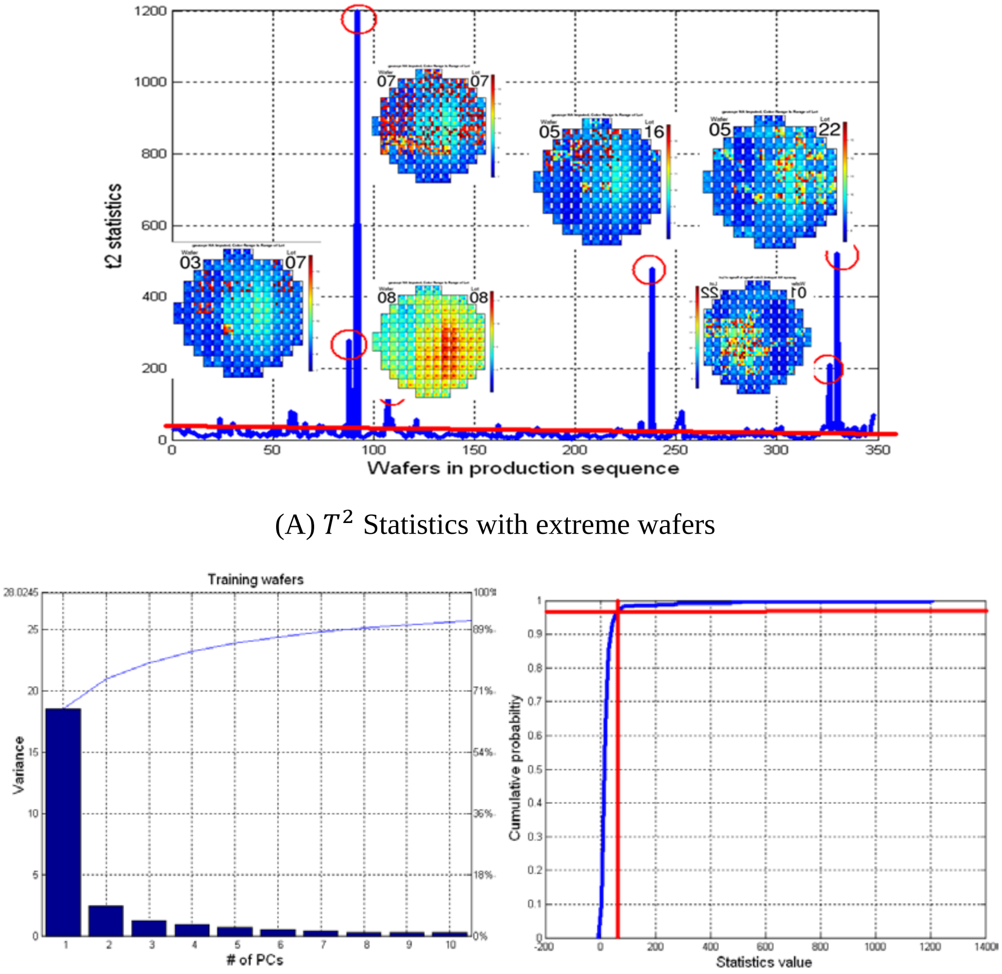


Figure 4.32 Similarity index for spatial variogram with smoothed wafers

4.5.2. Detecting Outlier Wafers with PCA and ICA-PCA

In this section, PCA and ICA-PCA are both applied to the 348-by-117 data matrix denoted X, which contains 348 wafers from all 25 lots. At each site on the wafer, the median 14PSROs are used as the attribute value of the site. Thus, a wafer is characterized by a 117-by-1 dimension vector. X is first normalized against the reference wafer. Since we do not have a set of "normal" wafers to train a PCA or ICA-PCA model, the entire dataset is used for constructing the PCA or ICA-PCA model. Results for both approaches are shown in Figure 4.33 and Figure 4.34. The top 10 PCs explaining above 90% of total variance are retained to compute T^2 statistics in both cases. A Kernel density estimator is adopted to select the threshold for outlier wafer detection. However, in this case, the distribution of T^2 statistics is skewed by several outlier wafers with extreme high T^2 values. In real practice, one can train an SPC chart based on historical wafers for better detection accuracy.

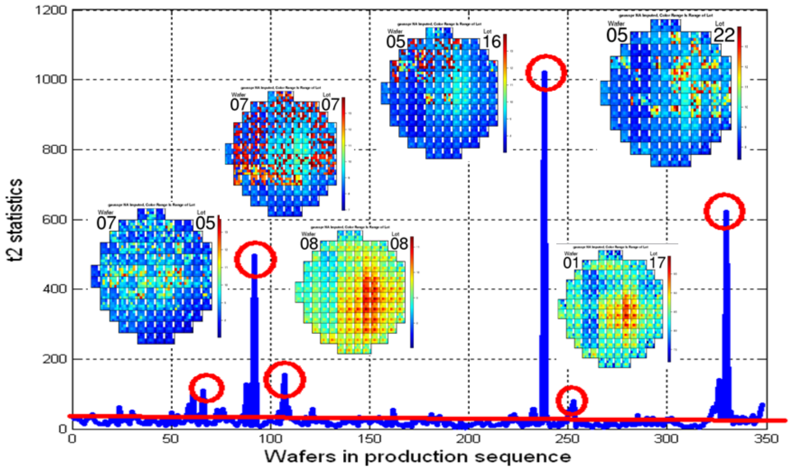


(A) T^2 Statistics with extreme wafers

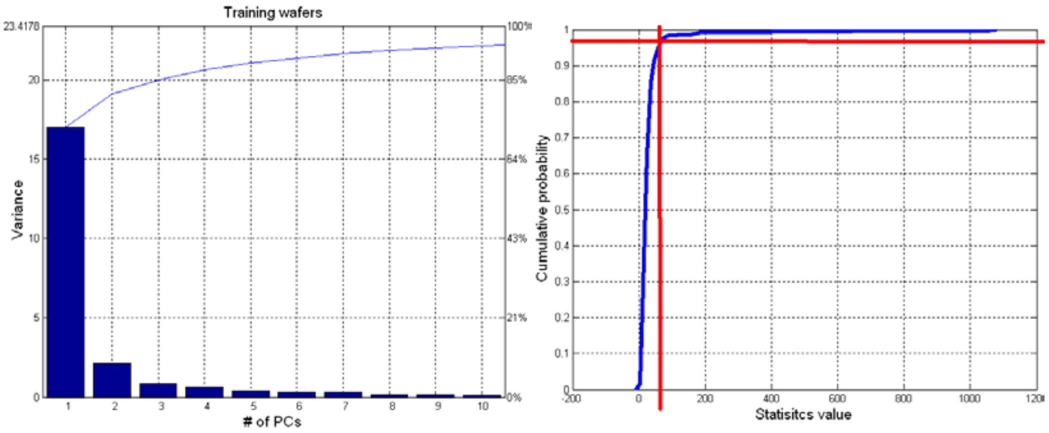
(B) PCA variance plot

(C) CDF plot

Figure 4.33 PCA based Hotelling's T2 statistics for outlier wafer detection



(A) T^2 Statistics with extreme wafer



(B) PCA variance plot (C) CDF plot

Figure 4.34 ICA-PCA based Hotelling's T2 statistics for outlier wafer detection

4.5.3. FDC Results Comparison

In this section, we compare the performance of different extreme wafer detection methods. Only the multivariate statistics-based similarity indices are used for testing here, since Moran I plot and MCD by Dies do not provide statistical alarms. Since there are 348 wafers available and no labeling information regarding which wafers are considered outliers or normal, the median wafer described in Equation 4.12 is referred to as a "normal" wafer. We first manually labeled the wafers that appear to have disrupted spatial patterns as "outlier" wafers. There are 36 wafers labeled "outliers"; the remaining 308 are considered "normal." The definition of an "outlier" is that the wafer has disrupted spatial patterns in comparison to the median wafer used for "normal" definition. Then, we randomly selected 50% of the wafers as our training set, while the remaining 50% wafers were used for validation. The final combination of training and validation sets was chosen to retain 20 "outlier" wafers in the training set and 18 "outlier" wafers in the validation set. We present the results of different detection methods in table 4.1. Each method is first applied to the training set wafers. A statistical threshold based on a certain percentage of cumulative probability from the training set is then selected to detect if a wafer is an outlier or normal. This threshold is then applied to classified wafers of the testing set.

The evaluated detection methods include the following: PCA based similarity index, ICA based similarity index, and Mahalanobis distance and Hotelling's T^2 from PCA models. DTW based variogram and correlogram were not applied due to high level noise associated with larger lag distance. In table 4.1, the misclassified rate is defined as the number of wafers being classified wrongly. The true alarm rate indicates the number of actual "outlier" wafers classified as "normal." The false alarm rate indicates the number of actual "normal" wafers classified as "outliers."

Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm

$S_{\mathrm{PCA}}^{\lambda}$ (95%)	7/174	1/20	6/154	13/174	1/18	12/156
S_{ICA} (95%)	12/174	2/20	10/154	20/174	3/18	17/156
MD (95%)	8/174	3/20	5/154	18/174	5/18	13/156
PCA (95%)	15/174	2/20	13/154	20/174	4/18	16/156
ICA-PCA (95%)	12/174	2/20	10/154	16/174	3/18	13/156

Table 4.1 Evaluation results

The method that best performed is the $\mathrm{S}_{\mathrm{PCA}}^{\lambda}$ based approach, where the statistical threshold is selected to be 95% of cumulative probability.

4.6. Optimal Site Selection for Process Monitoring

Given the strong spatial correlation among dies, it is insightful to group dies with similar performances together, and characterize the process according to such groupings. This assumes that different process can result in different spatial correlation structure among dies. Changes in spatial correlation structure can indicate variations in process. It is also beneficial to know what dies or sites on wafers tend to have similar performances, then process engineers can perform metrology optimally across wafers based on the grouping patterns. Metrology results are either used for process monitoring or for interpolating wafer yield. In the process monitoring case, the high collinear structure among sites allows us to extract a few sites that carry the most information for process variation and only measure these extracted sites in the future. In the case of wafer yield interpolation, the spatial correlation structure allows us to extract the sites that provide the most capability for interpolating the entire wafer map with the smallest error. Thus metrology costs can be significantly reduced by selecting optimal sites for different objectives, especially in the case of electrical measurements, where engineers often need to measure all the dies across an entire wafer.

In this section, we present several filter and wrapper based variable selection methods for selecting optimal sites adapted to both process monitoring and wafer map interpolation objectives. The performance of selection results is then evaluated by the FDC and prediction methods discussed in Section 4.5 and Chapter 2.

4.6.1. Site Clustering

In this section, we address how to partition sites into subclasses where they share similar performance using a clustering algorithm. Given the high dimensionality of our dataset, we have adopted the spectral clustering concept to partition the dies into different classes based on the extracted spatial correlation matrix. Given that there are 117 dies per wafer, we have a similarity matrix of 117-by-117. In this case, we have represented each die using the median value of 14 PSROs on the die. In figure 4.35, the heat map of die-to-die similarity matrix along with its histogram is shown. In this work, we have computed the similarity index for die-to-die as Pearson correlation coefficients without removing the lot-level and wafer-level averages from each die.

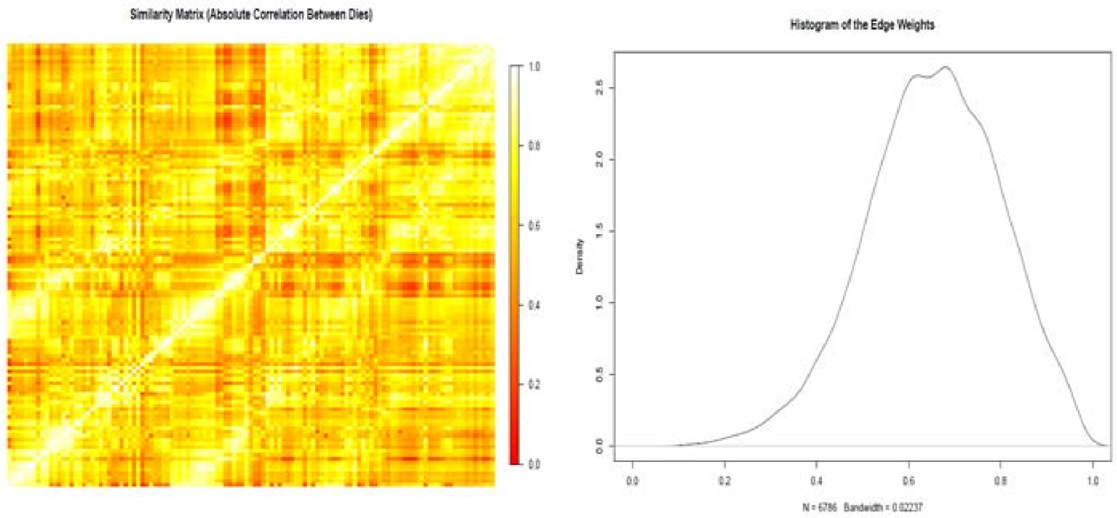


Figure 4.35 Die-to-Die similarity matrix and corresponding histogram

The spectral clustering algorithm [4.21] can be described as follows:

1) Given a set of N samples, $D = \{x_1, x_2, \dots, x_n\}$, x_i is usually highly dimensional
2) Compute the positive definite similarity matrix $S \in R^{nxn}$, where s_{ij} is the similarity between the x_i and x_j. The most general similarity function is the Gaussian function,

$$S(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right) \qquad (4.20)$$

where the parameter σ controls the width of the neighborhood in the graph.

3) Normalize the affinity matrix S to obtain the Laplacian matrix:

$$\bar{S} = L^{-\frac{1}{2}} S L^{-\frac{1}{2}} \qquad (4.21)$$

$L = L_{ij}$ is $N\ x\ N$ diagonal matrix with $L_{ii} = \sum_j S_{ij}$.

4) Compute the first j generalized eigenvectors $u_1, u_2, \dots, u_k$ of $\bar{S}$
5) Retain the first k eigenvectors with smallest eigenvalues, and then we can represent each observation in eigenspace with k-dimensions.
6) Cluster the observations with the chosen clustering technique; the common ones are k-means and GMM.

The first 4 eigenvectors with smallest eigenvalues are retained. They are shown in figure 4.36 for all the dies. There are many variants of spectral clustering, where different sets of eigenvectors are retained to partition the dataset [4.22].

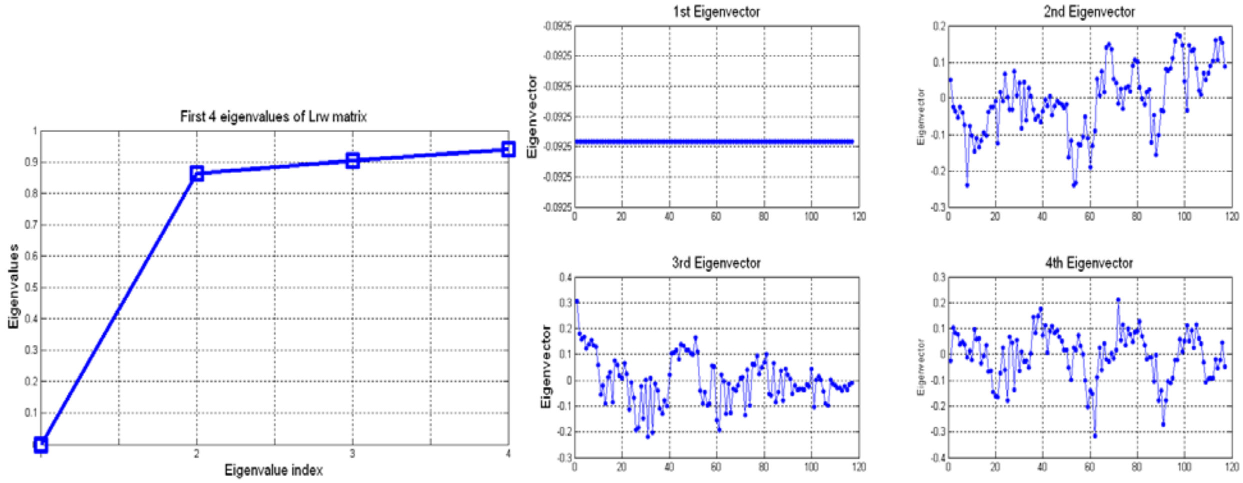


Figure 4.36 First 4 Eigenvalues and Eigenvectors

In this work, K-means clustering is then applied to the 117-by-4 input matrix. Due to variation of the K-means clustering algorithm, multiple runs of K-means are performed, after which a majority voting scheme is applied to assign the final class label to the dies. K-means clustering with an optimal *k* equal to 4 was chosen. 50 variations of K-means and final result of voting schemes are shown in figure 4.37.

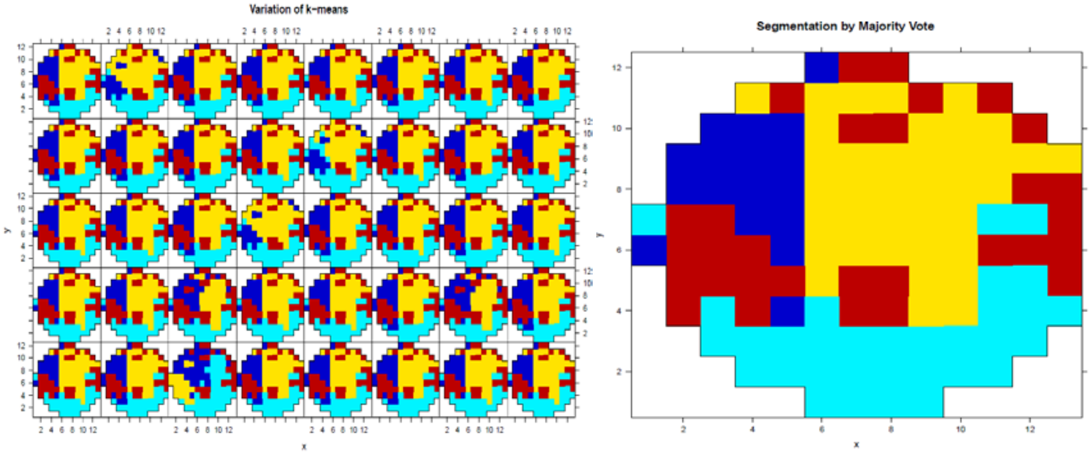


Figure 4.37 Variation of K-means and result of segmentation

As shown in figure 4.37, the 117 sites are partitioned into 4 clusters. The partition results indicate that sites at the bottom of wafers tend to share similar performance temporally, whereas sites near the center also share similar performances. The partition results match the PCA decomposition results shown in figure 4.20, where the main variations are contributed by dies at bottom edge and center of wafers. The clustering of sites can provide engineers with insights as to spatial correlation among sites, and also can become a preliminary phase leading engineers to sample where on a wafer to measure optimally for process information.

4.6.2. Site Selection for Process Monitoring

In this section, we present sites selection methods for process monitoring purposes. The objective is to select sites carrying the most information about the process. The assumption here is that information contained in selected sites is robust enough to be used for FDC model input. The highly collinear structure of the sites makes it possible for a few sites to preserve the multivariate structure and information carried by a complete set of sites.

Let us assume we have n observations with p dimensional measurement vectors for each observation, thus the data matrix is of dimensions $n \times p$. We can then partition the $p \times p$ dimension sample covariance matrix $\sum$ as follow [4.23]:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{4.22}$$

where $\sum_{11}$ is the $k \times k$ covariance matrix of selected k measurements and $\sum_{22}$ is the covariance matrix of the remaining set. The selection criterion for finding sites containing the most information [4.24] is:

$$\max_{k \in p} |\Sigma_{11}| \ or \ \min_{k \in p} |\Sigma_{22|11}| \tag{4.23}$$

where $|\cdot|$ denotes the determinant of matrix and $\sum_{22|11}$ is the conditional covariance matrix of the remaining set, given the selected ones. $\sum_{22|11}$ can be computed as:

$$\Sigma_{22|11} = \Sigma_{22} - \frac{\Sigma_{21}}{\Sigma_{11}} \Sigma_{12} \tag{4.24}$$

There are $2^p - 1$ choices to select $\binom{p}{k}$, thus it is not possible to find the global optimal set of variables in most applications. In this work, heuristic methods are explored to find the near-optimal set of sites. We tackle this as a variable selection problem with filter methods, since there is no need for an evaluation method.

There are many variants of filter methods for variable selection, such as selecting variables sequentially to maximize entropy. The filter method we propose is a two-stage Clustering-PCA based selection method. We compare this approach with wrapper methods based on site selections done with stepwise QDA and CART. The first method does not require a response variable as the later ones do, since it assumes the most informative sites can provide discriminative power for detecting “good” or “bad” wafers. In stepwise QDA and CART, our response variable is the class label of wafers, where the labels are the categorical values “good” and “bad.” The “good” and “bad” labels are assigned according to results from Section 4.5.

4.6.2.1. Clustering-PCA based Variable Selection

The concept of the two-stage approach is to first cluster the sites into subclasses, and then to perform PCA based variable selection within each class to select the most informative sites. PCA is preferred here given the highly collinear structure observed among the sites.

In this work, we have 348 wafers and 117 sites per wafer, thus it is desirable to select the appropriate clustering method for the clustering phase, since most distance metrics used in clustering suffer from the curse of dimensionality. In this work, spectral clustering is first applied as we described in section 4.6.1. A total of 4 classes of sites are retained. In the second phase, PCA based variable selection is then applied within each class. We denote this heuristic PCA based variable selection as an orthogonal sequential PCA approach. The details are as follows:

1) Given input matrix X_k of dimensions $n \times p_k$ for each class, n is the total number of wafers, and p_k is the number of sites in each k_{th} class resulting from clustering. We first transpose the X_k to ${X_k}^T$ with dimensions $p_k \times n$.
2) We then perform PCA on ${X_k}^T$ to obtain a set of loadings. The first loading vector corresponding to the first eigenvector of sample covariance matrix is retained as a selection basis, since it is the direction of largest variance among samples. The sample covariance matrix in this case is $n \times n$ rather than $p_k \times p_k$. Thus, the first loading vector is the $n \times 1$ dimension.
3) The retained loading vectors are then used to select the site that has highest Pearson correlation coefficient from the original input matrix X_k. We denote the selected site denoted as p_{kj}.
4) To ensure the remaining sites do not the carry same information as the selected p_{kj} have, we project the X_k onto the subspace orthogonal to the p_{kj}, and the projected matrix is denoted as $X_k^i \;\; i \in p_k$. The projection matrix P and X_k^i can be computed as follows:

$$P = I - p_{kj}{p_{kj}}^T / {p_{kj}}^T p_{kj} \tag{4.24}$$

$$X_k^i = X_k P^T \tag{4.25}$$

This projection removes the component in the remaining sites that can be linearly explained by selected sites.

5) Steps 1 and 4 are iterated until all the sites within each class are ranked. The sites within each class are ranked in their selection order from 1 to p_k with p_k being the total number of sites available in the class.

The selection of top ranked sites from all 4 classes can be chosen based the number of PCs needed to explain more than 95% of variance for samples in each class. In this work, we have retained the top 30 of 117 sites as the ones carrying significant amounts of information.

4.6.2.2. Selecting Sites for Wafer and Lot Identification

In this section, we utilize QDA and CART to select sites for two objectives: our first objective is to select sites that can discriminate a wafer's relative position within a lot, and the second is to select sites that can discriminate the relative lot position the wafer is allocated.

As concerns the first objective, the idea is that some sites may have significantly different performance from wafer to wafer while other sites within the same wafer tend to remain similar performance from wafer to wafer. This concept is visualized in figure 4.38 using the dataset discussed in section 4.3, where the die averages for wafers with same Wafer ID are overlaid on a wafer map. From the plot, we can observe that the center dies for wafers with Wafer ID 25 tend to have different performances from the ones with Wafer IDs 1, 10 and 20. In a sense, the selected sites are the ones that contribute the most to variance at the wafer level, and they can capture wafer-to-wafer process variations after the lot effect is removed. The response variable in this case is the wafer ID for each wafer, while the predictor variables are the sites on wafer.

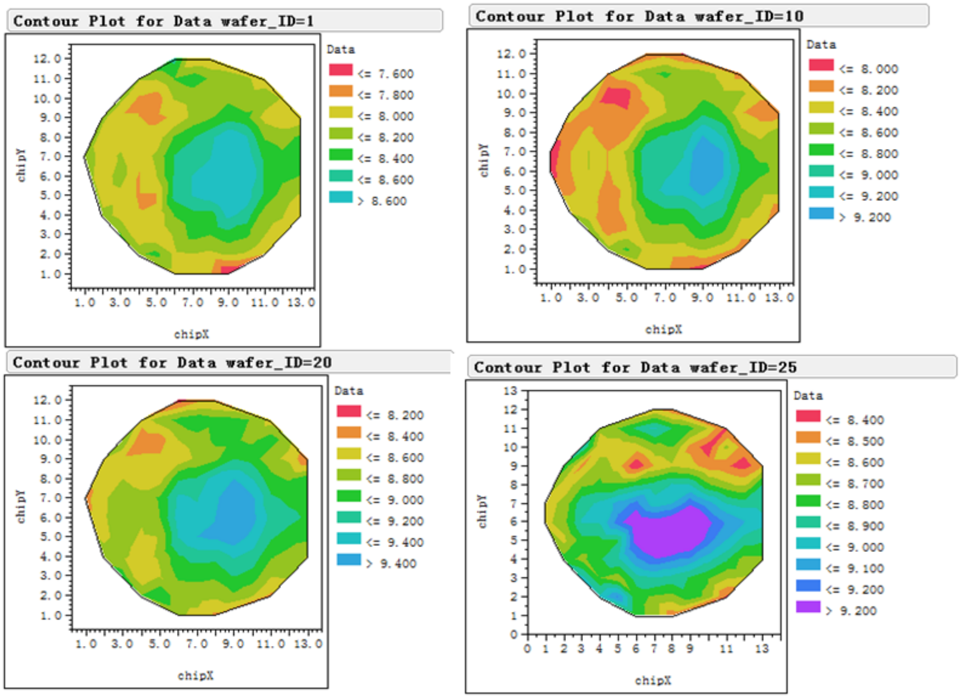


Figure 4.38 wafer average for all the wafers with same Wafer ID

For the second objective, the idea is that some sites may have significantly different performances from lot–to-lot, while other sites remain relatively similar from lot-to-lot. This concept is visualized in figure 4.39, where the die averages of wafers sharing the same Lot ID are overlaid on a wafer map. We can observe that some dies near the center of the wafer for Lot 23 have different performances from Lots 1, 10 and 20. Thus, these selected sites are the ones that contribute most to variance at the lot level and capture lot-to-lot process variations. The success of identifying such sites depends on the nature of the applied dataset.

