

# **PROCESS MODELING AND OPTIMIZATION USING INDUSTRIAL SEMICONDUCTOR FABRICATION DATA**

A Dissertation  
Presented to  
The Academic Faculty

By

Zubin Mevawalla

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy  
in  
Electrical and Computer Engineering



School of Electrical and Computer Engineering  
Georgia Institute of Technology  
May 2015

Copyright © 2015 by Zubin Mevawalla

# PROCESS MODELING AND OPTIMIZATION USING INDUSTRIAL SEMICONDUCTOR FABRICATION DATA

Approved by:

Dr. Gary S. May, Advisor  
*Dean, College of Engineering*  
*Georgia Institute of Technology*

Dr. Muhannad S. Bakir  
*Associate Professor, School of Electrical and*  
*Computer Engineering*  
*Georgia Institute of Technology*

Dr. Albert B. Frazier  
*Professor, School of Electrical and Computer*  
*Engineering*  
*Georgia Institute of Technology*

Dr. Paul A. Kohl  
*Regents Professor, School of Chemical and*  
*Biomolecular Engineering*  
*Georgia Institute of Technology*

Dr. Linda S. Milor  
*Professor, School of Electrical and Computer*  
*Engineering*  
*Georgia Institute of Technology*

Date Approved: Decemer 16, 2014



## ACKNOWLEDGMENT

I would like to thank Dr. May for giving me the opportunity to perform this research and engage in academia, and supporting me throughout.

Thank you to Dr. Frazier, Dr. Milor, Dr. Bakir, and Dr. Kohl for being on my committee and attending my defense.

Thank you to Mark Kiehlbauch and Masuji Honjo for your help and guidance at Micron Technology, Inc. Thanks also to Thien Lai, Benoit Andreani, Aida Frimel, and Ed Martin for helping me with data and experiments. Thanks to the several others who helped me navigate the many systems at Micron.

Thanks to LaJauna Ellis for your help in scheduling the defense, and in managing lots of Georgia Tech administrative procedure over the course of getting the degree.

Thanks to everyone in the ECE academic office, and OIE.

Thank you Van Nguyen.

Thank you my parents Kiran and Noshir, and my brother Anosh for your love and support.

# TABLE OF CONTENTS

<b>ACKNOWLEDGMENT</b> . . . . .	iii
<b>LIST OF TABLES</b> . . . . .	vii
<b>LIST OF FIGURES</b> . . . . .	viii
<b>CHAPTER 1 INTRODUCTION</b> . . . . .	1
1.1 Motivation . . . . .	1
1.2 Summary of work . . . . .	2
1.3 Thesis organization . . . . .	3
<b>CHAPTER 2 BACKGROUND</b> . . . . .	4
2.1 Semiconductor manufacturing industry . . . . .	4
2.2 Neural networks . . . . .	4
2.2.1 Neural Networks Applied to Semiconductor Manufacturing Issues . . . . .	8
2.2.1.1 Process Characterization, Process Modeling, and Recipe Generation . . . . .	8
2.2.1.2 Control . . . . .	11
Run-to-run . . . . .	11
Real-time . . . . .	12
Failure Detection and Diagnosis . . . . .	13
Virtual Metrology . . . . .	14
2.2.1.3 Summary . . . . .	15
<b>CHAPTER 3 RESEARCH AND DEVELOPMENT FAB MODELS</b> . . . . .	17
3.1 Introduction . . . . .	17
3.2 Data collection . . . . .	20
3.3 Processes modeled . . . . .	21
3.3.1 Hardmask etch 1 . . . . .	21
3.3.2 Hardmask etch 2 . . . . .	22
3.3.3 High-aspect-ratio etch . . . . .	22
3.4 Results . . . . .	23
3.4.1 Hardmask etch 1 . . . . .	23
3.4.2 Hardmask etch 2 . . . . .	25
3.4.3 High-aspect-ratio etch . . . . .	28
3.5 Sensitivity analysis . . . . .	30
3.6 Summary . . . . .	30
<b>CHAPTER 4 RAMPING STAGE MODELS</b> . . . . .	34
4.1 Introduction . . . . .	34
4.2 Method . . . . .	35
4.2.1 Feature selection . . . . .	35

4.2.1.1	Correlation based filter method, followed by SVM modeling . . . . .	35
4.2.1.2	Stepwise regression . . . . .	36
4.2.1.3	Minimum redundancy maximum relevance (mRMR) . . . . .	39
4.2.2	Modeling . . . . .	41
4.2.3	Genetic algorithm for optimized process recipe . . . . .	41
4.3	Results . . . . .	41
4.3.1	Correlation based filter method, followed by SVM modeling . . . . .	41
4.3.2	Stepwise regression . . . . .	44
4.3.3	Minimum redundancy maximum relevance (mRMR) . . . . .	48
4.3.4	Genetic algorithm for optimized process recipe . . . . .	48
4.4	Discussion . . . . .	53
4.5	Conclusion . . . . .	55
<b>CHAPTER 5</b>	<b>MANUFACTURING FAB MODELS . . . . .</b>	<b>56</b>
5.1	Introduction . . . . .	56
5.2	Data collection . . . . .	56
5.3	Modeling . . . . .	58
5.3.1	High-aspect-ratio etch . . . . .	58
5.3.2	Yield metrics . . . . .	59
5.3.2.1	Defect counts . . . . .	62
5.3.2.2	Probability of die failure . . . . .	62
5.4	Results . . . . .	62
5.4.1	High-aspect-ratio etch . . . . .	62
5.4.2	Yield metrics . . . . .	65
5.4.2.1	Defect counts . . . . .	65
5.4.2.2	Probability of die failure . . . . .	65
5.5	Sensitivity analysis . . . . .	67
5.6	Discussion and summary . . . . .	68
<b>CHAPTER 6</b>	<b>CHAMBER MATCHING EXERCISE . . . . .</b>	<b>70</b>
6.1	Introduction . . . . .	70
6.2	Method . . . . .	71
6.2.1	Identifying the outlying chamber . . . . .	71
6.2.2	Chamber modeling . . . . .	73
6.2.3	Process recipe optimization with Genetic Algorithms . . . . .	75
6.2.4	Dataset . . . . .	79
6.2.5	Verification of the experimental recipe on the production line . . . . .	79
6.3	Results . . . . .	79
6.3.1	Modeling results . . . . .	79
6.3.2	Verification of experimental recipe . . . . .	81
6.4	Discussion . . . . .	82
6.5	Summary . . . . .	84

<b>CHAPTER 7</b>	<b>COMPARISON TO LINEAR REGRESSION MODELS</b>	85
7.1	Introduction	85
7.2	Method	85
7.2.1	Linear regression	86
7.2.2	Hardmask etch 1	86
7.2.3	Hardmask etch 2	87
7.2.4	High-aspect-ratio etch	88
7.3	Results	88
7.3.1	Hardmask etch 1	88
7.3.1.1	Regression summary	88
7.3.1.2	Regression diagnostics	90
7.3.1.3	Comparison to neural network predictions	93
7.3.2	Hardmask etch 2	93
7.3.2.1	Regression summaries	95
7.3.2.2	Regression diagnostics	96
7.3.2.3	Comparison to neural network predictions	99
7.3.3	High-aspect-ratio etch	100
7.3.3.1	Regression summaries	100
7.3.3.2	Regression diagnostics	102
7.3.3.3	Comparison to neural network predictions	104
7.4	Discussion	107
<b>CHAPTER 8</b>	<b>COMPARISON TO SUPPORT-VECTOR-REGRESSION MODELS</b>	108
8.1	Introduction	108
8.1.1	Support vector regression	108
8.2	Method	110
8.3	Results	110
8.3.1	Hardmask etch 1	110
8.3.2	Hardmask etch 2	112
8.3.3	High-aspect-ratio etch	116
8.4	Discussion & conclusion	124
<b>CHAPTER 9</b>	<b>CONCLUSION</b>	127
9.1	Future work	130
<b>REFERENCES</b>		132

## LIST OF TABLES

Table 1	Comparison of the average prediction errors for neural network models trained with datasets chosen by the three different feature selection algorithms. The dataset chosen by stepwise regression resulted in models with the most accurate predictions. . . . .	48
Table 2	Comparison of the average prediction errors for the linear regression and neural network models. . . . .	93
Table 3	Comparison of the mean absolute percentage prediction errors for neural network and support vector regression models. . . . .	117

## LIST OF FIGURES

Figure 1	Overview of manufacturers response to demands of the market for semiconductor products. The arrow can be thought of as process of knowledge transfer between R&D and manufacturing fabs. The bullet points provide a comparison of the priorities of the two fabrication environments.	5
Figure 2	A 3-4-2 feed-forward neural network.	7
Figure 3	Schematic of an array of high aspect ratio holes in an oxide layer.	18
Figure 4	An overview of the process flow used to create the array of high-aspect-ratio holes in the oxide layer.	19
Figure 5	Overview of the creation of the hardmask	19
Figure 6	A top down view of the hardmask that is created.	20
Figure 7	Overview of model for hardmask etch 1.	21
Figure 8	Overview of model for hardmask etch 2.	22
Figure 9	Overview of the model for the high-aspect-ratio etch.	23
Figure 10	Comparison of measured (left) and predicted (right) CD (CDs were normalized). The x-axis and y-axis represent wafer location in mm.	24
Figure 11	The absolute percentage error for the neural network predictions. The legend on the right presents the different levels of mean absolute percentage error.	25
Figure 12	Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask (CDs were normalized). The x-axis and y-axis represent wafer location in mm.	26
Figure 13	Absolute percentage error for neural network predictions.	26
Figure 14	Comparison of measured (left) and predicted (right) output for y-direction CD in the hardmask (CDs were normalized). The x-axis and y-axis represent wafer location in mm.	27
Figure 15	Absolute percentage error for neural network predictions.	27
Figure 16	Comparison of measured (left) and predicted (right) output for x-direction CD in the oxide layer (CDs were normalized). The x-axis and y-axis represent wafer location in mm.	28
Figure 17	Absolute percentage error for neural network predictions.	29

Figure 18	Comparison of measured (left) and predicted (right) output for y-direction CD in the oxide layer (CDs were normalized). The x-axis and y-axis represent wafer location in mm. . . . .	29
Figure 19	Absolute percentage error for neural network predictions. . . . .	29
Figure 20	Sensitivity analysis for hard mask etch 2. Results are for x-direction CD in the hard mask. . . . .	31
Figure 21	A schematic of the data available for model building in a research and development fabrication environment. The fabrication process to be modeled is in the dashed blue rectangle. . . . .	32
Figure 22	Mean of the absolute percentage error for models with different numbers of input variables. . . . .	37
Figure 23	A plot of the correlation matrix for the reduced feature set. . . . .	38
Figure 24	A plot of the correlation matrix for the stepwise-regression reduced feature set. The algorithm selected 68 variables, but the first 25 are shown to allow comparison to Figure 23 . . . . .	40
Figure 25	A plot of the number of neurons in the hidden layer <i>versus</i> the RMSE for respective model. Graph is for the dataset chosen by stepwise regression. The testing RMSE trends upward with the addition of hidden layer neurons. . . . .	42
Figure 26	Density plots of the prediction errors of the same model, trained with different training algorithms. . . . .	43
Figure 27	Correlation + SVR. Plots of network output <i>versus</i> measured (Target) output for the training, testing, and their combination datasets. . . . .	45
Figure 28	mRMR. Plot of the training, testing, and validation errors <i>versus</i> training epoch. . . . .	46
Figure 29	Correlation + SVR. A histogram of the network prediction errors. . . . .	47
Figure 30	Regression results . . . . .	49
Figure 31	Error histogram . . . . .	50
Figure 32	mRMR. Plots of network output <i>versus</i> measured (Target) output for the training, testing, and their combination datasets. . . . .	51
Figure 33	mRMR. A histogram of the network prediction errors. . . . .	52
Figure 34	Genetic algorithm optimization plots. . . . .	54

Figure 35	A schematic of the data available for model building in a manufacturing environment. The fabrication process to be modeled is in the dashed blue rectangle. The lots whose data can be used to build a model for the yield metric are in the dashed red rectangles. . . . .	57
Figure 36	An overview of the model created for the high-aspect-ratio etch in the manufacturing fab. . . . .	59
Figure 37	An overview of the models created to predict yield metrics from inline measurements. . . . .	60
Figure 38	Sequential neural network model. . . . .	61
Figure 39	Comparison of measured (left) and predicted (right) CDs. The x-axis and y-axis represent wafer location in mm. . . . .	63
Figure 40	Absolute percentage error for neural network predictions for x-direction CD. . . . .	63
Figure 41	Comparison of measured (left) and predicted (right) CDs. The x-axis and y-axis represent wafer location in mm. . . . .	64
Figure 42	Absolute percentage error for neural network predictions for y-direction CD. . . . .	64
Figure 43	Comparison of measured (left) and predicted (right) under-etch defect counts. The x-axis and y-axis represent wafer location in mm. . . . .	65
Figure 44	Absolute percentage error for neural network predictions for under-etch defect counts. . . . .	66
Figure 45	Comparison of measured (left) and predicted (right) over-etch defect counts. The x-axis and y-axis represent wafer location in mm. . . . .	66
Figure 46	Absolute percentage error for neural network predictions for over-etch defect counts. . . . .	66
Figure 47	Comparison of measured (left) and predicted (right) probability of die failure. The x-axis and y-axis represent wafer location in mm. . . . .	67
Figure 48	Absolute percentage error for neural network predictions for probability of die failure. . . . .	68
Figure 49	Sensitivity analysis. The bars represent the percentage change in under-etch defects in response to a uniform change in their respective inputs . . .	69
Figure 50	Moving average chart of post-process metrology. The different lines represent the moving averages of the different chambers. Chambers whose lines were furthest from the target CD were candidates for modeling. . .	72



Figure 51	Overview of the model of the high-aspect-ratio etch. Numbers following the input description indicate etch step. . . . .	74
Figure 52	Example of a point in the input space encoded as a binary string. Points like these are labeled “individuals.” . . . .	76
Figure 53	A collection of possible solutions, encoded as binary strings, is labeled a population. . . . .	76
Figure 54	Illustration of crossover. The two individuals exchange their data to the right of the arrow. . . . .	77
Figure 55	An example of mutation where one bit is “flipped.” . . . .	77
Figure 56	Flowchart of a genetic algorithm. . . . .	78
Figure 57	Contour plots using measured and predicted data for the x-direction CD. . . . .	80
Figure 58	Contour plot of the percentage prediction error for x-direction CD. . . . .	81
Figure 59	Contour plots using measured and predicted data for the y-direction CD. . . . .	82
Figure 60	Contour plot of the percentage prediction error for y-direction CD. . . . .	83
Figure 61	Residual plots for diagnosis of regression for hardmask etch 1 . . . . .	92
Figure 62	Comparison of the residuals for regression and neural network models of hardmask etch 1. . . . .	94
Figure 63	Residual plots for diagnosis of regression model for x-direction CD for hardmask etch 2. . . . .	97
Figure 64	Distribution of the values of x-direction CD in the training dataset. . . . .	98
Figure 65	Residual plots for diagnosis of regression model for y-direction CD for hardmask etch 2. . . . .	99
Figure 66	Comparison of the distributions of the residuals for regression and neural network models for the hardmask etch 2 . . . . .	100
Figure 67	Residual plots for diagnosis of regression model for x-direction CD for high-aspect-ratio etch. . . . .	103
Figure 68	Residual plots for diagnosis of regression model for y-direction CD for high-aspect-ratio etch. . . . .	104
Figure 69	Comparison of the distributions of the percentage prediction error for regression and neural network models for the high-aspect-ratio etch . . .	105
Figure 70	Contour plot of x-direction CDs measured after the high-aspect-ratio etch	106

Figure 71	Contour plots for the x-direction CDs after the hardmask etch as predicted by the linear regression and neural network models. . . . .	107
Figure 72	Plot of the residuals <i>versus</i> the fitted values for the dependent variable, for each of the data points in the training dataset. . . . .	111
Figure 73	Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm. . . . .	111
Figure 74	Contour plot for absolute percentage error of the support-vector-regression predictions. . . . .	112
Figure 75	Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for hardmask etch 1. . . . .	113
Figure 76	Plot of the residuals <i>versus</i> the fitted values for the dependent variable, for each of the data points in the training dataset. . . . .	114
Figure 77	Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm. . . . .	115
Figure 78	Contour plot for absolute percentage error of the support-vector-regression predictions. . . . .	116
Figure 79	Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for hardmask etch 2. . . . .	117
Figure 80	Plot of the residuals <i>versus</i> the fitted values for the dependent variable, for each of the data points in the training dataset. . . . .	118
Figure 81	Comparison of measured (left) and predicted (right) output for y-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm. . . . .	118
Figure 82	Contour plot for absolute percentage error of the support-vector-regression predictions. . . . .	119
Figure 83	Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for hardmask etch 2. . . . .	120
Figure 84	Plot of the residuals <i>versus</i> the fitted values for the dependent variable, for each of the data points in the training dataset. . . . .	121

Figure 85	Comparison of measured (left) and predicted (right) output for x-direction CD in the oxide layer. The x-axis and y-axis represent wafer location in mm. . . . .	121
Figure 86	Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for the high-aspect-ratio etch. . . . .	122
Figure 87	Plot of the residuals <i>versus</i> the fitted values for the dependent variable, for each of the data points in the training dataset. . . . .	123
Figure 88	Comparison of measured (left) and predicted (right) output for y-direction CD in the oxide layer. The x-axis and y-axis represent wafer location in mm. . . . .	124
Figure 89	Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for the high-aspect-ratio etch. . . . .	125

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

The proliferation and uptake of integrated circuits has contributed significantly to what we recognize as our modern lives. It has allowed for unprecedented access to computation, information, and tools for communication. It is an enabler for the advancement culture, and science. This proliferation is underpinned by an industry of semiconductor manufacturers producing at the volume and cost that allow widespread availability.

The semiconductor manufacturing industry is uniquely characterized by Moore's Law. This was an observation made in 1965 that the complexity of semiconductor products increased exponentially and a prediction that this would continue [1]. This was revised a decade later to complexity doubling every two years [2], and it became a self-fulfilling prophecy as manufacturers strove to stay the course [3]. There is a debate to be had about this single-minded strategy for growth, as the industry approaches fundamental limits to device miniaturization, and Moore's law looks to be revised again to complexity doubling every three years [4]. For now the result is that semiconductor manufacturing is governed by short product life cycles and by the need for increasingly large capital expenditure for the increasingly sophisticated PP&E required to produce semiconductor products [5]. This places pressure on manufacturers to innovate quickly on products, and also to maintain high yields, throughputs, efficiencies, and overall equipment effectiveness (OEE) [6].

Happening alongside and as a result of this increased complexity and performance of semiconductor products is the increased complexity and performance of the fabrication equipment used to make these same products. Semiconductor products are manufactured through a series of chemical and physical fabrication processes performed upon a semiconductor substrate, typically a wafer of silicon. These fabrication processes and the equipment performing them have evolved faster than the science that can be used to explain them

[7] and thus cannot be characterized fully from first principles. Therefore, process development, investigation, and optimization has historically been carried out using statistical and experimental methods [7, 8] and “art and know-how” [9]. However, manufacturers are keen to do even better than this by leveraging the increasing amounts of data generated by fabrication equipment and collected in the typical semiconductor fabrication environment.

The nature of the data makes this task non-trivial and one that warrants academic research. Some aspects of this data, as enumerated very well in [8], are its high dimensionality in terms of observations and variables, highly non-linear relationships, noise, outliers, missing data, unusual distributions, and mixtures of categorical and numerical data. The desire of semiconductor manufacturers to unearth useful information and knowledge about their fabrication processes from these fab databases, containing large difficult datasets, formed the impetus for this research.

## **1.2 Summary of work**

Manufacturers address the distinct operational objectives of product innovation and manufacturing efficiency by having separate fabrication facilities (“fabs”) for development and manufacturing. Additionally, the industrial manufacture of a semiconductor product proceeds through several stages of production. These are typically a research and development (R&D) stage, a ramping stage, and a manufacturing stage. These production stages are distributed over the different fabs.

These differences in fabrication environment and stage of production result in differences in the characteristics of production of a semiconductor product over its manufacturing lifetime. Some examples of these differences are device yield, breadth of processing conditions, throughput, number of reaction chambers operating in parallel, metrology, and data collection. These differences are reflected in the data available in the fab databases.

This research explores the use of a neural network modeling and genetic algorithm optimization method with these different datasets. The focus is on a high-aspect-ratio etch

process across the different fabs and production stages. Models are built from process input variables to post-process metrology, and from process input variables to yield metrics. In the latter case, there can be tens of processes occurring between the model input and output variables.

I demonstrate the usefulness and industrial application of neural network process modeling and genetic algorithm recipe optimization by performing a reaction chamber matching exercise on a manufacturing line. The performance of a reaction chamber can deviate from target, either in terms of its post-process metrology or its associated yield metrics. The method developed herein generated an optimized recipe that brought the outlying behavior of a chamber closer to target and closer to that of the other chambers (“chamber matching”). This is one of many possible applications. It was chosen because it demonstrates both the fidelity of the process models and the effectiveness of the optimization algorithm.