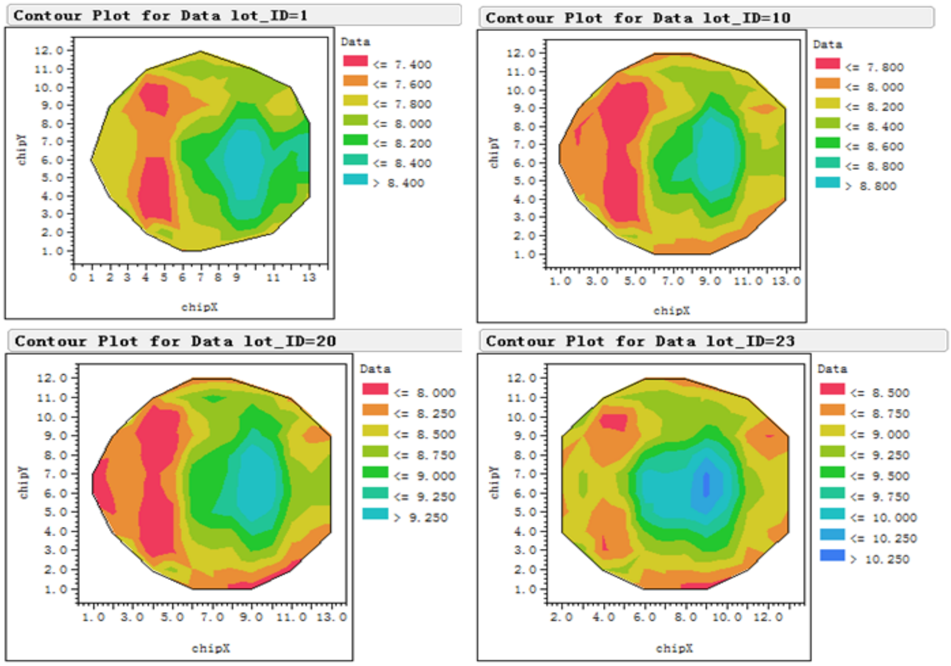


Figure 4.39 wafer average for all the wafers with same Lot ID

As we have discussed concerning the available dataset in section 4.3, there are a total of 348 wafers that span 23 lots. For each lot, there can be a maximum of 25 wafers. However, the number of wafers for our dataset is uneven. Therefore, we used the entire dataset of 348 wafers as a training set to extract the top 30 sites for Wafer ID discrimination and Lot ID discrimination. In a general application, one could collect more wafers to ensure that there are enough wafers for validation.

For the objective of Wafer ID discrimination, we will need to remove the lot effect first. To remove the lot effect for each site, we first define p_{ijk} as the median of the original 14PSROs at i_{th} site of j_{th} wafer of k_{th} lot and process the data as follows:

$$\tilde{p}_{ijk} = p_{ijk} - \bar{p}_{ij\cdot} \tag{4.26}$$

where $\bar{p}_{ij\cdot}$ denotes the average values of all the sites within the same lot and $\tilde{p}_{ijk}$ is the site value after lot average is removed. The input matrix X has dimensions of 348×117 and outputs 1-dimensional vector with categorical values from 1 to 25. Here, we denote the output as Wafer ID, and we have 25 classes of wafers. CART is applied to select the sites that provide discriminative power to Wafer ID.

For the second objective, p_{ijk} is used as input without any processing. Forward regularized QDA and CART are applied to select relevant sites for this objective. In this work, due to the small number of wafers available for each class and the high number of sites per wafer, we adopted a regularized QDA to avoid the curse of dimensionality. QDA assumes the 117 sites from each wafer class follow multivariate Normal distributions with different covariance matrices. Thus QDA seeks to perform dimensionality reduction where the objective is to minimize within-wafer class scatter and maximize between-class scatter [4.25].

For both objectives, the 30 top-ranked sites are retained as measure sites. The performance of selected sets is evaluated using the FDC schemes discussed in section 4.5.

4.6.3. Site selection for Wafer Map Interpolation with Wrapper methods

The objective of site selection for wafer map interpolation is to select sites that offer the highest predictive power over the remaining sites. This is different from the case of site selection for process monitoring, where sites explaining the most variances about the dataset are selected. Suppose we also have n wafers with p dimensional measurement vectors, we can slip the available p sites into two sets, U and M, where U contains the sites not selected, and M contains the selected sites for measurement. The objective is then to minimize the sum of the square of prediction error for all unselected sites across all wafers:

$$\min \sum_{k}^{n} \left(\hat{y}_i^U - y_i^U\right)^2 \tag{4.27}$$

where $\hat{y}_i^U$ is the predicted site measurement value for the unselected set, and y_i^U is the actual value. Thus, it is intuitive to use wrapper-based methods to perform site selection, since the selected set is dependent on the applied evaluation methods. We have adopted bootstrapping with a stepwise framework using prediction methods including PLS, and BPNN for site selection in this section.

4.6.3.1. Bootstrapped stepwise PLS

The combination of bootstrapping with stepwise is to reduce the high variance associated with stepwise models. Due to the high collinear structure of sites introduced by spatial correlation, PLS is the preferred prediction method, as it is capable of projecting high dimensional input and output variables onto lower subspace while maintaining the multivariate structure. The details of this approach are as follows:

1) Suppose we have an input matrix X of dimensions $n \times p$, n is the number of wafers, and p is the total number of sites per wafer. We first bootstrapped out a sample set $X_b$having dimensions $n \times p$.
   a. Let us define U as the set of unmeasured sites and M as the set of measured sites and where p_j denotes a site on wafer.
   b. Setting $\mathrm{M} = 0$ and $\mathrm{U} = \{p_1, p_2, p_3, \dots, p_{117}\}$, we first compute p number of PLS model, where input is 1-dimensional vector p_j and output is a $n \times (p-1)$ matrix with column vectors $\{p_{j+1}, p_{j+2}, \dots, p_{j+116}\}$. The p_j giving the best prediction error as shown in equation 4.26 is selected as the first measured site. This p_j is then removed from the U and added to M.
   c. Setting $\mathrm{M} = 1$ and $\mathrm{U} = \{p_{j+1}, p_{j+2}, \dots, p_{j+116}\}$, we then compute $(p-1)$ number of PLS model, where input is a $n \times 2$ matrix and output is a $n \times (p-2)$ matrix with column vectors $\{p_{j+2}, p_{j+3}, \dots, p_{j+116}\}$. The p_{j+1} offering the best prediction error as shown in equation 4.26 is selected as the first measured site. This p_{j+1} is then removed from the U and added to M.
   d. The forward selection procedure stops when $\mathrm{M} = 30$, since we limit the maximum number of measured site to 30 of the 117 available sites.

2) We perform *B* number of bootstrapped models. For each bootstrapped model, we obtain 30 sites with rankings from 30 to 1. To unselected sites, we assign a "0". Thus we can have distributions of ranks for each site; the median statistics for each distribution are then extracted to rank the predictive power of each site. The top 30 sites with higher median statistics are retained as selected sites.

4.6.4. Selected Sites Visualization and Performance Comparison

In this section, we visualize the selected top 30 sites for five selection schemes on the wafer map in figure 4.40. The color indicates the importance and selection order of selected sites where "1" indicates the first selected one and "30" indicated the last selected one. The five selection schemes are: (A) Clustering-PCA variable selection; (B) CART on wafer ID; (C) Stepwise QDA on wafer ID; (D) CART on Lot ID; (E) Bootstrapped stepwise PLS.

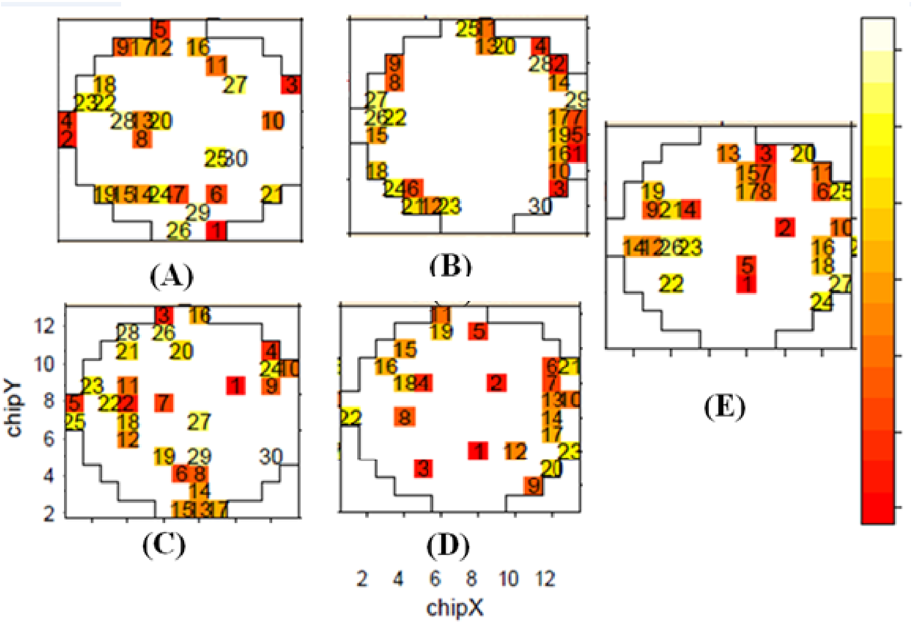


Figure 4.40 Visualization of site selection schemes

The majority of sites selected by each scheme are located at the edge of the wafer. This indicates that wafers manufactured using this specific process have performance variations at the edges from wafer to wafer and lot-to-lot. The selected sites are used as inputs for outlier wafer detection with the multivariate methods discussed in Section 4.5. The results are presented in table 4.2. For each method, we compare its training model accuracy and its testing set accuracy. The same training and testing sets are applied, in which the training set contains 20 "outlier" wafers and the testing set contains 18 "outlier" wafers.

(A) Clustering-PCA variable selection						
Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm
S_{PCA}^{λ} (95%)	10/174	1/20	9/154	13/174	1/18	12/156

S_{ICA} (95%)	15/174	3/20	12/154	18/174	2/18	15/156
MD (95%)	12/174	3/20	9/154	15/174	2/18	13/156
PCA (95%)	14/174	2/20	12/154	14/174	2/18	11/156
ICA-PCA (95%)	13/174	2/20	11/154	16/174	1/18	12/156
(B)CART on wafer ID						
Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm
S^{λ}_{PCA} (95%)	15/174	2/20	13/154	16/174	1/18	15/156
S_{ICA} (95%)	17/174	4/20	13/154	19/174	3/18	16/156
MD (95%)	18/174	3/20	15/154	20/174	4/18	16/156
PCA (95%)	16/174	2/20	14/154	18/174	3/18	15/156
ICA-PCA (95%)	14/174	1/20	13/154	17/174	3/18	14/156
(C)Stepwise QDA on wafer ID						
Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm
S^{λ}_{PCA} (95%)	15/174	0/20	15/154	16/174	1/18	15/156
S_{ICA} (95%)	19/174	3/20	16/154	20/174	2/18	18/156
MD (95%)	18/174	3/20	15/154	18/174	3/18	15/156
PCA (95%)	16/174	2/20	14/154	17/174	2/18	15/156
ICA-PCA (95%)	15/174	2/20	13/154	15/174	2/18	13/156
(D) CART on Lot ID						
Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm

S_{PCA}^{λ} (95%)	14/174	1/20	13/154	14/174	1/18	13/156
S_{ICA} (95%)	16/174	4/20	12/154	16/174	3/18	13/156
MD (95%)	17/174	3/20	14/154	16/174	2/18	14/156
PCA (95%)	14/174	2/20	12/154	15/174	2/18	13/156
ICA-PCA (95%)	13/174	1/20	12/154	13/174	1/18	12/156
(E) Bootstrapped stepwise PLS						
Methods Training: 174 20 Validation: 174 18	Training model accuracy			Testing set accuracy		
	Misclassified	True Alarm	False Alarm	Misclassified	True Alarm	False Alarm
S_{PCA}^{λ} (95%)	14/174	1/20	13/154	13/174	1/18	12/156
S_{ICA} (95%)	16/174	3/20	13/154	16/174	3/18	13/156
MD (95%)	15/174	2/20	13/154	16/174	2/18	14/156
PCA (95%)	15/174	3/20	10/154	15/174	4/18	11/156
ICA-PCA (95%)	15/174	2/20	13/154	14/174	2/18	13/156

Table 4.2 Evaluation of site selection using FDC models

The results of our analysis presented in table 4.2 suggest that clustering-PCA based site selection tends to outperform other methods. The sites selected using the clustering-PCA based method provide monitoring capabilities more or less equivalent to measuring all the sites across the wafer.

4.7. Conclusion and Future work

In this Chapter, we have discussed a variety of methods to explore FDC model creation using hierarchical wafer data. These methods were then explored employing different site selection approaches to reduce manufacturing costs by allowing engineers to measure fewer sites while achieving the same process monitoring capability at the production line level. Similarity factors based on PCA S_{PCA}^{λ} tend to perform best among the described techniques. Similarity factor for consecutive wafers can also be derived to track within-lot variation as shown in figure 4.41, where there is a spike in statistic values when the lot is changed. Future work can be extended to create an online FDC model that employs a state space approach in which the selected sites can be robust enough to capture process dynamics.

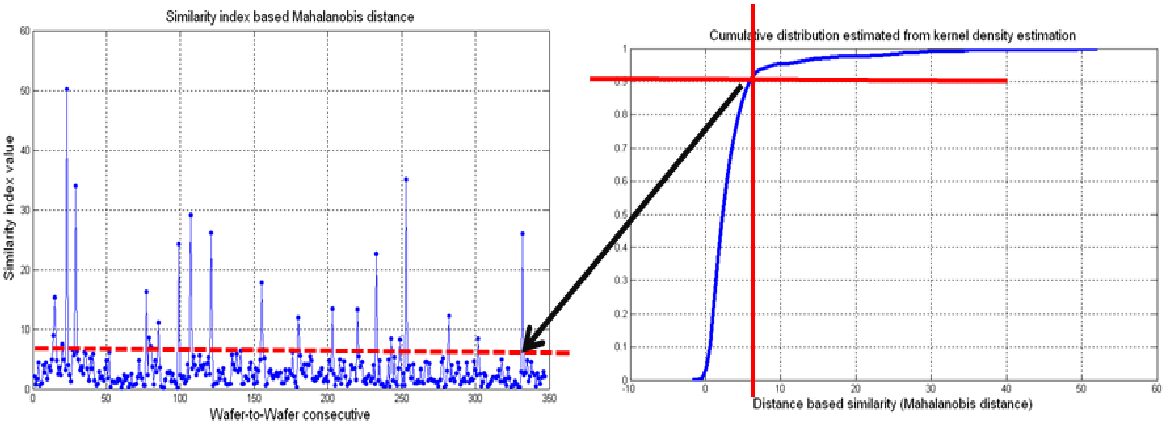


Figure 4.41 Consecutive Wafer Similarity index from MD

References

[4.1] Vincent, Tyrone, Broc Stirton, and Kameshwar Poola. (2011) “Sampling Approaches to Metrology in Semiconductor Manufacturing.” *IEEE Transaction of Manufacturing*.

[4.2] Friedberg, P., Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. (2005). “Modeling within-die spatial correlation effects for process-design cooptimization.” in *Proc. 6th Int. Symp. Quality Electron. Des.*, Mar. 2005, pp. 516–521.

[4.3] Boning, D., J. Chung, D. Ouma, and R. Divecha,(1997). “Spatial variation in semiconductor processes: Modeling for control.” in *Proceedings Process Control Diagnostics and Modeling in Semiconductor Manufacturing II Electrochemical Society Meeting*, May 1997.

[4.4] Cheng, L. , P. Gupta, C. Spanos, K. Qian and L. He.(2009). “Physically Justifiable Die-Level modeling of Spatial Variation in View of Systematic Across Wafer Variability.” *IEEE Design Automation Conference*, pp. 104-108, July 2009.

[4.5] K. Qian and C. J. Spanos. (2009). “Hierarchical modeling of spatial variability of a 45 nm test chip.” *Proc. SPIE*, vol. 7275, pp. 234–238, Mar. 2009.

[4.6] Rossi, R.E., Mulla DJ, Journel AG, Franz EH. (1992). “Geostatistical tools for modeling and interpreting ecological spatial dependence.” *Ecol Monogr* 62:277–314.

[4.7] Sherman, Michael. (2011).*Spatial Statistics and Spatio-Temporal Data: Covariance Functions and Directional Properties.* New York: John Wiley & Sons, Ltd.

[4.8] Cressie, N. and D. M. Hawkins. 1980. “Robust Estimation of the Variogram.” *Mathematical Geology* Vol. 12, No. 2, pp. 115-125.

[4.9] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D.& Altman, R.B.(2001). “Missing value estimation methods for dna microarrays.” *Bioinformatics* 17(6), 520–525.

[4.10] Sani Nassif Analyzing the impact of process variations on parametric measurements: novel models and applications Proceedings if the Conference on Design, Automation and Test in Europe 2009

[4.11] Müller, W.G. (1999). “Least-squares fitting from the variogram cloud.” *Stat. Probabil. Lett.* 43, pp. 93–98.

[4.12] Anselin, L. (1996). “The Moran Scatterplot as an ESDA Tool to Assess Local Instability in Spatial Association.” In *Spatial Analytical Perspectives on GIS in Environmental and Socio-Economic Sciences*, 111–25, edited by M. Fischer, H. Scholten, and D. Unwin. London: Taylor and Francis.

[4.13] Frommelt, Ester, M., A., Kriegel, H.-P., and Sander, J. (1998). Algorithms for Characterization and Trend Detection in Spatial Databases, *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY.

[4.14] Bivand, R. S., and B. A. Portnov. (2004). “Exploring Spatial Data Analysis Techniques Using R: The Case of Observations with No Neighbors.” In *Advances in Spatial Econometrics: Methodology, Tools and Applications*, 121–42, edited by L. Anselin, R. J. Florax, and S. J. Rey. Berlin: Springer-Verlag.

[4.15] Lowry, C. A. and D. C. Montgomery. (1995) "A review of multivariate control charts." *IIE Transactions*, vol. 27, pp. 800-810.

[4.16] Gnanadesikan, R., and J.R. Kettenring (1972). “Robust estimates, residuals, and outlier detection with multi-response data.” *Biometrics* 28:81-124.

[4.17] Singhal A., Seborg D. E. (2002). “Pattern matching in historical batch data using PCA.” *IEEE Control Systems Mag*. 22: 53–63.

[4.18] Ge, Z., Z. Song. (2007). “Process monitoring based on independent component analysis–principal component analysis (ica–pca) and similarity factors.” *Industrial & Engineering Chemistry Research* 46 (7) 2054–2063.

[4.19] Rousseeuw P. J., and A.M. Leroy. (1987). *Robust Regression and Outlier Detection*. New York: John Wiley and Sons Inc.

[4.20] Myers, C.S. and Rabiner, L.R. (1981). “A comparative study of several dynamic time-warping algorithms for connected word recognition.” *The Bell System Technical J.* 60 (7), 1389--1409.

[4.21] von Luxburg, U. (2006). “A tutorial on spectral clustering.” *Tech. Rep. 149, Max Planck Institute for Biological Cybernetics*, August 2006.

[4.22] Bach F., and M. Jordan. (2004). “Learning spectral clustering.” In *Proc. of NIPS-16.* Cambridge MA: MIT Press.

[4.23] McCabe, G. P. (1984). “Principal Variables.” *Technometrics*, 26(2), 137-144.

[4.24] Sharma, S. (1996). *Applied Multivariate Techniques*. New York: John Wiley & Sons, Inc.

[4.25] Rencher, A. C. (1995). *Methods of Multivariate Analysis.* New York: John Wiley & Sons Inc.

Chapter 5

Predictive Modeling of Solar Cell Variability

Photovoltaic (PV) cells manufactured with different technologies and process flows are expected to exhibit variability in performance. Variations in the current and voltage characteristics lead to power loss known as mismatch when the cells are connected in series and in parallel within a network. The result of mismatch is that the maximum output power of the network will be less than the sum of the maximum output powers of the individual cells in the network [5.1].

It is essential to characterize such performance distribution in order to reduce mismatch induced power losses. If the distributions of cell characteristics such as short circuit current (I_{sc}), open circuit voltage (V_{oc}) and maximum power output (P_{mp}) are known, then one can design a cost effective PV plant layout with the help of a PV array performance simulation produced using a SPICE simulator. In figure 5.1, we show a performance distribution plot obtained from BP solar. It is clear that the performance of cells follows a Gaussian distribution.

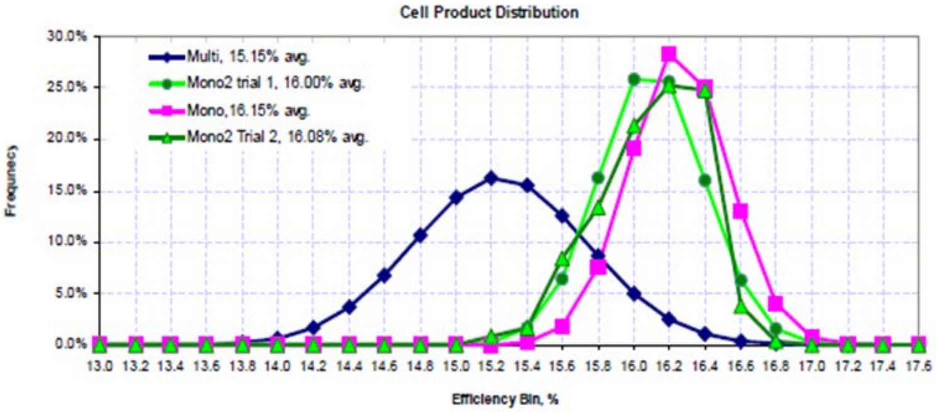


Figure 5.1 Distribution of finished cells at BP solar [source: BP solar]

In this chapter, our goals are to combine statistical techniques and the SPICE simulator to understand how environmental effects and process variations can impact final solar power output.

In Section 5.1, we briefly describe how solar cells operate and their dependence on material properties and environmental conditions. The dependencies are shown using simulation results from SPICE and PC1D. In Section 5.2, we perform a simulation in SPICE to understand how mismatch depends on variations in cell's characteristics. Then a statistical model is created to predict mismatch power loss, given cell network configuration and cell performance. The model results are compared with the results from a Bucciarelli [5.11] model. The possible binning of cells using a multivariate technique is also discussed alongside the simulation results. In Section 5.3, a SPICE based diode array simulator is implemented to understand how nonuniformities at the cell level can impact final cell electrical performance. The nonuniformities considered in this thesis are defect concentration dependent. A variety of defect patterns are created according to different spatial probability classes. Then, spatial statistics are extracted for each pattern, and a statistical model is created to capture the how spatial nonuniformities can affect cell performance. In figure 5.2, we show how different variability components get lumped into the solar power system.

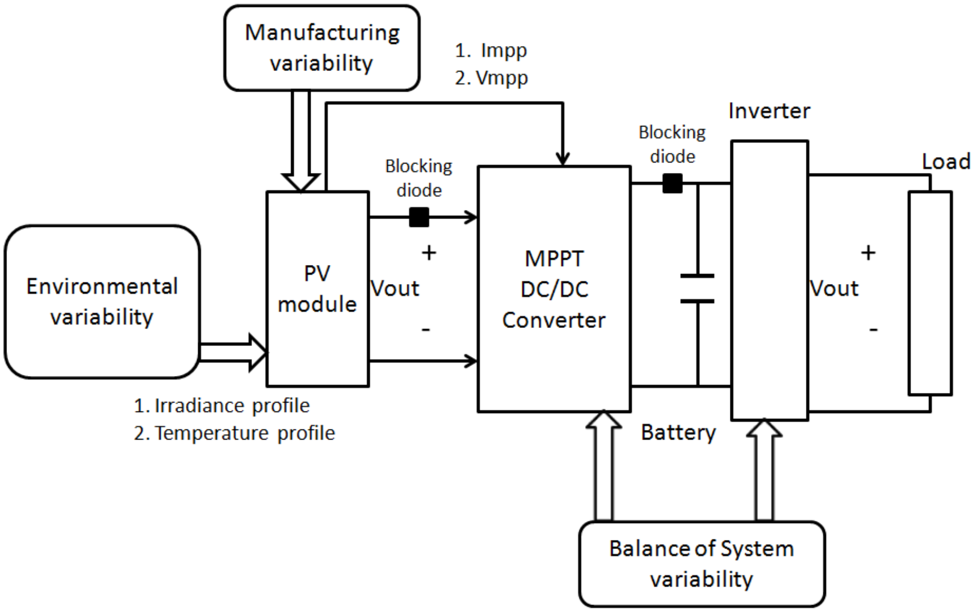


Figure 5.2 Solar power system components

From figure 5.2, we can see there are three primary types of variability source contributing to PV power output variability. *Environmental variability* includes weather, PV module layout orientation and site locations. *Manufacturing variability* includes fluctuations in electrical properties and the reliability of PV modules resulting from process variations. *Balance of system variability* includes variations in inverter performance, power cable resistance fluctuations and instability of maximum power point tracking (MPPT) control algorithms.

5.1. Solar Cell Characteristics and Operation

Figure 5.3 represents a typical silicon based solar cell structure with its corresponding operation under irradiation.

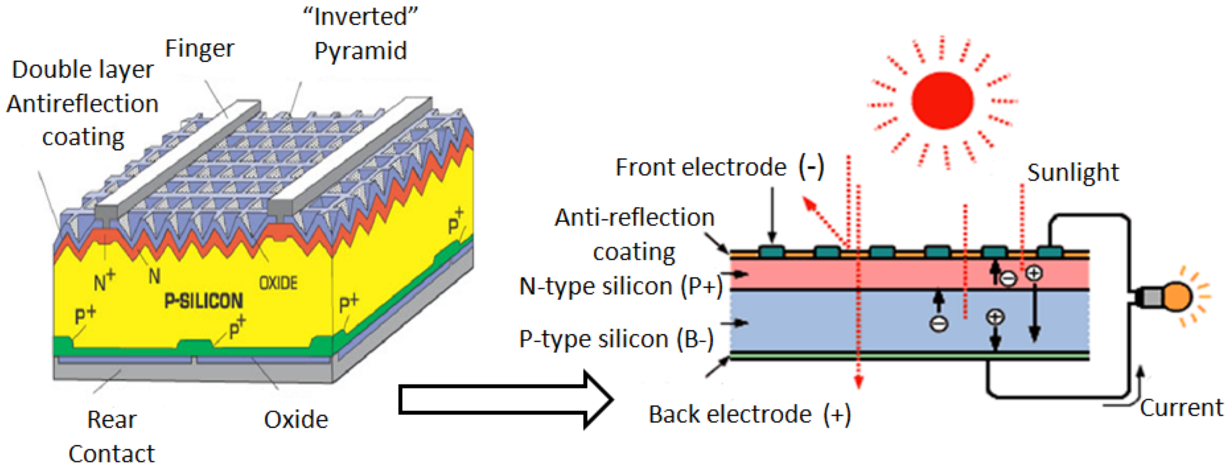


Figure 5.3 Typical silicon based solar cells [5.2]

In order to understand the behavior of solar power output under different operating conditions, it is necessary for a PV power plant designer to simulate solar cell performance in a circuit simulator for optimal design. The common equivalent circuit model applied to represent a solar cell is a single-diode model, but increased understanding of current loss due to recombination mechanisms has led to a 2-diode model being adopted for modeling. Especially in the case of thin film solar cells, the 2-diodes model is commonly chosen, since thin films are rich in grains and defects, and current loss due to recombination is more dominant [5.3]. The standard testing conditions (STC) for solar cell measurement are 1kW/m^2 irradiation and Air Mass 1.5 [5.4] spectrum at 25℃. For all existing PV technologies, it is common to observe a substantial discrepancy, around 30 to 40%, between "best of breed" laboratory cell efficiency and production cells. In figure 5.4, we show the IV curve of solar cells with critical performance parameters labeled.