### **1.3 Thesis organization**

Chapter 2 presents some background about the semiconductor manufacturing industry, and some theoretical background for the primary data mining tools and algorithms employed in this research. Chapter 3 details the use of neural networks to model three dry etch processes in a R&D fab. Chapter 4 details the use of neural networks to model a high-aspect-ratio etch process that has recently been introduced to a manufacturing fab for a product that is being ramped to manufacturing volumes. Chapter 5 details the use of neural networks to model a high-aspect-ratio etch process that has reached a level of maturity in a manufacturing fab. Chapter 6 details a chamber matching exercise, which is a potential industrial application of neural network modeling. Chapter 7 presents a comparison of the performance of the neural network models to that of linear regression models. Chapter 8 presents a comparison of the performance of the neural network models to that of support vector regression models. Chapter 9 presents some concluding thoughts and a discussion on possible future research.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Semiconductor manufacturing industry**

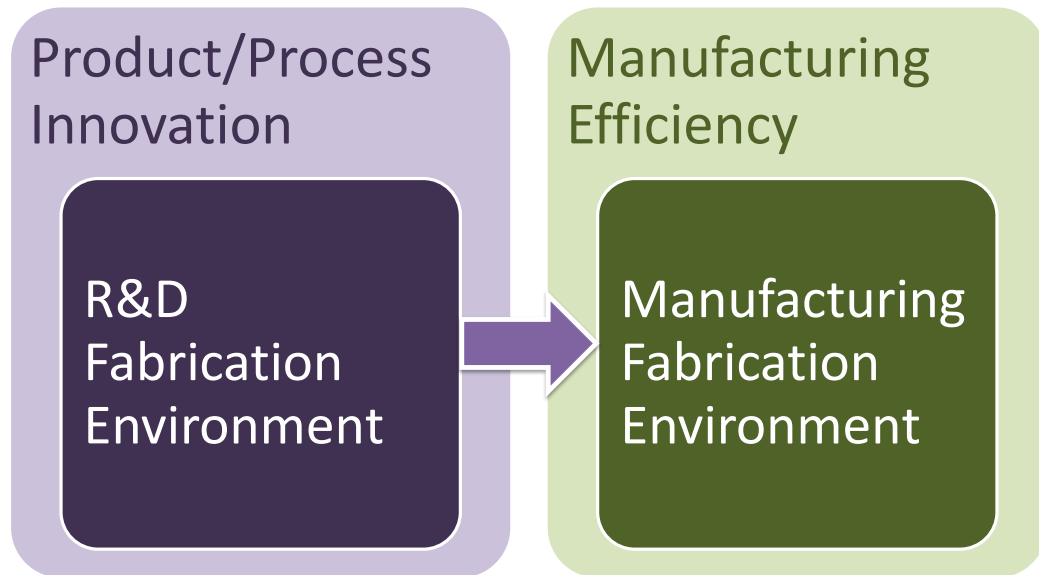
Manufacturers address the distinct operational objectives, of product innovation and manufacturing efficiency, by having separate fabrication facilities for development and commercial scale manufacturing. See Figure 1. This concept became widespread in the mid-1970s [7] and has been cited in [9] as the most effective management tool for bringing new processes to manufacturing. New process introduction is important because product innovation is dependent on process innovation in semiconductor manufacturing. This is so to a greater degree than in other manufacturing industries like automotive [7].

A firm’s ability to develop and introduce new processes to manufacturing forms part of its competitive advantage. According to the model created in [5], the time at which a manufacturer starts ramping to production volumes is the largest contributor to profitability. Thus methods that accelerate characterization of fabrication processes, yield learning, or ramping of production are very valuable to a manufacturer.

#### **2.2 Neural networks**

Artificial neural networks are empirical modeling tools. They are non-linear computational systems that can “learn” from data presented to them. These properties make them attractive for representing complex semiconductor manufacturing processes, as they can be used to create sophisticated models even when deep knowledge of the underlying physics and chemistry of a process is unavailable.

An artificial neural network is made up of simple processing units called “neurons”, which are interconnected through weighted connections. A neuron performs a simple mathematical procedure: summing its inputs and passing that result through a sigmoidal function to form its output. Interconnection in these networks enables certain functionality



- Process characterization
- Metrology
- Low yields
- Yield issues NOT ascribable to upstream processes
- Process control
- Throughput
- High yields
- Yield issues ascribable to upstream processes

Figure 1: Overview of manufacturers response to demands of the market for semiconductor products. The arrow can be thought of as process of knowledge transfer between R&D and manufacturing fabs. The bullet points provide a comparison of the priorities of the two fabrication environments.



[10]. The rules determining how these neurons are positioned and connected to each other determine the network architecture. Network architectures have significant effect on the performance of the network and the type of function the network is best suited for. Several different architectures have been studied in terms of their use toward semiconductor manufacturing including: multilayer feed-forward neural networks [11], self-organizing maps [12], and radial basis function networks [13].

Feed-forward neural networks were used for this research. The feed-forward neural network architecture is commonly used for data fitting. In a feed-forward neural network, the neurons are arranged in layers: an input layer, one or more hidden layers, and an output layer. Figure 2 depicts a feed-forward neural network. Neurons in a feed-forward network can pass their output to neurons in a subsequent layer only. A feed-forward neural network with a single hidden layer with enough neurons can approximate any measurable function to the desired level of accuracy [14]. Thus, a large enough neural network can accurately represent the complex mapping between the input variables and output variables of a fabrication process. The number of input and output layer neurons are determined by the number of input and output variables to the model respectively.

Training is the process of adjusting the network parameters (connection weights), using process data (empirical knowledge [15]), so that network behavior matches that of the process being modeled. The error back-propagation algorithm is used to train the network in this study. In this algorithm, the weights are initially randomized. Data is presented to the network in input-output pairs (measured data). The input data is presented to the input layer of the network, where the input layer neurons perform their calculations and propagate the results to the subsequent layer (hidden layer) through the weighted connections. Similarly the hidden layer neurons then perform their calculations and propagate the results to the next layer through weighted connections, and so on and so forth until the information reaches the output layer. The resulting network output is compared to the measured output data. The network weights are adjusted so as to decrease this error. This is repeated until

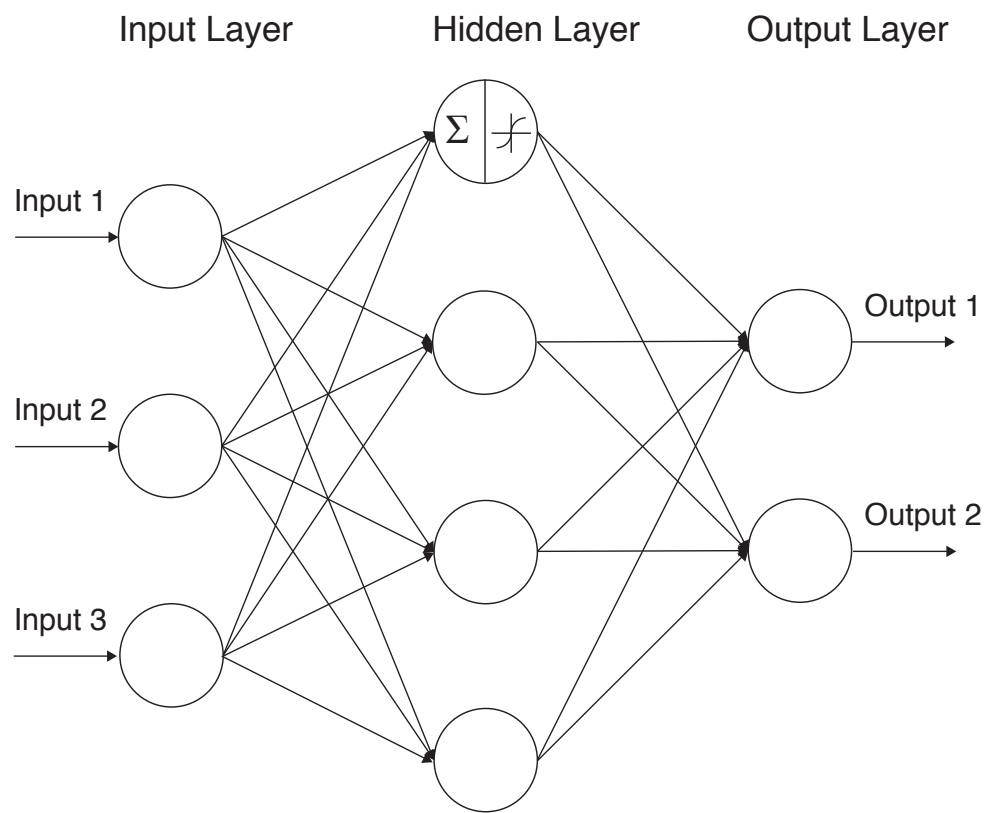


Figure 2: A 3-4-2 feed-forward neural network.

the error reaches an acceptable level or until the error converges to some asymptote. A detailed description of the error back-propagation algorithm can be found in [16].

The majority of the data collected (~90%) was used for training the neural network models. The remaining subset was used for testing. The test data is used to gauge the predictive abilities of the neural network models on datasets they have not been trained on. The resulting output (predicted output) is then compared to the real world data obtained by experiment (the output of the testing dataset). In this way, a judgment can be made about how well the models are performing [17].

### **2.2.1 Neural Networks Applied to Semiconductor Manufacturing Issues**

The literature provides several investigations into the use of neural networks for semiconductor manufacturing. For example, neural networks have been used for process characterization, recipe generation, control, failure detection, failure diagnosis, and circuit design. They have also been used across the many process modules like etch, implant for example. The type of neural network and the kinds of data used to train them, vary across the different applications. In many instances, the methods using neural networks as part of the solution equal or outperform more traditional approaches.

#### *2.2.1.1 Process Characterization, Process Modeling, and Recipe Generation*

Process characterization refers to how well a firm understands a fabrication process in terms of its response to its input variables [7]. Neural networks have been used to characterize a variety of fabrication processes, and have found particular use in modeling plasma etch processes [18]. Several examples of their use with plasma etch processes are cited in a literature review by Ringwood *et al.* [19].

In [17], Himmel and May use the data from an earlier project, that modeled the removal of a polysilicon film with plasma etch using response surface methodology (RSM). They used the data to make neural network models of the same process and compared modeling and prediction errors to those for the RSM models. They found that the experimental errors

were 38.3% lower for the neural network models. They also found that neural networks outperformed the RSM models when trained with a smaller dataset. The reason cited for this increased predictive capability was the non-parametric nature of neural network modeling.

Once there is a model of a process, from its manipulatable input variables to its output, it can be used for recipe generation. In [20], Han and May model a plasma enhanced chemical vapor deposition process with neural networks, then use the model with genetic algorithms to generate an optimal process recipe. Training data was created for the model using a fractional factorial experiment. The genetic algorithm (GA) searched the input space to the model to find the settings for the five input variables that resulted in the optimal process responses. In [21], they verify experimentally a recipe created in this way. They also compared recipes generated with genetic algorithms to those created with other optimization routines like Nelder and Mead's simplex method, Powell's algorithm, and hybrid algorithms (GA + simplex or GA + Powell). The recipes generated by genetic algorithms had the best performance, as determined by the quality of the resulting films.

In [22], Rietman and Frye also use genetic algorithms with neural network models for recipe generation. They use production data, not data generated from designed experiments, to train the neural network models.

There are also some examples of researchers using the same methodology, but with different methods. For example they might use neural network models, but with particle swarm optimization (PSO) for recipe generation. Or they may use a different type of neural network for the process model. In [23], Ahn *et al.* describe software they created that facilitates the creation of neural network models, and recipe generation using PSO with those models. In [24], Bay *et al.* apply the methodology to solar cell fabrication. They used both PSO and GAs. They made two models. The first model was for an amalgam of a texturing and drive in process. The results of that process are reflectance and sheet resistance, which are known to affect solar cell efficiency. They form the input variables

to the second model, whose output is efficiency. This is an example of using a cascade of neural networks, where the output of one neural network becomes the input to another. They found that PSO performed more consistently than GAs for their optimization task. This meant that there was less variance in the suggested values for the input variables, over multiple optimization runs, when using PSO as compared with GAs.

In the research mentioned so far, manipulatable process input variables were used as input to the model and some kind of downstream metrology was used as model output. However, online sensor data can also be used to characterize a process. This data tends to have a much higher dimensionality. It is usually preprocessed with a data dimensionality reduction method, like principal component analysis (PCA) or autoencoder neural networks (AENN), before being used for process modeling with neural networks.

In [25], Hong *et al.* modeled etch rate, uniformity, and anisotropy of a reactive ion etch (RIE) using optical emission spectroscopy (OES) data. OES-data dimension was reduced with PCA and AENNs, and then the smaller dimension dataset was used as neural network model input. The models using the AENN-reduced input data made slightly better predictions. One of the possible reasons for this is cited to be that AENNs do not discard any information during the data compression whereas PCA does. The AENN methods require more computation, however.

In [26], Triplett *et al.* made models where the output was the electron mobility at the “inverted interface” of a high electron mobility (HEMT) device created by molecular beam epitaxy (MBE). The HEMT device does not use silicon but uses InAs and AlSb. The inverted interface is the first interface created by MBE. They made two models, the first with growth conditions as model input and the second with reflection high energy electron diffraction (RHEED) data for input. They used PCA to reduce the dimensionality of the RHEED data before modeling, achieving a 100:6 reduction ratio. Both models performed well, achieving prediction accuracies of over 90%.

### 2.2.1.2 Control

Historically, control in semiconductor manufacturing consisted of statistical process control (SPC) [18]. Means and variances were calculated with historical fab data to define control limits. Then the Western Electric Rules were applied to more current data, collected from the processes, to generate an alarm for those that were out of control [6, 27]. SPC only draws attention to processes that may be out of control. It does not suggest corrective actions [6]. In the late 1980s and the 1990s the industry evolved the concept of advanced process control (APC) [6, 18]. The aim of APC was to use data to suggest corrective actions [6].

APC can be subdivided by function into fault detection, fault classification, fault prognosis, and process control [18]. APC can also be subdivided in the time-domain into real-time control, and run-to-run (R2R) control [19]. Real-time control requires *in situ* data, or data collected as the process is happening. This data is used to adjust manipulatable input variables as the process is happening. The collection of this data is dependent on the availability of sensors that can function in the prohibitive environments found in many semiconductor manufacturing processes [6]. Run-to-run control uses measurements made downstream of a process, to adjust the input variables for the next wafer of batch [19]. Neural network models have been used across these functional and time-domain subdivisions for APC.

**Run-to-run** In [28], Rietman and Patel developed artificial neural network model based controller for a plasma etch process. They used production data from about 6000 wafers from a MOS fabrication line to train the model. They did not remove any outliers in the data. The plasma etch consisted of three etch steps. The first step was a timed etch. The second had a different (slower) etch chemistry with higher selectivity and OES endpoint detection. The third etch step is a timed over-etch step. The third etch step is the one to be controlled. The established method was for an operator to determine etch time based on the etch time for the previous cassette. The model fused data from many additional data sources, for both the previous cassette and the current wafer up to the second etch step, to

make predictions for the ideal etch time for the third etch step. The use of historical data to determine etch time for the current wafer “closes the loop,” in contrast to traditional SPC. The controller was tested in a production environment and reduced the standard deviation of remaining oxide thickness after the third etch step by 40%.

In [29], Card *et al.* create a cascade neural network model of a plasma etch with 15 months of data from Digital Equipment Corp. The paper presents a feasibility analysis. They chose input variables for the model that were independent of each other, because the control algorithms might need to change each independent of the others. Model variables were categorized as continuous, and replacement or calibration. The outputs were etch rate, standard deviation of the etch rate across the wafer, and etch selectivity. Predictive ability was mixed, 73% and 76% for standard deviation and selectivity, and better than 87% for the rest. They present two optimization algorithms, one for least cost (LC) and one for least effort (LE). Least cost uses cost functions and the predictive model to find the lowest total cost solution. Least effort favors using inputs that have the most impact on achieving desirable output values, and uses as few as possible. They analyzed the optimization off line by comparing the optimization method’s suggestions with what was actually done at the fab. LC agreed more often with fab behavior, and LE highlighted that the model had learned certain relationships that had taken the fab team a while to utilize.

**Real-time** In [30], Davis and May use neural networks for the control of variable frequency microwave (VFM) curing of polymer dielectrics. They use a neural network for system identification, which is similar to the models used in the research mentioned earlier. But they also use a neural network to model an inverse of the “plant” or process. Plant modeling proceeds with the familiar factorial experiment to generate training data, then using the error back-propagation algorithm to train the model. They used the distal learning method [31] to train the inverse plant model. They implemented the controller and found it was able to regulate the temperature set point with about 7% error.

In [32], Stokes and May use neural networks for the control of reactive ion etching.

The control scheme is labeled indirect adaptive control. It is similar to the Davis and May paper [30], in that it uses an inverse model of the plant for the controller and uses the distal supervised learning approach for training it. They compared the neural controller to a more traditional LQG/LTR controller in software simulations and found that the neural controller performed faster, in terms of tracking and reacting to sudden changes in process parameters.

**Failure Detection and Diagnosis** Traditionally SPC has been used for failure detection, and the expertise of the fab engineers was used for diagnosis. The literature provides examples of how neural networks in conjunction with the theory of evidence can perform these same functions.

In [27], Hong and May perform real-time malfunction diagnosis for a plasma etch using neural networks and Dempster-Shafer (D-S) theory. They use OES and residual gas analysis (RGA) data for input to the model, and the input settings for the etch process as the output to the model. The dimensionality of the OES data is reduced with PCA before use with the model. They pick six atomic masses for the RGA data. The evidential reasoning works on the principal that if the neural-network-predicted input settings differ from the settings in reality, then there is evidence of malfunction. They compared diagnostic performance when using OES data only, RGA data only, and both OES and RGA data (sensor fusion). The method worked well for detecting and diagnosing faults in the RF system when using OES data alone. The method worked well for detecting and diagnosing faults in the gas flows when using RGA data alone. The method performed better when using both OES and RGA data than when using either separately.

In [33], Setia and May use a similar method (neural network models and D-S theory) as [27] for failure detection and diagnosis for via creation with excimer laser ablation. There was interesting data preprocessing for one of the neural network inputs. They performed a logarithmic transformation on this input and it improved the detection accuracy.

In [34], Setia and May use the same data as they did for the previous study, but this



time used an adaptive neuro fuzzy inference system (ANFIS) for detection and diagnosis. An ANFIS model uses neural network theory to parameterize a fuzzy system. The ANFIS model performed well, with 100% accuracy in detection and 90% in diagnosis (*versus* 95% for the neural network and D-S method).

**Virtual Metrology** There is a delay between when a process occurs and when its results are measured. This delay can be in the order of days with typical inline metrology. Virtual metrology (VM) uses process data that is measured *in situ* to make predictions about the wafer as soon as its processed [35]. Neural networks are a useful tool to create these predictive models for virtual metrology.

In [35], Lynn *et al.* compare the use of neural network models to that of multiple linear regression models for virtual metrology for an industrial plasma etch process. They do so with different variable selection techniques and data disaggregation methods. There were three variable selection techniques. These were PCA, correlation methodology, and stepwise selection. Correlation methodology compares the correlation input of each input and the powers of each input to the output. The most correlated variables are used as model input. Stepwise selection comprises both forward and backward selection. These work by adding a variable to a minimal model or removing a variable from an all inclusive model, based on an F-test of the improvement of adding or removing the respective variable. They found that the neural network model with inputs chosen by the correlation methodology performed best. The data disaggregation that attempted to address the effects of process drift and preventative maintenance did not help the modeling.

In [36], Lynn *et al.* use design of experiments (DOE) to generate data for neural networks and other modeling methods to make VM models of plasma electron density and etch rate for a plasma etch. They use a model-based predictive control scheme to control these two responses that are difficult to measure. The input for the model is taken from a plasma impedance monitor (PIM), that provides information about the power supply to the

plasma etch chamber. They also use an invasive “hairpin resonator” to collect plasma electron density response data during their designed experiments. Control is achieved using the VM models and predictive functional control (PFC). They found that the neural networks were the most accurate VM models with test data. They also found that they could perform setpoint tracking for the plasma electron density with time constants of less than a second and could perform setpoint tracking for etch rate to within 1% of the desired etch rate.

In [37], Lynn *et al.* use six months of industrial data to make VM models for a plasma etch. They compared three modeling methods, including neural networks, partial least squares (PLS) regression, and Gaussian process (GP) regression. Another level of comparison was global *versus* local modeling. The three local modeling schemes were regional, clusters, and windowed methods. PM cycles change chamber behavior, so local models might prove more accurate in certain “operating regimes.” Regional modeling meant dividing the data between PM events into regions, and making models using data from a particular region only. Regional modeling did not improve upon global modeling but added complexity. The cluster method meant using PCA on the PIM data to find clusters that could be labeled “operating regimes.” They used data from a cluster to make predictions when operating in that cluster. They could not conclude that it is better than global modeling. Windowed modeling meant using data from the 30 to 300 previously processed wafers to make predictions about the next wafer. They found that this method performed best, out of the local methods and global modeling, but that if the window became too small it was detrimental to the neural network model predictions.