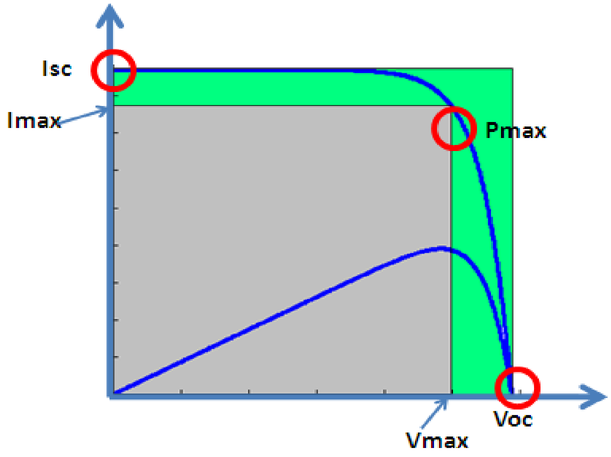


Figure 5.4 IV curve for solar cell

As shown in figure 5.4, the electrical performance of a solar cell can be characterized by five electrical parameters: I_{sc}, V_{oc}, I_{mp}, V_{mp}, FF [5.4]. To model the electrical behavior of a solar cell, an equivalent circuit model is needed. In this work, we have employed the 2-diode model for modeling individual cells. The 2-diode model shown in figure 5.5 comprises two diodes to represent the loss of current due to the physical properties of PV cells, one lumped series resistor (R_s) to account for resistive loss of photo-generated current due to bulk resistivity, contact resistance, a metal grid and current collection bus bar.

The lumped series resistor might work well for assuming uniformity across the solar cell, but decomposition of this resistor into a distributive network of resistors for different loss mechanisms might be necessary for an accurate modeling of thin film solar cells. R_s needs to be as small as possible. Another lump parameter is R_{sh}, which is used to account for the leakage currents and shunt connection of PN junctions of the cell. The value of R_{sh} needs to be as large as possible.

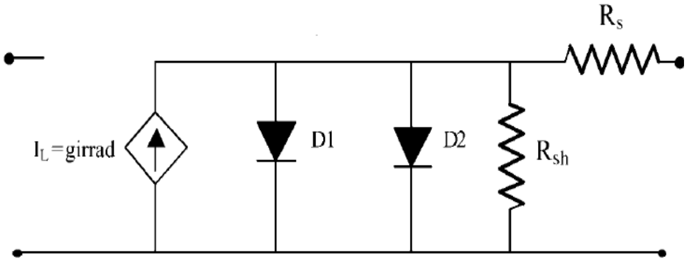


Figure 5.5 2-diode equivalent circuit model for solar cell

The diode *D1* represents the diffusion current loss of a solar cell; the relative non-ideality of a solar cell compared to an ideal diode is accounted for by adding an *n* factor to the diode current equation. The diode *D2* represents the current loss due to recombination at the space charge region for a solar cell. The recombination loss is more severe for thin film cells.

The IV curve for solar cells can be described with following equations [5.5].

$$\mathrm{I} = \mathrm{I_{ph}} - \mathrm{I_{01}}\left(\mathrm{e}^{\frac{\mathrm{q(V+IRs)}}{\mathrm{nkT}}} - 1\right) - \mathrm{I_{02}}\left(\mathrm{e}^{\frac{\mathrm{q(V+IRs)}}{\mathrm{nkT}}} - 1\right) - \frac{\mathrm{V + IRs}}{\mathrm{Rsh}} \tag{5.1}$$

$$\mathrm{FF} = \frac{\mathrm{I_{mp} * V_{mp}}}{\mathrm{P_{in}}} \tag{5.2}$$

$$\eta = \frac{\mathrm{I_{mp} * V_{mp}}}{\mathrm{P_{in}}} = \frac{\mathrm{V_{oc} I_{sc} FF}}{\mathrm{P_{in}}} \tag{5.3}$$

Parameter definitions for a solar cell are listed below:

η Efficiency of the solar cell (%)

$\mathrm{P_{in}}$ Total incident power (W)

$\mathrm{I_{sc}}$ Short circuit current (A)

$\mathrm{I_{ph}}$ Photo-generated current (A)

$\mathrm{I_{01}}$ Saturation current of Diode 1 (diffusion diode) (A)

$\mathrm{I_{02}}$ Saturation current of Diode 2 (recombination diode) (A)

$\mathrm{I_{mp}}$ Current at the maximum power point (A)

$\mathrm{V_{oc}}$ Open circuit voltage is also a function of temperature (V)

$\mathrm{V_{mp}}$ Voltage at the maximum power point (V)

$\mathrm{P_{mp}}$ Power at maximum power point (W)

FF Fill factor (dimensionless)

K Boltzmann's constant, 1.381E-23(J/K)

Q Electron charge, 1.60218E-19 (coulomb)

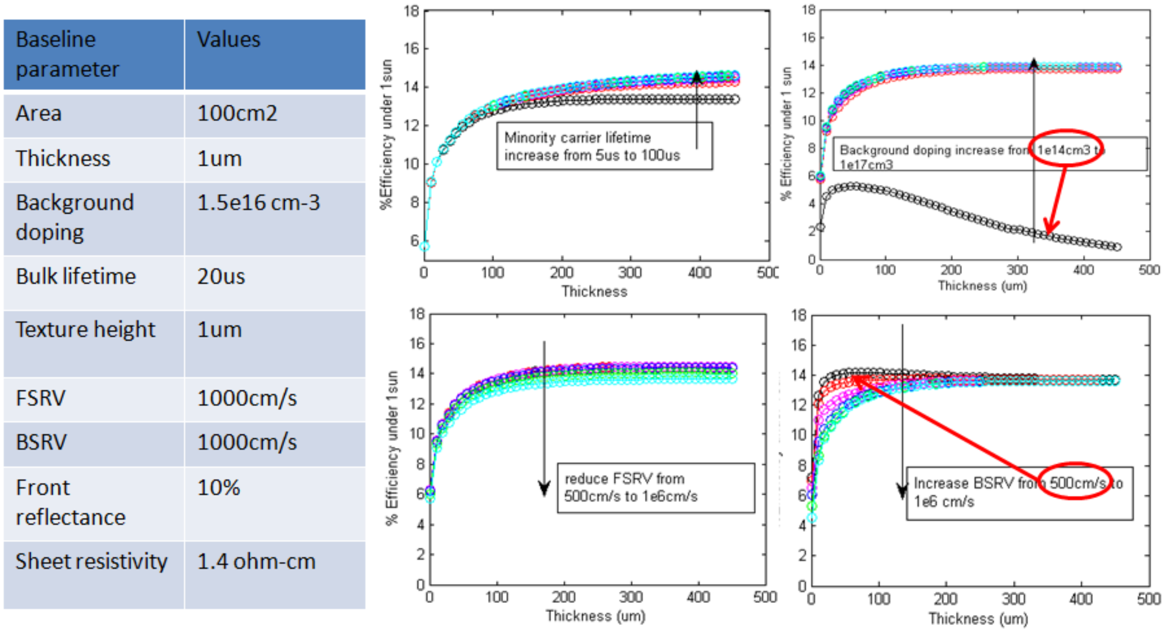
The illumination of PV cells generates a free charge carrier, which allows current flow through the connected load. The I_{ph} is proportional to the incident irradiance and ambient temperature of the cell, and thus, PV modules are subjected to environmental variability, which can be characterized by spatial variations of I_{ph} for individual cells across entire PV array. I_{ph} is also directly proportional to $\mathrm{I_{sc}}$, and they are ideally identical. $\mathrm{I_{sc}}$ and $\mathrm{V_{oc}}$ are both material properties dependent, while $\mathrm{V_{oc}}$ can be also expressed in terms of I_{ph} [5.6],

$$\mathrm{V_{oc}} = \frac{\mathrm{nkt}}{q} \ln\left(\frac{\mathrm{I_{ph}}}{\mathrm{I_{01}}} + 1\right) \tag{5.4}$$

where we have assumed the current loss due to the recombination diode, and where series and shunt resistances are relatively small. This is generally true in practice, given that $\mathrm{I_{02}}$ is often smaller than $\mathrm{I_{01}}$ by several orders of magnitude.

Manufacturing induced variability for solar cells is reflected in the variation of a cell's physical properties. To understand how variation in physical material properties can impact final cell efficiency, we employed a PC1D [5.7] device simulator to model the impact of physical properties on solar cell efficiency. The simulation was done with an N+PP+ single crystalline silicon-based solar cell with an aluminum back surface field [5.7]. The area was kept at 100 cm^2, and front internal reflectance was set to 10%, to indicate that only 90% of incident light would pass through the interface. Texture height was kept at 1μm to trap more light. The emitter sheet resistivity of all simulated devices was 1.4 ohm-cm^2. Five physical parameters were varied across a large range: thickness, background doping concentration, bulk lifetime, front surface recombination velocity (FSRV) and back surface recombination velocity (BSRV).

The simulation results are shown in figure 5.6. The left table in figure 5.6 shows the baseline cell parameters, while the right plots are the simulation results of varying each physical property independently. We plot the efficiency vs. thickness of cell due to the fact that one of the limiting costs for Si-wafer based PV cells is the wafer thickness.



Baseline parameter	Values
Area	100cm2
Thickness	1um
Background doping	1.5e16 cm-3
Bulk lifetime	20us
Texture height	1um
FSRV	1000cm/s
BSRV	1000cm/s
Front reflectance	10%
Sheet resistivity	1.4 ohm-cm

Figure 5.6 Effect of device parameter on solar cell efficiency

The simulation results shown in figure 5.6 indicate that the bulk minority carrier lifetime is directly proportional to the final cell efficiency in the higher thickness region, and that BSRV is inversely proportional to final cell efficiency in the lower thickness region. Higher bulk minority carrier lifetime refers to material of higher purity, and lower BSRV means the back contact surface of the cell has a lower defect density. We have summarized the relationship between physical properties and electrical performance of cells in table 5.1 below, where + sign indicates directly proportional relationship and – sign indicates inversely proportional relationship.

Physical parameter	Isc	Voc	Pmax	FF	Efficiency
Cell thickness	+	+	+	+	+
Device area	+	+	+	+	+
Background doping	–	+	+	+	+
Texturing	+	NA	+	+	+
Junction depth	–	–	–	–	–
Bulk lifetime	+	+	+	+	+
FSRV/BSRV	+	+	+	+	+
Metal Contact	+	+	+	+	+

Table 5.1 Material properties of solar cell vs. electrical performance

In order to understand the dependence of cell performance on equivalent circuit parameters, the effects of series resistance, shunt resistance, temperature, and irradiance profiles are investigated by performing a simulation in SPICE with the 2-diode model mentioned above. In Figure 5.7, we show the IV and PV curves of a solar cell when we sweep R_{sh} from 1000 Ω to 1 Ω and R_s from 0.001 Ω to 1 Ω. And in figure 5.8, we show the IV and PV curves of a solar cell when sweeping I_{ph} from $1.34A$ to $4.35A$ and sweeping I_{01} from $1e^{-5}\ A$ to $1e^{-8}A$. Other parameters are assumed as follows for all sweeping conditions: $I_{01} = 1e^{-6}A$, $I_{02} = 1e^{-12}\ A$, $R_s = 0.001\ \Omega$, $R_{sh} = 1000\ \Omega$ and $I_{ph} = 4.35\ A$. The simulation results suggest that contact resistance related to R_s and I_{ph} has the most significant impact on final power output.

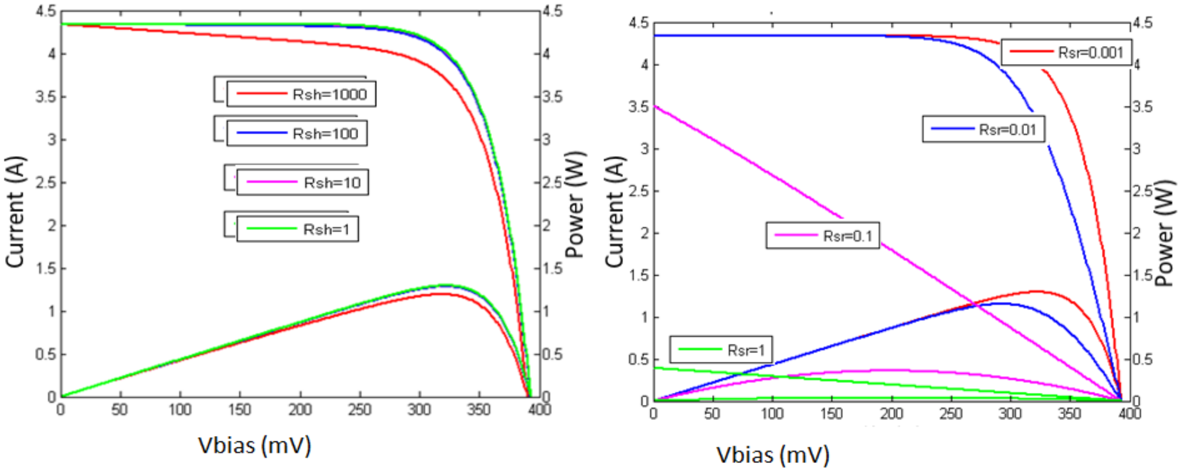


Figure 5.7 Effect of R_{sh} and R_s on IV curve

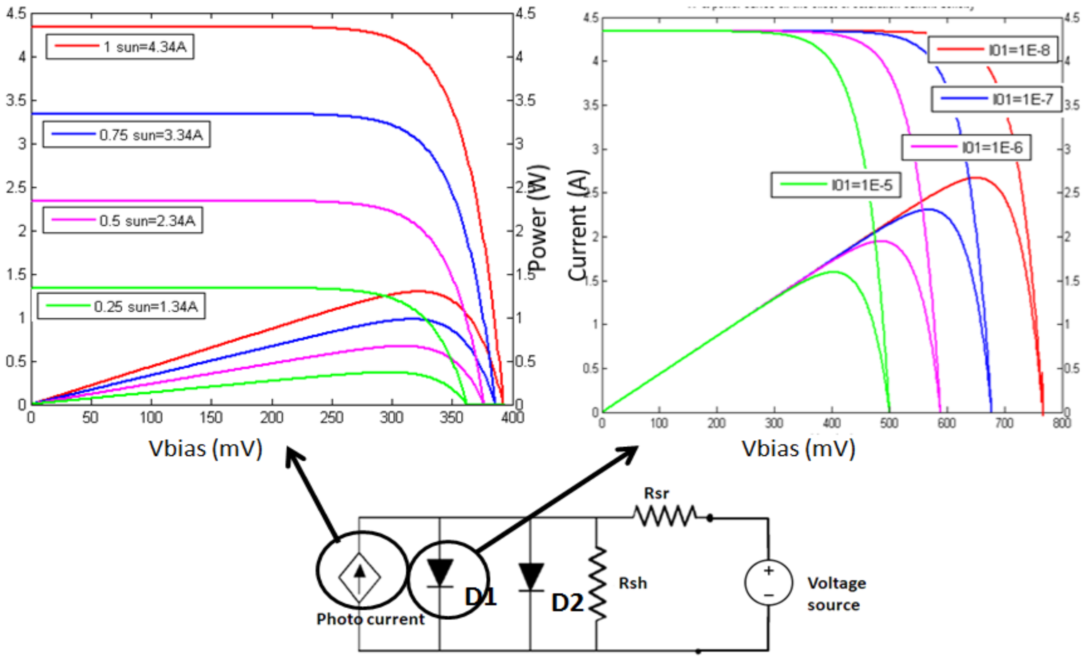


Figure 5.8 Effect of saturation current and photo-current on IV curve

The final output power of a solar cell is the result of different loss mechanisms, including limitations of material properties and environmental conditions. The summarized efficiency loss mechanisms are presented in figure 5.9. The loss of efficiency for a typical solar cell can be categorized into three factors: loss in V_{oc}, loss in I_{sc}, and loss due to parasitic resistance. V_{oc} is highly correlated to the defect density across cells and thickness of cell, while I_{sc} is dependent on how well the cell material can absorb sunlight. Parasitic resistance accounts for contact resistance and shunt path across junction.

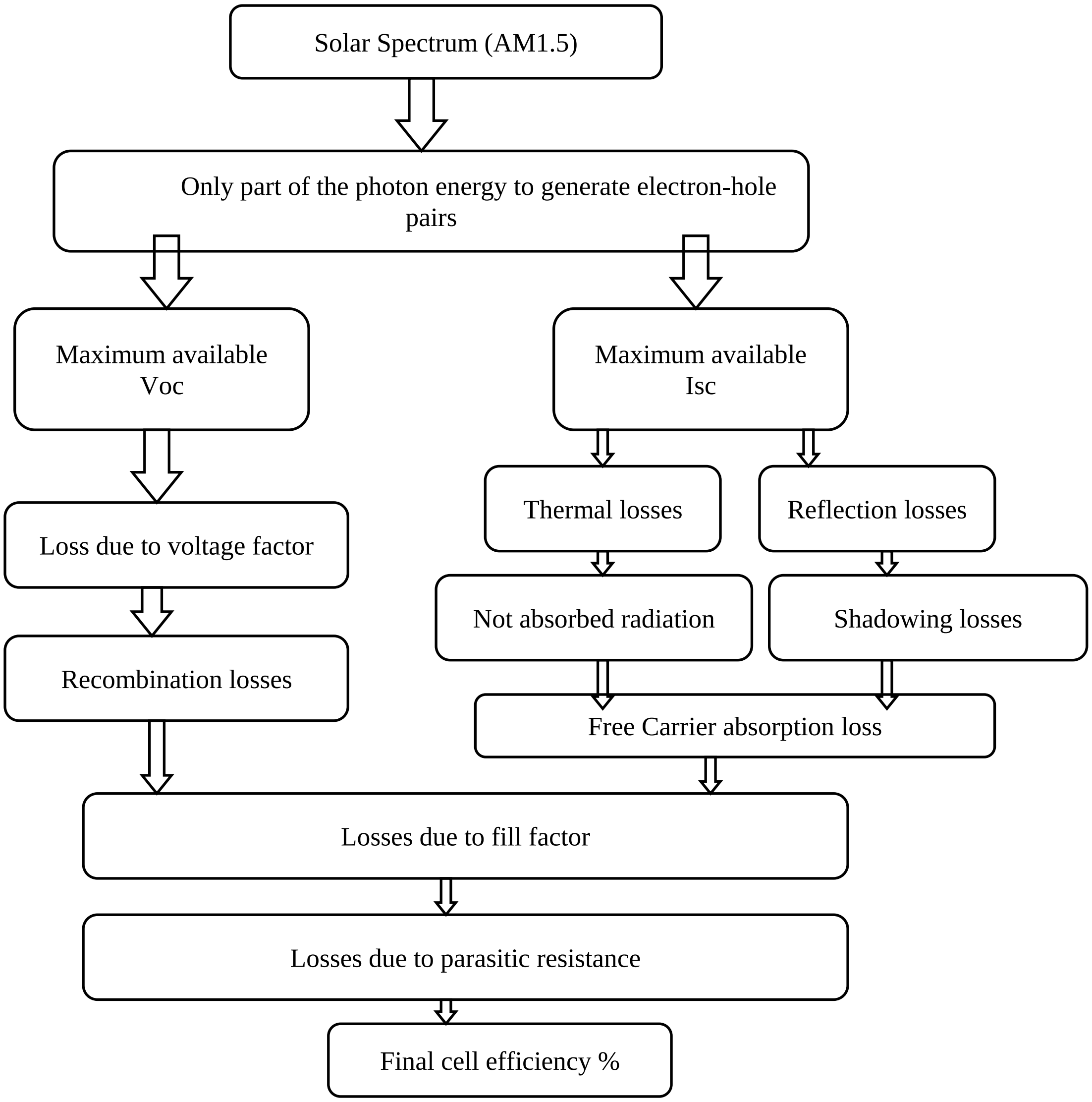


Figure 5.9 Efficiency losses mechanisms for solar cell [5.8]

5.2. Investigation of Mismatch for Solar PVs

As we have shown in figure 5.1, the performance of cells coming from the same production line can vary significantly. Even though binning has been done to group cells with similar performance together in order to construct a solar array, mismatch is still inevitable, given the statistical nature of cell characteristics. In addition to the manufacturing mismatch, environmental conditions also impose mismatch loss on cell arrays. It is important to have full insight as to how these variations can impact final power output performance when forecasting the energy output of solar power facilities.

To reduce the effects of mismatch induced losses, PV engineers have focused primarily on three alternatives. The first is to perform binning to reduce the variation of cell characteristics due to production. The second one is to utilize different PV interconnect configurations for cell arrays. The third alternative is to create prediction models that can capture the relationship between cell characteristics and final PV power output. Kaushika [5.10] investigated the mismatch losses of a network of cells with experimental measurements of series cell strings and parallel strings where the number of cells connected in a series string varies, as does the number of cells connected in a parallel string. Bucciarelli's model [5.11] was used to estimate the mismatch losses of fresh and aging cells. In this section, we investigate the mismatch induced power loss at the array level for different cell network configurations, as well as varying environmental conditions and propose a statistical model for predicting the mismatch induced power loss. We also explore statistical methods, including Kriging and NN, for accurate irradiance profile interpolation where the environmental variations can be coupled into PV power output forecasting cost effectively.

5.2.1. Concept of Mismatch for Solar Cells

In this section, we illustrate how cell variation can impact the final performance of parallel and series arrays using a SPICE simulation. The electrical parameters tuned to reflect changes in cell characteristics include: I_{ph}, R_{sh}, R_{sr}, and V_{oc}. For serial array simulation, we have two cells connected in series, one with higher $\mathrm{I_{ph}}$ and one with lower $\mathrm{I_{ph}}$. The other electrical parameters were carefully tuned to ensure that we have significantly different $\mathrm{I_{sc}}$ and similar $\mathrm{V_{oc}}$. For the parallel array simulation, we have two cells connected in parallel, where the electrical parameters were tuned to ensure we have significantly different $\mathrm{V_{oc}}$ and similar $\mathrm{I_{sc}}$. The results are shown in figure 5.10.

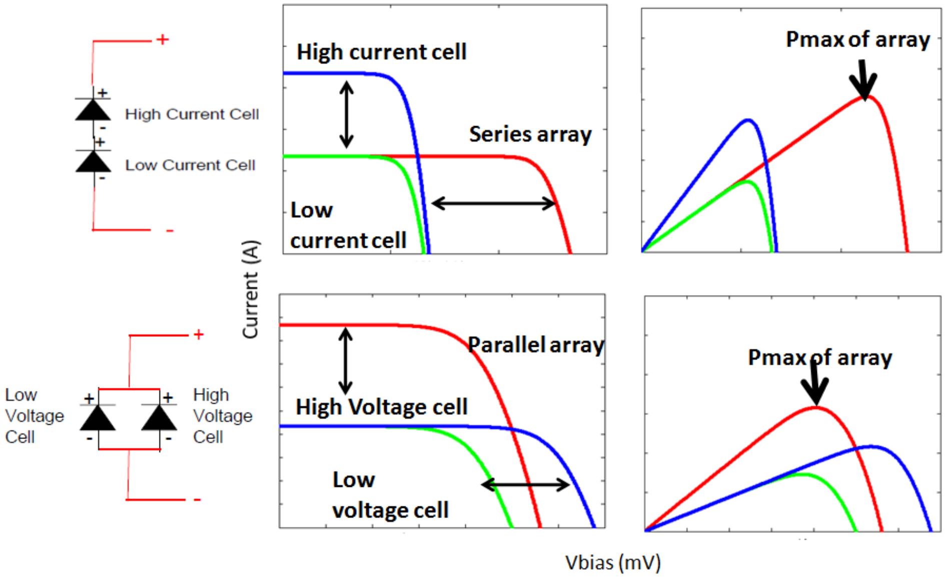


Figure 5.10 IV curves for mismatched combinations of two solar cells

In figure 5.10, the top two plots represent the serial array, while the bottom two plots depict the parallel array. In the serial array, we can see that the main limiting factor for mismatch induced power loss is the minimum maximum current of cells. In the parallel array, the main limiting factor of mismatch effects is the minimum maximum voltage of cells. Therefore, various configurations of cell networks have been constructed to address mismatch effects, as well as to maintain specific levels of output voltage and current to meet regulations. The most common PV interconnect configurations are shown in a later section.

5.2.2. Environmental Variation

In addition to the mismatch effect originating in process variation, environmental variations also impose mismatch losses. Such environmental variations include uneven distribution of the solar irradiance profile, and the degradation of cell properties over time due to field exposure. Uneven distribution of the solar irradiance profile generally results from the shading of cell surfaces as well as distributive nature of incident solar beams.

Mismatch induced power loss due to degradation of cells generally increases with the amount of time the solar cells are used in the field. There are four main causes of the degradation of solar cells, including: material property-related, hot spot heating, surface soiling, and optical degradation of the module. It is difficult to obtain field data on degradation, because most investigations have been done using accelerated field tests in a laboratory. However, Ries [5.9], at the Schatz Energy research center, CA, compared PV module performance before and after 11 years of field exposure. In that study, 191 PV modules were measured at normal operating cell temperature (NOCT) in 1990 and 2001. The data indicated that the variability in PV module performance increases after 11 years of field exposure. The results of this experiment are shown in figure 5.9. From the distribution of $\mathrm{P_{mp}}$ in figure 5.11, we can see that the variance of distribution becomes larger for p_{mp}; this is mainly due to increased variance in the I_{sc}. There have also been several review papers discussing how the performances of in-field solar panels vary over time and evoking possible causes for this variability. These findings suggest that PV modules do not fail catastrophically, but tend to degrade over time in a two stage process: rapid degradation ranging from 1% to 3% within the first couple years after exposure to sunlight, and then a slower degradation rate ranging between 0.25 to 1%/year. Based on this observation, one can propose a probabilistic approach to the model to predict the degradation of PV modules in the field and its impact on mismatch-induced power loss over time.