### 2.2.1.3 Summary

The literature has shown that neural network models have been used with data from a variety of fabrication processes, and for a variety of manufacturing tasks. They have been used for process characterization, recipe generation, real-time process control, and virtual metrology.

However, the literature has not examined how the usefulness of neural network models

might change over the different stages of production for a semiconductor product. There is also a lack of granularity in the depiction of industrial fabrication environment and the data collected across the different environments. Furthermore, there is little investigation of how the research presented, that use neural networks on upstream processes, affect yield metrics. This thesis will attempt to address some of these issues, and do so in particular with regards to process characterization, process modeling, and recipe generation.

## **CHAPTER 3**

### **RESEARCH AND DEVELOPMENT FAB MODELS**

#### **3.1 Introduction**

Neural network models have been developed and tested for fabrication processes at an industrial research-and-development (R&D) fab. The models predict post-process metrology from process input variables. The models were built with data obtained from the databases already present at the industrial fabrication facilities, as opposed to from experiments designed to generate modeling data.

The operational objective of the fabrication environment determines the nature of the data available for modeling. To illustrate with an example, first note the trade-off between throughput and metrology. Fewer wafers pass through the production line if more time is spent examining each one. At the R&D stage, a greater value is placed on process understanding, thus a greater value on process metrology. This in turn is reflected in the amount of metrology data collected in an R&D fab. A manufacturing environment places more weight on throughput. Thus some forms of process metrology, like cross-section SEMs, are eliminated. Differences in the data available mean differences in the models that can be made and differences in how they can be best used.

There were three processes modeled. The first two processes create a hardmask that is used during the third process, a high-aspect-ratio etch. The combined effect of these three processes is to create an array of high-aspect-ratio cylindrical holes in an oxide layer, see Figure 3. They act like a mold for device structures created in subsequent fabrication processes. Thus, their dimensions are important. Of particular importance are the dimensions between these holes in the x-direction and y-direction. These are referred to as the x-direction CD and y-direction CD respectively. These particular processes were chosen because of the challenges they presented and because they were the most likely to benefit from optimization and tighter control (the high-aspect-ratio etch, in particular). An

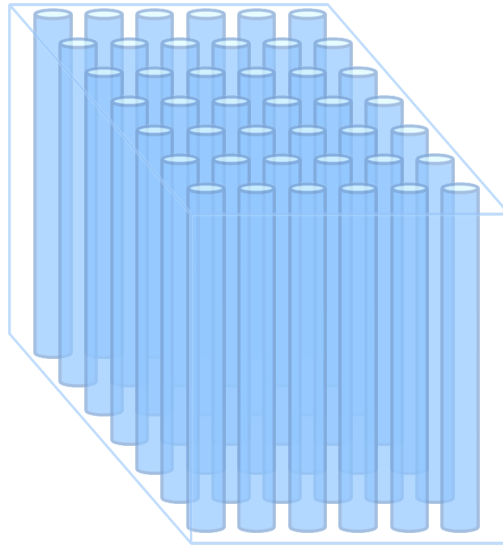


Figure 3: Schematic of an array of high aspect ratio holes in an oxide layer.

overview of the process flow appears in Fig. 4.

The hardmask is a layer of dielectric anti-reflective coating (DARC<sup>®</sup>) that is etched to reveal the openings through which the cylindrical holes are subsequently etched. The hardmask is created by first etching a set of parallel trenches in the DARC<sup>®</sup> layer that are not deep enough to reveal the underlying material. These trenches are then filled with a conformal layer of bottom anti-reflective coating (BARC). Then another set of parallel trenches are etched, but this time with the trenches running perpendicular to the first set of trenches. This etch, through the BARC and DARC<sup>®</sup> layers, continues deep enough to reveal the underlying material (an oxide layer) in the areas where the two sets of trenches overlap. These are the openings through which the cylindrical holes are etched. Figure 5 presents an overview of the creation of the hardmask. Figure 6 presents a top down view of the hardmask that results.

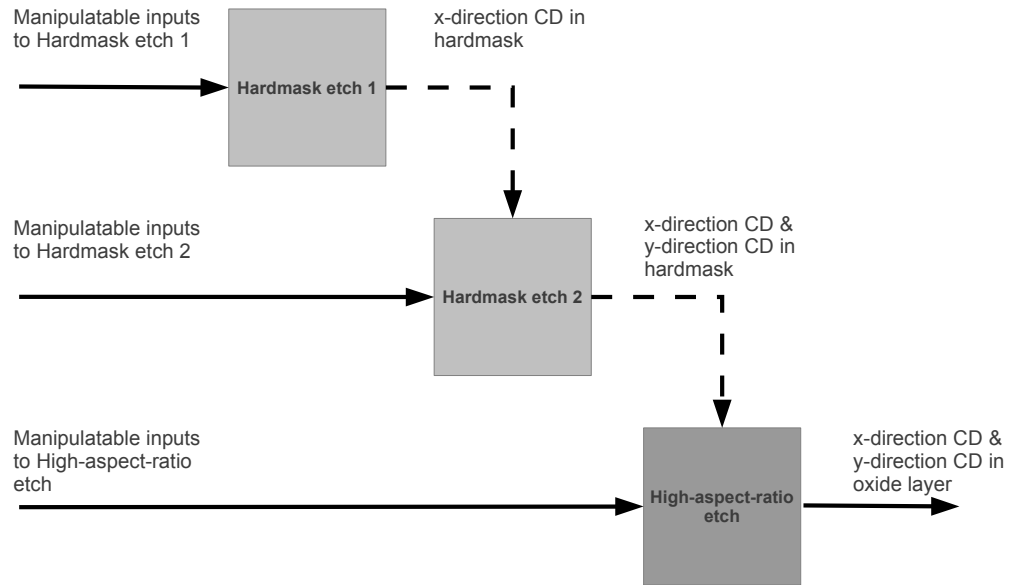


Figure 4: An overview of the process flow used to create the array of high-aspect-ratio holes in the oxide layer.

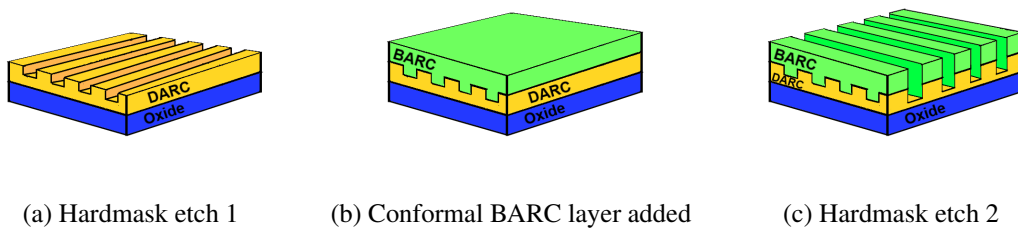


Figure 5: Overview of the creation of the hardmask

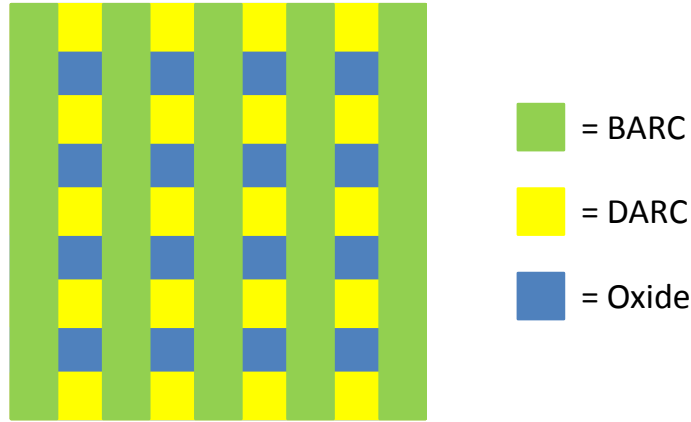


Figure 6: A top down view of the hardmask that is created.

### 3.2 Data collection

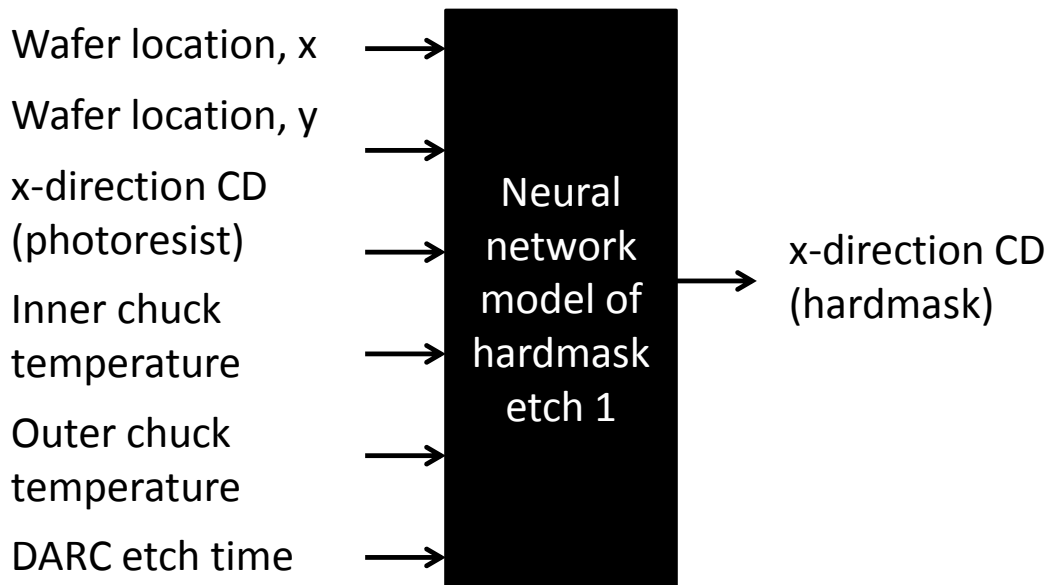
The data was gathered from a preexisting database. The data comes from experiments designed to understand the processes under investigation, but not specifically designed for modeling. Thus, it is possibly not the most thorough exploration of the input space, but the data contains information about process behavior over a larger subspace of the input space than in a manufacturing environment.

The majority of the data collected ( 90%) was used for training the neural network models. The remaining subset was used for testing. The test data is used to gauge the predictive abilities of the neural network models on datasets they have not been trained on. The resulting output (predicted output) is then compared to the real world data obtained by experiment (the output of the testing dataset). In this way, a judgment can be made about how well the models are performing [17].

### 3.3 Processes modeled

#### 3.3.1 Hardmask etch 1

Hardmask etch 1 creates the first set of trenches. Its purpose is to define the distance between cylindrical holes in the x-direction. Thus, the x-direction CD forms the output of the model (Fig.7). Wafer location was known to have an effect on the x-direction CD, and thus it is included among the model inputs (Wafer X and Wafer Y). Additionally, adding wafer location as an input allows the models to make predictions across the wafer, allowing the user to judge the model's ability to predict uniformity. The x-direction CD in the photoresist, the inner and outer chuck temperatures, and the DARC® etch time were included as inputs. Inputs were chosen based on their availability in the database, and on the expertise of the fab engineers as those known to affect x-direction CD in the hard mask.



1

Figure 7: Overview of model for hardmask etch 1.



### 3.3.2 Hardmask etch 2

Hardmask etch 2 creates the second set of trenches, running perpendicular to the first set. Its purpose is to define the distance between the cylindrical holes in the y-direction. Thus y-direction CD forms part of the model output (Fig. 8). The etch is not perfectly anisotropic and thus has an effect on the x- direction CD, which is also part of the output. The inputs were chosen because they all affect the output.

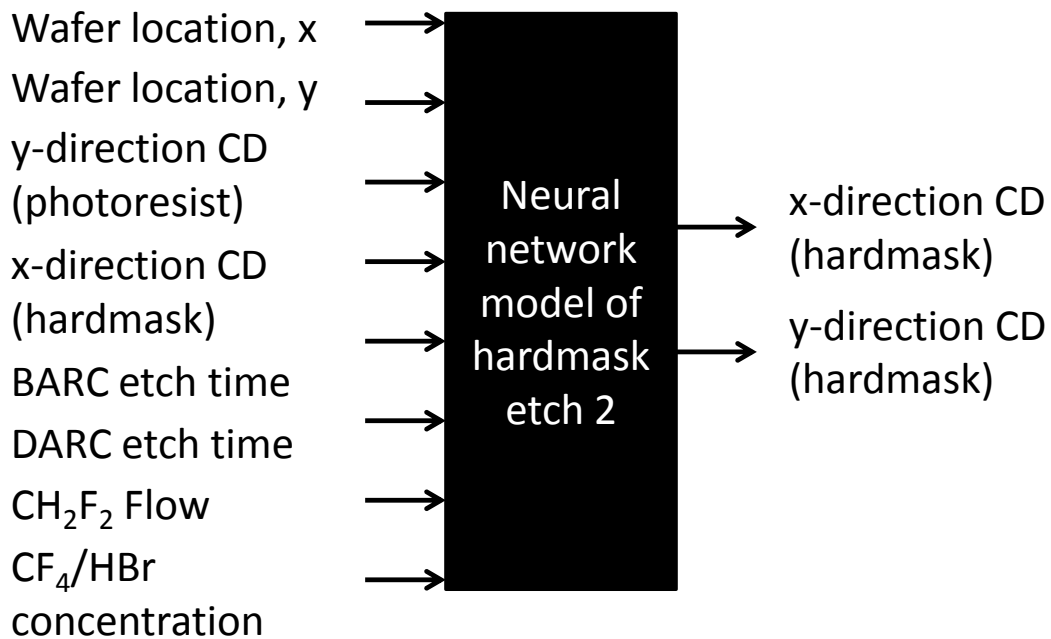


Figure 8: Overview of model for hardmask etch 2.

### 3.3.3 High-aspect-ratio etch

The overall etch is made of three etch steps, each with its own gas flow and etch time. These, together with two measurements of the critical dimension in the masking layer and

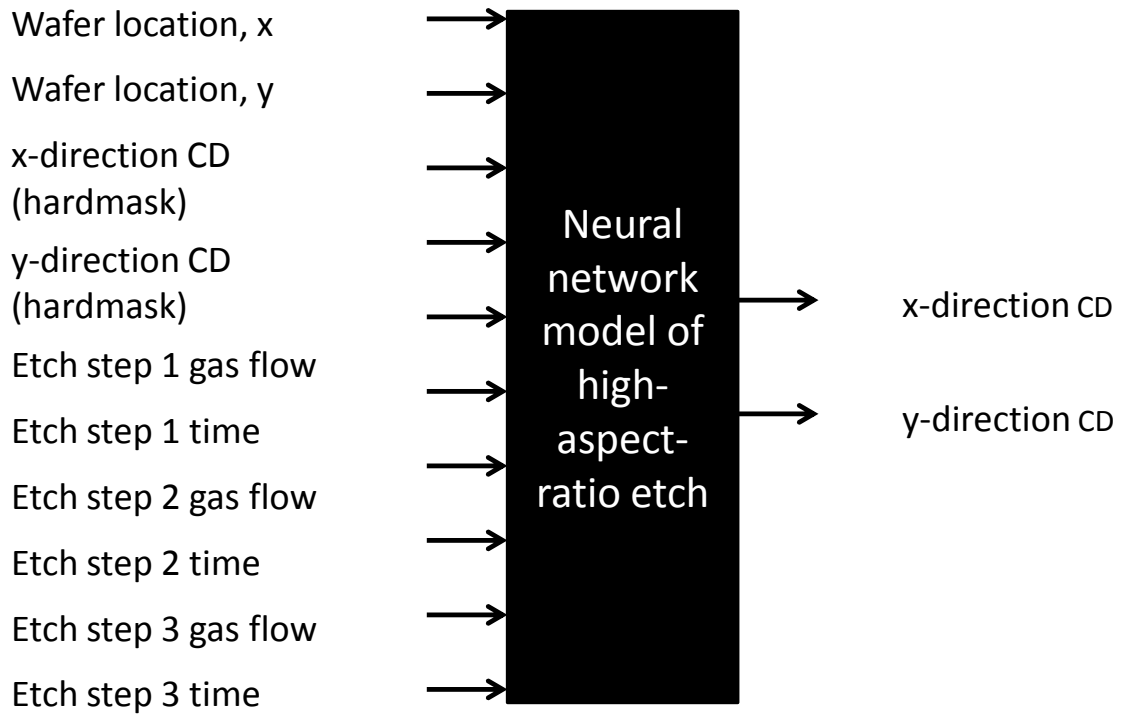


Figure 9: Overview of the model for the high-aspect-ratio etch.

the wafer location of these measurements, form the model input. The outputs are the x-direction and y-direction CD between the cylindrical holes. An overview of the high-aspect-ratio etch is presented in Fig. 9.

### 3.4 Results

#### 3.4.1 Hardmask etch 1

The model for the first hard mask etch was created with a 6-6-1 network. There were 368 data points in the training dataset and 30 in the testing dataset. The model was trained with the error back-propagation algorithm, with a learning rate of 0.001 and a momentum of 0. Modeling results are presented graphically as contour plots. This allows comparison of both the values of the output (x-direction CD in the hardmask) and its variation across the wafer (Figure 10). CD was measured at 15 wafer locations. The model was used to try

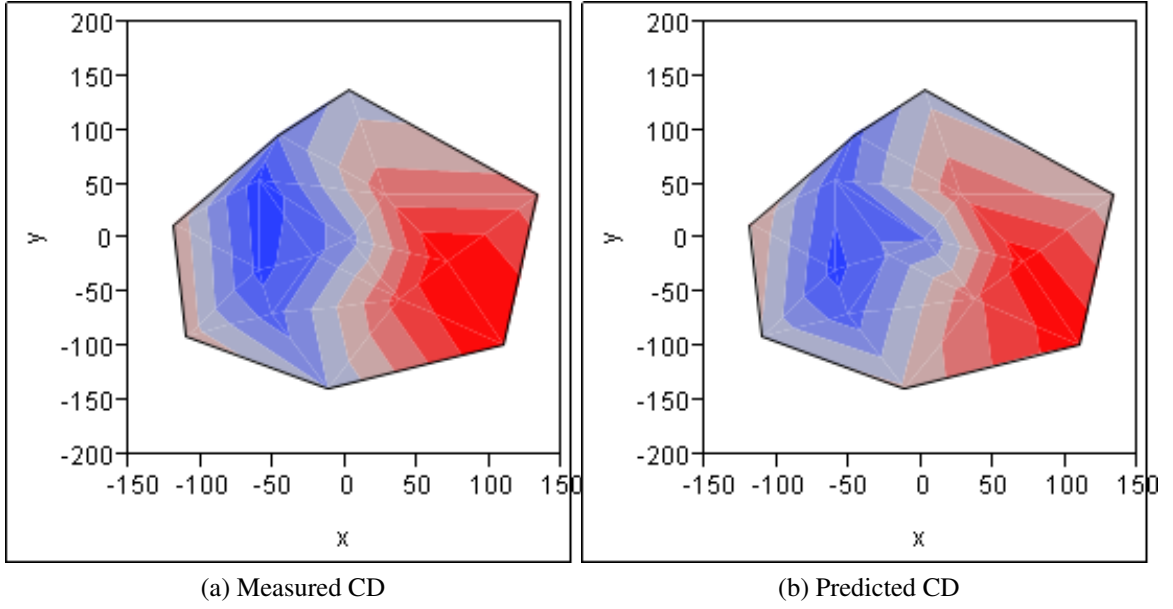


Figure 10: Comparison of measured (left) and predicted (right) CD (CDs were normalized). The x-axis and y-axis represent wafer location in mm.

and predict CDs at these same 15 wafer locations. The contour plots are created using the normalized CD values at the 15 wafer locations and by interpolating the normalized CD values between these 15 wafer locations. There is no extrapolation of CD values to the wafer edges, and for this reason the results presented are not circular like the wafer. The results presented are the average values over two wafers. This helps to mitigate the effect of the occasional outlier from the experimental data.

The model has accurately captured the input-output relationship for Hardmask Etch 1, as seen from the similarity between the output measured from experiment (left) and the output predicted by the model (right). This is further illustrated by a contour plot of the percentage error, which quantifies the differences between measured and predicted values of the output at different places on the wafer. Figure 11 presents the contour plot for absolute percentage error for the neural network predictions. It is calculated for each wafer location as

$$M = \frac{1}{n} \sum_{i=1}^n \left\| \frac{A_i - P_i}{A_i} \right\| * 100\%, \quad (1)$$

where  $A_i$  is the actual or measured value,  $P_i$  is the model predicted value, and  $n$  is the

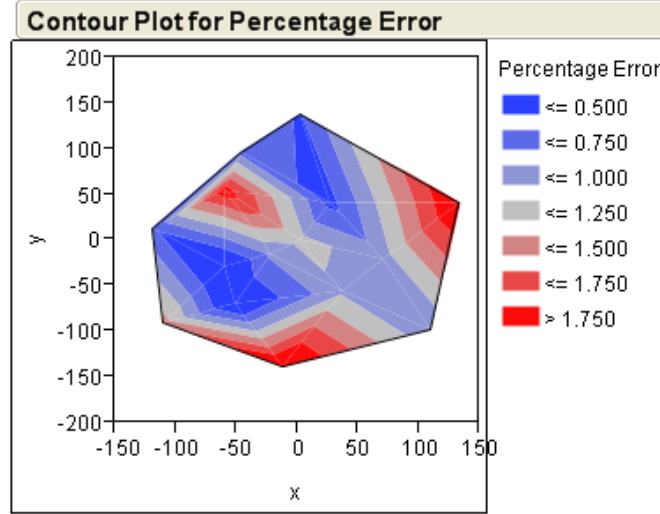


Figure 11: The absolute percentage error for the neural network predictions. The legend on the right presents the different levels of mean absolute percentage error.

number of observations for that wafer location in the data set. As can be seen in Figure 11, error remained below 2% across most of the wafer.

### 3.4.2 Hardmask etch 2

The model for the second hard mask etch was created with a 8-8-2 network. There were 403 data points in the training dataset and 48 in the testing dataset. The model was trained with the error back-propagation algorithm, with a learning rate of 0.001 and a momentum of 0. Results are again presented as contour plots and are the average results of four wafers. Because there are two outputs to Hardmask Etch 2, two sets of results are presented. The model has captured the variation of the x-direction CD across the wafer (high in the middle low at the edges), as seen in Figure 12. There is a deviation between measured and predicted values for the critical dimension at the bottom of the wafer. This is also reflected in the error plot shown in Figure 13.

The results for the y-direction critical dimension are presented in Figure 14. The error remained low, with slight deviations in the center and at the bottom of the wafer, as seen in Figure 15.

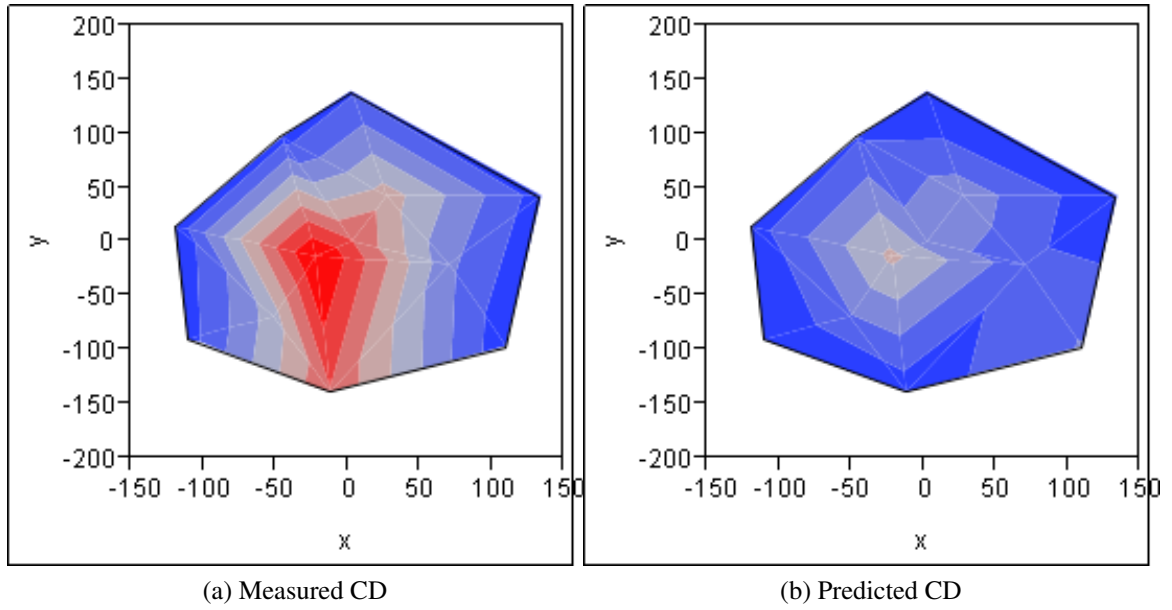


Figure 12: Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask (CDs were normalized). The x-axis and y-axis represent wafer location in mm.

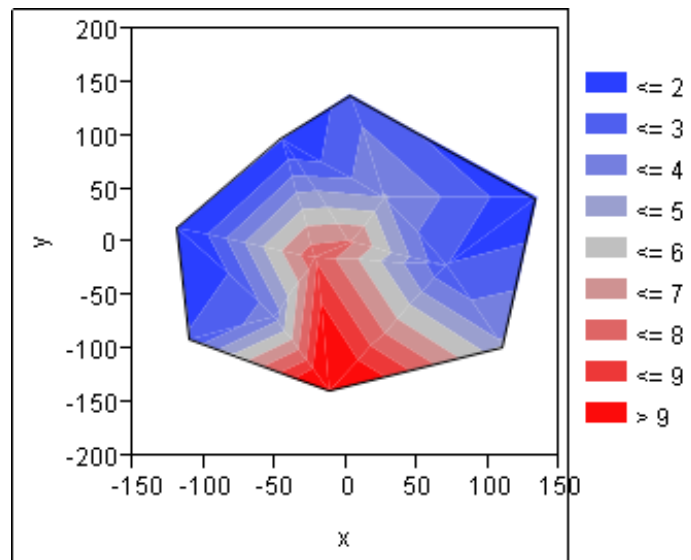


Figure 13: Absolute percentage error for neural network predictions.

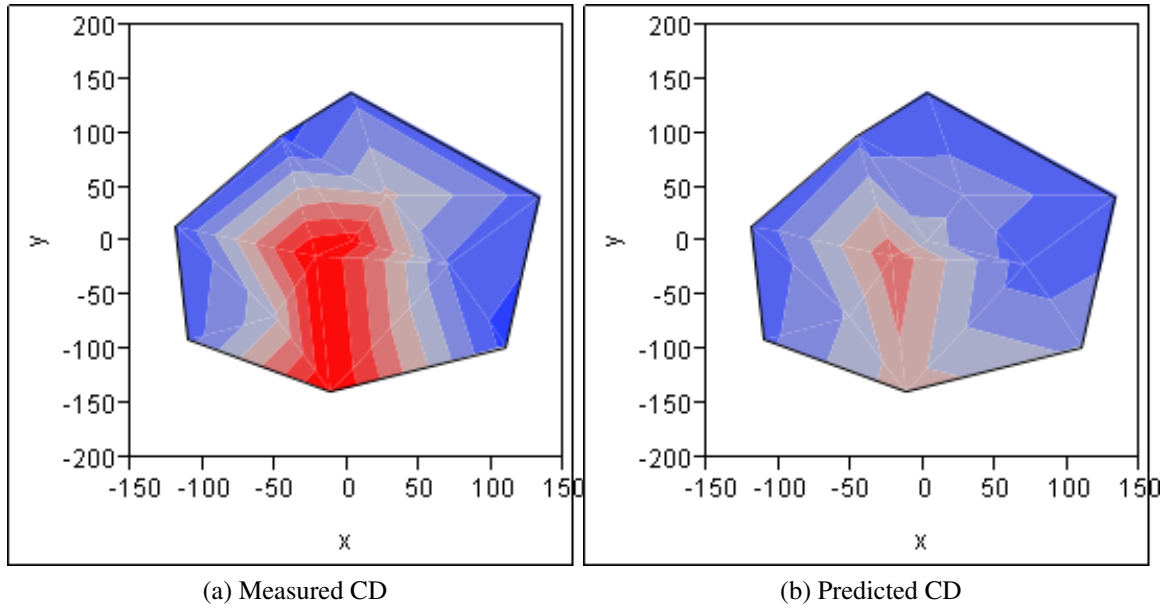


Figure 14: Comparison of measured (left) and predicted (right) output for y-direction CD in the hardmask (CDs were normalized). The x-axis and y-axis represent wafer location in mm.

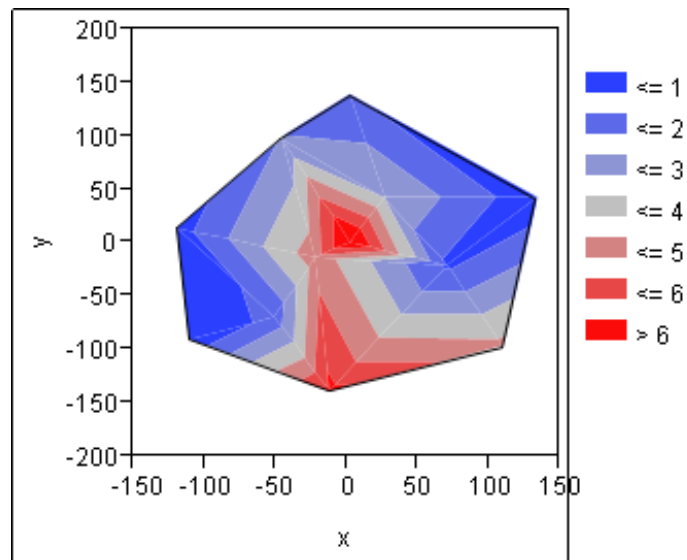


Figure 15: Absolute percentage error for neural network predictions.

### 3.4.3 High-aspect-ratio etch

The model for the high aspect ratio etch was created with a 10-10-2 network. There were 639 data points in the training dataset and 45 in the testing dataset. The model was trained with the error back-propagation algorithm with a learning rate of 0.001 and a momentum of 0. These results presented are the average results of two wafers. Two sets of results are presented below because there are two outputs to the high-aspect-ratio etch. The first (Figure 16) is for the x-direction critical dimension between holes. The model accurately captures the variation. The shape of the graph is slightly different from previous results because of missing or incomplete data in the database (*i.e.*, there were data points missing for the lower left portion of the wafer). The images for x-direction CD were created using eleven data points on each wafer. The percentage error is presented in Figure 17.

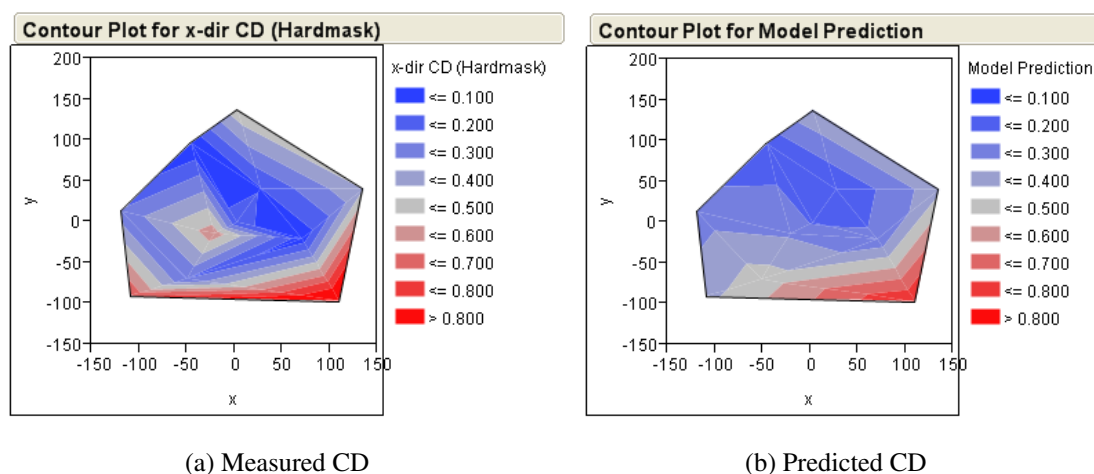


Figure 16: Comparison of measured (left) and predicted (right) output for x-direction CD in the oxide layer (CDs were normalized). The x-axis and y-axis represent wafer location in mm.

The results for the y-direction critical dimension are presented in Figure 18. The images were created with ten data points on each wafer. There is correlation between measured and predicted CD uniformity, particularly in the top part of the wafer. The error remained low, with the largest deviation coming near the right edge of the wafer (see Figure 19).

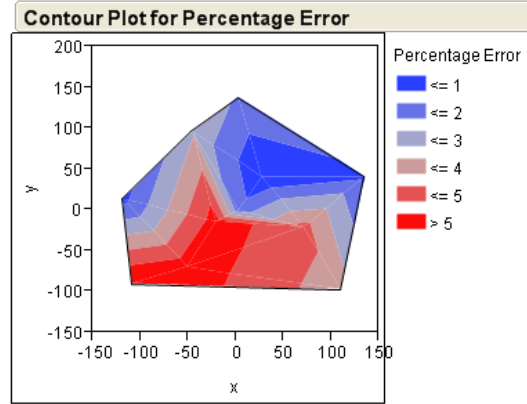


Figure 17: Absolute percentage error for neural network predictions.

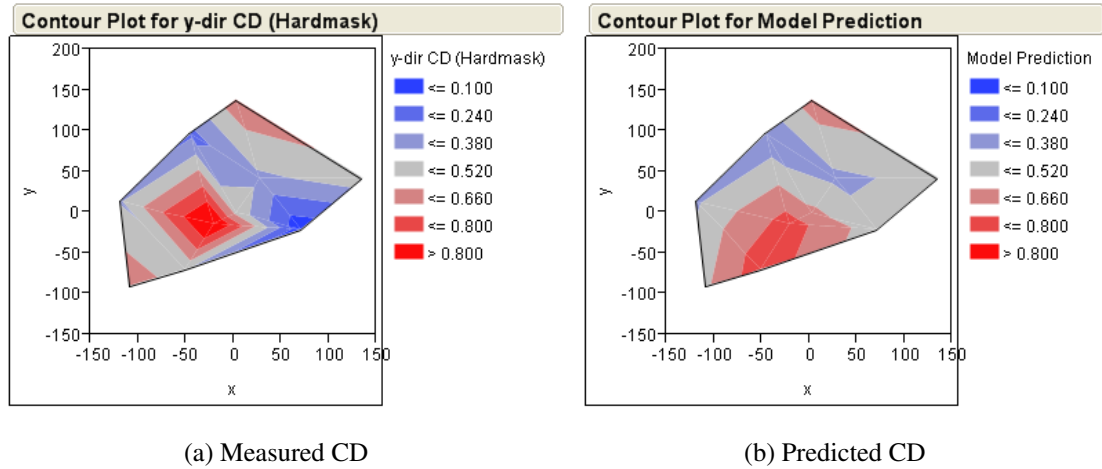


Figure 18: Comparison of measured (left) and predicted (right) output for y-direction CD in the oxide layer (CDs were normalized). The x-axis and y-axis represent wafer location in mm.

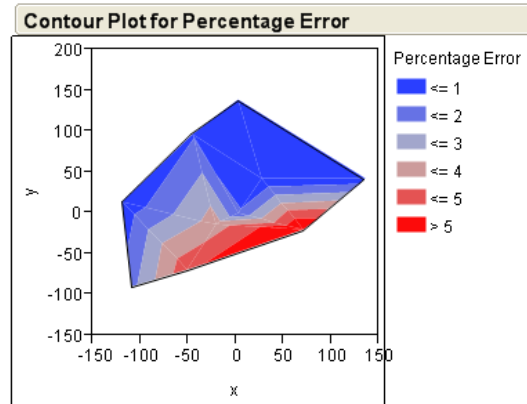


Figure 19: Absolute percentage error for neural network predictions.



### 3.5 Sensitivity analysis

A neural network model that has accurately captured the input-output relationship of a process can be used to see how the output of a process responds to changes in input at various points in the input space. One way of studying these responses is sensitivity analysis, in which a small change is made in one of the input variables. A small change would be 10% (or less) of the total variation of that input variable in the input space, for example. All other input variables are kept constant. The modified input vector and the unmodified input vector are passed through the network, and their outputs compared. The ratio of the percentage difference of the outputs to the percentage difference in the modified input variable is defined as the sensitivity [38].

The analysis can be repeated for all input variables in a process step, allowing their relative effects on the output to be studied. This analysis is not ideal because of the nonlinearities of semiconductor manufacturing processes, but can still provide some insight [11]. The relationship between the input variables and the output is typically nonlinear. In addition, there are typically interaction effects between the input variables. These are not accounted for in the sensitivity analysis. However, when examining a specific point in the input space, a particular process recipe for example, a sensitivity analysis can provide an insight into the process in a local region around the point of interest. An example might be an instance where minor adjustments are needed for an established process recipe.

A sensitivity analysis was performed on each of the processes modeled. A representative example would be to look at the relative effects the inputs of Hardmask Etch 2 have on the x-direction critical dimension in the hard mask (see Fig. 20).

### 3.6 Summary

The neural network models were proven to be accurate and flexible in this study. They were able to accurately predict post-process metrology of fabrication processes in an R&D environment, using data that preexisted in the fab database. Thus this study highlights

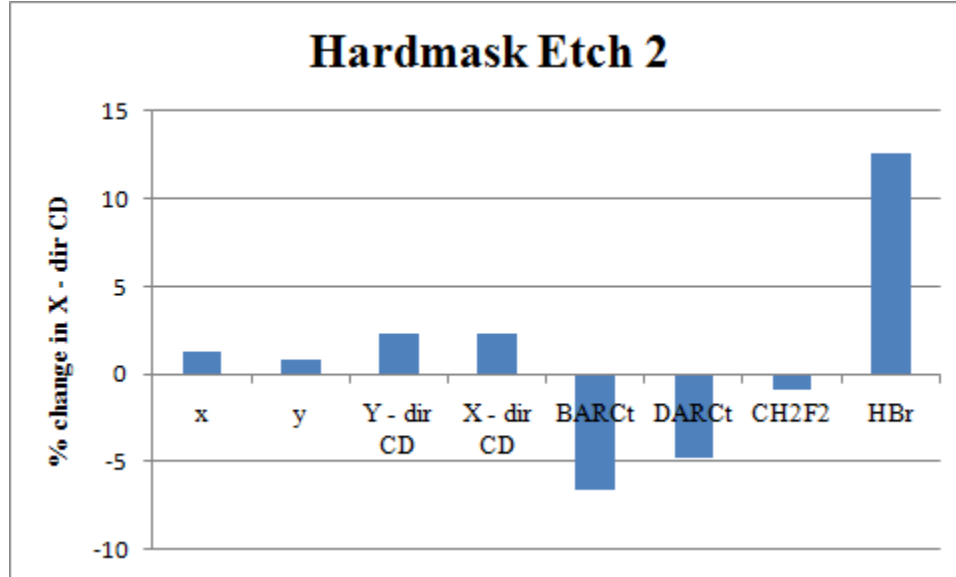


Figure 20: Sensitivity analysis for hard mask etch 2. Results are for x-direction CD in the hard mask.

how neural networks can be used to extract more value from this large, complicated, but underutilized resource that is the fab database.

However, neural networks were not as successful in predicting yield metrics from fabrication process input variables, in such an environment. Yield can be quite low, 40-50% die yield for example, when a product is transitioned from an R&D environment to a manufacturing one. When yields are at these levels and below, there is less clarity in identifying or assigning upstream fabrication processes as causes for sub-optimal yield metrics. This is because there are fabrication errors at too many levels of the product as illustrated in Figure 21. Each blue square in Figure 21 represents a fabrication process. Each row is a lot that undergoes the same set of fabrication processes. It is difficult to find lots where there is a clear causal link between the fabrication process to be modeled and the yield metric. Thus using the input variables from one fabrication process in a model does not provide enough information to make predictions for yield metrics.

Chapter 5 presents the exploration and evaluation of the use of the neural network modeling method in a manufacturing fabrication environment, and highlights some contrasts with its use in the R&D fabrication environment. The focus was on a high-aspect-ratio

# Data for yield models during development

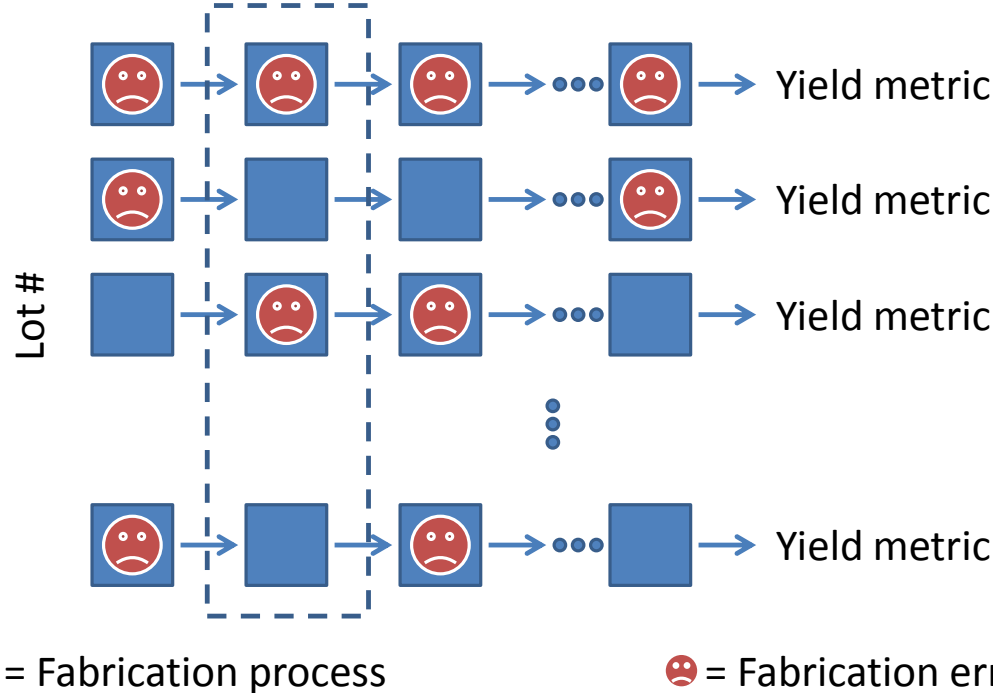


Figure 21: A schematic of the data available for model building in a research and development fabrication environment. The fabrication process to be modeled is in the dashed blue rectangle.

plasma etch process.