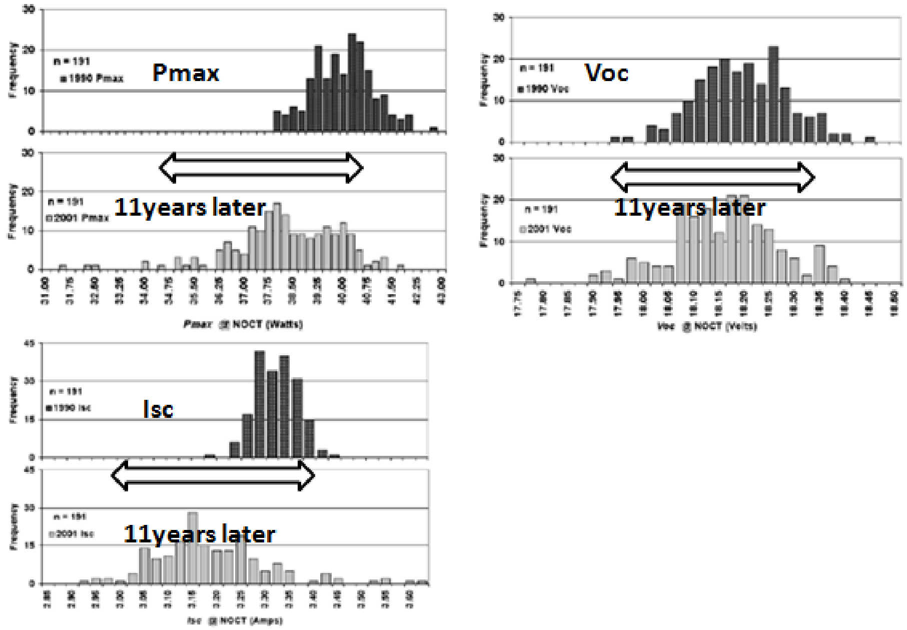


Figure 5.11 In-field measurement of solar cell characteristic over time [5.9]

5.2.3. Simulation of Mismatch Losses for Lumped Variations

In this section, we use MC simulations to show that that it is necessary to combine manufacturing and environmental variability at the small level to obtain accurate power output. The simulation contains 2000 runs of a 21x41-cells array, 21 cells for each series string and 41 parallel strings. Environmental variability is introduced for every run by imposing a diffused irradiance profile generated by a uniform distribution with upper and lower bounds at $[0.64\ sun\ 1\ sun]$, while temperature at each cell level is interpolated based on a linear relationship with the cell-level irradiation. Manufacturing variability is introduced by sampling each equivalent circuit parameter $I_{ph}, I_{01}, I_{02}, R_{sr}$, and R_{sh} from Normal distributions. The P_{mp} of the 21x41-cells array is then computed using the SPICE simulator. The results are then compared against simulation results with a uniform irradiance profile [0.8 sun] across the array, and identical manufacturing variability. The simulation concept is shown in figure 5.12 and distributions of output P_{mp} are shown for both cases in figure 5.13.

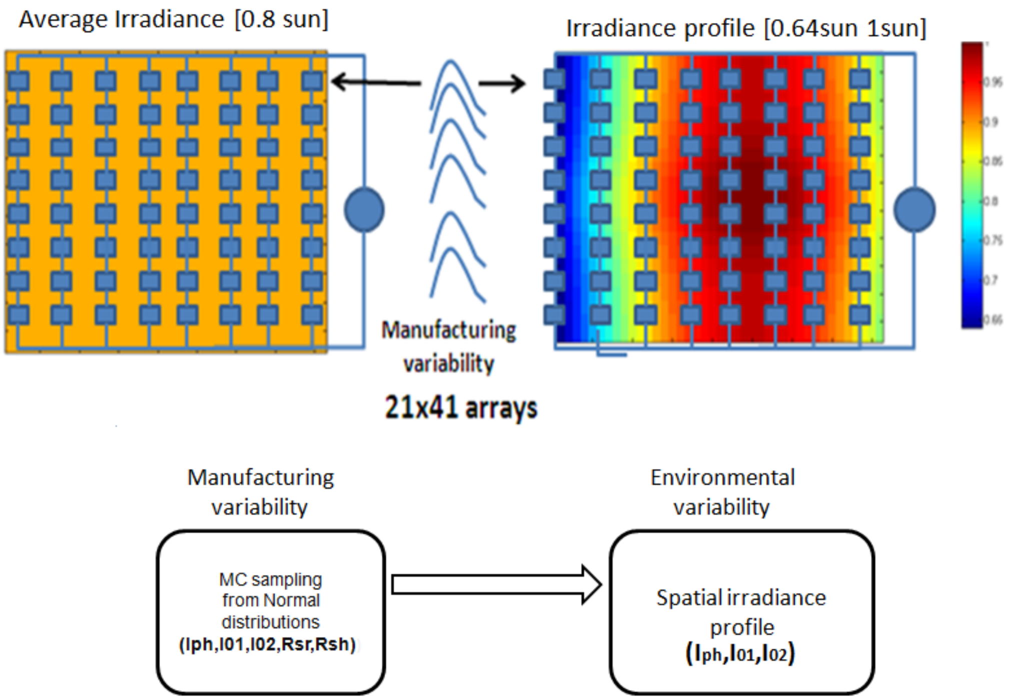

Average Irradiance [0.8 sun]
Irradiance profile [0.64sun 1sun]
0.95
0.9
0.85
0.8
0.75
0.7
0.65
Manufacturing variability
21x41 arrays
Manufacturing variability
Environmental variability
MC sampling from Normal distributions (Iph,I01,I02,Rsr,Rsh)
Spatial irradiance profile (Iph,I01,I02)

Figure 5.12 Simulation of manufacturing variability and irradiation variability

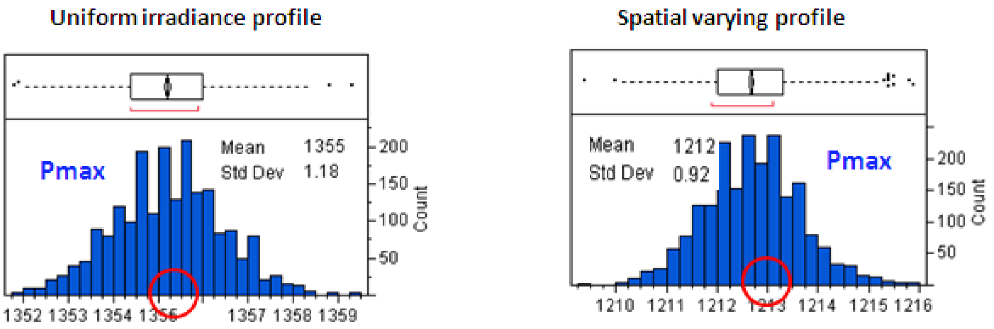


Figure 5.13 Distribution of electrical performance of arrays

We can observe, looking at figure 5.13, that without consideration for environmental variability, the P_{mp} is overestimated by 10% power. This significant reduction in output power suggests one should consider such irradiance profile changes when forecasting the energy power output for PV arrays.

5.2.4. Simulation of Alternative Interconnect Configuration

To increase tolerance for mismatch-induced loss, different PV configurations can be adopted, subject to cost considerations. Bridge-link and Total-Cross-Tied are two common alternate interconnect configurations.

In figure 5.14, we present the IV curves of several configurations for partially shaded and unshaded cells, to illustrate how electrical performance can be protected using alternative configuration. All unshaded cells are assumed to have the same characteristics, with $I_{ph} =$

$4.35A$, $R_{sh} = 1000$ ohm, $R_s = 0.001$ ohm and $V_{oc} = 0.58V$, while shaded cells are assumed to have $I_{ph} = 2.18A$ and all other characteristics remaining the same. It is clear that the Bridge-link alternative offers slightly higher P_{mp} in the presence of shaded cells. For a large PV plant, such a configuration can avoid a significant amount of mismatch loss caused by the presence of shading.

In this thesis, we have also simulated 12 common configurations that have been discussed in the literature. Instead of simulating at the module level, we simulated at the cell level, where each unit in the configuration is a cell rather than module. The 12 configurations are shown in figure 5.15.

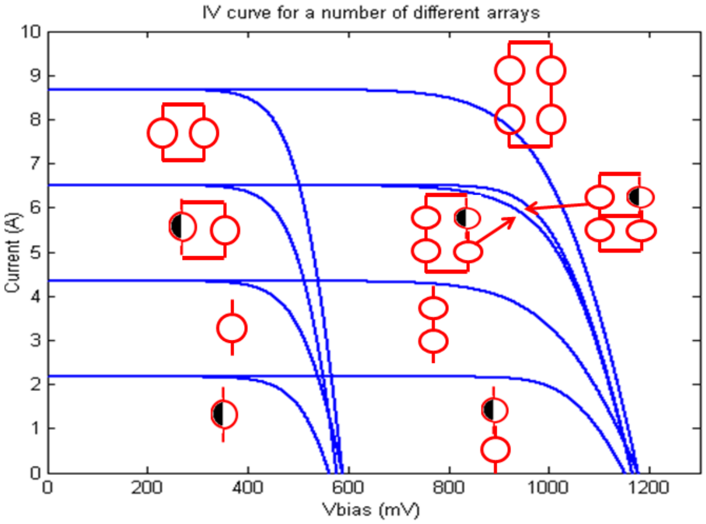
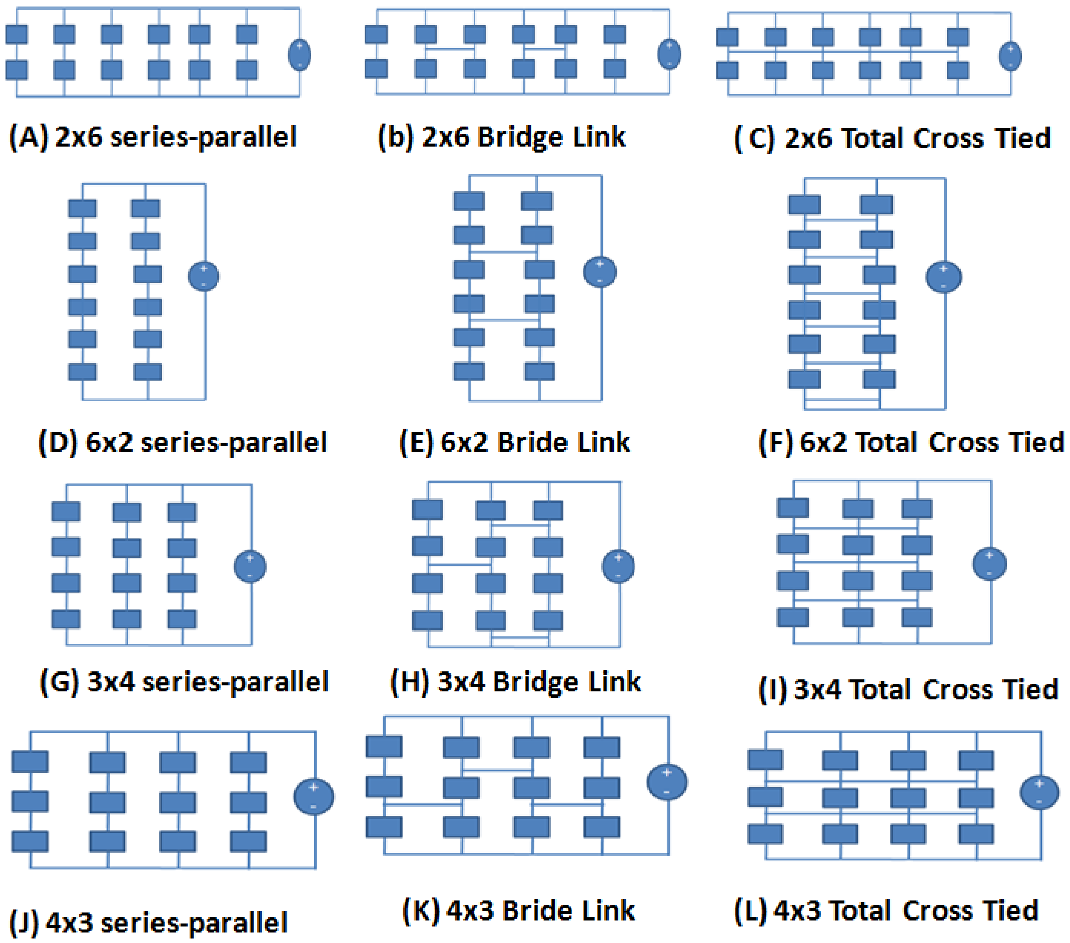


Figure 5.14 IV curves for different network configurations



(A) 2x6 series-parallel

(b) 2x6 Bridge Link

(C) 2x6 Total Cross Tied

(D) 6x2 series-parallel

(E) 6x2 Bride Link

(F) 6x2 Total Cross Tied

(G) 3x4 series-parallel

(H) 3x4 Bridge Link

(I) 3x4 Total Cross Tied

(J) 4x3 series-parallel

(K) 4x3 Bride Link

(L) 4x3 Total Cross Tied

Figure 5.15 PV interconnect configuration or mismatch loss reduction

In order to simulate the effects of mismatch for each configuration, we have sampled each cell characteristic randomly from among the following normal distributions with $[\mu\ \sigma]$ as: $I_{ph} = [4.35A\ \ 0.435A]$,$I_{01} = [1e^{-6}A\ 1e^{-7}A]$,$R_s = [0.001\ ohm\ 0.0001ohm]$,$R_{sh} = [1000ohm\ 100ohm]$, $V_{oc} = [0.58V\ 0.058V]$. Each PV configuration was sampled 2000 times; the means of the performance parameters for each PV array are shown in table 5.2 below.

Interconnect configuration	I_{SC}	V_{OC}	P_{mp}	% Mismatch
(A)	23.9A	1.04V	19.8W	10.5%
(B)	24.9A	1.09V	21.4W	11.1%
(C)	24.5A	1.08V	20.3W	10.7%
(D)	7.5A	3.15V	18.9W	13.2%
(E)	7.9A	3.38V	20.0W	10.3%
(F)	8.1A	3.41V	21.5W	9.9%
(G)	12.1A	2.11V	20.6W	10.4%
(H)	13.9A	2.32V	23.2W	8.5%
(I)	13.1A	2.42V	22.5W	9.3%
(J)	16.4A	1.58V	22.8W	9.2%

(K)	17.1A	1.69V	23.6W	9.1%
(L)	17.5A	1.71V	23.3W	9.3%

Table 5.2. Electrical performance of different PV interconnect configurations

The results in table 5.2 suggest that serial connections cause PV performance to deteriorate most when there is mismatch among cells. In practice, it is often necessary to reach a certain level of open circuit voltage to meet regulations, and thus serial connection is often used at the module level to connect cells.

5.2.5. Statistical Model for Mismatch Prediction

Given the variations in cell characteristics that result from production and environmental variations, it is desirable to be able to predict the mismatch induced power loss of solar arrays for specific networks.

Bucciarelli [5.11] estimated the mismatch losses of series string, parallel string and derived an explicit equation for more complex networks. In the Bucciarelli model, the variation between cell characteristics and their mean values is assumed to be very small. Also, a constant, ***c*,** related to FF, was introduced to estimate mismatch induced power loss. The equation 5.5 below explains the relationship between ***c*** and FF. A graphic plot is also used to illustrate the nonlinear relationship between ***c*** and FF in figure 5.16.

$$FF = \frac{P_{mp}}{I_{sc}V_{oc}} = \frac{c^2}{(1+c)[c+\ln(1+c)]} \tag{5.5}$$

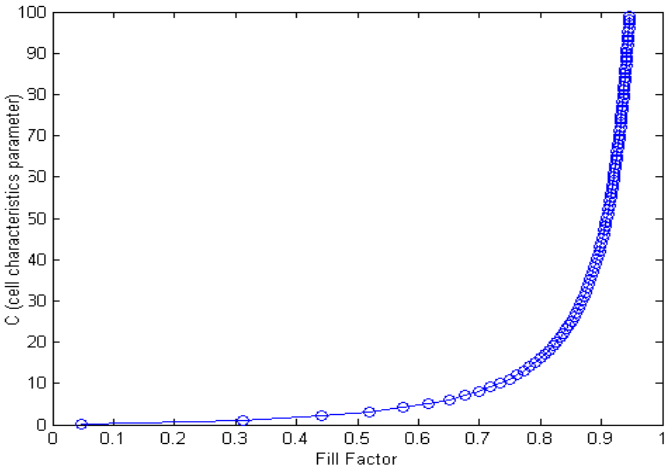


Figure 5.16 c vs. FF

Bucciarelli then derived the estimates of the expected values of maximum power and fractional power loss for a series string, parallel string and complex networks. The estimate equations are shown below.

$$E[\Delta P]_{series} = \frac{(c+2)}{2}\sigma_\eta^2\left(1-\frac{1}{N}\right) \tag{5.6}$$

$$E[\Delta P]_{parallel} = \frac{(c+2)}{2}\sigma_{\xi}^{2}\left(1-\frac{1}{M}\right) \tag{5.7}$$

$$T = LMN \tag{5.8}$$

$$E[\Delta P]_{network} = \frac{(c+2)}{2}\left\{\sigma_{\eta}^{2}\left(1-\frac{1}{T}\right)-\left(\sigma_{\eta}^{2}-\sigma_{\xi}^{2}\right)\frac{N}{T}(M-1)\right\} \tag{5.9}$$

ΔP is the mismatch loss, σ_{η}^{2} is the variance of $\mathrm{I_{mp}}$, σ_{ξ}^{2} is the variance of $\mathrm{V_{mp}}$, L is the number of cells in a single string, N is the number of cells in the series string, and M is number of cells in the parallel string. In the Bucciarelli series array model, individual cells are assumed to exhibit a Gaussian probability density for $\mathrm{I_{mp}}$, with no variations in $\mathrm{V_{mp}}$. For the parallel array mode, Bucciarelli assumes that the cells exhibit variations in $\mathrm{V_{mp}}$ with no variation in $\mathrm{I_{mp}}$. In practice, cells can have variations in both $\mathrm{I_{mp}}$ and $\mathrm{V_{mp}}$, as well as variations in other electrical parameters. Therefore our goal in this section is to create a statistical model that can capture the relationship between variations to electrical parameters and corresponding mismatch losses.

In this work, we have used a Monte Carlo simulation to randomly sample cell characteristics from a defined Gaussian distribution. The samples were generated to ensure that we can cover a range of mismatch loss between 0 % and 95%. In practice, the mismatch loss for a solar array after binning is less than 10%. The variations in cell characteristics can be introduced by imposing a Gaussian distribution on $\mathrm{I_{ph}}$, $\mathrm{I_{01}}$, $\mathrm{I_{02}}$, $\mathrm{R_{sh}}$, and $\mathrm{R_{sr}}$. We then defined several metrics for cell variation, and mismatch loss as a function of these variation metrics;

$$MPL = \frac{\sum_{i=1}^{N} P_{mp\_i} - \sum_{i=1}^{N} P_{mp\_array}}{\sum_{i=1}^{N} P_{mp\_i}} \tag{5.10}$$

$$\Delta X_{max} = \frac{Max(X_{max}) - Min(X_{max})}{Max(X_{max}) + Min(X_{max})} \tag{5.11}$$

$$MPL_{serial} = f(\Delta I_{max}, \Delta V_{max}, \Delta P_{max}, \Delta I_{sc}, \Delta V_{oc}) \tag{5.12}$$

$$\ln\left(MPL_{parallel}\right) = f(\Delta I_{max}, \Delta V_{max}, \Delta P_{max}, \Delta I_{sc}, \Delta V_{oc}) \tag{5.13}$$

Equation 5.10 defines the mismatch loss of a solar array in %. ΔX_{max} indicates the variation of cell characteristics. For a parallel array, logarithmic transformation of the mismatch loss % is used as a response variable. This is due to heteroscedasticity in the generated data, specifically, the tendency for higher value responses to have more variation. Log transform can stabilize the variance of response variables.

In this work, we have utilized linear regression and neural network as our modeling techniques. We first show the results of linear regressions for serial and parallel arrays in figure 5.13. For the proposed linear models, we simulated 2000 samples with MC; 1000 samples are used as a training set, while the other 1000 are designated as a validation set. Each set of samples consists of two cells connected in serial or parallel. As shown in figure 5.17, the validation results of the proposed linear models suggest better prediction power with the new extracted metrics, in comparison to Bucciarelli model, especially when the mismatch loss is located in a small range, which is where most commercial cell arrays would exhibit mismatch loss after the binning process.

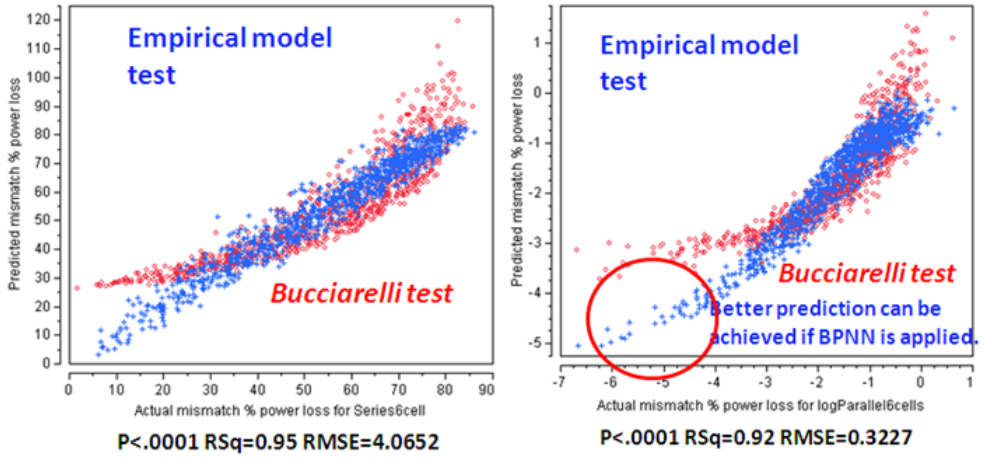
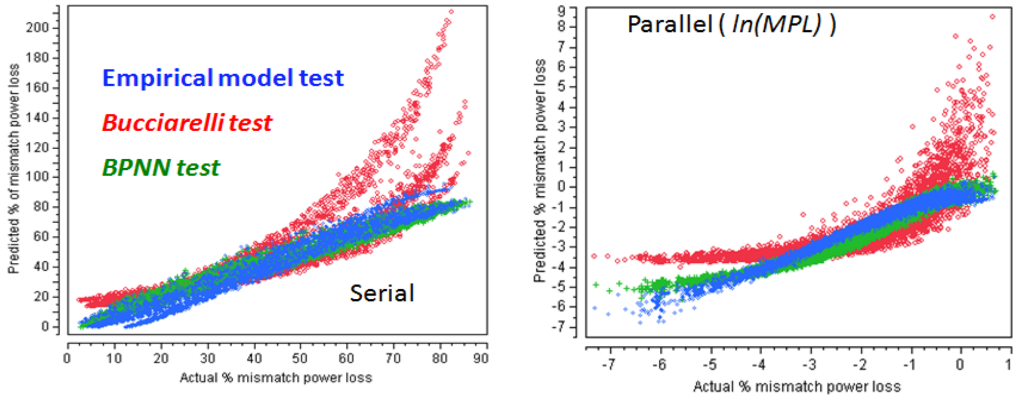


Figure 5.17 Results of linear models for serial (Left) and parallel (Right) array

BPNN was also applied with the new extracted metrics to capture nonlinearity in a PV system. To demonstrate the relative performances of NN and linear models against the Bucciarelli model, we simulated 10,000 samples with MC, of which 5000 samples were used as a training set while the other 5000 were used as a validation set. Each set of samples consisted of two cells connected in serial or parallel. The results of the NN, linear and the Bucciarelli models are shown in figure 5.18.



	Empirical	Bucciarelli	BP Neural Net
Serial	0.98	0.69	0.96
Parallel	0.94	0.58	0.97

Figure 5.18 Results of NN models for serial and parallel array (2-cells case)

The results of both the linear and NN models, as shown in figure 5.14, compare favorably with results of the Bucciarelli model, especially in the region of small mismatch loss, which is the most frequent actual case in practice. In table 5.3, we show the relative performance of each model with relatively simple interconnects. In table 5.4, we show the relative performance of each model with the alternative PV interconnect configuration discussed in Section 5.2.4. Each prediction model is based on 10,000 generated samples, with 5000 being a training set and the remainder designated as testing. R^2 is chosen as the performance index in this case.

PV interconnect	Serial			Parallel		
	Linear	NN	Bucciarelli	Linear	NN	Bucciarelli
2 cells	0.99	0.96	0.92	0.99	0.97	0.65
3 cells	0.97	0.96	0.89	0.98	0.96	0.62
4 cells	0.96	0.96	0.86	0.96	0.95	0.62
5 cells	0.95	0.95	0.83	0.94	0.95	0.59
6 cells	0.95	0.95	0.83	0.92	0.95	0.58

Table 5.3. Performance of Mismatch prediction models for serial and parallel PV interconnect configuration

PV interconnect (parallel cases)	Methods		
	Linear	NN	Bucciarelli
A	0.92	0.91	0.67
B	0.91	0.92	0.63
C	0.88	0.88	0.61
D	0.87	0.91	0.58
E	0.84	0.91	0.59
F	0.87	0.87	0.58
G	0.85	0.90	0.56
H	0.86	0.89	0.57
I	0.84	0.91	0.53
J	0.83	0.88	0.54
K	0.82	0.87	0.53
L	0.83	0.85	0.53

Table 5.4. Performance of Mismatch prediction models for alternative PV interconnect configuration

The results shown in tables 2 and 3 indicate that we can achieve a relatively good prediction of mismatch with linear and NN models. This suggests that one can adopt simulation for creating a statistical prediction model for mismatch and achieve acceptable performance rather than using analytical equations such as the Bucciarelli model. In the case of alternative PV interconnection configurations, the Bucciarelli model is not able to capture the differences between Serial-parallel, Bridge-link and Total-cross-tied networks.

5.2.6. Interpolate Solar Irradiance Profile

To be able to predict the mismatch of PV arrays under operating conditions, one would need to characterize the individual cell submitted to environmental variations. Since most environmental variations result in changes to irradiance profiles across PV arrays, it is essential to incorporate accurate irradiance profile into the simulation when forecasting energy output for PV arrays.

The common approaches for quantifying solar irradiance during array performance measurements have thus far entailed using solar irradiance sensors such as thermopile-based pyranometers and pyrheliometers. However, such approaches come with high costs, require rigorous calibration, produce lumped spectral and optical effects, and installation procedures have often introduced difficulty in deploying the sensors densely across entire PV arrays or large-scale plants. Thus, it is desirable to be able to install a few irradiance sensors in a systematic way, and then interpolate an irradiance profile across entire PV arrays or plants using the measured irradiance locations. Accurate estimation of irradiance profiles can lead to better optimization of MPPT and correct forecasting of PV system energy output.

Given the diffusive nature of irradiance profiles, one can apply statistical models to predict the diffused irradiance as function of locations. In this section, we utilize the Kriging and NN methods to interpolate a simulated irradiance profile. A spatially varying irradiance profile was first generated with a bimodal Gaussian function where the irradiance is dependent on (x, y) coordinates [5.14]:

$$G(x,y) = 1 - \left[exp\left(-\frac{(x-x_c)^2}{\sigma_x} - \frac{(y-y_c)^2}{\sigma_y}\right)\right] \tag{5.14}$$

where (x_c, y_c) are the coordinates of the center of irradiance profile and (σ_x, σ_y) are the width of irradiance profile. One can adjust the parameters to obtain different irradiance profiles. The bimodal Gaussian function is preferred due to the fact that sunlight usually decreases gradually from a hot spot to the edge of an illuminated region.

In this work, we have simulated a 21x41 PV array, where 20 cells are connected in series for each string, and 40 strings are connected in parallel. In figure 5.19, we present two layouts for sensor placement: the left one includes 25% of cells with irradiance sensors, while in the right one 15% of cells have irradiance sensors. The generated irradiance profile is shown in figure 5.20.