## CHAPTER 4

### RAMPING STAGE MODELS

#### 4.1 Introduction

As illustrated in Figure 1, manufacturers address their two distinct organizational objectives of product/process innovation and manufacturing efficiency by having a fab for each. Because the product/process development must happen first, there must be a period of transition whereby knowledge and production move from the R&D fab to manufacturing. The complexities inherent in semiconductor manufacturing and the precision it requires mean that this is not as trivial as handing over process recipes.

In the context of this work, this period of transition when production has begun to happen in the manufacturing fab but volumes and yields have not reached their plateau is referred to as the ramping stage. The time at which the stage starts and the steepness of the ramp are very important to a manufacturers profitability, given the rate of price decay for semiconductor products [5].

The distinct nature of this phase of the production process is observable in the fab database. Given that product is being manufactured, there are larger volumes of data than available in the R&D fab. Failure rates are also decreasing to the point where failures may be ascribable to upstream processes. Also variation is greater in the datasets, in terms of recipes used and process outcomes, than for more mature products/processes. This middle ground between development and manufacture, in terms of size and variation of fab datasets, could prove fruitful for model building and recipe optimization.

This chapter will test this theory by building neural network models of the high-aspect-ratio plasma etch process. First the model inputs were chosen using a variety of feature selection algorithms. Models were built with the different datasets and their prediction accuracies compared. Then using the most accurate model, an optimized process recipe was found using a genetic algorithm.

## 4.2 Method

### 4.2.1 Feature selection

The fab database contains data for several hundred features (variables) pertaining to any fabrication process. A subset of these is relevant for modeling. The process of finding this subset is called feature selection. The assumption is that the total dataset contains redundant, or irrelevant variables. Removing these from the dataset, results in a simpler model that can be parameterized using fewer observations.

Feature selection algorithms can be roughly divided into three categories. These are wrapper, filter, and embedded methods [39, 40, 41]. Wrapper methods ‘wrap’ a modeling algorithm with the search for a feature subset. A model is built for many different feature subsets. Subsequently the subset whose associated model performs best among these is chosen. This process is computationally intensive as a model needs to be created and trained for each feature subset.

Filter methods calculate a score for each feature in the dataset. Subsequently, the highest scoring features - the top ten percentile for example - are chosen as inputs to the model. The correlation-coefficient between a feature and the output variable is an example of a scoring function. Filter methods cannot account for the model that will subsequently be created, or feature interactions, or redundancies when picking a subset [40]. They are, however, less computationally intensive than wrapper methods.

Embedded methods are those where feature selection is embedded in the process of creating a model. This is not the case with neural networks. A combination of the filter and wrapper methods were used to find the input variables for the models in this chapter.

#### 4.2.1.1 *Correlation based filter method, followed by SVM modeling*

A filter method with correlation-coefficient scoring function was used to create a feature subset of a specified number of features. A support-vector-machine (SVM) model was then built with this subset. This process is repeated for several feature subsets containing different numbers of features. The reduced dataset associated with the model with the

minimum prediction errors was chosen for further analysis. Pseudocode to represent this process is presented below.

#### Feature Selection Pseudocode

```
# Start with entire feature set.
entire_feature_set = import(data_from_database)

# Build a model for a range of feature subsets with
# different numbers of features.
for num_features in range(1,50):
    reduced_feature_set = filterMethod(entire_feature_set)
    model = buildModel(reduced_feature_set)
    error = error.append(model.meanSquareError())

# Choose the feature subset that resulted in the
# model with minimum prediction error.
size_of_best_feature_subset = returnMinIndex(error)
```

A plot of model prediction error *versus* the number of features in the model is presented in Figure 22. Prediction error drops rapidly from a model having one input to having ten. Prediction error remained relatively flat beyond ten features, with the minimum occurring at 26 features.

The dataset can possibly be pared down even further. Looking at a plot of the correlation matrix for the input variables, Figure 23, one can see that there are some variables are highly correlated and therefore perhaps do not contain any extra information.

#### 4.2.1.2 Stepwise regression

Stepwise regression is a wrapper method that incrementally adds or removes variables from a linear regression model, based on a performance measurement of the resulting models. It starts by making single variable regression models for all possible input variables in the dataset. After picking the best of this initial set, variables are added or removed incrementally from the model, based on whether their addition or removal improves the Akaike

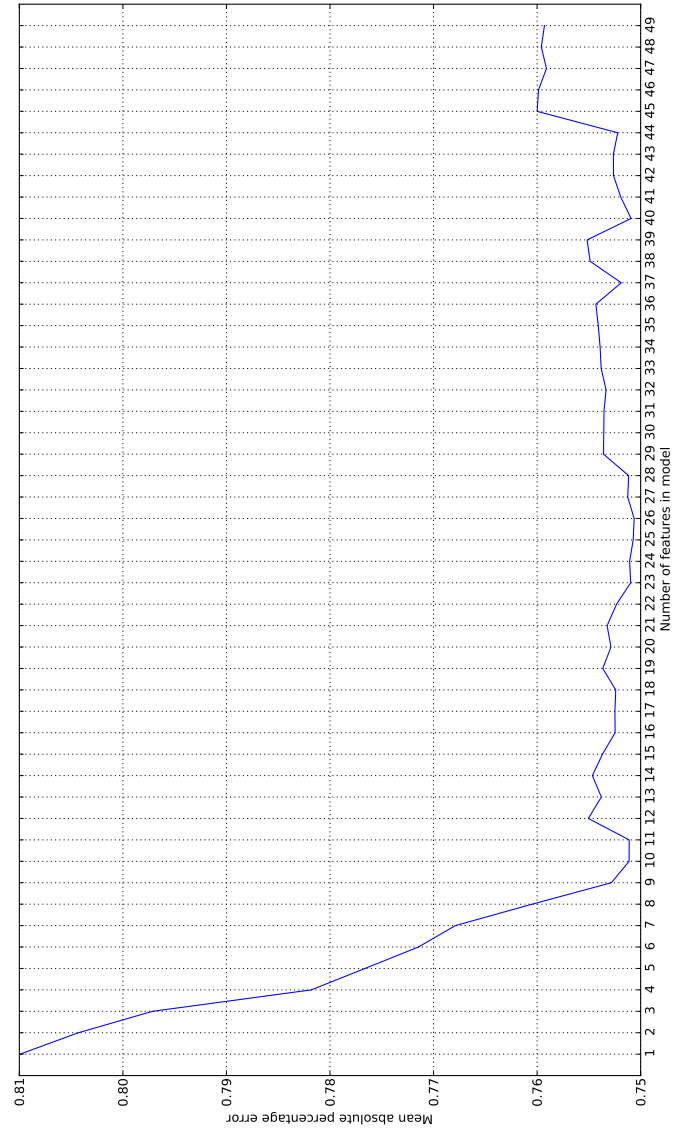


Figure 22: Mean of the absolute percentage error for models with different numbers of input variables.



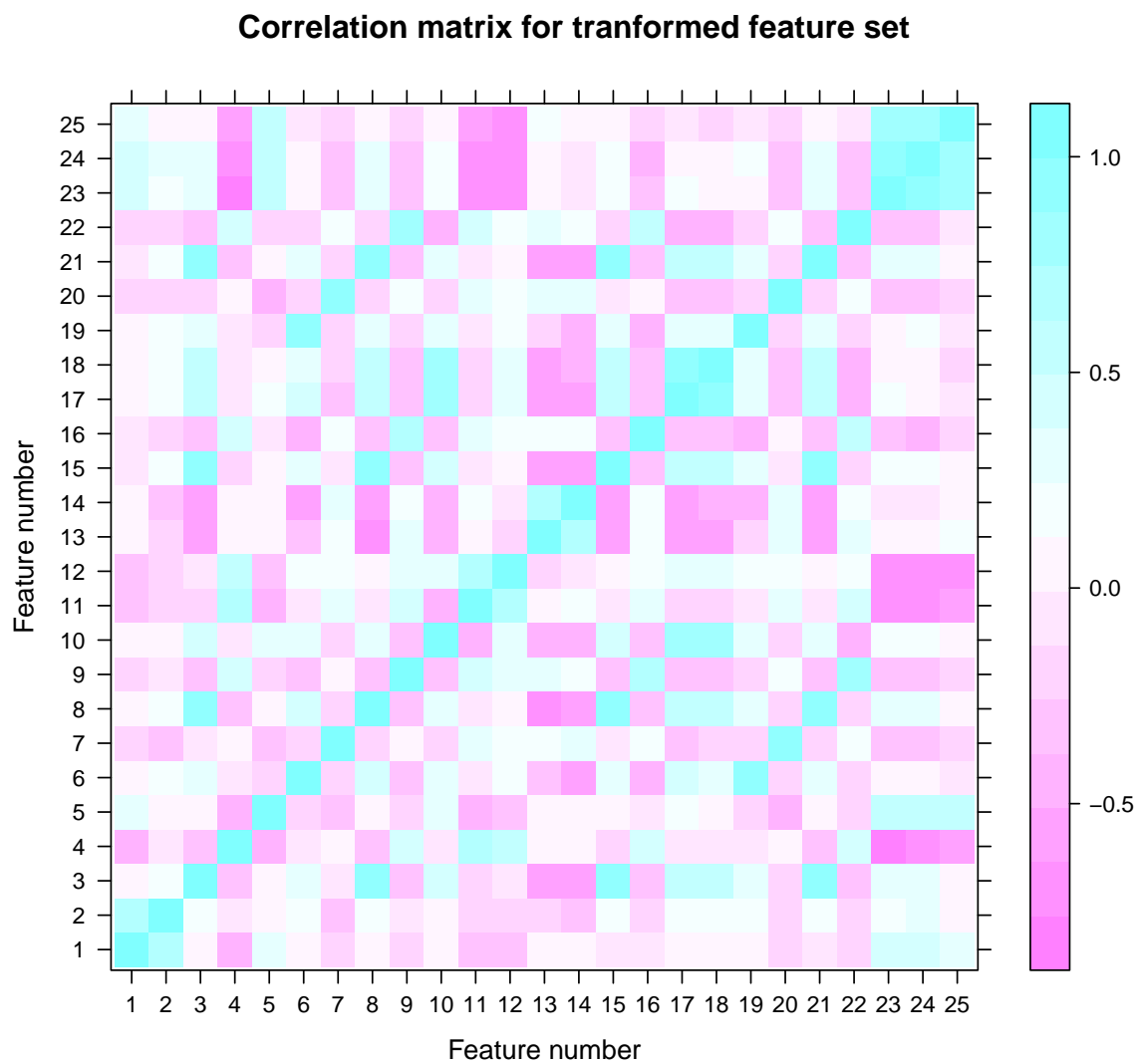


Figure 23: A plot of the correlation matrix for the reduced feature set.

Information Criterion (AIC) [42]. The process stops when the addition or removal of a variable makes a negligible difference to the AIC. The formula for AIC is

$$AIC = N * \ln\left(\frac{SS_{error}}{N}\right) + 2K \quad (2)$$

where  $N$  is the number of observations, and  $K$  is one more than the number of parameters fit.

Neural network models created and trained with a variable subset chosen by stepwise regression performed more accurately than those created and trained with variable subsets chosen by the other feature selection algorithms used in this study. Stepwise regression also performed well in [35], and as noted there, the variables in the resulting feature subset are less correlated than those chosen with a correlation filter method. See Figure 24 and compare to Figure 23.

#### 4.2.1.3 Minimum redundancy maximum relevance (mRMR)

Minimum redundancy maximum relevance (mRMR), is an information theoretic approach to feature selection developed in [43, 44]. It uses a measure of the information shared between two random variables known as mutual information:

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \quad (3)$$

where  $I(X; Y)$  is the mutual information for  $X$  and  $Y$ ,  $p(x, y)$  is the joint probability distribution, and  $p(x)$  and  $p(y)$  are the marginal probability distributions. The advantage of using mutual information is that no assumption is made about the nature of the dependence between the variables, as is the case with correlation [40].

Features that share a lot of information with the output variable are relevant. However, if they share a lot of information with other relevant variables, then they are redundant. The algorithms searches for the feature subset where the average mutual information between individual features and the output variable is maximized, and the average mutual information between individual features and other features is minimized.

**Correlation matrix for stepwise-regression reduced feature set**

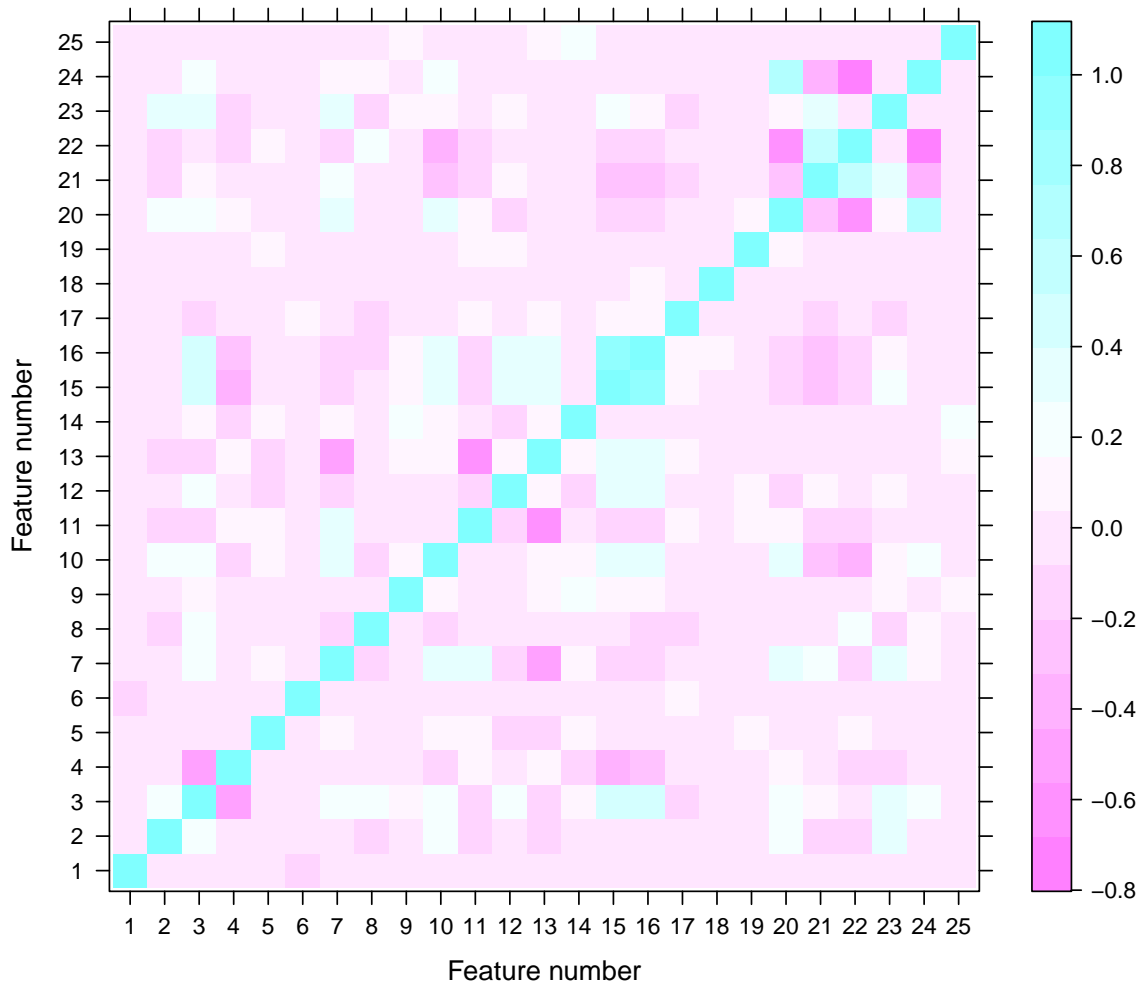


Figure 24: A plot of the correlation matrix for the stepwise-regression reduced feature set. The algorithm selected 68 variables, but the first 25 are shown to allow comparison to Figure 23

### 4.2.2 Modeling

Neural network models were created using MATLAB's Neural Network toolbox. The datasets supplied by the feature selection algorithms contained differing numbers of input variables. Hence, the input layers of the models created for each were of different sizes. Models with varying numbers of hidden layer neurons were created and trained with the Levenberg-Marquardt algorithm, as it converges quickly. The performance of the different models, as measured by RMSE, was then compared as in Figure 25.

The model with the best performance was then retrained with Bayesian regularization as the training algorithm. This resulted in models with better fit, as seen in Figure 26. The errors for the model trained with Bayesian regularization were more concentrated around zero, although the algorithm typically needs more time to converge.

### 4.2.3 Genetic algorithm for optimized process recipe

A genetic algorithm was used to search for an input recipe that produced the desired output in the model. Assuming the model is an accurate reflection of reality, this input recipe should result in the desired output when applied to the actual fabrication process.

If one thinks of the neural network as a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  where  $n$  is the dimension of the input space, then the genetic algorithm can be thought of as a search for the vector  $\mathbf{X}$  that minimizes a predefined fitness function:

$$fitness(\mathbf{X}) = (d - f(\mathbf{X}))^2. \quad (4)$$

This is a measure of the distance between network output  $f(\mathbf{X})$  and the desired output  $d$ . The MATLAB Global Optimization toolbox was used to implement the genetic algorithm.

## 4.3 Results

### 4.3.1 Correlation based filter method, followed by SVM modeling

Figure 27 presents graphs of neural network model output against the measured data for the model trained with features chosen by the correlation + SVM feature selection algorithm.

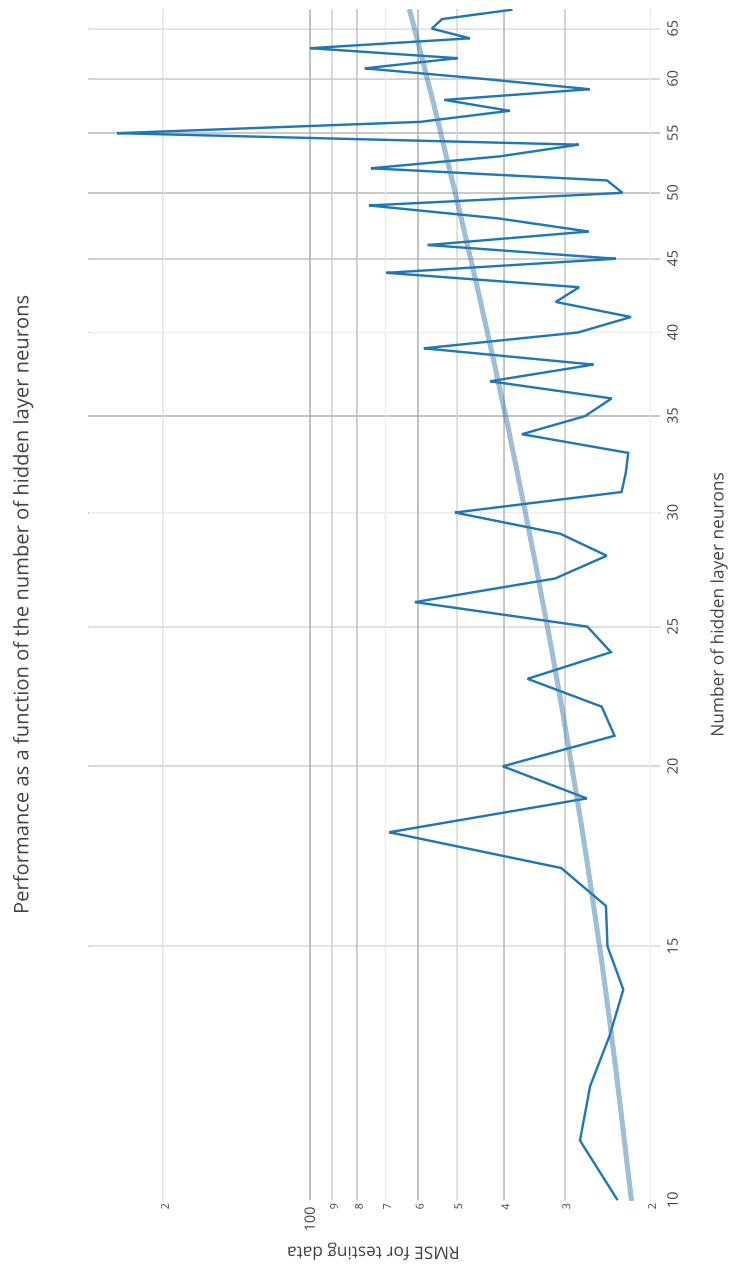


Figure 25: A plot of the number of neurons in the hidden layer *versus* the RMSE for respective model. Graph is for the dataset chosen by stepwise regression. The testing RMSE trends upward with the addition of hidden layer neurons.