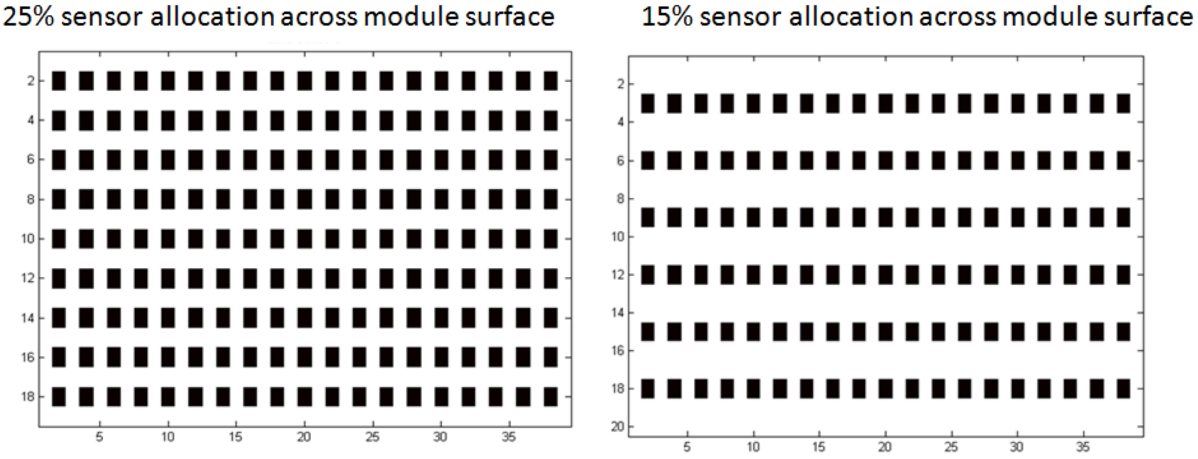

25% sensor allocation across module surface
15% sensor allocation across module surface
2
4
6
8
10
12
14
16
18
20
5
10
15
20
25
30
35

Figure 5.19 Layout of sensors

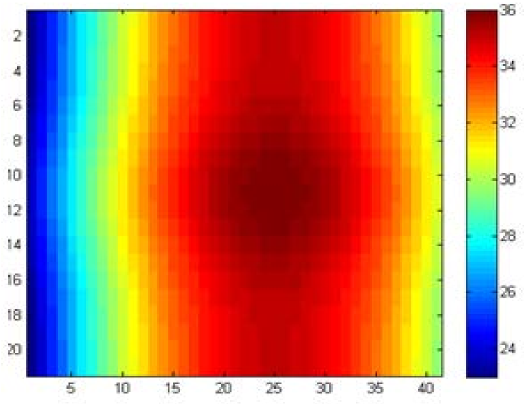


Figure 5.20 Generated irradiance profile for 21x41 sized PV array

Prior to apply Kriging to interpolate the entire irradiance profile, we first computed a semivariogram for each layout and fit a theoretical variogram model to the semivariogram. The results are shown in figure 5.21 and indicate that with 25% sensor coverage we can capture most of the spatial variance of the irradiance profile.

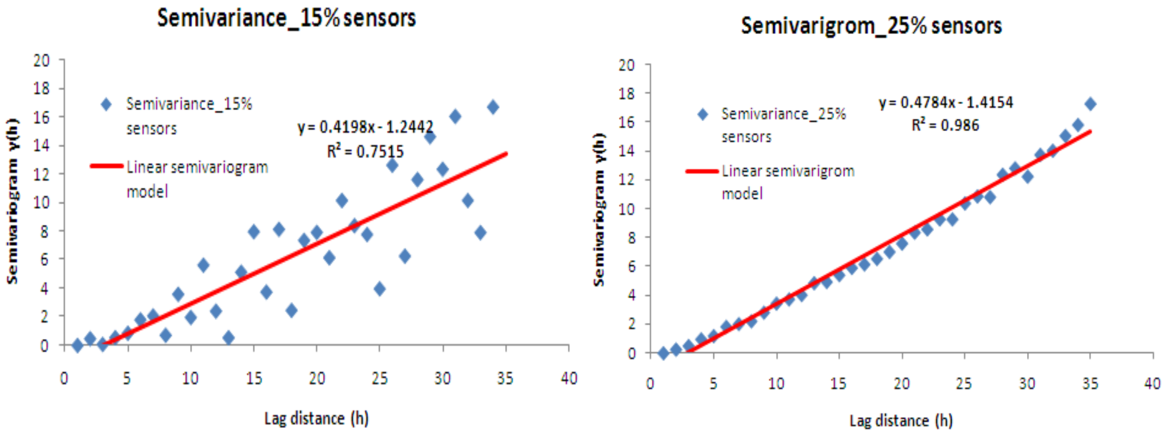


Figure 5.21 Spatial variogram models for kriging interpolation

After the semivariogram model was computed, we applied Kriging to interpolate the entire irradiance profile. The performance of Kriging was then compared with the NN approach. In the NN approach, a nonlinear function is trained to capture the relationship between irradiance and location coordinates. The results are shown in figure 5.23.

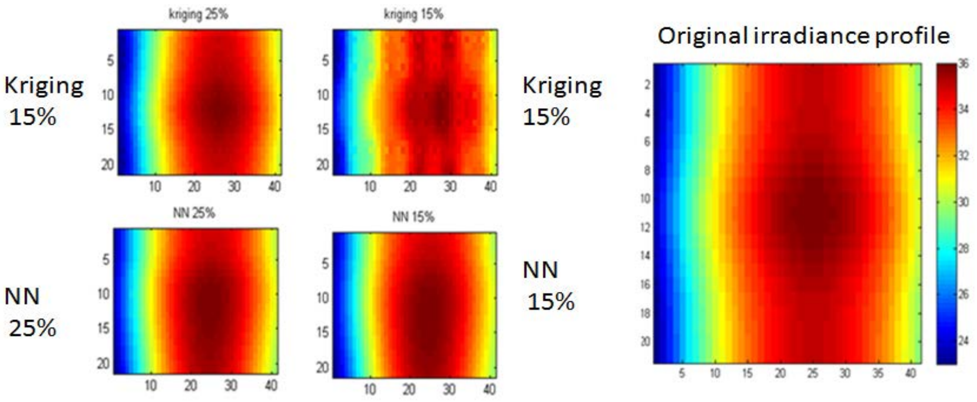


Figure 5.22 Interpolation results for simulated irradiance profile

Based on the plots shown in figure 5.22, we can see that if the irradiance profile exhibits a strong spatial dependence, then NN outperforms the Kriging approach with a lesser number of sensors. After the entire irradiance profile is interpolated, we then simulated the interpolated profiles in SPICE with corresponding $I_{ph}, I_{01}, I_{02}, R_{sr},$ and R_{sh}, where I_{ph} I_{01}, I_{02} are adjusted based on the each cell's irradiance and temperature, and R_{sr}, R_{sh} are identical for all cells in the PV array. The IV and power curves for all four interpolated profiles are compared against the original ones in figure 5.23 The results suggest that NN outperforms Kriging in this case due to the fact that the data is generated from an explicit location based function. However, there can be measurement noise in actual field measurements, where the irradiance profile might not have a strong spatial dependence. In such cases, NN will underperform the Kriging approach.

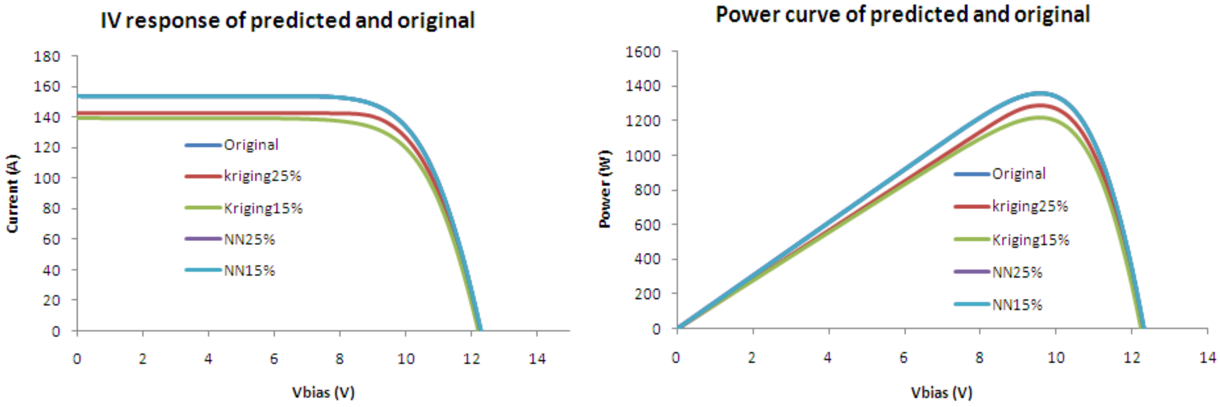


Figure 5.23 Simulated electrical performance for interpolated results

5.2.7. Binning with Clustering and Multivariate Parameters

In order to reduce mismatch power losses resulting from production, cell binning has become a common industrial practice, along with varying PV module interconnect configurations. Binning is done by grouping cells into different bins based on their performance, as shown in figure 5.24. There are three common cell binning methods: binning by (1) I_{mp}; (2) P_{mp}; (3) I at fixed V [5.15]. These three common methods all adopt a single parameter for grouping cells. As we have seen in simulation results from the previous section, this matches the expectation that a majority of mismatch losses result from current mismatch. However, ignoring voltage mismatch might result in significant power losses, given that most PV arrays are connected in serial-parallel configuration to form modules. In this thesis, we assume the binning is done to ensure that mismatch losses are minimized at the module level. This assumption follows from the fact that cell manufacturer have different concerns than module manufacturers. A module manufacturer would have concerns about mismatch losses, reliability and conversion losses.

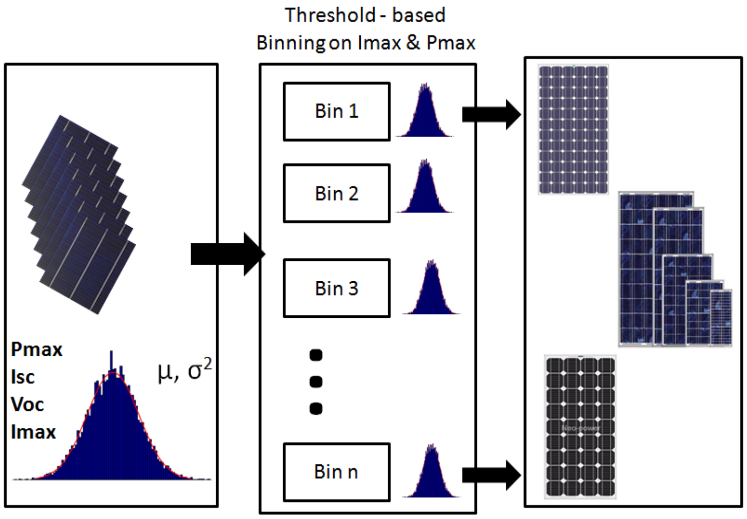


Figure 5.24 Cell Binning process

In this work, we propose a cell binning method that utilizes more information rather than relying on a single metric. This method is based on multivariate parameters including: $I_{mp}, V_{mp}, P_{mp}, I_{sc}$ and V_{oc}, while a K-means clustering technique is applied to assign cells into optimal bins. We argue that by incorporating more parameters into the binning process, the mismatch losses resulted from V_{oc} are also accounted for. In order to compare the performance of our proposed binning method with conventional approaches, we evaluate mismatch loss due to manufacturing variability and degradation according to each binning methods with a serial-36 cells and the 12 alternative PV interconnect configurations we showed in figure 5.15. To observe which binning method is more tolerant for degradation over time, an annual degradation rate for the electrical performance of each cell was sampled from a bounded uniform distribution with upper and lower bounds as $[0.25\%, 0.75\%]$. The uniform distribution based degradation rate is assumed to reflect the in-field degradation rate of a PV array over a 25-year life-time. This degradation rate is then applied to I_{sc}, V_{oc} and FF to correspondingly adjust the electrical parameters of cells.

We first applied an MC simulation along with the PC1D device simulator to generate 10,000 silicon based N+PP+ solar cells with manufacturing variability introduced to cells' physical parameters. Each of the physical parameters is assumed to follow specific Normal distribution and be independently identical. Then, corresponding equivalent circuit electrical parameters including $I_{ph}, I_{01}, I_{02}, R_{sr}$, and R_{sh} were extracted for each cell from their IV curves and their electrical performance parameters $P_{max}, I_{mp}, V_{mp}, I_{sc}, V_{oc}$ and FF. The corresponding distributions of physical parameters and generated electrical parameters are shown in figure 5.25.

- Background doping cm-3 N(1e16,1e15)
- Area cm2 N(100 ,10)
- Thickness um N(300,30)
- Surface concentration cm-3 N(2.8e20,2.8e19)
- FSRV cm/s N(10e6,10e5)
- BSRV cm/s N(10e5,10e4)
- Front Reflectance % N(10, 2)
- Bulk carriers lifetime us N(7.3,1.4)

Each physical parameter is assumed to be IIND

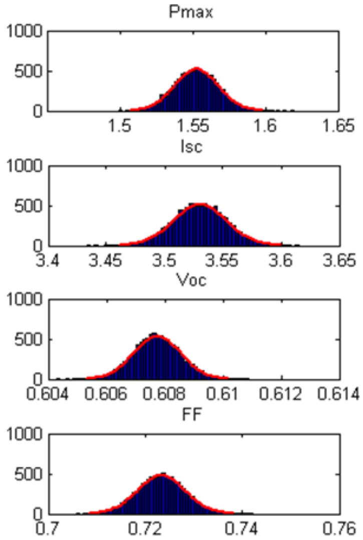


Figure 5.25 10,000 cells generated from Monte Carlo simulation

The 10,000 simulated cells were then used as input samples for alternative cell binning. We show the simulation and mismatch evaluation flow in figure 5.26. The chosen binning methods to evaluate included: (1) Binning with 25% and 75% quantile of I_{mp} (2) K-means clustering with I_{mp} (3) K-means clustering with P_{mp} (4) K-means clustering with $I_{mp}, P_{mp}, V_{mp}, I_{sc}$ and V_{OC}.

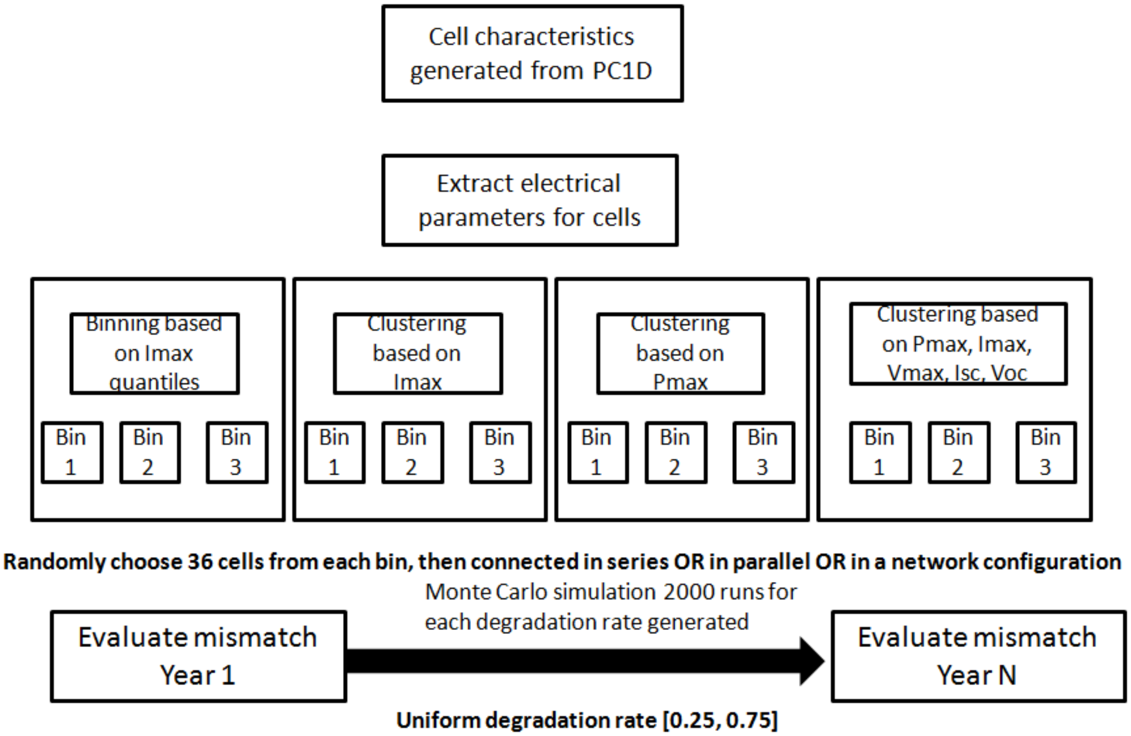

Cell characteristics generated from PC1D
Extract electrical parameters for cells
Binning based on Imax quantiles
Clustering based on Imax
Clustering based on Pmax
Clustering based on Pmax, Imax, Vmax, Isc, Voc
Bin 1
Bin 2
Bin 3
Randomly choose 36 cells from each bin, then connected in series OR in parallel OR in a network configuration
Monte Carlo simulation 2000 runs for each degradation rate generated
Evaluate mismatch Year 1
Evaluate mismatch Year N
Uniform degradation rate [0.25, 0.75]

Figure 5.26 Simulation and evaluation flow for alternative binning

For each binning method, we then evaluated the binning performance by sampling cells from each bin and connecting them in the desired PV array configuration. For each sampled set, another run of MC was generated to simulate the effect of annual degradation introduced into each cell. The sampled degradation rate from uniform distribution was applied to I_{sc} and V_{oc}. The degraded FF, I_{sc} and V_{oc} were then used to adjust $I_{ph}, I_{01}$for degradation effects on the electrical performance of solar cell. The adjusted I_{ph} and I_{01} were then fed into the SPICE simulator to extract performance for degraded cells. In this work, we simulated the degradation effect across a 25-year time period. In figure 5.27, the % of power loss in a 36-cells serial PV array due to mismatch is plotted from year 1 to year 25 according to different binning methods. The % of power loss shown is the average of all the MC runs for a specific combination of bin assignment and sampled degradation rate.

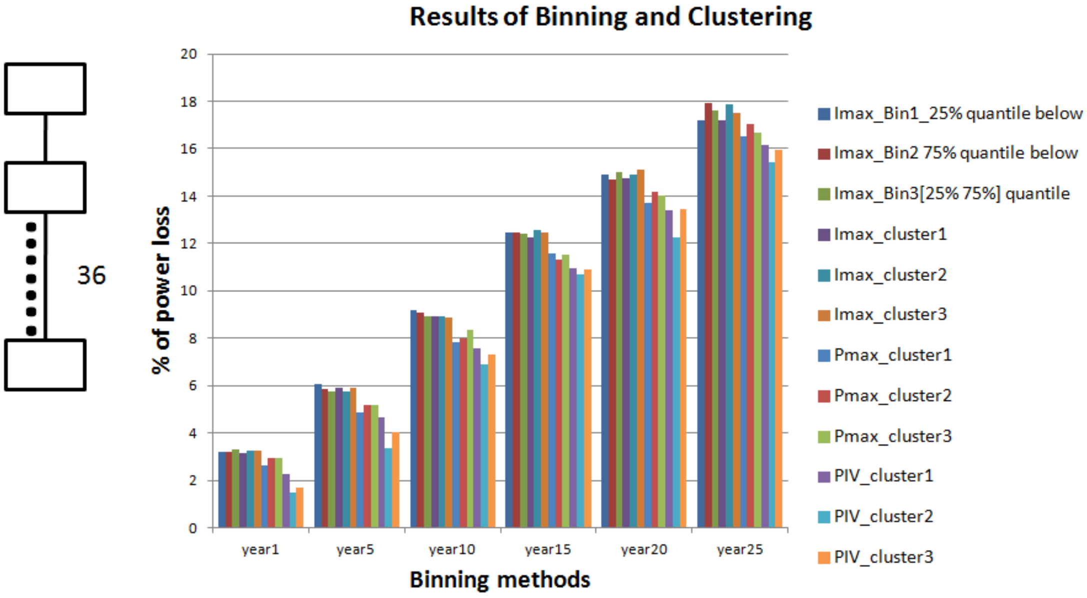


Figure 5.27 Performance evaluations for different binning method with serial array

In figure 5.28, we present the year-25 mismatch results for all 12 alternative PV interconnect configurations according to different binning methods.

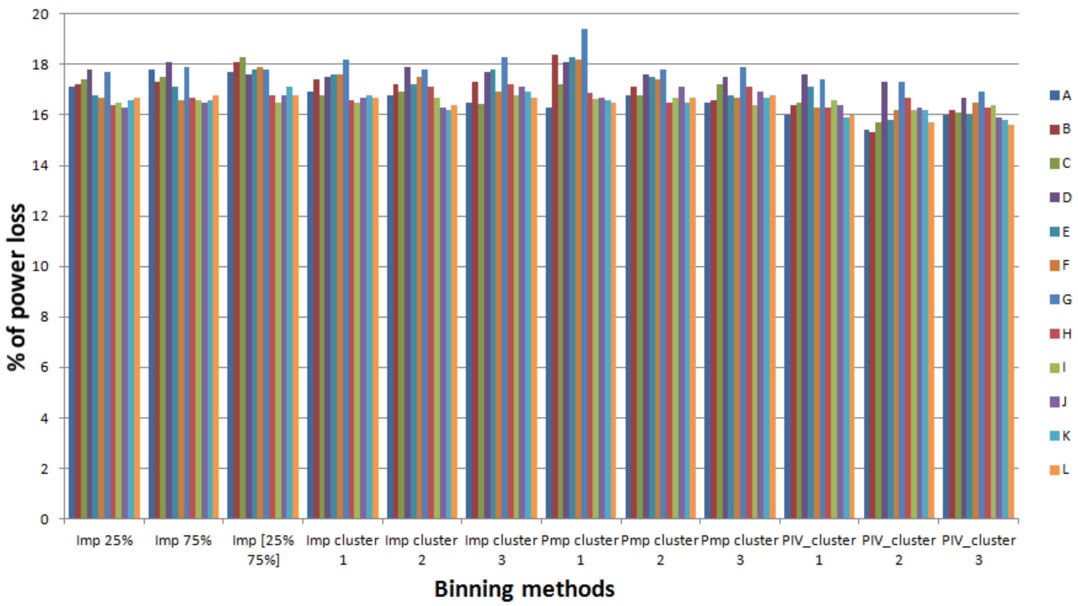


Figure 5.28 Performance evaluations for different binning method across different configuration

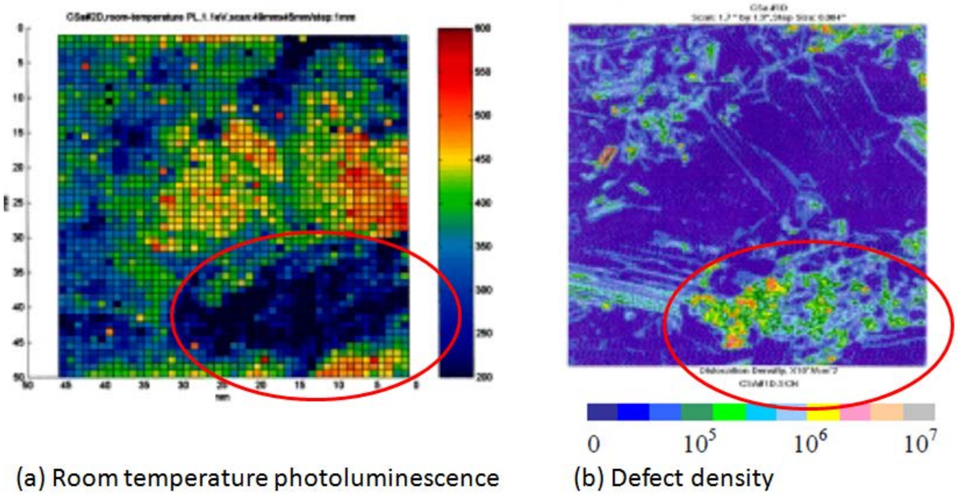
Both figures 5.27 and 5.28 suggest that mismatch loss can be reduced across the entire lifecycle of a solar cell array using clustering binning methods that employ multiple performance parameters.

5.3. Modeling Nonuniformities of Solar Cells

Nonuniformities of the material properties of a solar cell have been known to impact final solar cell performance. Degrees of nonuniformity, including local variations in defect density and junction thickness variation are all usually measured using local minority carrier lifetime across the entire cell. It has been shown in the literature that there is a correlation between the spatially resolved carrier lifetime of cells and spatially resolved solar cell efficiency.

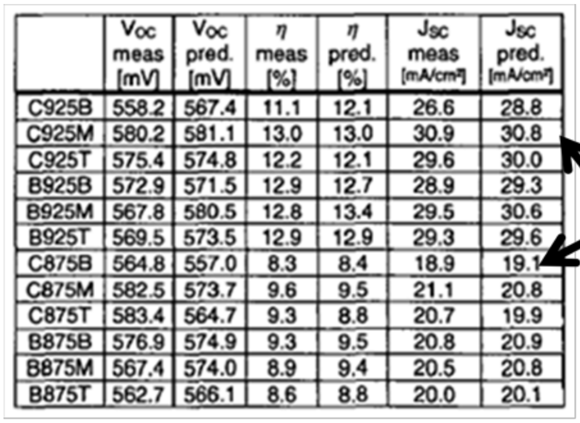
Warta [5.16] investigated the correlation of local carrier lifetime of multicrystallene (MC) silicon and the corresponding monochromatic solar cell efficiency using microwave-detected photo-conductance decay (MW-PCD) measurements and illuminated lock-in thermography (ILIT). Good qualitative correlations between areas of reduced lifetime values and reduced efficiency of the finished cell were observed. However, the prediction of local and global solar cell efficiency from measured lifetimes on unprocessed wafers is of limited use, due to the incorrect results of lifetime measurements. MC Si solar cells are also subject to efficiency limitations due to defect clusters. A study presented by Sopori [5.17] shows that MC Si wafers used in commercial solar cell fabrication exhibit a tendency to form large clusters of defects, which remain laterally separated from each other. Defect clusters are often sites of impurity precipitation, and become the low-performing regions in the cell. They tend to shunt the device and constitute the primary efficiency limiting mechanism in current solar cells. Sopori's paper [5.17] describes the nature of defect clusters, their formation mechanisms, and their effects on solar cell performance. An electronic model of a solar cell containing defect clusters is simulated and used to estimate the limitation of cell efficiency attributable to defect clusters.

In figure 5.29, we can see how local defect density (b) directly corresponds to local photocurrent (a). This suggests that higher defect density can lower the electrical performance parameters of a cell. This phenomenon is validated by the results of an experimental study shown in figure 5.30, where we can observe that cells with distributions around lower means have better electrical performance. A lower mean for distributions of minority carrier lifetime indicates that the cell has a lower defect density.



(a) Room temperature photoluminescence (b) Defect density

Figure 5.29 Defects vs. solar cell efficiency [5.16]



	Voc meas [mV]	Voc pred. [mV]	η meas [%]	η pred. [%]	Jsc meas [mA/cm²]	Jsc pred. [mA/cm²]
C925B	558.2	567.4	11.1	12.1	26.6	28.8
C925M	580.2	581.1	13.0	13.0	30.9	30.8
C925T	575.4	574.8	12.2	12.1	29.6	30.0
B925B	572.9	571.5	12.9	12.7	28.9	29.3
B925M	567.8	580.5	12.8	13.4	29.5	30.6
B925T	569.5	573.5	12.9	12.9	29.3	29.6
C875B	564.8	557.0	8.3	8.4	18.9	19.1
C875M	582.5	573.7	9.6	9.5	21.1	20.8
C875T	583.4	564.7	9.3	8.8	20.7	19.9
B875B	576.9	574.9	9.3	9.5	20.8	20.9
B875M	567.4	574.0	8.9	9.4	20.5	20.8
B875T	562.7	566.1	8.6	8.8	20.0	20.1

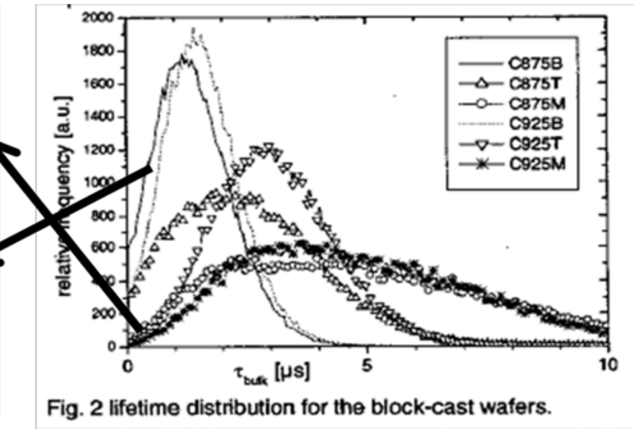


Fig. 2 lifetime distribution for the block-cast wafers.

Figure 5.30 Minority carrier lifetime distribution vs. electrical performance of cell [5.17]

Along with experimental studies examining how these nonuniformities impact solar cell performance symmetrically, there have been multiple studies modeling these nonuniformities with distributive diode arrays, where an entire cell is divided into mini-cells that reflect local variations. However, they all lack a predictive model to characterize how different defect patterns can impact final cell performance.

In this section, we first review the past approaches that have been used to model nonuniformities. Then we propose a simulation flow that utilizes a 3-D distributive diode array to model different nonuniformities patterns. This simulation flow is shown in figure 5.31. The first step in the simulation flow is to generate different defect patterns, then populate these patterns into a SPICE simulator where the distributive diode array is defined. Then, the corresponding cell parameters, including I_{ph}, I_{01}, I_{02},R_{sh}, R_{sr}, and V_{oc}, were extracted with PC1D for corresponding minority carrier-lifetime. The final solar cell performance is then extracted from the SPICE results. The entire simulation flow is automated using Perl scripts.

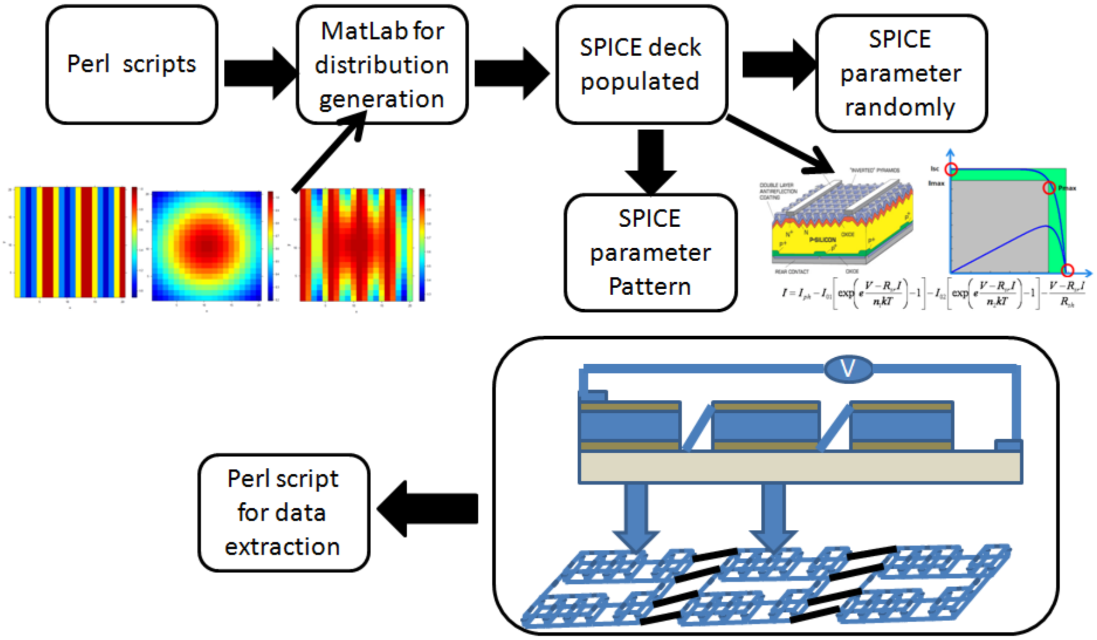

Perl scripts
MatLab for distribution generation
SPICE deck populated
SPICE parameter randomly
SPICE parameter Pattern
"INVERTED" PYRAMIDS
DOUBLE LAYER ANTIREFLECTION COATING
N+
N
P-SILICON
OXIDE
p+
REAR CONTACT
OXIDE
Isc
Imax
Pmax
$I = I_{ph} - I_{01}\left[\exp\left(e\frac{V-R_{sr}I}{n_1kT}\right)-1\right] - I_{02}\left[\exp\left(e\frac{V-R_{sr}I}{n_2kT}\right)-1\right] - \frac{V-R_{sr}I}{R_{sh}}$
V
Perl script for data extraction

Figure 5.31 Simulation flow for nonuniformities in cells

With the simulator, we then create a statistical prediction model to capture the relationship between different defect patterns and final cell performance. This statistical model can be adopted in a production environment, as shown in figure 5.32, to predict final cell performance based on the defect image map obtained at an earlier production stage.

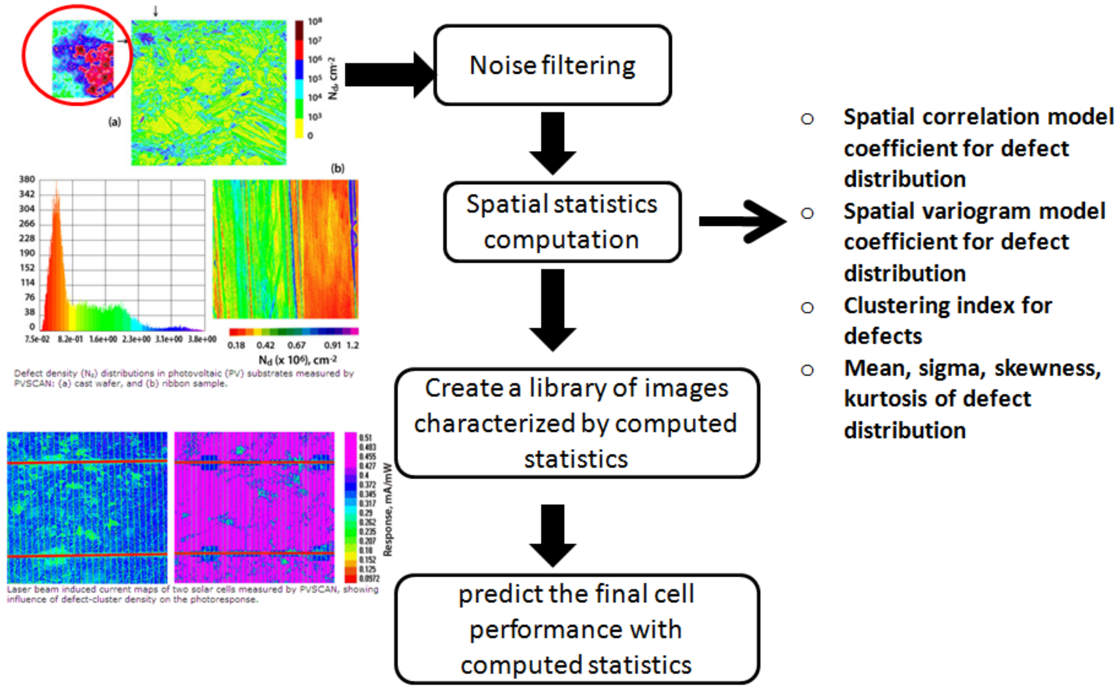

10⁸
10⁷
10⁶
10⁵
10⁴
10³
0
N_d, cm⁻²
(a)
(b)
380
342
304
266
228
190
152
114
76
38
0
7.5e-02
8.2e-01
1.6e+00
2.3e+00
3.1e+00
3.8e+00
0.18
0.42
0.67
0.91
1.2
N_d (x 10⁶), cm⁻²
Defect density (N_d) distributions in photovoltaic (PV) substrates measured by PVSCAN: (a) cast wafer, and (b) ribbon sample.
0.51
0.483
0.455
0.427
0.4
0.372
0.345
0.317
0.29
0.262
0.235
0.207
0.18
0.152
0.125
0.0972
Response, mA/mW
Laser beam induced current maps of two solar cells measured by PVSCAN, showing influence of defect-cluster density on the photoresponse.
Noise filtering
Spatial statistics computation
Create a library of images characterized by computed statistics
predict the final cell performance with computed statistics
○ Spatial correlation model coefficient for defect distribution
○ Spatial variogram model coefficient for defect distribution
○ Clustering index for defects
○ Mean, sigma, skewness, kurtosis of defect distribution

Figure 5.32 Statistical prediction models for cell performance

As figure 5.32 illustrates, the first step of the statistical model is to filter out the noise from defect images, then statistics are computed to characterize these defect patterns. A library of defect images, along with corresponding statistics and cell performances, are then stored. This library can be used to train a prediction model for incoming solar cells. We will discuss the simulation flow and prediction model creation in the following sections.

5.3.1. Diode Arrays For Nonuniformities

The conventional approach in defect modeling for calculating the influence of defects on the material quality is to estimate the average carrier recombination and express it in terms of an average minority carrier-lifetime or diffusion length. However, a solar cell is a distributed device in which each region connects to other regions, and defect clusters at different locations affect the entire cell. Thus, it is necessary to calculate the electrical characteristics of each local region based on its local material properties, and then calculate the influence of each local region on the entire cell. We can thus calculate the influence of any defect distribution on final cell performance and determine the loss introduced by various defect distributions.

The entire cell can be modeled as a network model. The network model consists of an array of any size of small area, that is: local cells that are interconnected through a common junction and a bus. Each small-area cell is assigned a defect density corresponding to that in the actual wafer for the corresponding location. The modeling process has two steps. In the first step, each local cell is represented in terms of the recombination properties associated with its defect density, which yields values of I_{ph}, minority carrier-lifetime τ, and diode saturation currents I_{01} and I_{02}. In the second step, the diode array is interconnected using resistive elements to account for sheet resistance of junctions and metal contact resistance.

Different distributed solar cell models have been proposed in the literature for modeling nonuniformities. We have shown the 4 most cited models in figure 5.34. Karpov proposed a random diode array to model a thin-film device structure with a random distribution quality [5.18]. This model is shown in figure 5.33 (A). Abdelhalim [5.19] proposed a 1-dimensional distributed SPICE model for a solar cell, where the cell is divided laterally into identical segments in which the current distribution is the same. Each of the segments comprises the cell area from the center-line of a metallization finger to the line half-way to the neighboring finger. The number of segments is twice the number of metal fingers. All segments are connected in parallel. This model is shown in figure 5.34 (B). The distributed model presented in Palma's work [5.20] is for simulating the voltage drop on the front contact grid. This model is shown in figure 5.34 (C). The model shown in figure 5.34 (D) is a 2.5D distributed SPICE model for thin-film cell structure [5.22].

All these proposed distributed models focus on explaining specific parts of the loss mechanism, such as series loss due to front contact, or voltage loss due to junction quality. Also, these works fail to statistically describe how spatial nonuniformities impact the final cell performance when all these effects are lumped together.

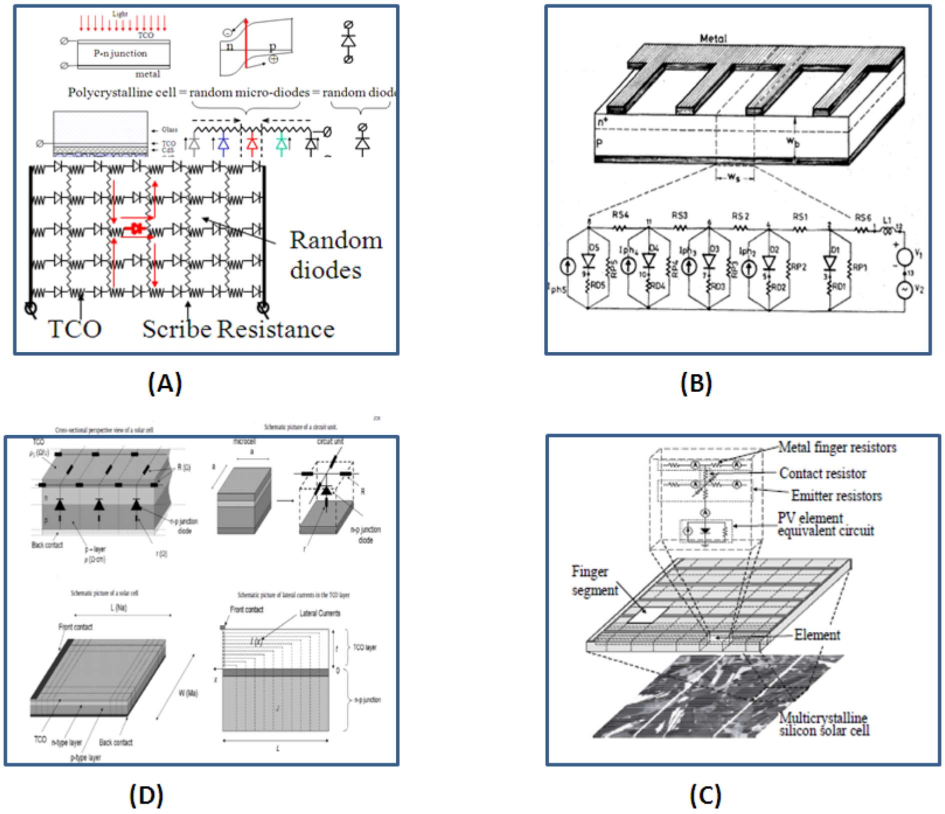


Figure 5.33 Distributed solar cell models [5.18, 5.19, and 5.20]

The model we propose in this thesis is a 3-dimensional distributed solar cell model where the entire cell is modeled as an array of mini-cells. We have adopted SPICE to simulate this proposed cell model. This 3-D cell model is shown in figure 5.34. Each mini-cell is represented with a 2-diode equivalent circuit. The mini-cells are interconnected with metal contact resistors at the top and transparent conductive oxide (TCO) resistors at the bottom. The metal contact resistors and TCO resistors are for modeling thin-film cells, since they are more subject to nonuniformities. In the case of silicon-based solar cells, we can eliminate these resistors. It is important to include the top and bottom contact resistance in a distributive manner for thin-film cells, because it has been shown that higher contact resistance in thin-film cells mitigates the impact of nonuniformities [5.18].

We demonstrate the mitigation effect with simulation results shown in figure 5.35. The simulation is based on two scenarios applied to the 3-D distributive cell model, where each cell is a 10×10 distributive array. In both scenarios, we have the 4 mini-cells at the center of the array be defective according to the following parameters: $I_{01} = 1e^{-5}A$; $I_{02} = 1e^{-9}A$; $R_{sh} = 0.1\Omega$; $I_{ph} = 1.34\,A$, while the remaining non-defective cells all have identical performances with parameters set to: $I_{01} = 1e^{-7}A$; $I_{02} = 1e^{-12}A$; $R_{sh} = 1000\Omega$; $I_{ph} = 4.34\,A$. We then set the $R_{tco} = 0.1\Omega$ in scenario 1 and; $R_{tco} = 0.01\Omega$ in scenario 2. The electrical potentials across the entire cell are shown for both scenarios in figure 5.35. The left plot corresponds to scenario 1 with $R_{tco} = 0.1$ and the right one is for scenario 2 with $R_{tco} = 0.01$. It is expected that the "defective" mini-cells at the center will suck in power from adjacent mini-cells. With relatively higher TCO resistance, this effect can be mitigated and the overall V_{oc} of entire cell will not be significantly affected. This phenomenon is validated with electrical potential plots in figure 5.36 where higher contact resistances can prevent mini-cells near the defective mini-cell from suffering lower electrical potential. Thus relative higher V_{oc} can be retained.

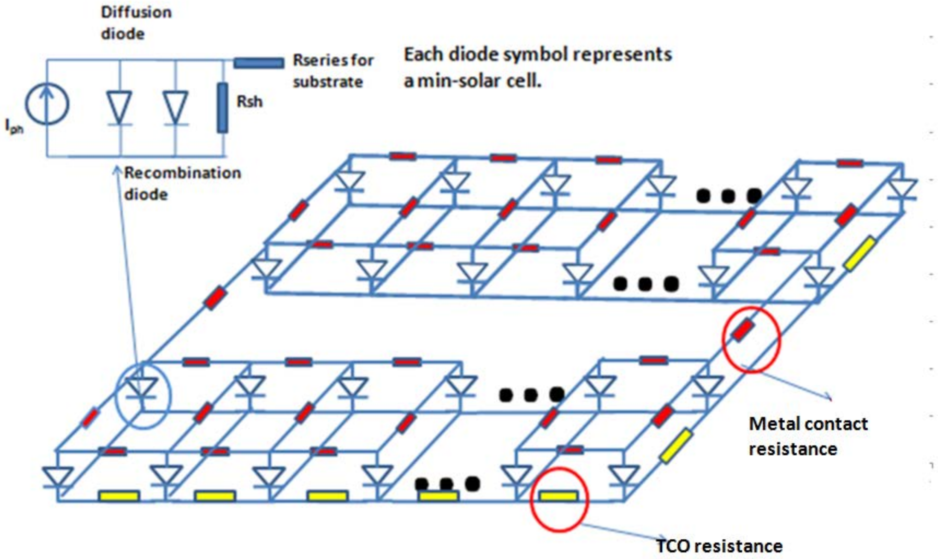


Figure 5.34 Diode arrays used for nonuniformities modeling

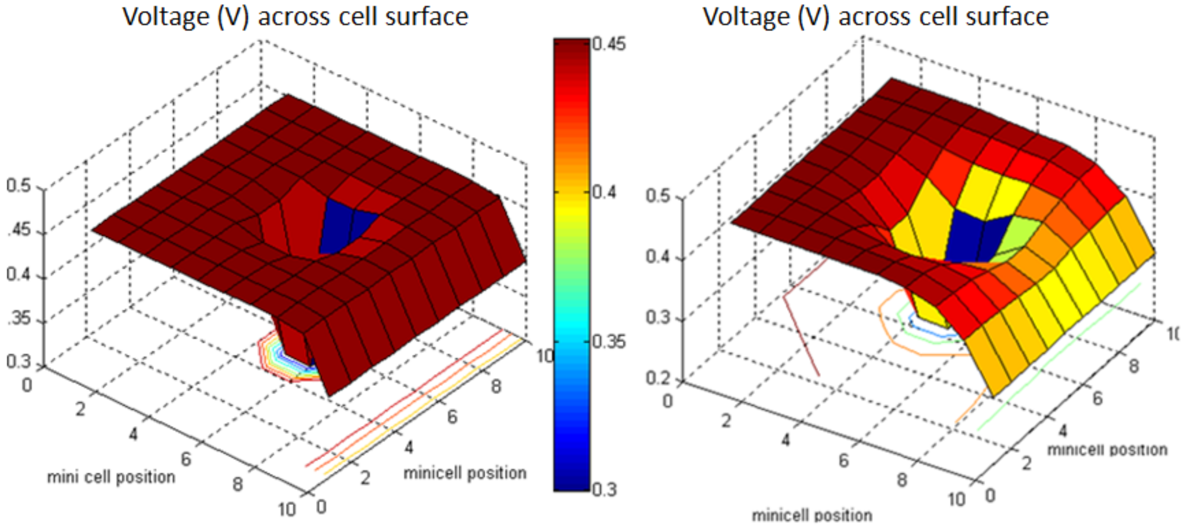


Figure 5.35 Electrical potential for mini-cells across cell surface

In order to understand how distributions of local properties impact entire cell performances, we utilized a baseline cell represented by a 20×20 array for simulation. This 20×20 array was used for all later analysis. In this array, each mini-cell has variability according to five electrical parameters : I_{01},I_{02},I_{ph},R_{sh},R_{sr}. We combined the R_{tco} from contact resistance and R_m from material resistance as a lumped parameter R_{sr}. In this simulation, we varied the distribution of each local electrical parameter independently. Gaussian distribution was applied in this case. The distribution of each parameter is listed in table 5.5a below:

$I_{01} = [\mu, \sigma]$		$I_{02} = [\mu, \sigma]$		$I_{ph} = [\mu, \sigma]$		$R_{sh} = [\mu, \sigma]$		$R_{sr} = [\mu, \sigma]$	
$[\mathbf{1e^{-5}, 0.1}$	$[\mathbf{1e^{-5}, 0.2}$	$[\mathbf{1e^{-9}, 0.1\mu}$	$[\mathbf{1e^{-9}, 0.2\mu}$	$[\mathbf{0.34, 0.1}$	$[\mathbf{0.34, 0.2}$	$[\mathbf{0.1, 0.1\mu}]$	$[\mathbf{0.1, 0.2\mu}]$	$[\mathbf{0.01, 0.1}$	$[\mathbf{0.01, 0.2}$
$[\mathbf{1e^{-6}, 0.1}$	$[\mathbf{1e^{-6}, 0.2}$	$[\mathbf{1e^{-10}, 0.1}$	$[\mathbf{1e^{-10}, 0.2}$	$[\mathbf{1.34, 0.1}$	$[\mathbf{1.34, 0.2}$	$[\mathbf{1, 0.1\mu}]$	$[\mathbf{1, 0.2\mu}]$	$[\mathbf{0.1, 0.1\mu}$	$[\mathbf{0.1, 0.2\mu}$
$[\mathbf{1e^{-7}, 0.1}$	$[\mathbf{1e^{-7}, 0.2}$	$[\mathbf{1e^{-11}, 0.1}$	$[\mathbf{1e^{-11}, 0.2}$	$[\mathbf{2.34, 0.1}$	$[\mathbf{2.34, 0.2}$	$[\mathbf{10, 0.1\mu}]$	$[\mathbf{10, 0.2\mu}]$	$[\mathbf{1, 0.1\mu}]$	$[\mathbf{1, 0.2\mu}]$
$[\mathbf{1e^{-8}, 0.1}$	$[\mathbf{1e^{-6}, 0.2}$	$[\mathbf{1e^{-12}, 0.1}$	$[\mathbf{1e^{-12}, 0.2}$	$[\mathbf{3.34, 0.1}$	$[\mathbf{3.34, 0.2}$	$[\mathbf{1000, 0.1}$	$[\mathbf{1000, 0.2}$	$[\mathbf{10, 0.1\mu}]$	$[\mathbf{10, 0.2\mu}]$

Table 5.5a. Parameters for simulation

2000 Monte Carlo simulation runs were executed, varying individual parameters with Gaussian distribution while keeping other mini-cells constant using following values: $I_{01} = 1e^{-6}A$, $I_{02} = 1e^{-10}A$, $R_{sh} = 1000\ \Omega$, $R_{tco} = 0.01\ \Omega$, $I_{ph} = 3.34\ A$ as means, with sigma being 10% of mean. The applied variations are summarized in table 5.5b.

	I_{01}	I_{02}	I_{ph}	R_{sh}	$R_{sh} + R_{sr}$
Baseline	$[1e^{-6}A, 10\%]$	$[1e^{-10}A, 10\%]$	$[3.34\ A, 10\%]$	$[1000\ \Omega, 10\%]$	$[0.1\ \Omega, 10\%]$
Monte Carlo	$[1e^{-8}, 1e^{-5}]$	$[1e^{-12}, 1e^{-9}]$	$[0.34\ ,3.34]$	$[0.1, 1000]$	$[0.001, 1]$

Table 5.5b. MC simulation parameters for cell variability defined by Gaussian distribution with mean values

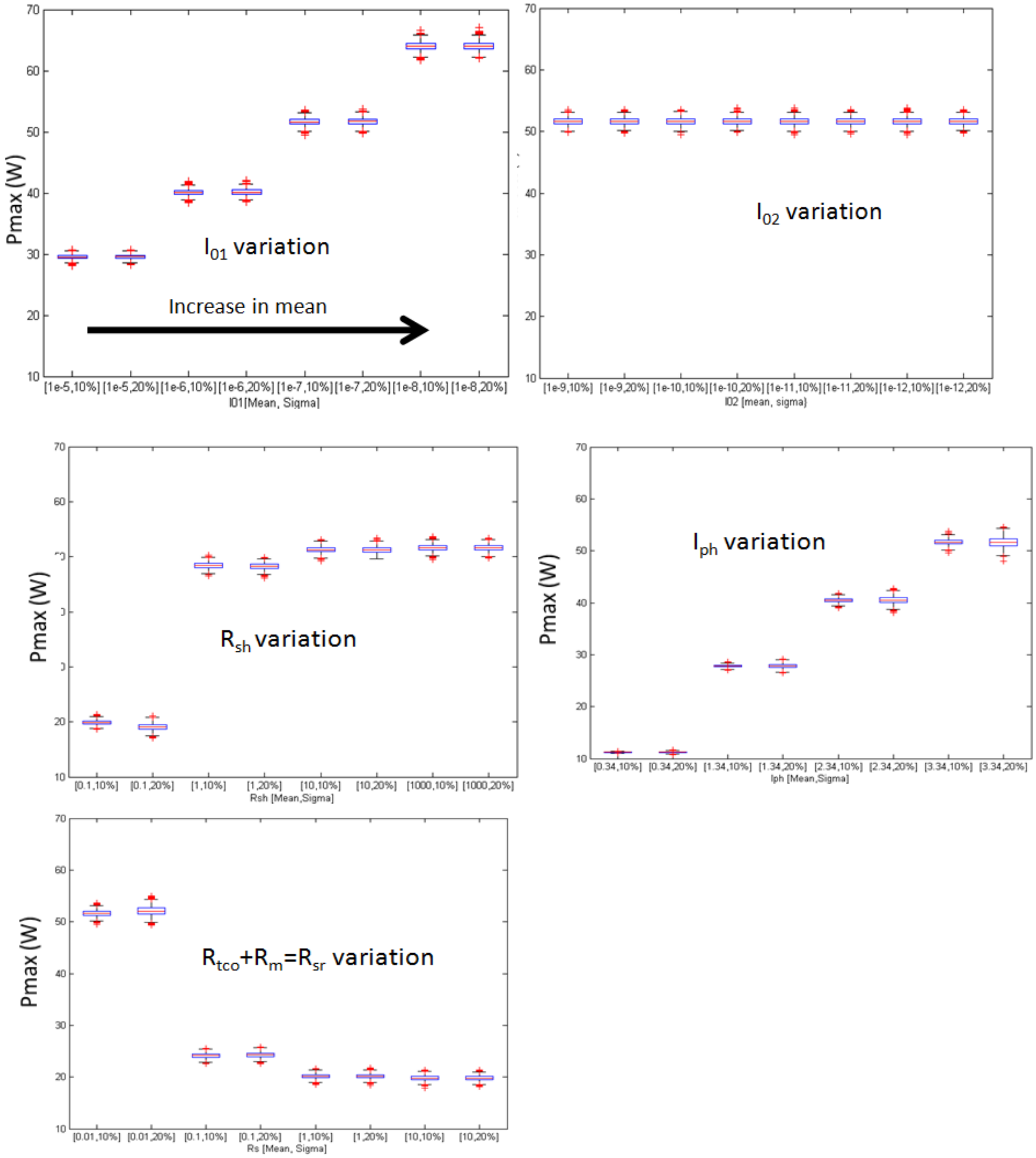


Figure 5.36 Monte Carlo simulation results for individual parameter fluctuations

We analyze the simulation results in figure 5.36 with 5 plots where quintiles of the maximum power of entire cell performance are plotted against different conditions. The results suggest that the distributions of $I_{01}$and I_{ph} produce the largest impact on final cell performance. These two parameters are directly correlated with levels of defect density in cell. The simulation results shown in figure 5.36 imply that the spread of distribution does not impact the final output power of the cell as significantly as the mean does. However, the results also suggest that an identical distribution of local variations can result in cells that have final power output vary consistently from the norm by 15%.. This indicates that the location of defects can play an important role in final cell performance. In production, it is quite normal for cells to have similar distributions in terms of physical properties such as defect density, but the location of defects can vary significantly. This phenomenon is illustrated in figure 5.37.