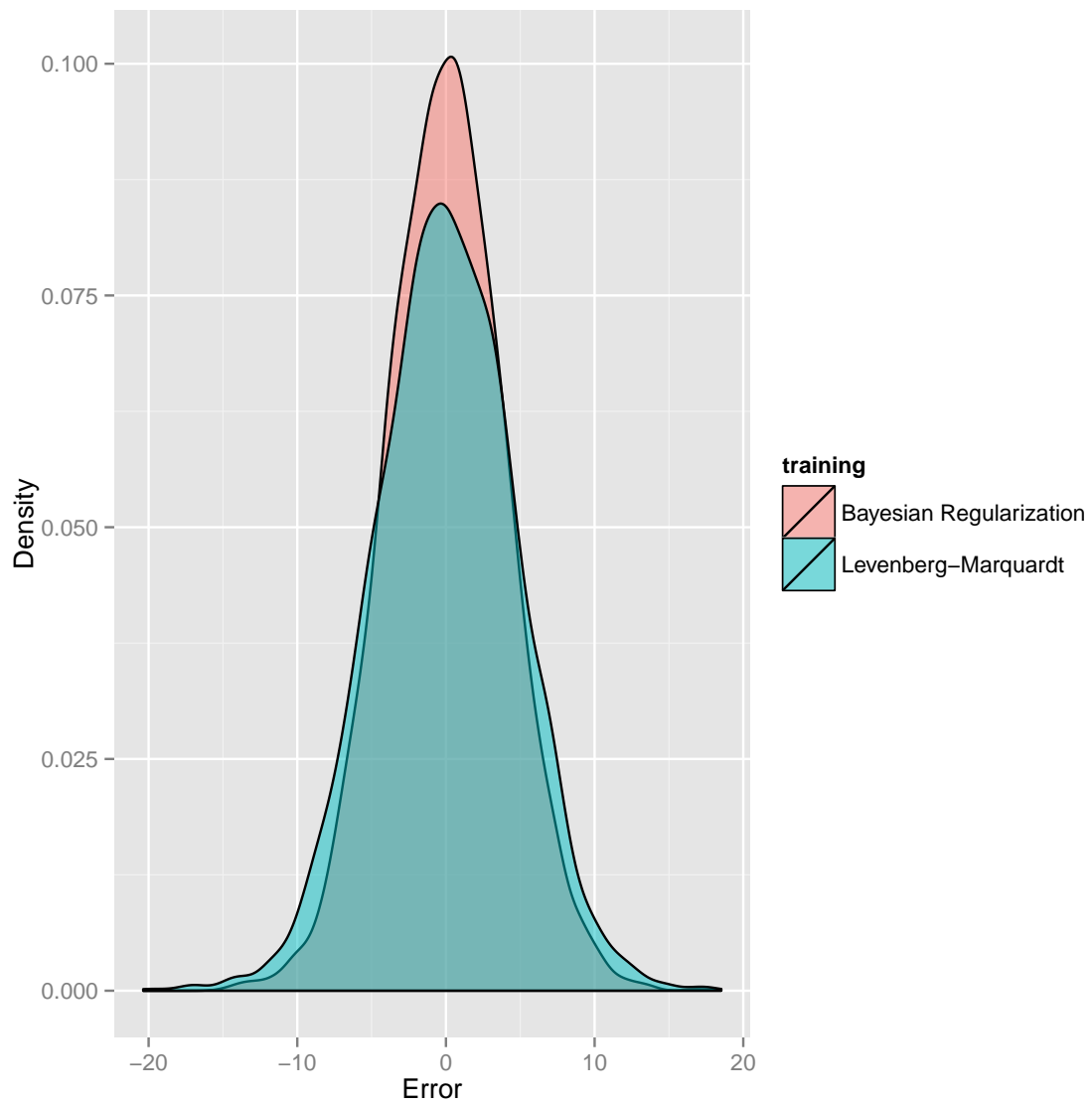


Figure 26: Density plots of the prediction errors of the same model, trained with different training algorithms.

There is a plot for the training data, validation data, testing data, and the combination of the three.

The graph for training data shows how close the model could get to the relationship in the data presented to it. The R value of 0.64148 above the graph represents the sample correlation coefficient between measured and predicted values.

The validation data set is not used to train the models (adjust model parameters), but is used to stop the training process. The model prediction error for the validation dataset is calculated after every training epoch. This error typically decreases early on as the training algorithm adjusts model parameters to reflect the relationship in the training data. It reaches a minimum then begins to increase as the algorithm starts to overfit the training data. Model parameters are set to be the values where the validation error was at its minimum. This is illustrated in Figure 28. The green line shows the validation error as the training progresses.

The graph for the testing data provides information about how generalizable the model is. The correlation coefficient between model predictions and measurements is 0.54567 for testing data. Figure 29 shows a histogram for the prediction errors of the network. The errors are centered around zero. The mean absolute percentage error (MAPE) was 0.6534%.

#### **4.3.2 Stepwise regression**

The graphs in Figure 30 plot the output of the neural network model against the target (measured) data, for the model trained with features chosen by the stepwise regression feature selection algorithm. There is a plot for the training data, testing data, and the combination of the two.

The sample correlation coefficient of 0.72595 for the training data is higher than it is for the dataset chosen with the correlation + SVR method. The sample correlation coefficient of 0.5571 for the testing data is also higher than it is for the dataset chosen with the correlation + SVR method.

Figure 31 shows a histogram for the prediction errors of the network. The errors are

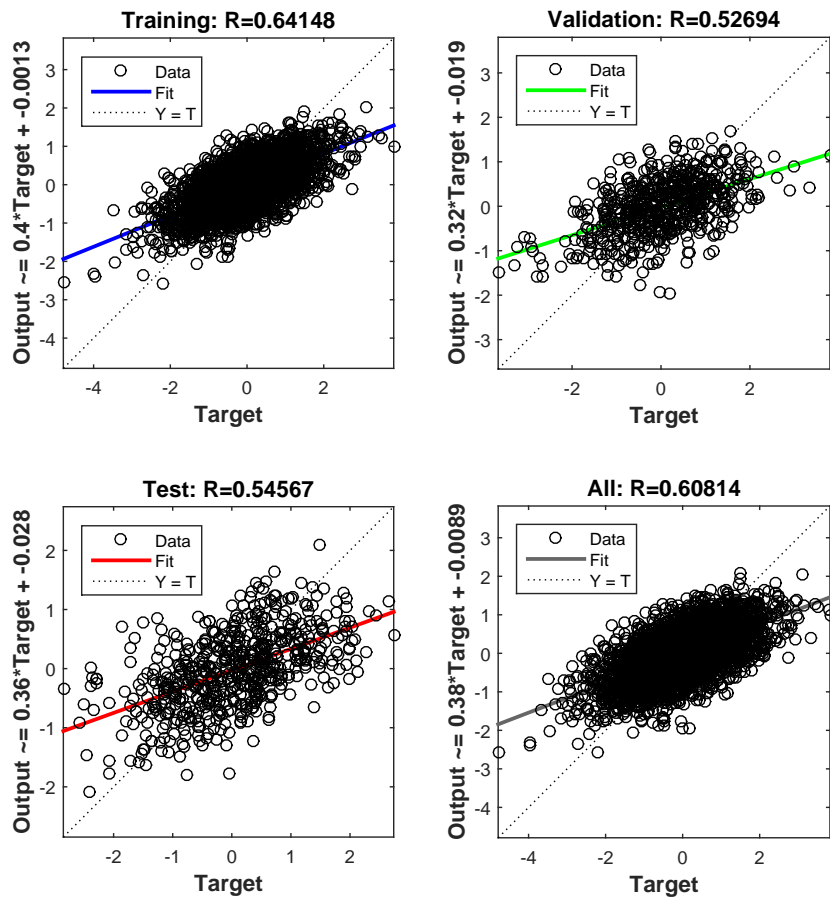


Figure 27: Correlation + SVR. Plots of network output *versus* measured (Target) output for the training, testing, and their combination datasets.



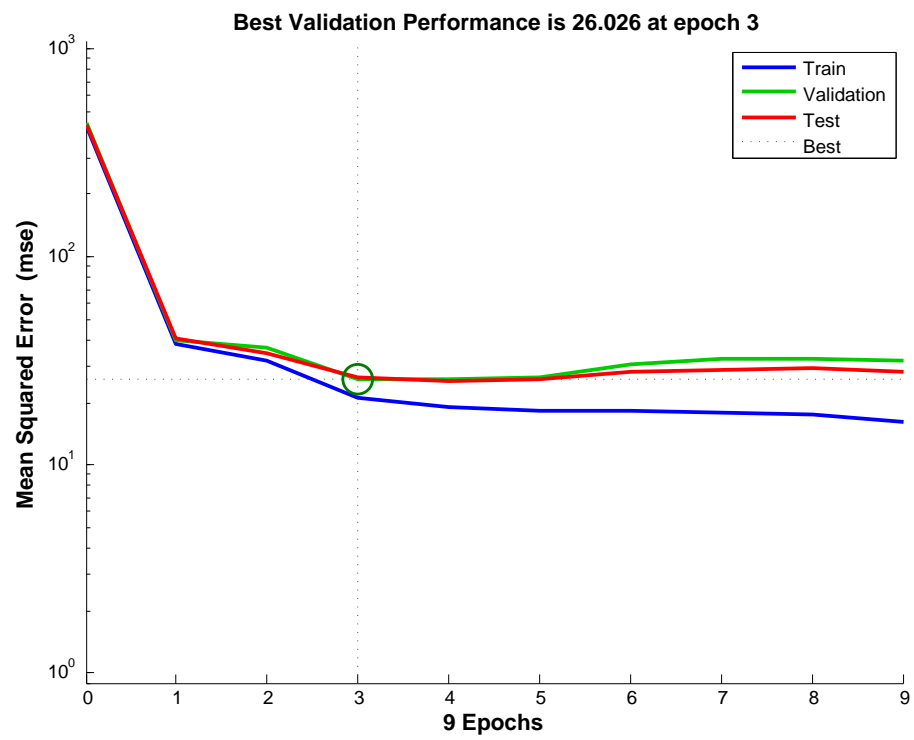


Figure 28: mRMR. Plot of the training, testing, and validation errors *versus* training epoch.

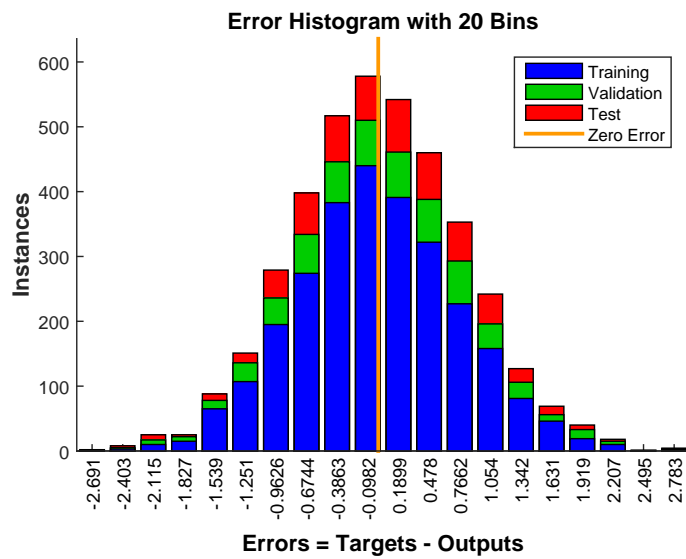


Figure 29: Correlation + SVR. A histogram of the network prediction errors.

Table 1: Comparison of the average prediction errors for neural network models trained with datasets chosen by the three different feature selection algorithms. The dataset chosen by stepwise regression resulted in models with the most accurate predictions.

	<b>MAPE</b>	<b>RMSE</b>
Corr + SVR	0.6534%	4.5450
Stepwise	<b>0.6473%</b>	<b>4.4332</b>
mRMR	0.6877%	4.6503

centered around zero. The mean absolute percentage error (MAPE) was 0.6473%.

#### 4.3.3 Minimum redundancy maximum relevance (mRMR)

The graphs in Figure 32 plot the output of the neural network model against the target (measured) data, for the model trained with features chosen by the mRMR feature selection algorithm. There is a plot for the training data, validation data, testing data, and the combination of the three.

The sample correlation coefficient of 0.59418 for the training data is lower than that for the models trained with data chosen by the other two feature selection algorithms. The sample correlation coefficient of 0.49651 for the testing data is also lower than it is for the other two datasets.

Figure 33 shows a histogram for the prediction errors of the network. The errors are centered around zero. The mean absolute percentage error (MAPE) was 0.6877%.

#### 4.3.4 Genetic algorithm for optimized process recipe

The graphs in Figure 34 depict how the genetic algorithm evolved towards an acceptable solution. The fitness value *versus* generation graph plots both the average fitness value of the population and the best (smallest distance between predicted and desired output) fitness of any individual in the population. It can be seen that there existed, from the outset, at least one individual with a fitness close to zero. The average fitness value of the population quickly decreased, becoming almost zero in nearly twenty generations.

The second graph plots the average distance between individuals as the algorithm progresses. The population can be seen to evolve towards and become concentrated around an

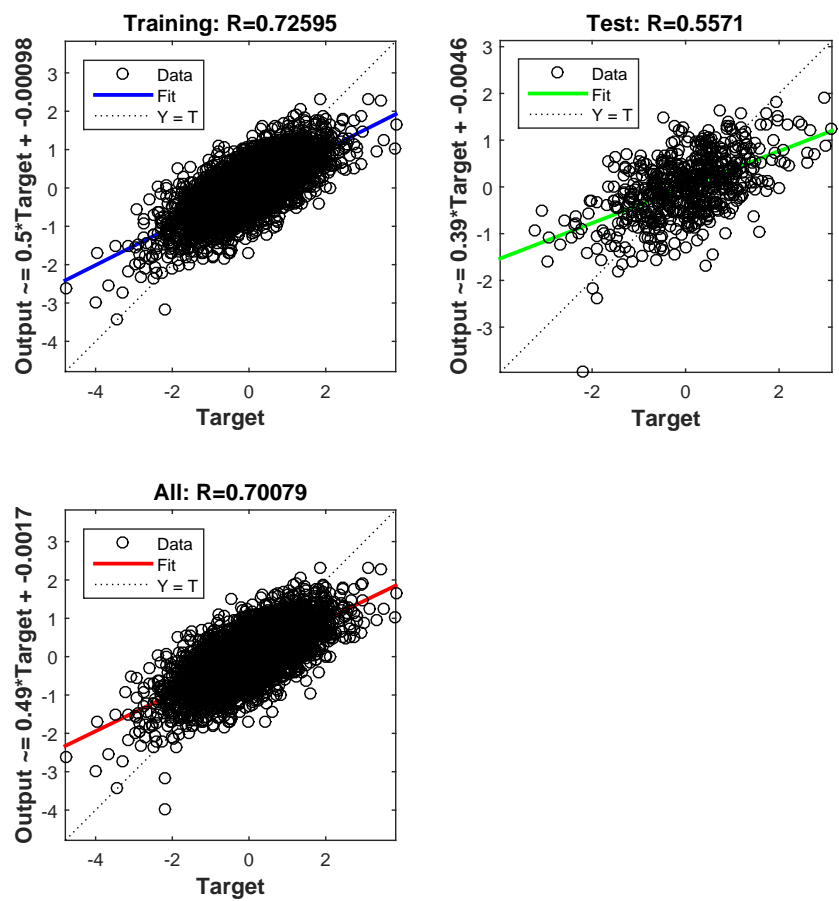


Figure 30: Regression results

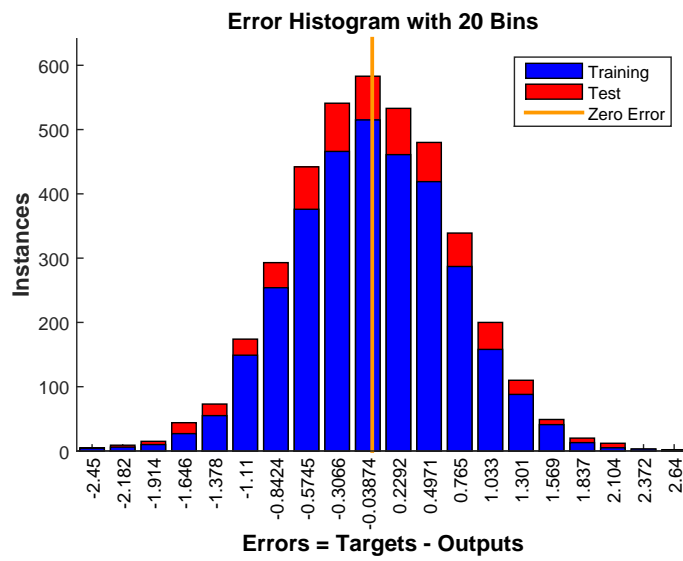


Figure 31: Error histogram

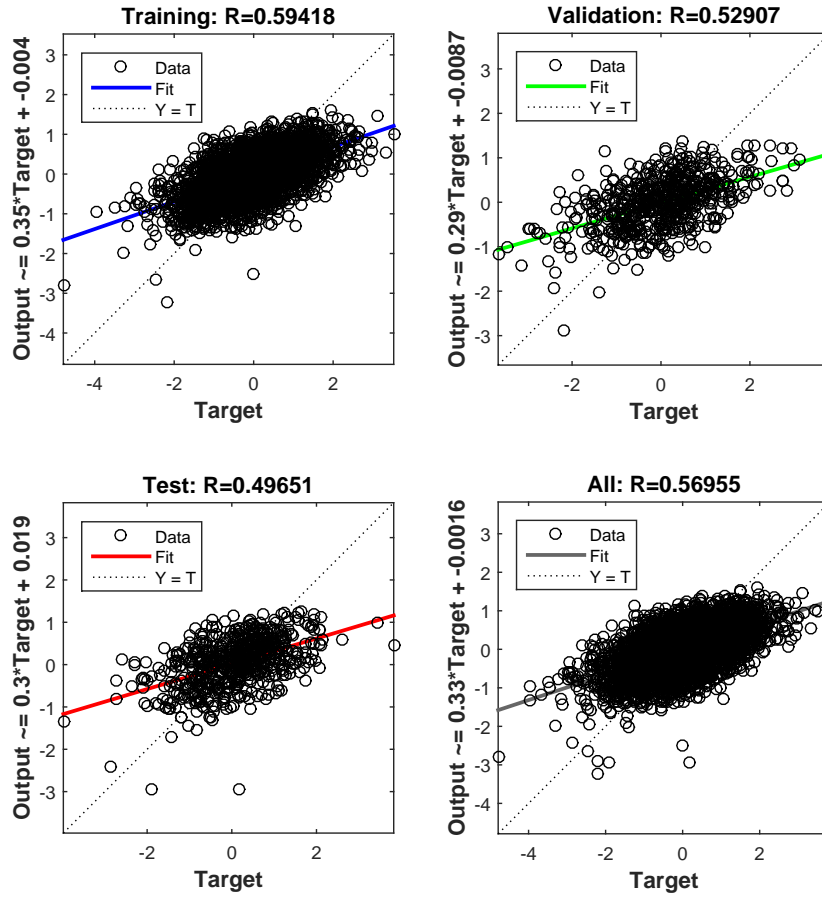


Figure 32: mRMR. Plots of network output *versus* measured (Target) output for the training, testing, and their combination datasets.

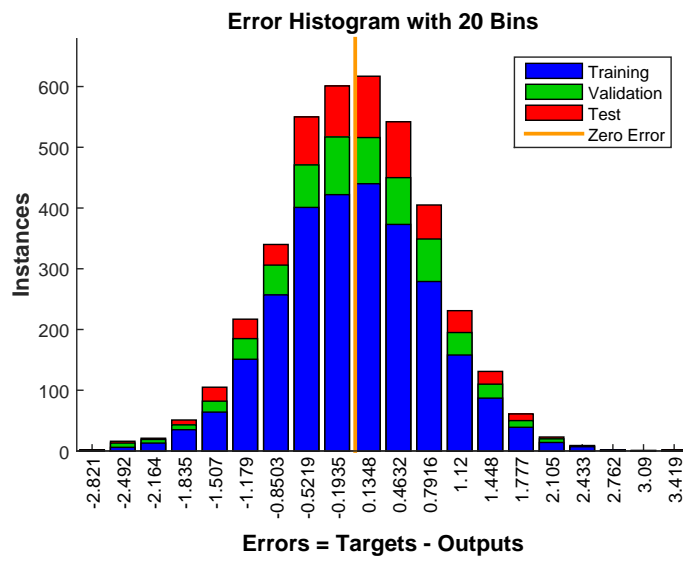


Figure 33: mRMR. A histogram of the network prediction errors.

optimum.

The score histogram shows the distribution of the scores for the individuals in the final population. Most of the scores are seen to be close to zero. The last graph plot the fitness for each individual in the population. The x-axis shows that there were 200 individuals in the population. It can be seen that all individuals have a fitness value below 0.004.

## 4.4 Discussion

Model performance was similar for datasets chosen by all feature selection algorithms. One possible explanation could be that only a few features contributed the majority of information needed to model the variation in CD. Thus differences in features chosen by the different algorithms would not contribute significantly to differences in model performance if the most relevant features are present in every dataset.

In the best case, where the feature subset was chosen by stepwise regression, the model explained about 53% of the variance ( $R^2 = 0.5270$ ). See Figure 30. Ideally this would be higher.