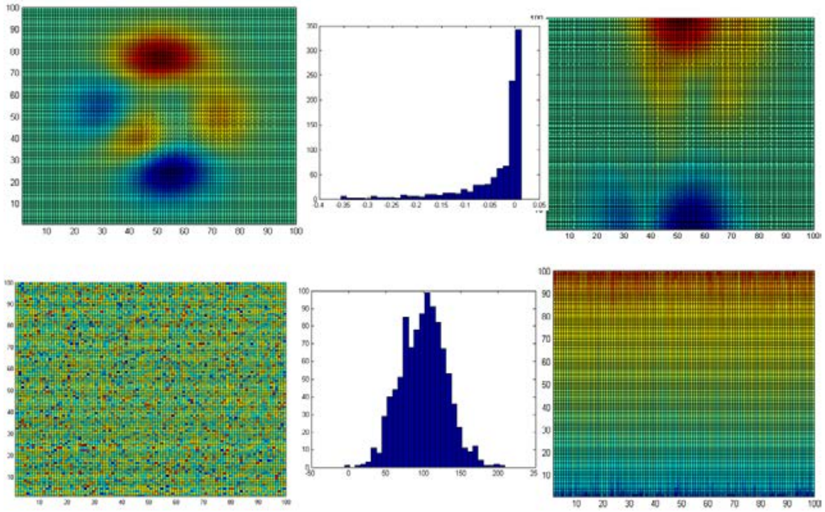


Figure 5.37 Illustration of spatial distribution with same histogram

5.3.2. Defect Concentration to Electrical Parameters

In order to produce an accurate simulation model, it is necessary to be able to incorporate the variations in electrical behavior of material properties. In practice, one would need to measure cells with certain defect concentrations; the obtained IV curves would then be fed into optimization routines to extract equivalent circuit parameters. In this work, our focus is creating a predictive model for cell performance based on nonuniformity patterns. Due to the lack of experimental data, we have utilized binary defects in this work, where the mini-cell is defined to be defective or non-defective. The mini-cell is defined as defective if corresponding local defect concentration or minority carrier-lifetime is above a defined threshold level. In production, one would need to adopt optimization methods discussed below to extract corresponding local mini-cell parameters based on measured minority carrier-lifetime.

There exist multiple approaches to extracting electrical parameters from solar cell I-V curves. They can be grouped into three categories: direct approaches according to I-V characteristics, gradient based methods and global stochastic optimization methods [5.23].

Among these methods, the direct approaches are based on the use of I-V curve features, such as axis intercepts and gradients at selected points, to determine specific cell parameters. The accuracy of these methods is thus limited by the numerical differentiation as well as by the simplified formulas used in parameter extraction. In addition, several different conventional nonlinear fitting algorithms, such as the quasi-Newton method and its variations, have been proposed to solve solar cell parameter extraction. However, nonlinear optimization also introduces a difficulty in that the extracting process cannot guarantee accurate results for the global convergence if it starts from an arbitrarily chosen initial guess. Therefore, using conventional gradient-based methods, we still cannot characterize the nonlinear behavior of the solar cell very well [5.24].

In the literature, global optimizations such as GA, particle swarm optimization (PSO) and simulated annealing (SA) have been proposed to extract parameters with higher accuracy [5.25]. However, GA suffers some deficiencies in performance when the parameters being optimized are highly correlated. In such cases, the crossover and mutation operations cannot ensure better fitness of offspring because chromosomes in the population have similar structures, and their average fitness is high toward the end of the evolutionary process. In the case of simulated annealing, it is relatively hard to determine the temperature steps that allow fast converging to global minimum. PSO has been shown to have better accuracy than GAs and other gradient-based methods.

5.3.3. Generation of Defect Patterns

In practice, one would obtain an image map of carrier lifetime that correlates with defect concentration spatially across the cell, after which imaging process techniques could be applied to filter out image noise. Then, defect patterns could be characterized for each cell and corresponding cell efficiency inferred from defect patterns. In this work, due to the limited availability of industrial data, we have adopted a spatial probability approach to generate different spatial patterns. It is assumed that we have identical sizes of cells, and that the defect concentration can be categorized into binary levels. That means we can label certain locations of cell with “1” if the defect concentration exceeds our threshold level and “0” if the location is below our threshold. We then apply a spatial probability function, with the assumption that the defect level of a location is independent of the defect levels of other locations across the cell.

The spatial probability function states that the defect level of a location is a function of its relative position within the cell: $P(1\ or\ 0) = p(x,y)$, where x and y are coordinates of the location. In this work, we have assumed each solar cell be represented by a 20x20-grid, each grid has a binary value of 0 or 1. The expressions of spatial probability $p(x,y)$ for some common patterns are listed below [5.26]:

$$Spot: p(x,y) = exp^{\left(-r^2/\sigma^2\right)}, r^2 = (x-x_c)^2 + (y-y_c)^2 \tag{5.15}$$

$$Ring: p(x,y) = 1 - exp^{\left(-r^2/\sigma^2\right)}, r^2 = (x-x_c)^2 + (y-y_c)^2 \tag{5.16}$$

$$Repetitive(horizontal): p(x,y) = {(1+\sin((2\pi y\ /\ T)+\theta))}/{2} \tag{5.17}$$

$$Repetive(vertical): p(x,y) = {\left(1+\sin\left(\left({2\pi x}/{T}\right)+\theta\right)\right)}/{2} \tag{5.18}$$

$$Mixed\ pattern\ ("OR"): p(x,y) = p_1(x,y) + p_2(x,y) - p_1(x,y)p_2(x,y) \tag{5.19}$$

$$Mixed\ pattern\ ("AND"): p(x,y) = p_1(x,y)p_2(x,y) \tag{5.20}$$

x_c and y_c are assumed to be the origins of cells. For spot and ring patterns, the width is controlled by the parameter σ. The repetitive horizontal or vertical pattern is characterized by repetitive rows or columns of defects whose positions are controlled by the parameters $T$and θ. For mixed patterns, the defect pattern is the result of a logical “OR” or logical “AND” of the pattern 1 and 2 whose probabilities are $p_1(x,y)$ and $p_2(x,y)$. Once the spatial probability is generated for a specific pattern, binomial inversion is performed to convert the grids into “1” and “0.” In this work we have simulated 800 cells with 8 defect patterns. They are listed in table 5.6 below.

A:mixAnd_spot_xc10.5_yc10.5_sigma8_repVer_T5_phi0	x_c=10.5 y_c=10.5 σ=8,T=5, θ=0
B:mixOr_spot_xc10.5_yc10.5_sigma8_repVer_T5_phi0	x_c=10.5 y_c=10.5 σ=8,T=5, θ=0
C:repHor_T5_phi0	T=5, θ=0
D:repVer_T7_phi2	T=7, θ=2
E:ring_xc0.5_yc12.5_sigma10	x_c=0.5 y_c=12.5 σ=10
F:spot_xc10.5_yc10.5_sigma8	x_c=10.5 y_c=10.5 σ=8
G: binom_rand_p0.7	P=0.7 (70% defective rate)
H: binom_rand_p0.3	P=0.3 (30% defective rate)

Table 5.6. Generated defect patterns

In figure 5.38, we show the spatial probabilities corresponding to Equations 5.15 to 5.20.

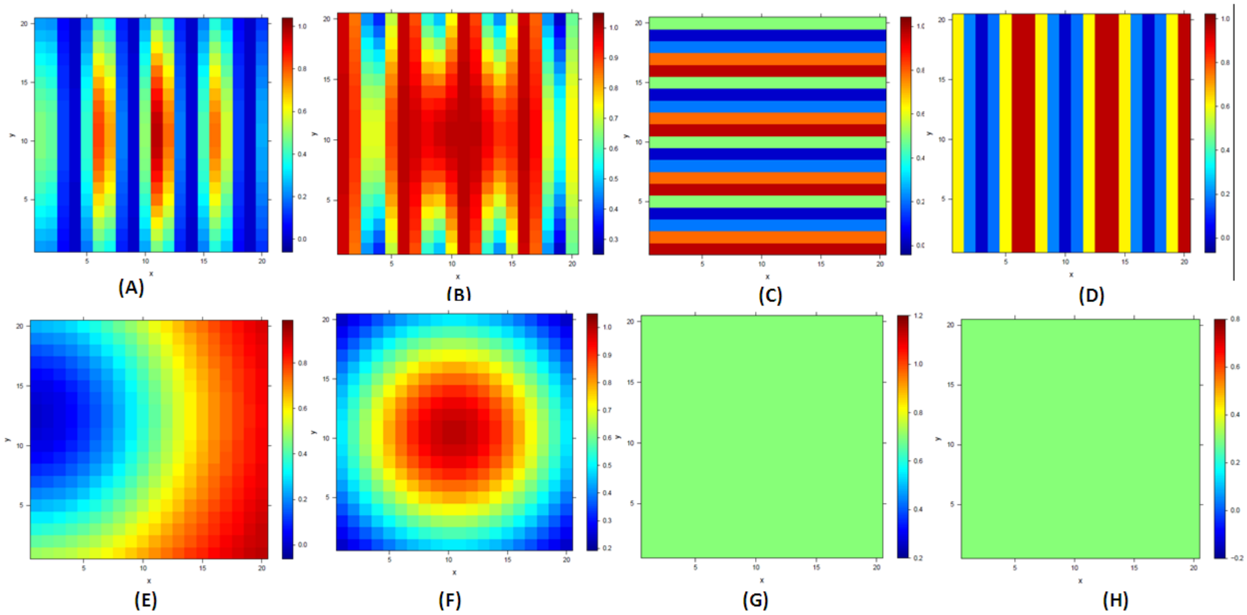


Figure 5.38 Spatial probabilities of defect patterns

For each class of spatial probabilities, 100 cells are generated. In figure 5.39, we show corresponding defect patterns generated using the above 8 spatial probabilities. 25 patterns are shown for each spatial probability. Each defective mini-cell is coded with a “black” color.

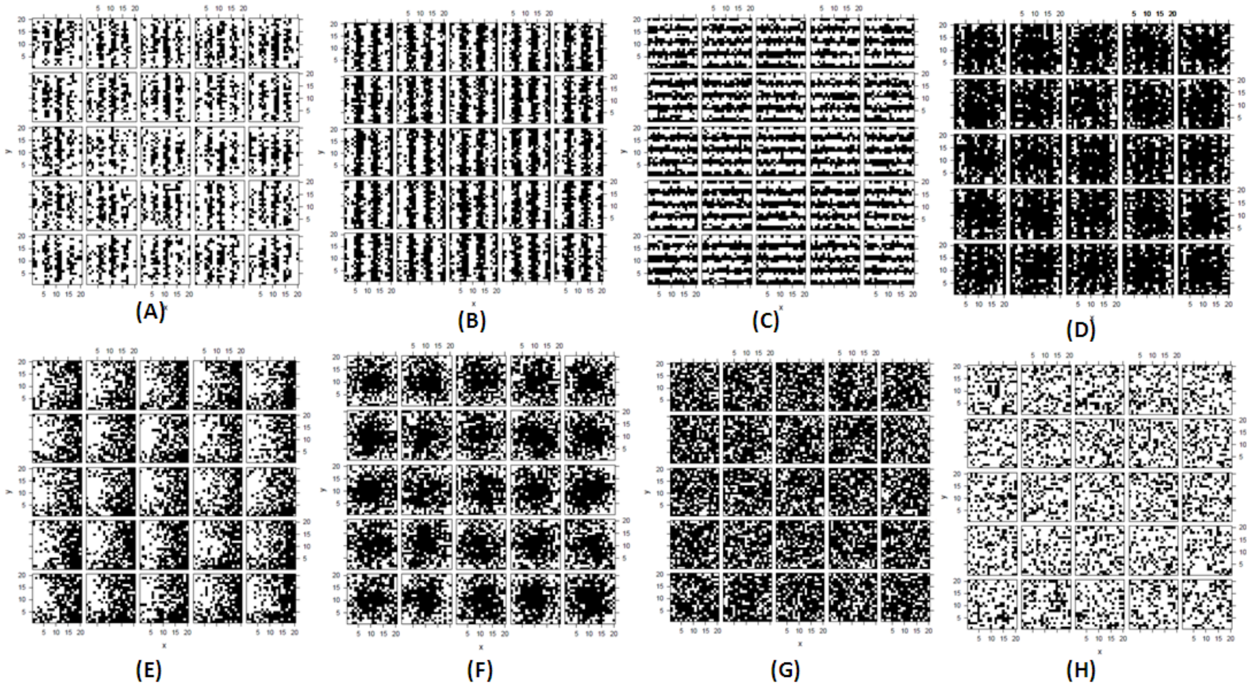


Figure 5.39 Binary defect patterns for different spatial probabilities

5.3.4. Extraction of Spatial Statistics for Binary Defect Patterns

After the defect patterns are generated, we need to extract statistics that can capture the characteristics of each defect pattern. There are two types of statistics we want to extract for each defect pattern: spatial statistics to capture the spatial characteristics of defects, and the degree of clustering exhibited by defects. For the first, we have adopted joint count statistics (JCs) [5.27] to compute the adjacency degree of defects. The spatial variogram and correlogram are then computed for JCs. The purpose of JCs is to convert the binary defect map into numerical continuous values. For the second objective, we compute a cluster index for each defect pattern. The assumption here is that the degree of defect clustering affects final cell performance. We then utilize statistical prediction methods to chart the relationship between these extracted statistics and cell performances.

5.3.4.1. Computing Join-Count Statistics Using Spatial Correlogram and Variograms

Join-count statistics (JCs) have been applied in the field of defect pattern identification for IC manufacturing [5.28]. JCs measure the degree of adjacency between different levels of an attribute. For this work, joint count measures the adjacency of good and defective grids, where each grid is a mini-cell.

A join-count statistic contains two T statistics, one that measures the number of defective grids neighboring defective grids and another that measures the number of good grids neighboring good grids [5.29]. A join is formed when the two grids are located in the same neighborhood. Let H denote a set of neighbors, and let n denote the total number of grids per cell. Under certain neighborhood constructions system, the number of possible joins is given by:

$$c = \sum_{i<j} w_{ij} \tag{5.21}$$

Where $w_{ij} = 1$ if (i,j) are in the same neighborhood, 0 elsewhere.

We can have three types of joins: 0-to-0 join (between good grids), 0-to1 join between functional and defective, and 1-to-1 join. To discriminate among the three joins, an indicator variable is introduced for grid i as $x_i = 1$ for defective, 0 for good. Let c_{00}, c_{01} and c_{11} denote the numbers of 0-to-0, 0-to-1, and 1-to-1 joins, then:

$$c_{00} = \sum_{i<j} w_{ij}(1-x_i)(1-x_j) \tag{5.22}$$

$$c_{01} = \sum_{i<j} w_{ij}(x_i - x_j)^2 \tag{5.23}$$

$$c_{11} = \sum_{i<j} w_{ij} x_i x_j \tag{5.24}$$

c_{00}, c_{01} and c_{11} depend on the neighborhood construction rules. The King's move and Rook's move neighborhood construction rules are the most popular. The former is adopted in this work.

Let $H(g)$ denote a set of *g-th* order neighbors, defined as grids that are g distant from each other. Join length g corresponds to the distance between the two grids involved. In this work, Manhattan distance is used as the distance metric. If the distance between two grids is denoted by $d(i,j)$, $H(g)$ can be written as:

$$H(g) = \{(i,j) \in W | d(i,j) = g\} \quad for\ g = 1,2,\ldots.,m \tag{5.25}$$

where W is a collection of all possible joins within the 20x20 grids map and *m* is the maximum length of join. The number of *g-th* order joins is:

$$c(g) = \sum_{i<j} w_{ij}(g) \tag{5.26}$$

with $w_{ij}(g) = 1 \quad if\ (i,j) \in H(g)$ else 0.

A generalized JC-based statistic with g_{th} order neighbors as a measure of spatial autocorrelation can be obtained as follows:

$$T(g) = \alpha_0 f\big(c_{00}(g)\big) + \alpha_1 f\big(c_{11}(g)\big) \tag{5.27}$$

Where $c_{00}(g)$ and $c_{11}(g)$ are the number of *g-th* order 0-to-0 and 1-to-1 joins and $f(\cdot)$ is a monotonic function. α_0 and α_1 are subjected to constraint that :

$$\alpha_0 + \alpha_1 = 1;\ (\alpha_0, \alpha_1) = (p, q) \tag{5.28}$$

p is the defective rate equal to $\left({}^{b}/_{N}\right)$, where b is the number of defective local cells and N is the number of total local cells, 400 in our work. q is equal to $1 - p$. Based on central limit theorem, the standardized statistic $T(g)$ can be approximated using standard normal distribution as follows:

$$Z_T(g) = \frac{T(g) - c(g)pq}{\sqrt{c(g)p^2q^2}} \tag{5.29}$$

where $(g) = c_{00}(g) + c_{11}(g) + c_{01}(g)$.

The $Z_T(g)$ is computed for all mini-cells across the 20x20 grid. Then, each cell with a different defective pattern can be characterized by a 20x20 numerical matrix. This 20x20 numerical matrix can then be viewed as a map with continuous values to which we can apply a spatial variogram or correlogram to capture spatial differences.

To characterize different defect patterns, empirical spatial variogram and correlogram were computed for each cell based on their JCs across the 20x20 grids. In figures 5.40- and 5.41, we show corresponding spatial correlograms and variograms for defect patterns shown in figure 5.39.

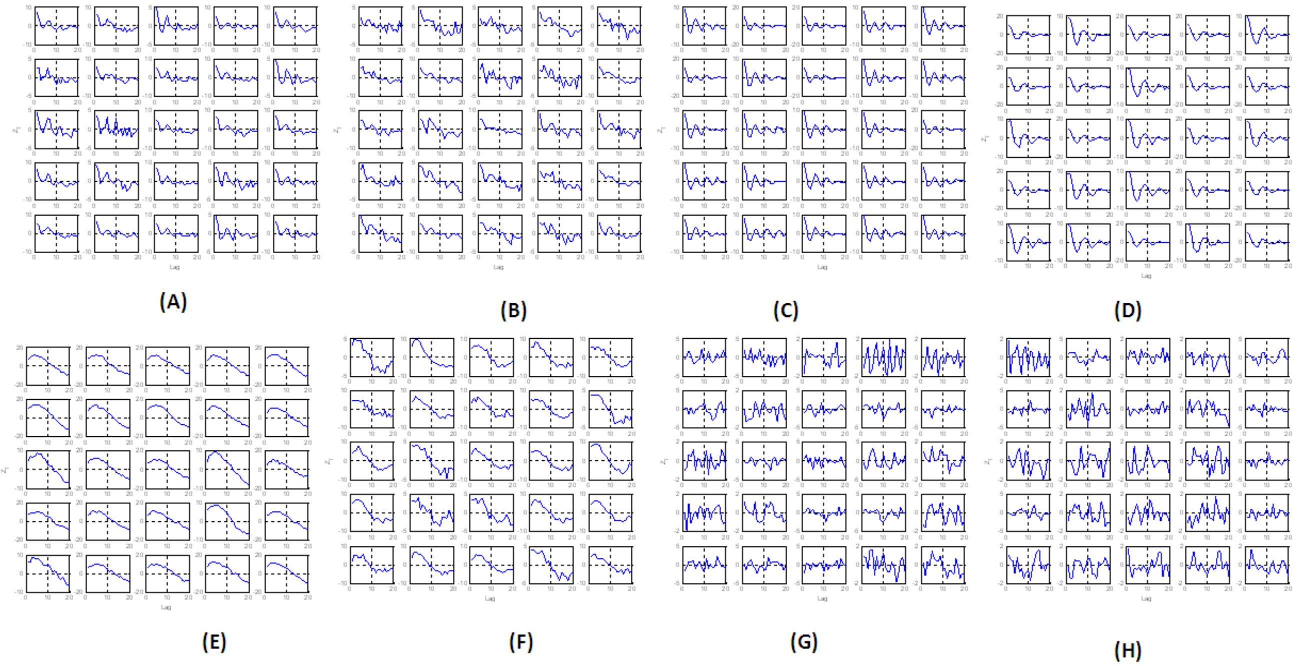


Figure 5.40 Corresponding spatial correlogram for spatial probabilities

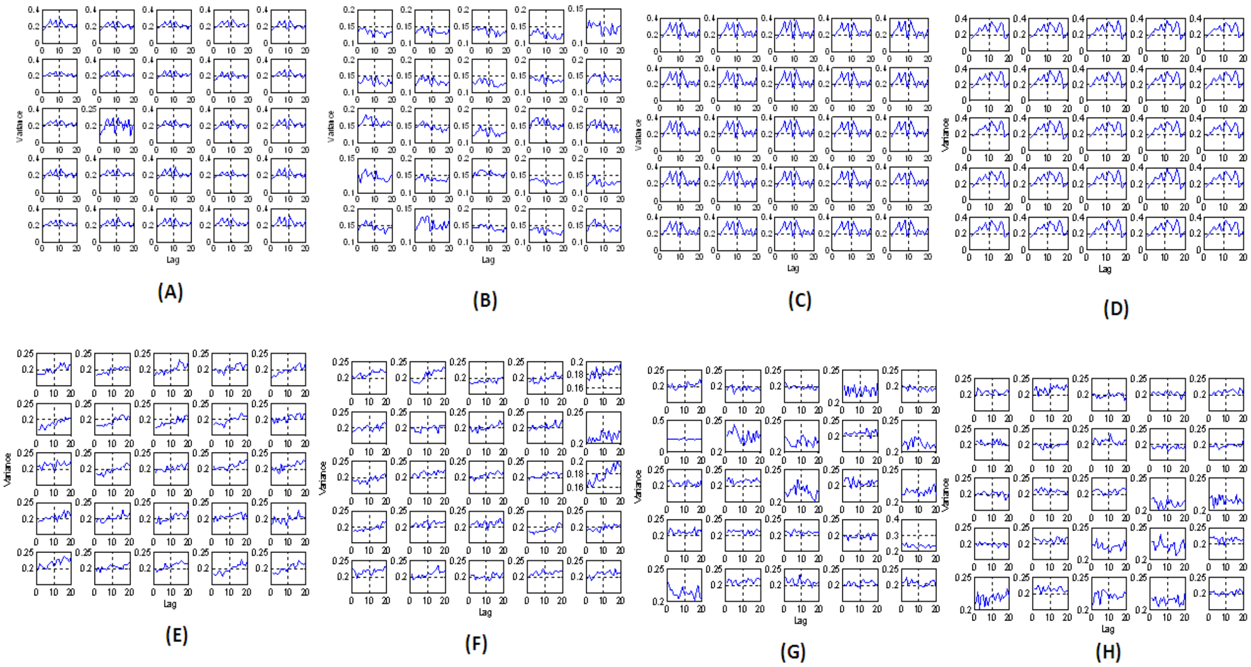


Figure 5.41 Corresponding spatial variogram for spatial probabilities

5.3.4.2. Computing Clustered Index for Defective mini-cells

Besides join-count statistics, we adopt three statistical indexes to compute the degree of clustering for the defective mini-cells. We expect that the degree of clustering of defective mini-cells is correlated with differences in the final electrical performance of entire cells. If we can characterize the clustering sufficiently with statistical indexes, then a regression or classification model can be trained to infer final solar cell performance based on defect images at an earlier stage of processing. The clustering indexes described in this section are nonparametric; there is no assumption as to the defective mini-cells' distribution. In this work, we have adopted three clustering indices to capture degrees of defective mini-cell clustering for each defect pattern.

Degree of defect clustering can be measured based on the locations of defects. Jun [5.30] utilized projected defect location to describe defect patterns on semiconductor wafers. Thus, CI_j is computed using the x and y and coordinates (x_i, y_i) of defects' locations on the wafer map. The mean and standard deviation of intervals between adjacent defects' coordinates are then used as metrics for measuring the degree of clustering. Despite the CI_j's advantage of requiring no statistical assumption about defect distribution, it tends to underestimate the degree of defect clustering.

Definition 1:

The first cluster index, CI_M, was proposed by Tong [5.31] to relieve the drawbacks of CI_J. It uses one dimensional rotation to compute the degree of clustering based on defect location coordinates. It is computed according to the following five steps:

1) Project the defect coordinates (x_i, y_i) onto a new axis obtained by rotating the *x*-axis counterclockwise using angle θ^0. The new coordinates for the *i*th defect with respect to θ^0 can be calculated as follows:

$$x^*_{i,\theta} = cos\theta * x_i + sin\theta * y_i \tag{5.30}$$

where *i* denotes the *i*th defect, and θ represents a rotating angle, where $0 \leq \theta \leq 180$.