The time taken to run the feature selection algorithms ranged from a few minutes to many hours. For the data set used in this work there was no correlation between the running time of an algorithm and its performance. Stepwise regression chose the feature subset that resulted in the most accurate predictions, and took the least amount of time to run. Given this result, and how common it is to find implementations of this algorithm in different software packages, it is a good starting point for feature selection.

The genetic algorithm converged very quickly and definitively to a solution. This is impressive given the dimensionality of the solution space and heuristic nature of the algorithm. The solution represents a process recipe that brings the value of the process output for the model closest to its desired value. Given an accurate model, this recipe should also bring the value of the output of the actual process to its desired level. Searching in the realm of a computer model, as opposed to experimenting on a manufacturing line, represents a



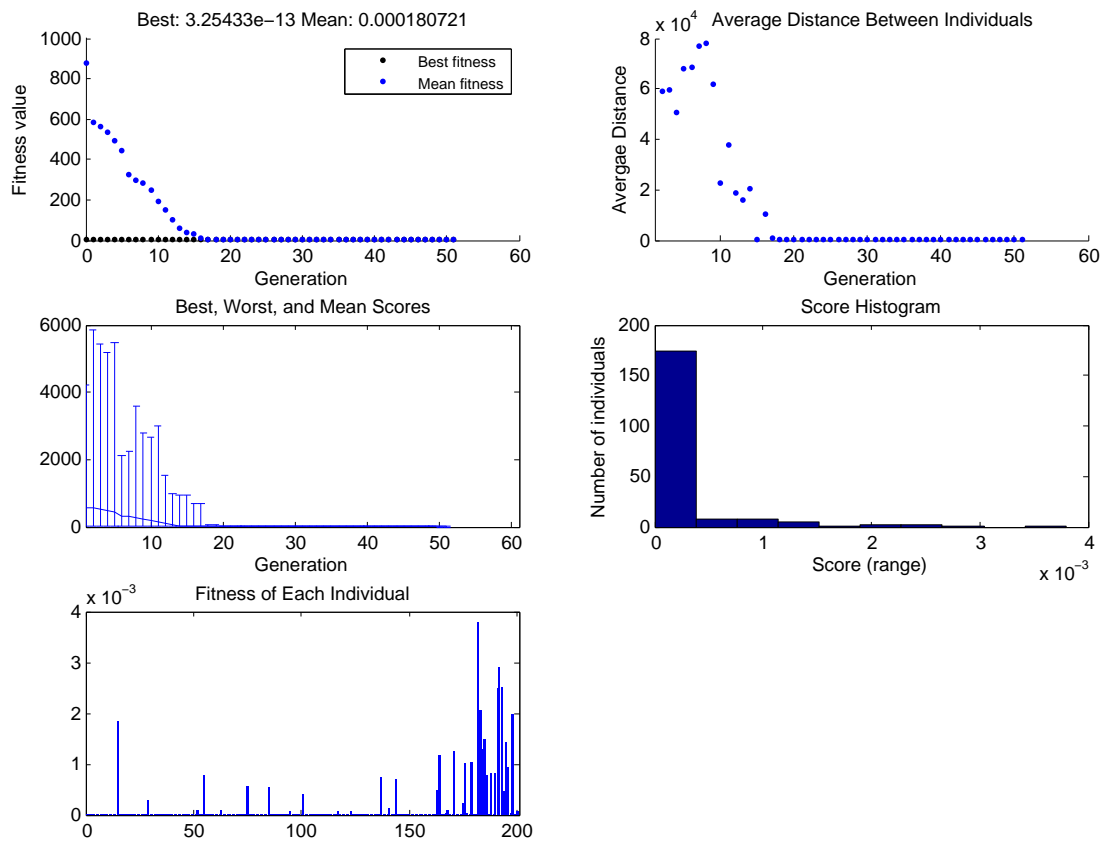


Figure 34: Genetic algorithm optimization plots.

significant savings in cost, time, and effort for a manufacturer.

## **4.5 Conclusion**

Model performance, in terms of the sample correlation coefficient between measured and predicted data, reached a plateau across the datasets chosen by the different feature selection algorithms. Given the breadth of feature selection algorithms and model sizes used, one could conclude that the dataset did not contain enough information to build a more accurate model. But this is valuable information in itself. It points to need for more, or adjusted metrology for this process step. For example, there might be some unmeasured influence on the CDs that is worth searching for. Or perhaps an algorithm for extracting CDs from the raw data could be modified.

## **CHAPTER 5**

### **MANUFACTURING FAB MODELS**

#### **5.1 Introduction**

This chapter will present an exploration and evaluation of the use of the neural network modeling method in a manufacturing fab. The different nature of the data available in a manufacturing fab database, as opposed to that in a development one, presents a set of trade-offs.

In a manufacturing environment at steady state, yields are typically much higher than in a research and development environment. Fab engineers are more able to analyze and identify which upstream processes are responsible for faulty yield metrics. Thus the right data, from lots where faulty yield metrics were caused by fabrication process of interest, can be isolated and used for model building as illustrated in Figure 35. Accurate models of several yield metrics were created from input variables to a high-aspect-ratio etch process. Models were also made of the post-etch metrology from input variables to the high-aspect-ratio etch.

#### **5.2 Data collection**

Data collection is an important part of the modeling process. The quality and amount of training data used greatly affects the performance of neural network models. In past studies, the data used typically originated from designed experiments [17, 20, 45], and only occasionally from industrial fabrication processes [11, 46]. Ironically, the operations of semiconductor fabs generate very large amounts of useful data. Although the data is typically generated for purposes other than process modeling [47], some of it like on-line sensor data (gas flows, RF power, in-line CD measurements, etc.) is nevertheless relevant. The data generated are usually stored in large databases that can be queried [11].

The advantage of using production data for modeling is that it saves the cost and time

## Data for yield models during manufacture

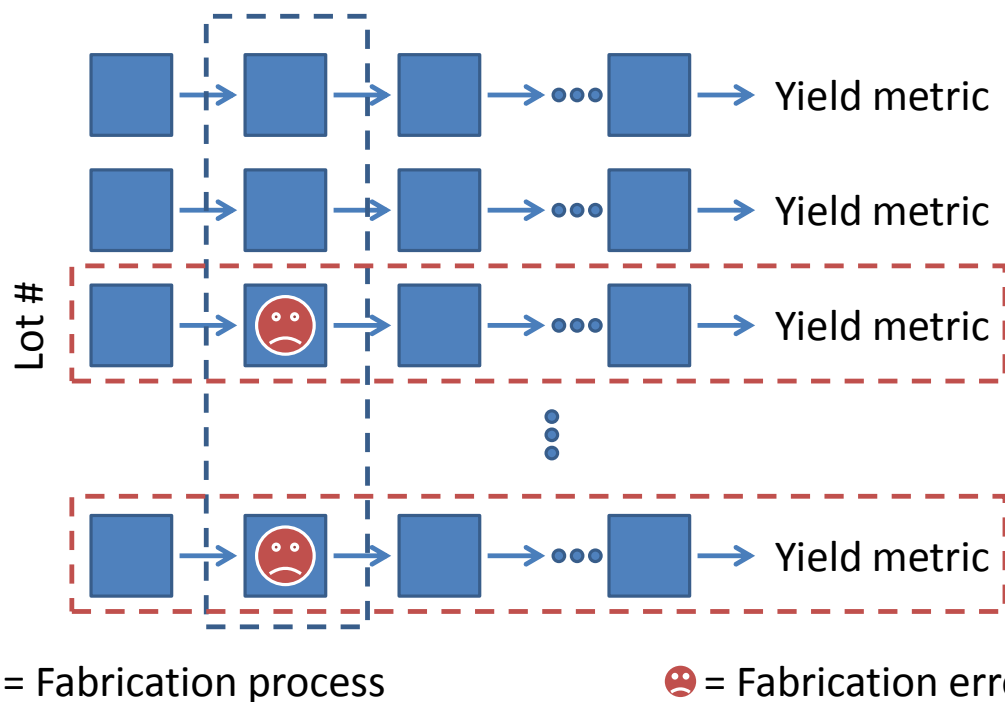


Figure 35: A schematic of the data available for model building in a manufacturing environment. The fabrication process to be modeled is in the dashed blue rectangle. The lots whose data can be used to build a model for the yield metric are in the dashed red rectangles.

spent to carry out designed experiments [47]. However, there are some disadvantages. Artificial neural network models are much better at interpolating than extrapolating [48]. With production data, there is much less variation in the process variables. The resulting models thus may not be as useful for optimization purposes, as one has to assume that the optimal recipe for a process is within that narrow range of inputs [48]. The other problem with using production data is the effort needed to get the data in a format ready for modeling. There can be inconsistencies in the way the data is collected, labeled, stored, etc. [48]. Training neural network models with designed experiments provides more variation in the training inputs, but incurs additional cost and will typically generate less training data. Less training data potentially limits the accuracy of the network models that can be created.

## **5.3 Modeling**

The models were created with the Object Oriented Neural Network Simulator (ObOrNNS), a program developed by the Intelligent Semiconductor Manufacturing Group at Georgia Tech. It allows for rapid creation, training and testing of neural network models [49]. The choice of inputs for all the models evolved iteratively through the creation of several models of each type. Results presented are from those that performed best.

### **5.3.1 High-aspect-ratio etch**

Similar to that in the R&D fab, the purpose of the high-aspect-ratio etch is to create an array of high-aspect-ratio holes in an oxide layer. These holes act like a mold for device structures created in subsequent fabrication steps. Therefore, the dimensions of these holes are important. Of particular importance are the distances between these holes in the x-direction and y-direction. They are the CDs we are interested in controlling and thus they form the output of the model. The overall etch is made of three smaller etches each with its own set of process parameters like gas flows, RF powers, bias powers, and charge species tuning unit (CSTU) settings. These, together with the measurements of the critical dimensions in

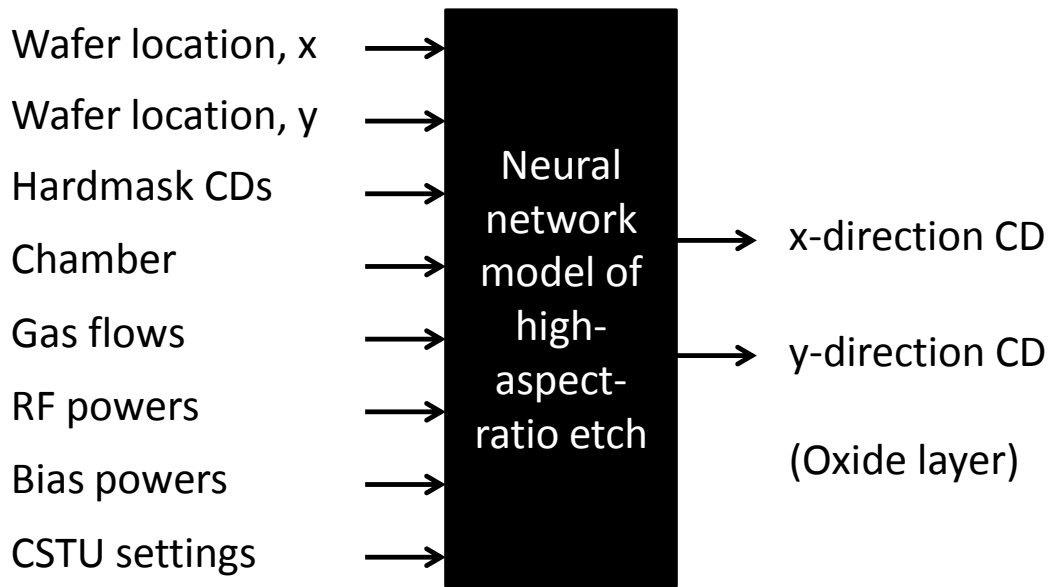


Figure 36: An overview of the model created for the high-aspect-ratio etch in the manufacturing fab.

the masking layer, the wafer location of these measurements, and the chamber in which the wafer is processed, form the model inputs. An overview of the model is presented in Figure 36.

A 16-16-2 network was created to model the process. Data from approximately 1100 wafers was used to train the model and data from another 50 wafers was used to test it.

### 5.3.2 Yield metrics

There were two models created to predict yield metrics. In-line CD measurements, the location of these in-line CD measurements, and an indicator of the process chamber form the inputs to these models (Figure 37). The in-line CD measurements are the same as those in the output to the oxide etch model. This fact is important for future work involving the optimization for yield because the models can be connected as in Figure 38.

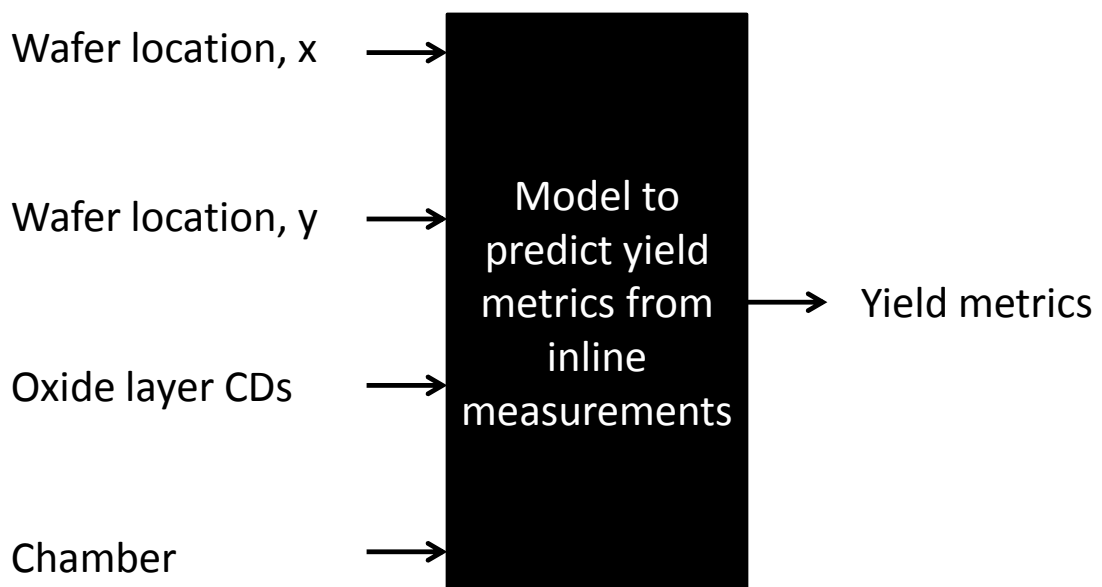


Figure 37: An overview of the models created to predict yield metrics from inline measurements.

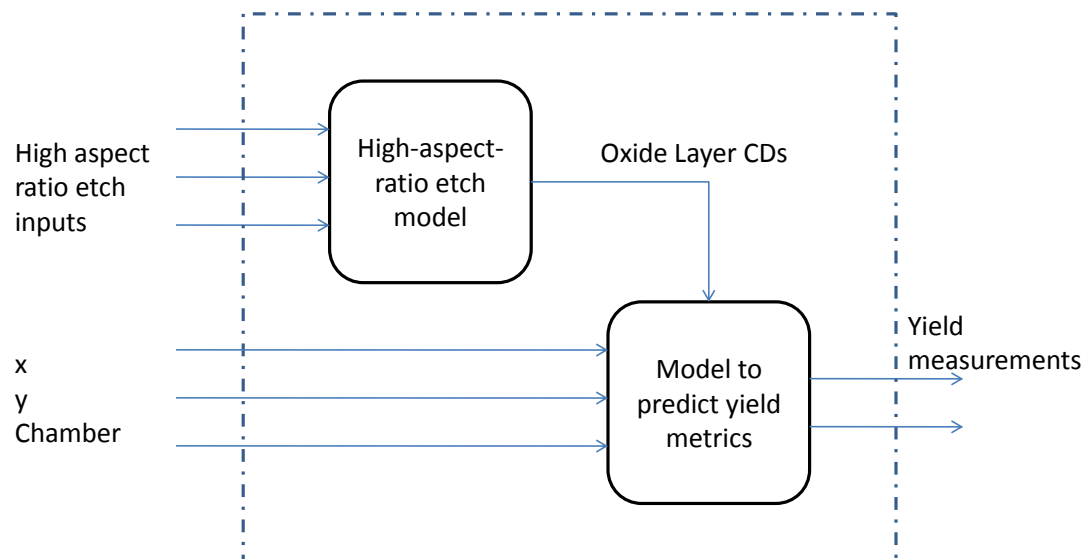


Figure 38: Sequential neural network model.



#### *5.3.2.1 Defect counts*

The first of the yield models predicts defect counts on the wafer. The model makes separate predictions for defects that are caused by over-etch and under-etch during the high aspect ratio etch step. The model was created with a 5-6-12 neural network. Data from approximately 650 wafers was used to train the model and data from another 50 wafers was used to test it.

#### *5.3.2.2 Probability of die failure*

The second of the yield models predicts the probability that a die will fail a test of functionality at probe as a result of non-ideality in the high aspect ratio etch step. A larger amount of preprocessing had to be performed for this model. This is because the probabilities of failure were not directly available from the database and had to be calculated and arranged properly into the modeling datasets. The model was created with a 5-5-6 neural network. Data from approximately 650 wafers was used to train the model and data from another 27 wafer was used to test it.

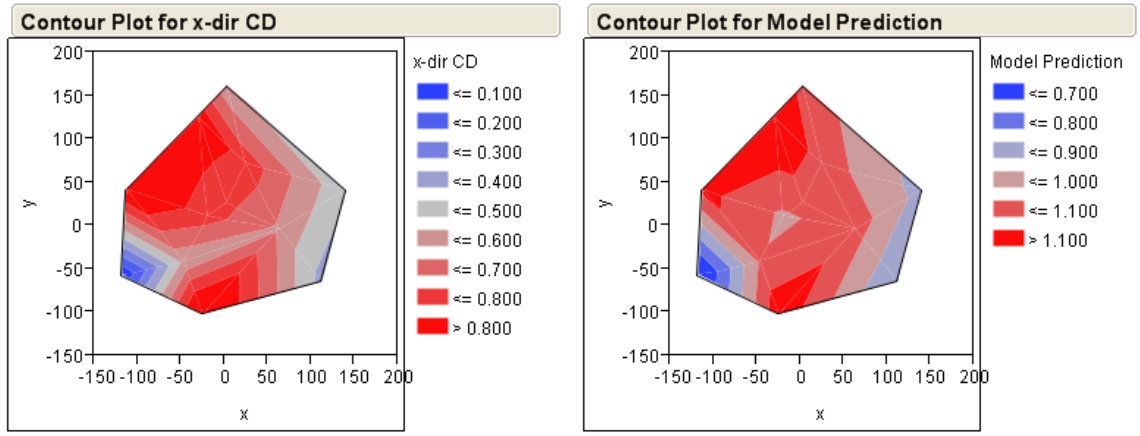
## **5.4 Results**

### **5.4.1 High-aspect-ratio etch**

The model performed very well with an average prediction error of 3.9%. Results are presented graphically as error contour plots. This allows comparison of both the values of the output and its variation across the wafer. Figure 39 shows the results for x-direction CD. The CD values have been normalized using the range of the measured CD values. The results presented are the average over 47 wafers. The contour plot for absolute percentage prediction error for x-direction CD is presented in Figure 40.

Figure 41 presents a similar comparison of measured and predicted CDs, but this time for the y-direction CD. The contour plot for absolute percentage prediction error for y-direction CD is presented in Figure 42.

In both cases, the model has accurately captured the distribution of the CDs as seen



(a) x-direction CD

(b) Neural network predictions for x-direction CD

Figure 39: Comparison of measured (left) and predicted (right) CDs. The x-axis and y-axis represent wafer location in mm.

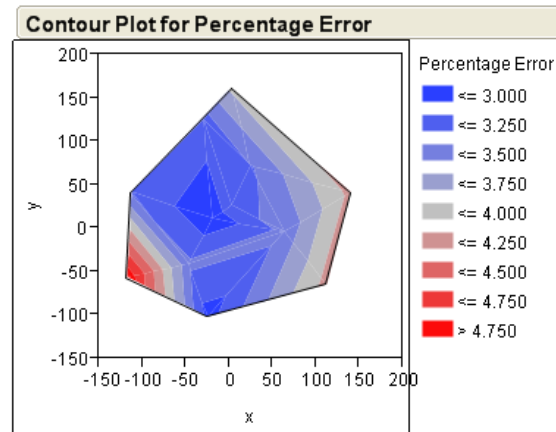
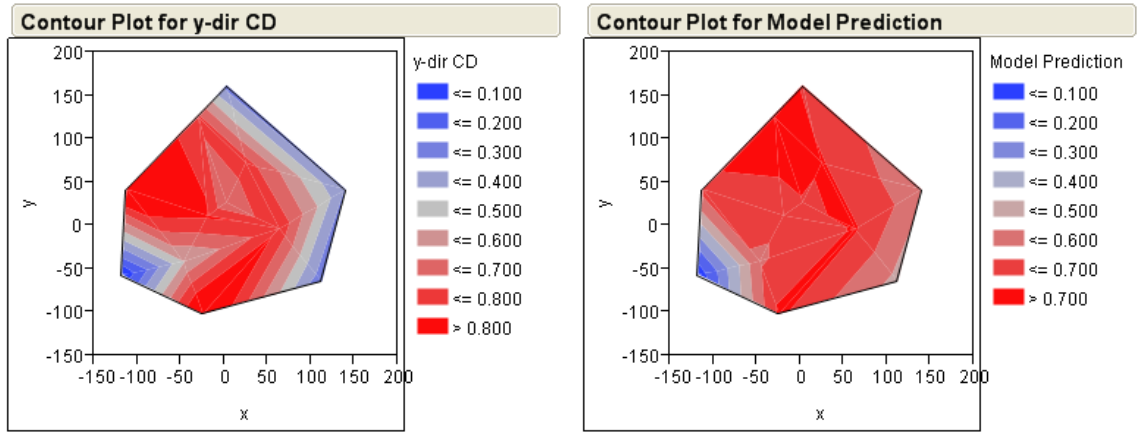


Figure 40: Absolute percentage error for neural network predictions for x-direction CD.