2) Sort the $x^*_{i,\theta}$ in ascending order and calculate the intervals between each adjacent coordinate value $x^*_{i,\theta}$. The intervals between each adjacent coordinate value $x^*_{i,\theta}$ then can be calculated as follows:

$$v_{i,\theta} = x^*_{i,\theta} - x^*_{i-1,\theta} \tag{5.31}$$

3) Calculate the square coefficient of variation (SCV) for $v_{i,\theta}$. The SCV for $v_{i,\theta}$ can be determined as follows:

$$SCV_\theta = \frac{s^2_{v,\theta}}{\bar{v}^2_\theta} \tag{5.32}$$

where SCV_θ represents the squared coefficient of variation for $v_{i,\theta}$:

$$\bar{v}^2_\theta = \left(\sum_{i=1}^{n} v_{i,\theta}\right) \Big/ n \tag{5.33}$$

$$s^2_{v,\theta} = \left(\sum_{i=1}^{n} \left(v_{i,\theta} - \bar{v}_\theta\right)^2\right) \Big/ (n-1) \tag{5.34}$$

4) Change the angle of θ and calculate the corresponding $\theta = 1^\circ$. The number of 180 SCV_θ values with respect to θ increased by $\theta = 1^\circ$, can be obtained through Steps 1-3.

5) Calculate the CI_M. According to the SCV_θ values obtained from step 4, the average SCV_θ values determine the clustering index CI_M as follows:

$$CI_M = \frac{\sum_{\theta=0}^{180} SCV_\theta}{180} \tag{5.35}$$

Definition 2:
The second clustering index [5.32] takes advantage of the angle between x and *y* coordinates for each defect on a wafer. It can be computed in the following steps:

1) Calculate the angle between defect coordinate with *x*-axis in the first quadrant. Because randomly distributed defects tend to have consistent angle intervals, then the variance of all angle intervals is expected to be small. The angles between each defect's coordinates are computed as follows:

$$\theta_i = \tan^{-1}\left({}^{y_i}\!/_{x_i}\right) \tag{5.36}$$

where $\theta_i$is the angle between the *i*th defect coordinates (x_i, y_i) with respect to *x*-axis in the first quadrant.

2) Rank $\theta_i s$ in ascending order and calculate their intervals.

$$A_i = \theta_i - \theta_{i-1} \tag{5.37}$$

where A_i is the angle interval and large variance of $A_i s$ indicates a higher degree of clustering for defects.

3) Determine the defect clustering index based on:

$$CI_A = \frac{S_A^2}{\bar{A}^2} \tag{5.38}$$

where $\bar{A} = \sum {}^{A_i}\!/_{n}$; $S_A^2 = \sum {}^{(A_i - \bar{A})^2}\!\big/_{(n-1)}$

Definition 3:
The third clustering index utilizes the distance between defect locations [5.31]. CI_D is computed as follows:

1) Calculate the Euclidean distance between each defect coordinate and origin of the cell map.

$$L_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{5.39}$$

Where L_i represents the Euclidean distance of the *i*th defect coordinates with respect to the origin (x_0, y_0).

2) Rank $L_i s$ in ascending order and calculate their intervals.

$$D_i = L_i - L_{i-1} \tag{5.40}$$

Where D_i represents the distance intervals, and large variance of $D_i s$ indicates the higher degree of clustering for defects.

3) Determine the defect clustering index based on:

$$CI_D = \frac{S_D^2}{\overline{D}^2} \tag{5.41}$$

where $\overline{D} = \sum {}^{D_i}\!/_{n}$; $S_D^2 = \sum {}^{(D_i - \overline{D})^2}\!\big/_{(n-1)}$.

In the case of randomly distributed defects, the CI_Dvalues for both angles and distance intervals tend to be small. Thus, a large CI_D indicates higher degree of clustering.

5.3.5. Prediction Model Results

In this section, we created a statistical prediction model that utilizes the defective rate, and computed spatial correlograms, variograms, as well as clustering indices including CI_M , CI_A

and CI_D as input to predict P_{mp} of solar cells. Each spatial variogram and correlogram contains information up to Lag 20 given the cell is simulated on a 20x20 grids. There were a total of 800 solar cells simulated, with 8 spatial probability defects described in Section 5.3.3, and 100 cells generated for each spatial probability class. Each spatial probability class is denoted as a defect class. The simulation assumes binary defects, where a defective mini-cell is assumed to have electrical parameters of: $I_{01} = 1e^{-5}A$; $I_{02} = 1e^{-9}A$; $R_{sr} = 0.01\Omega$; $R_{sh} = 0.1\Omega$; $I_{ph} = 1.34\ A$ and a non-defective mini-cell is assumed to have electrical parameters: $I_{01} = 1e^{-7}A$; $I_{02} = 1e^{-12}A$; $R_{sr} = 0.01\Omega$; $R_{sh} = 1000\Omega$; $I_{ph} = 4.34\ A$. The $R_{sr} = 0.01\Omega$ accounted for contact resistance and material body resistance, and is assumed to be constant for both defective and non-defective cases, since higher defect density mostly affects photo-current generation and shunting path across cell junctions.

The defective rate for generated cells is shown against their corresponding defect class in figure 5.42.

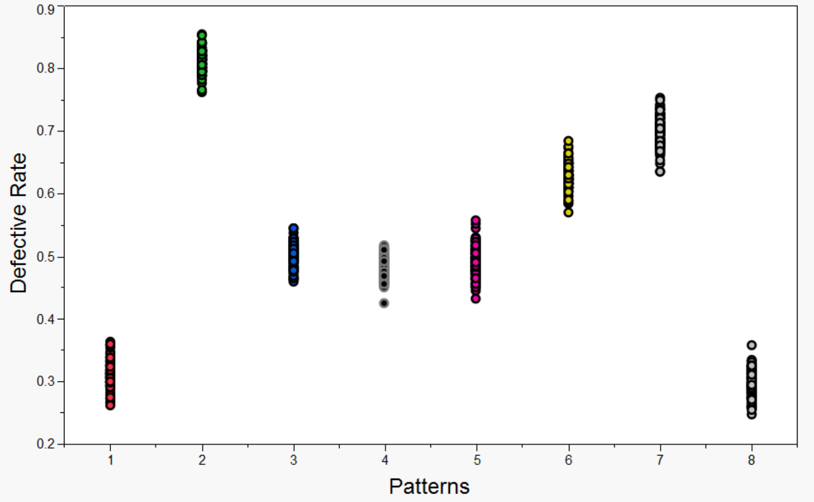
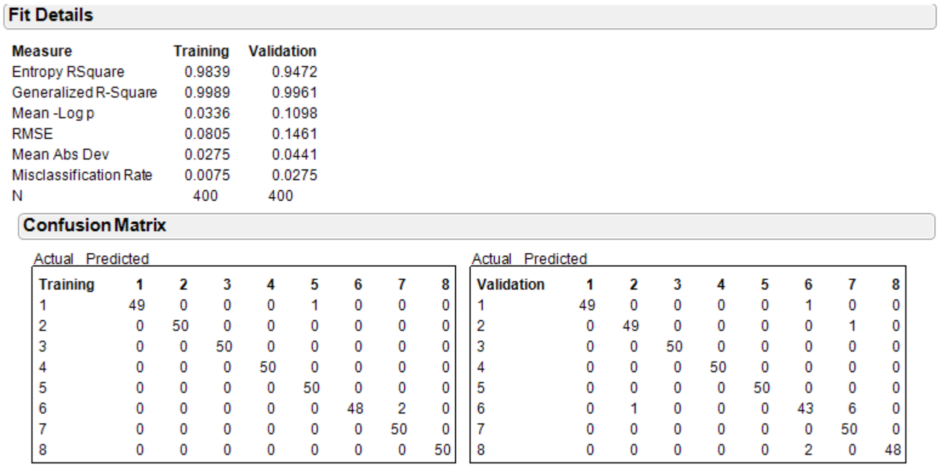


Figure 5.42 Corresponding Defective rate for each defect class

We first compare how well each cell can be classified into corresponding defect patterns using the extracted statistics described above. The classification methods used here are classification tree and BNN. 400 cells are sampled from 800 cells in such a way that there are 50 cells from each defect class. These sample cells are used as a training set while the remaining cells are used as a testing set. Both classification tree and BPNN models can achieve misclassification rates as low as 0.75%, meaning only 3 cells are misclassified among 400 testing cells. We present the result of classification tree model in figure 5.43. The variables providing the most classification power are:

Lag1_spatialcorrelogram	Lag2_spatialcorrelogram	Lag10_Spatialcorrelogram	Lag8_Variogram	Defective rate



Fit Details

Measure	Training	Validation
Entropy RSquare	0.9839	0.9472
Generalized R-Square	0.9989	0.9961
Mean -Log p	0.0336	0.1098
RMSE	0.0805	0.1461
Mean Abs Dev	0.0275	0.0441
Misclassification Rate	0.0075	0.0275
N	400	400

Confusion Matrix

Actual Predicted

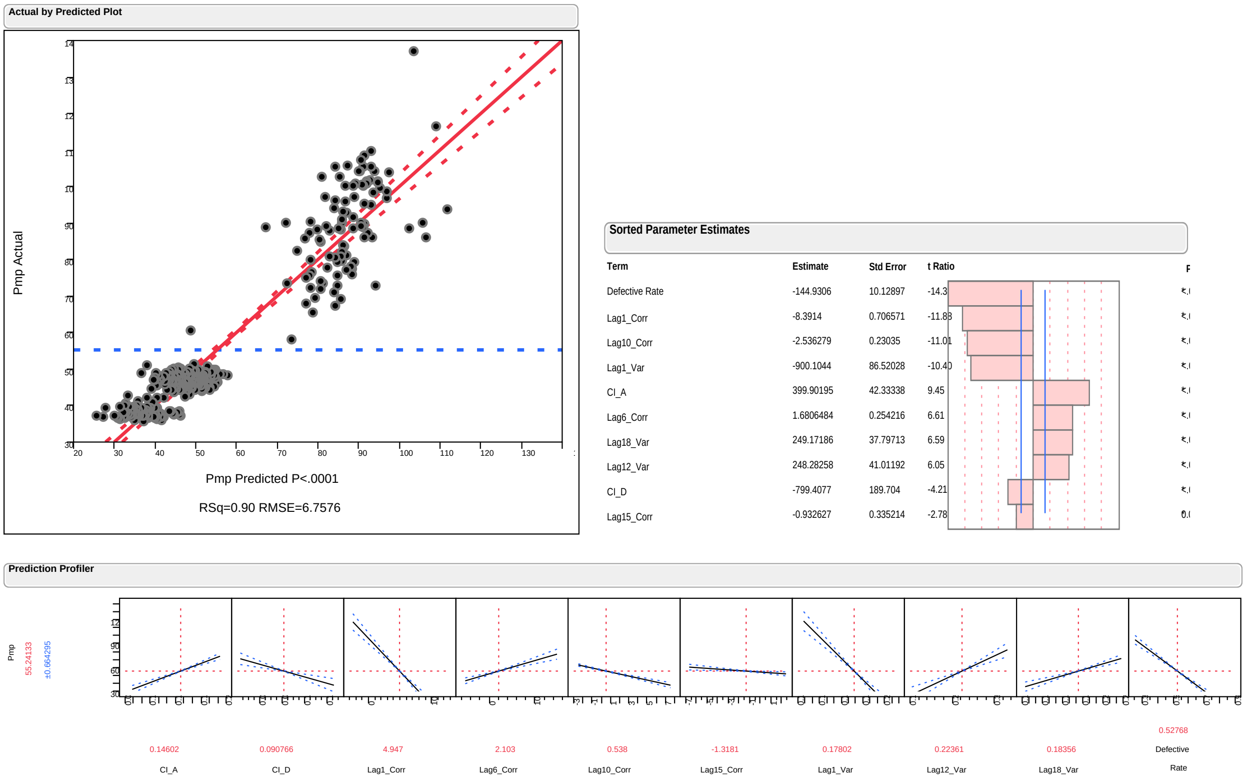
Training	1	2	3	4	5	6	7	8
1	49	0	0	0	1	0	0	0
2	0	50	0	0	0	0	0	0
3	0	0	50	0	0	0	0	0
4	0	0	0	50	0	0	0	0
5	0	0	0	0	50	0	0	0
6	0	0	0	0	0	48	2	0
7	0	0	0	0	0	0	50	0
8	0	0	0	0	0	0	0	50

Actual Predicted

Validation	1	2	3	4	5	6	7	8
1	49	0	0	0	0	1	0	0
2	0	49	0	0	0	0	1	0
3	0	0	50	0	0	0	0	0
4	0	0	0	50	0	0	0	0
5	0	0	0	0	50	0	0	0
6	0	1	0	0	0	43	6	0
7	0	0	0	0	0	0	50	0
8	0	0	0	0	0	2	0	48

Figure 5.43 Pattern Classification result of Classification tree model

For the prediction of maximum power output P_{mp}, we employed a stepwise variable selection in combination with BPNN to predict the P_{mp} of the 400 testing cells. The R^2 is approximately 0.9, however, this is due to leverage points created by the defective rate being the dominant factor. The fit result is shown in figure 5.44, and we overlaid predicted and actual results for the 400 testing cells in figure 5.45.



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	P
Defective Rate	-144.9306	10.12897	-14.3	<.(
Lag1_Corr	-8.3914	0.706571	-11.88	<.(
Lag10_Corr	-2.536279	0.23035	-11.01	<.(
Lag1_Var	-900.1044	86.52028	-10.40	<.(
CI_A	399.90195	42.33338	9.45	<.(
Lag6_Corr	1.6806484	0.254216	6.61	<.(
Lag18_Var	249.17186	37.79713	6.59	<.(
Lag12_Var	248.28258	41.01192	6.05	<.(
CI_D	-799.4077	189.704	-4.21	<.(
Lag15_Corr	-0.932627	0.335214	-2.78	0.(

Figure 5.44 Stepwise variable selection with BPNN results

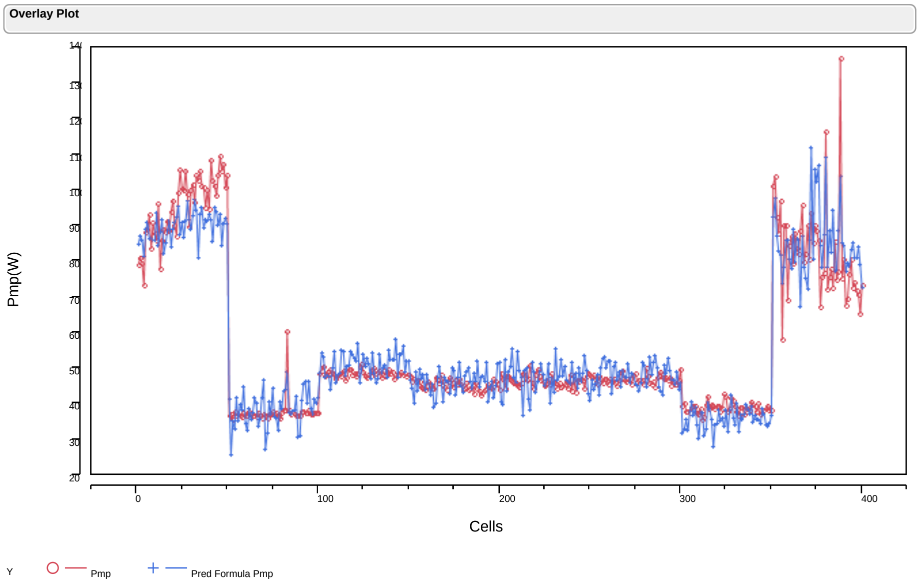


Figure 5.45 Overlay plot for Pmp prediction results from BPNN

The results in figure 5.44 show that the most dominant parameter for predicting p_{mp} is the defective rate, whereas all three cluster indices failed to provide predictive power. The overlay plot in Figure 5.45 indicates that the prediction model captures the differences of p_{mp} for different defect classes. However, it does not explain the variation of p_{mp} within a defect class, where defective rate is similar for all cells.

5.4. Conclusion and Future work

In this chapter, we investigated mismatch loss in solar PV networks due to manufacturing and environmental variations (Section 5.2). An empirical model based on extracted statistics was constructed to predict the mismatch loss of different PV networks. The empirical model was found to perform better than Bucciarelli's model. We also proposed using clustering with multivariate parameters. The results show that we can gain slightly better mismatch loss across the lifetime of solar PV networks. In Section 5.3, we employed a 3-D distributive diode array to model nonuniformities. We then created a statistical prediction model to explain the relationship between defect pattern and final cell performance. A R^2 of 0.9 was achieved with simulated data.

Future work can focus on using real defect maps with corresponding measured electrical performances. We might also explore simulating defect patterns with continuous defect concentrations using a device simulator such as Sentaurus, permitting extraction of more realistic defective cell performances.

References

[5.1] Wilson, K., D. De Ceuster, and R. A. Sinton. (2006). "Measuring the effect of cell mismatch on module output." In *Proc. 4th IEEE World Conf. Photovoltaic Energy Convers.*, May 2006, vol. 1, pp. 916–919.

[5.2] Swanson R., P. Verlinden, R. Crane, R. Sinton, C. Tilford. (1992). “High efficiency silicon solar cells.” *Proceedings of the 11th EVPSC*, Montreux; 35–40.

[5.3] Villalva, M. G., J. R. Gazoli, and E. R. Filho.(2009) “Comprehensive approach to modeling and Simulation of Photovoltaic Arrays.” *IEEE Trans. Power Electronics*, vol. 24, no. 5, pp. 1198–1208, May 2009.

[5.4] Green, Martin. (1981). *Solar Cells: Operating Principles, Technology and System Applications*. New York: Prentice Hall.

[5.5] Xiao, W., W. G. Dunford, and A. Capel. (2004) “A novel modeling method for photovoltaic cells.” in *Proc. IEEE 35th Annu. Power Electron. Spec. Conf. (PESC)*, vol. 3, pp. 1950 – 1956.

[5.6] Hezel, R. (2002) “Novel back contact silicon solar cells designed for very high efficiencies and low-cost mass production.” *Proceedings of the 29th IEEE PVSEC*, New Orleans; 114–117.

[5.7] Honsberg, C. and S. Bowden. (1999). *Photovoltaics CDROM: Devices, Systems, and Applications*. Photovoltaics Centre, The University of New South Wales, Australia.

[5.8] Nelson, J. (2003). *The Physics of Solar Cells*; Imperial College Press: London.

[5.9] Reis, A. M, N.T. Coleman, M.W. Marshall, P.A. Lehman, C.E. Chamberlain. (2002). “Comparison of PV module performance before and after 11-years of field exposure.” *Proceedings of the 29th IEEE Photovoltaic Specialists Conference,* New Orleans, Louisiana, USA.

[5.10] Kaushika N.D., A.K. Rai. (2007). “An investigation of mismatch losses in solar photovoltaic cell networks.” *Energy* 32:755e9.

[5.11] Bucciarelli, L. (1979). “Power loss in photovoltaic array due to mismatch in cell characteristics.” *Sol Energy* 23:277–88.

[5.12] Chouder, A. et al. (2009). Analysis model of mismatch losses in PV systems. *Journal of Solar Energy Engineering* 131 (May), 024504.

[5.13] Drif, M. et al., (2007). “Univer Project. A grid connected photovoltaic system of 200 kWp at Jae’n University. Overview and performance analysis.” *Solar Energy Materials and Solar Cells* 91, 670–683.

[5.14] Drif, M. et al., 2008. A new estimation method of irradiance on a partially shaded PV generator in grid-connected photovoltaic systems. Renewable Energy 33, 2048–2056.

[5.15] Field, Halden. (2002). “PV Measurements. “Cell binning method analysis to minimize mismatch losses and performance variation in Si-based modules.” *Photovoltaic Specialists Conference.. Conference Record of the Twenty-Ninth IEEE.*

[5.16] Schubert, M., S. Riepe, and W. Warta. (2006). “Spatially resolved trapping detection and correlation with material quality in multicrystallene silicon.” in *Proceedings of the 21st*

European Photovoltaic Solar Energy Conference, pp. 629–633, WIP, Dresden, Germany, September 2006.

[5.17] Tajima, M., M. Warashina, and T. Hisamatsu. Proc. 14th workshop on Crystalline Silicon Solar Cells and Modules, ed. B.L. Sopori (NREL/BK-520-36622, Golden, CO 2004), pp. 27-34.

[5.18] Karpov, V. G., A.D. Compaan, D. Shvydka. (2004). "Random diode arrays and mesoscale physics of large-area semiconductor devices." *Phys. Rev.* B 69, 435.

[5.19] A. Zekry and A. Al-Mazroo. (1996). "A Distributed SPICE Model of a Solar Cell." *IEEE Transactions on Electron Devices* Vol. 43, No. 5, May, pp. 691-700.

[5.20] Antonini A., M. Stefancich, D. Vincenzi, C. Malagú, F. Bizzi, A. Ronzoni, and G. Martinelli. (2003). "Contact grid optimization methodology for front contact concentration solar cells." *Sol. Energy Mater. Sol. Cells*, vol. 80,pp. 155–166.

[5.21] Galiana, B., C. Algora, I. Rey-Stolle, I.G. Vara.(2005). "A 3-D model for concentrator solar cells based on distributed circuit units." *IEEE Transactions on Electron Devices* 52(12): 2552–2558.

[5.22] Foss, S., B. Olaisen, E. Marstein, and A. Holt. (2006). "A new 2.5d distributed spice model of solar cells." in *Proc. Eur. Photovolt. Sol. Energy Conf .Exhib.*, Dresden, Germany, vol. 21.

[5.23] Yadir, S., M, Benhmida, M. Sidki, E. Assaid, M. Khaidar. (2009) "New method for extracting the model physical parameters of solar cells using explicit analytic solutions of current-voltage equation." *The International IEEE Conference on Microelectronics* (ICM), 19-22 Dec, 2009, pp. 390 – 393.

[5.24] Kimm H. S., B.G. Morris, S.S. Han, G.S. May. (2008). "A comparison of genetic and particle swarm optimization for contact formation in high-performance silicon solar cells." In: *Proceedings of the international joint conference on neural networks.* Hong Kong, China: Institute of Electrical and Electronics Engineers Inc. p. 1531–5.

[5.25] Ye, Meiying, Xiaodong Wang, Yousheng Xu. (2009). "Parameter extraction of solar cells using particle swarm optimization." *Journal of Applied Physics* , vol.105, no.9, pp.094502-094502-8, May.

[5.26] Palma, F. D., G. D. Nicolao, G. Miraglia, E. Pasquinetti, and F. Piccinini. (2005) "Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing." *Pattern Recognit. Lett.*, vol. 26, pp. 1857-1865.

[5.27] Cliff, A. D., and J. K. Ord. (1981) *Spatial Processes: Models & Applications*. New York: Pion.

[5.28] Hsieh, H. W., and F. L. Chen. (2004). "Recognition of defect spatial patterns in semiconductor fabrication." *lnt. J. Production Res.*, vol. 42, no. 19, pp.4153–4172.

[5.29] Jeong Y., S. Kim, M. Jeon. (2008). “Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping.” *IEEE Trans Semicond Manuf* 21(4): 625–637.

[5.30] Jun CH, Hong, Y., S.Y. Kim, K.S. Park, H. Park. (1999). “A simulation-based semiconductor chip yield model incorporating a new defect cluster index.” *Microelectron Reliab* 39:451–456.

[5.31] Tong L-I, C.H. Wang, D.L. Chen. (2007). “Development of a new cluster index for wafer defects”. *Int J Adv Manuf Technol* 31:705–715.

[5.32] Tsai, W.-J., L.-I. Tong, and C.-H. Wang. (2008). “Developing a new defect cluster index.” *J. Chin. Inst. Ind. Engineers*, vol. 25, no. 1, pp. 18–30.

Chapter 6

Conclusion and Future Work

In this Chapter, we discuss the conclusion of three main research projects in this thesis in Section 6.1, and provide future direction for each work in Section 6.2.

6.1 Thesis Summary

This thesis first presents study in wafer-to-wafer metrology model creation with specific focus on plasma etching operation in Chapter 3. The prediction of wafer metrology with process tool sensor data is called Virtual Metrology (VM) model. The study for VM model focus on variable selection and model creation approaches. High dimensional and drifting nature of sensor data collected from production wafers makes it difficult to have a robust static VM model. To address the redundancy of data, we have explored several variable selection schemes from filter and wrapper approaches to select sensor parameters that are relatively significant to the response variable. Drifting nature of sensor data due to chamber events introduces artificial clusters in input data while the response variables do not show corresponding clusters. Therefore, we have also discussed detrending techniques to account for this phenomenon. However, there is not a general solution or detrending sensor parameters, and one has to rely on expert domain knowledge to apply appropriate detrending. Various multivariate prediction methods such as Principal Component regression (PCR), Partial Least Square (PLS), Neural Network (NN), Regression Tree (CART), and K-nearest neighbor (KNN) regression have been employed to predict the etch bias of plasma etching operation with sensor parameters as input. The results show that we can obtain a testing R^2 around 0.76 with NN model.

The thesis then presented FDC (fault detection and classification) methods for wafers using site-to-site metrology as input. We have utilized spatial statistics such as spatial variogram and multivariate statistics to detect wafers with disrupted patterns. The multivariate methods discussed include Mahalanobis distance, PCA and ICA. Among them, similarity factor derived from PCA $S_{\mathrm{PCA}}^{\lambda}$ tends to perform better. We then performed site selection methods to select optimal sites for process monitoring. The idea is to capture most information needed for process monitoring with fewer measured sites. The best performing site selection method was based on clustering-PCA variable selection approach where one first partition the sites into subset then select the most informative sites of each subset.

In last section of thesis, we model the variability of solar PVs. SPICE simulator has been utilized to simulate different PV networks. We first investigated mismatch loss of different PV networks under environmental and manufacturing variations. Then an empirical model based on extracted statistics is then created to predict mismatch loss of different networks. The model result is then compared against analytical Bucciarelli's model. We then modeled nonuniformities of solar cell with distributive diode network. 800 cells with different defect patterns were simulated in SPICE, and spatial statistics based on join count along with clustering indices are computed for each cell. We then trained a statistical prediction model with BPNN to predict the maximum power output of each cell based on their defect patterns. A $R^2 \sim 0.9$ was achieved across cells from different defect classes. However, the model lacks of predictive power for cells coming from same defect class.

6.2 Future Work

Future work for creating robust VM model would require more focus on selecting variables and accommodation of process dynamics. The selected variables should capture within-lot and lot-to-lot variation for response variables such as CD bias or Etch rate. We have observed there are some predictor variables exhibit higher correlation with response variables within lot and across several lots, however this correlation is weaken if we sample wafers from lots ranging over a longer production period. One could utilize state space approach to adaptively update the model coefficients based on evolving correlation among variables. VM model can also be applied as a system health indicator rather than replacing real metrology measurement results. Thus threshold value can be set to trigger alarm for equipment maintenance.

The work for enhanced metrology with site-to-site metrology data can be extended into two directions. The first direction would be implementation of online FDC model where an adaptive approach can be applied to update process mean and correlation structure among sites. The model would also need to be validated against real faulty wafers. Another direction would be selecting optimal sites for entire wafer map interpolation. In such case, engineer can only measure a few sites then predict the metrology results of other sites on the same wafer.

The next step to extend the work of modeling variability of solar cell would be utilize a device simulator that can couple realistic material and physical properties to simulate how different defect concentration can impact final cell performance. In this thesis, we have assumed binary defect patterns where mini-cells are classified into defective and non-defective cases. This might lead to mistreatment of interaction among adjacent mini-cells.

Reference