(a) y-direction CD

(b) Neural network predictions for y-direction CD

Figure 41: Comparison of measured (left) and predicted (right) CDs. The x-axis and y-axis represent wafer location in mm.

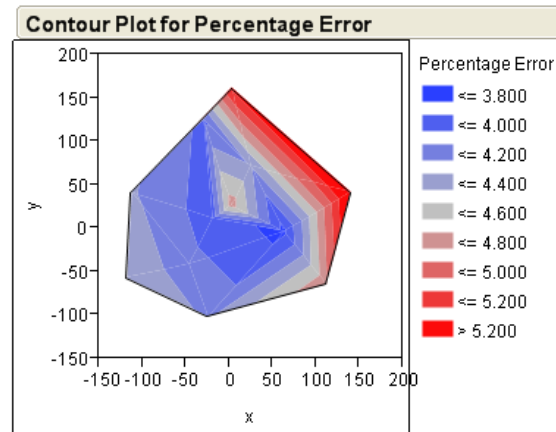
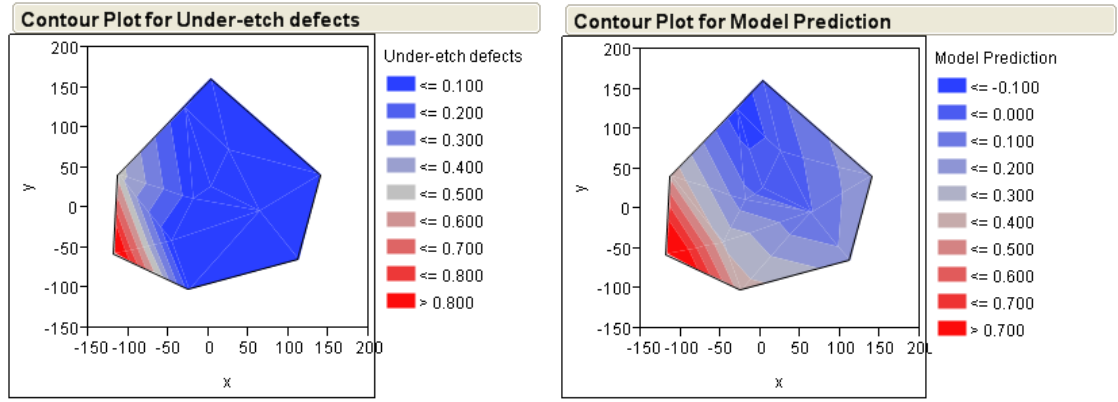


Figure 42: Absolute percentage error for neural network predictions for y-direction CD.



(a) Under-etch defect counts

(b) Neural network predictions for under-etch defect counts

Figure 43: Comparison of measured (left) and predicted (right) under-etch defect counts. The x-axis and y-axis represent wafer location in mm.

from the similarity between the plots of measured CDs and the model-predicted CDs in Figures 39 & 41. Also, as can be seen from figures 40 & 42, the error remained below 5% across most of the wafer for both sets of predictions.

## 5.4.2 Yield metrics

### 5.4.2.1 Defect counts

The average prediction error for defect counts was 14.9%. Figure 43 shows the results for defects as a result of under-etch. The model accurately captured the distribution of the defects as seen in the similarity between the plots of actual defects (left) and predicted defects (right). The prediction error remained mostly below 15% as seen in Figure 44.

Figure 45 shows the results for defects as a result of over-etch. The model accurately captured the distribution of the defects as seen in the similarity between the plots of actual defects (left) and predicted defects (right). The prediction error remained mostly below 25% as seen in Figure 46.

### 5.4.2.2 Probability of die failure

The average prediction error for probability of die failure was 21.8%. Figure 47 shows the results for probabilities of die failure as contour plots. The model captured the distribution

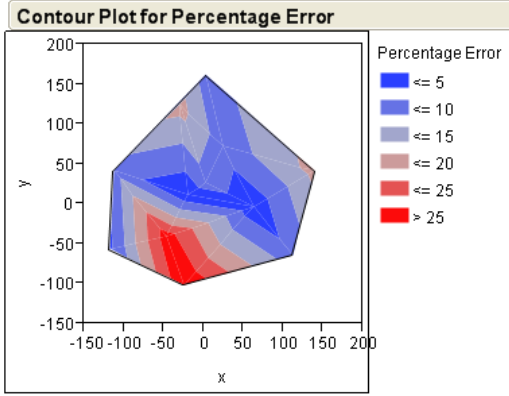
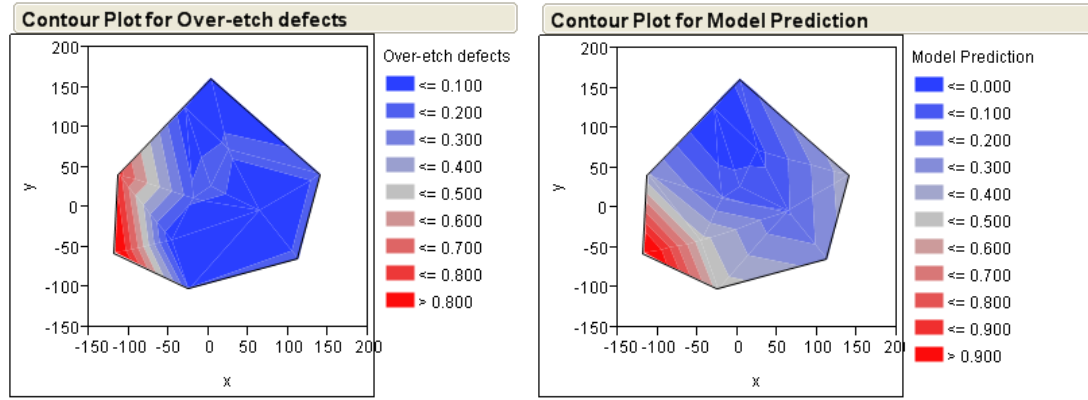


Figure 44: Absolute percentage error for neural network predictions for under-etch defect counts.



(a) Over-etch defect counts

(b) Neural network predictions for over-etch defect counts

Figure 45: Comparison of measured (left) and predicted (right) over-etch defect counts. The x-axis and y-axis represent wafer location in mm.

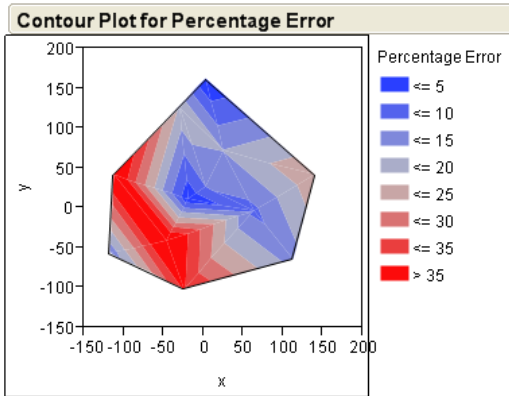
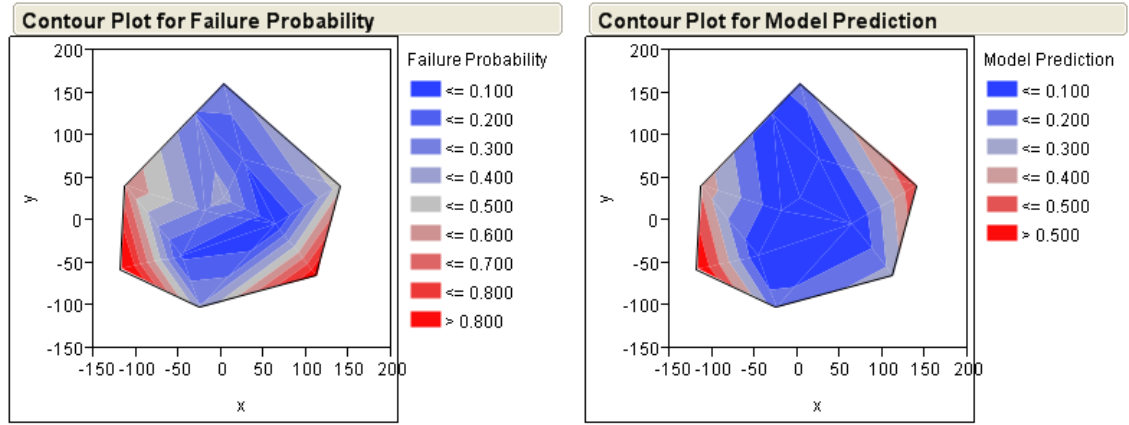


Figure 46: Absolute percentage error for neural network predictions for over-etch defect counts.



(a) Probability of die failure.

(b) Neural network predictions for probability of die failure.

Figure 47: Comparison of measured (left) and predicted (right) probability of die failure. The x-axis and y-axis represent wafer location in mm.

of the defects as seen in the similarity between the plots of actual defects (left) and predicted defects (right). There was a deviation in the predictions for area of high probability of failure on the right part of the wafer. Its location is predicted to be slightly higher up on the wafer than in the actual distribution (left). The prediction error remained mostly below 30% as seen in the Figure 48.

## 5.5 Sensitivity analysis

A sensitivity analysis was performed for the for both process models and yield models. Performing a sensitivity analysis for a mature manufacturing process is perhaps more useful than it is for one in development. Because of the costs associated with lost product, costs that are increasing with the tendency toward larger wafer sizes and smaller product geometries, manufacturers are very conservative in the changes they make to any process in a manufacturing line. This includes changes they make for the sake of experimentation. A sensitivity analysis is a systematic way for a manufacturer to explore opportunities for process improvement in the narrow window around the current operating point for a process. A representative example would be to look at the relative effects the inputs of the

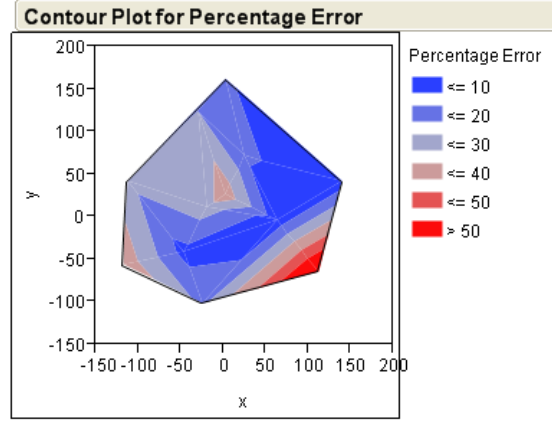


Figure 48: Absolute percentage error for neural network predictions for probability of die failure.

defect counts model have on under-etch related defects (see Figure 49).

## 5.6 Discussion and summary

The contrast between development and manufacturing data presented trade-offs for modeling. There is less input variance in manufacturing data. However, there is much more data to work with. The models were able to handle the increased complexity and the reduced input variance of the manufacturing data and were able to model the high-aspect-ratio etch process well.

What was not clear from modeling using R&D fabrication data was the neural networks ability to predict yield. With development data, it was difficult to attribute yield metrics to fabrication processes, or process parameters, or in-line measurements. It was difficult to collect enough yield data with enough accuracy for modeling.

There was dramatic improvement in the predictions for yield using manufacturing data, with the neural networks being able to make predictions about yield using in-line CD measurements. They captured the uniformities of the fails across wafer particularly well.

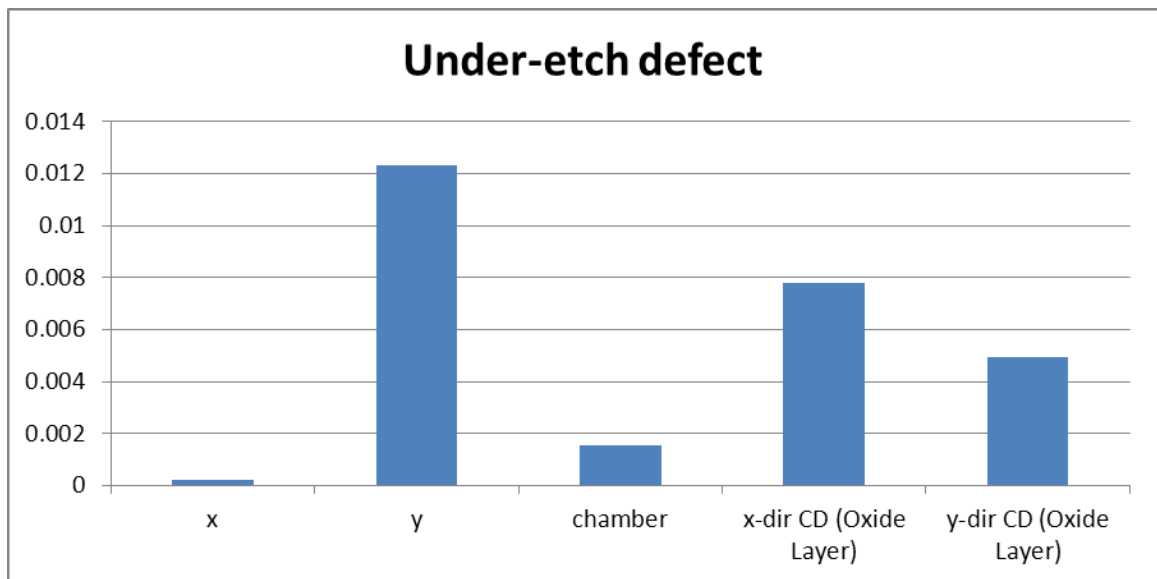


Figure 49: Sensitivity analysis. The bars represent the percentage change in under-etch defects in response to a uniform change in their respective inputs



## CHAPTER 6

### CHAMBER MATCHING EXERCISE

#### 6.1 Introduction

Semiconductor manufacturing, whether it is for memory or microprocessors, is high volume manufacturing. The manufacturing process is comprised of hundreds of distinct steps performed by sophisticated pieces of equipment. In order to realize high volume, there are typically several reaction chambers operating in parallel in which the same fabrication process can occur. There is unavoidable variation in a particular fabrication process step across chambers, even if the chambers are the same (manufacturer, model, configuration) and running the same recipe.

One critical process in modern semiconductor manufacturing is plasma etching. It uses an ionized gas (plasma) that is created in the reaction chamber using an RF electric field. The plasma reacts with the areas of the wafer surface that are exposed. The reaction products then diffuse away from the surface leaving a void. The plasma allows the reactions to occur at a lower temperature than they normally would, and plasma etching affords more etch anisotropy than wet etch methods [50].

The process variation between plasma etch chambers can have different effects that can occur anywhere downstream [51]. Process variation is reflected in the immediate post-process metrology, or it is observable in yield metrics which are calculated after possibly hundreds of downstream fabrication processes. This variation affects yield, product performance and reliability [52].

The purpose of this chapter is to explore the use of artificial neural networks and genetic algorithms to reduce chamber-to-chamber variation. Artificial neural networks are empirical modeling tools capable of modeling the complex nonlinear relationships found in many semiconductor manufacturing processes [53]. A genetic algorithm is a global search heuristic [54] that will be used to search the input space to the plasma etch model to find

the recipe that makes the chamber performance of the outlying chamber more consistent with the performance of the other chambers. The new recipe is subsequently verified on the production line. The method used is generally applicable to a wide range of fabrication steps.

This study focuses on a chamber matching exercise for a high-aspect-ratio<sup>1</sup> etch. A chamber is identified as an outlier by a comparison of the moving averages of the post-process metrology of all the chambers. An artificial neural network is used to model the plasma etch process in the outlying chamber. Genetic algorithms are then used to generate a process recipe that brings that outlying chambers<sup>2</sup> performance closer to target and thus reduce the process variation across chambers.

## **6.2 Method**

### **6.2.1 Identifying the outlying chamber**

Chambers were identified as outliers based on moving average charts of the post-process metrology and moving average charts of several yield metrics (see Figure 50). The yield metrics used were identified by fab engineers as those being affected by the process being modeled. The moving average chart contained data from a 45 day window. This guaranteed that the chart would contain data across a preventative maintenance (PM) event for each chamber. PM events need to be considered for plasma etch modeling, as addressed in the literature [37, 55], because chamber behavior can change across these events. A PM event is a step change in the chamber environment, in contrast to the slow drift in chamber environment that occurs due to accumulation of reaction by-products on the chamber walls [18]. The large data window helps to choose a chamber that is an outlier across PM events. The choice of the chamber to be modeled was corroborated using an analysis of means (ANOM) chart, created with JMP®.

---

<sup>1</sup>Aspect-ratio is the ratio of the depth of a structure to its width. A high-aspect-ratio etch could have a 40:1 ratio of etch depth to etch width for example.

<sup>2</sup>This method can be applied to any chamber. An outlying chamber was chosen for proof of concept to demonstrate the efficacy of the method more clearly.

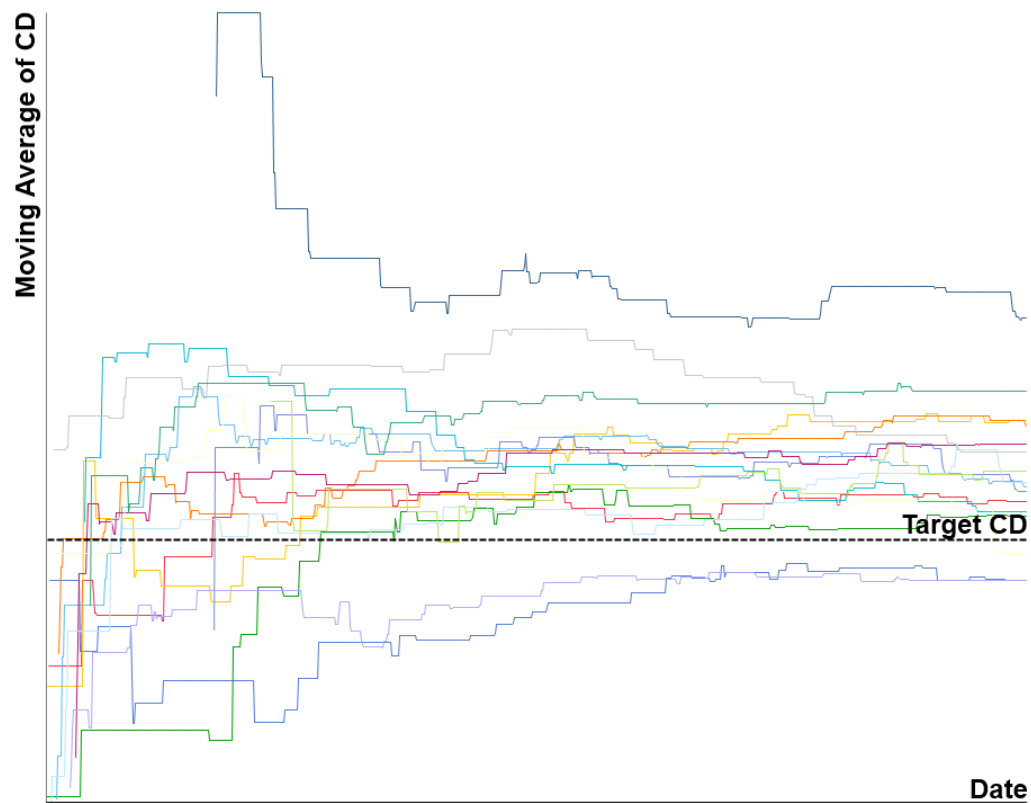


Figure 50: Moving average chart of post-process metrology. The different lines represent the moving averages of the different chambers. Chambers whose lines were furthest from the target CD were candidates for modeling.

### 6.2.2 Chamber modeling

A plasma etch process to etch high-aspect ratio features was modeled. The overall process was comprised of three etch steps, each with a different etch time and chemistry. The result of the overall process was to create an array of high-aspect-ratio holes. These holes act as molds for device structures that are fabricated downstream in the production sequence. The distances between the holes in the x and y direction, and their profiles (anisotropy of the etch) are important to product yield.

The distances between the holes in the x and y direction are measured post process, and are the responses to be modeled. These are labeled as the x-direction CD and y-direction CD respectively, and are also the variables to be controlled. There is no measurement of the profile of the etch that takes place in the production line. This is because of the inherent trade-off between metrology and throughput, and because measuring the profile of the etch would require cross sections (breaking wafers) and negatively impact yield. Issues with the etch profiles are reflected in yield metrics.

There are thirteen inputs for this model. These are the plasma source powers for each of the three etch steps, the O<sub>2</sub> flows for the second and third etch steps, the charged species tuning unit (CSTU) settings<sup>3</sup> for the second and third etch steps, CD measurements in the masking layer, and the wafer location (x and y) of these CD measurements. CD measurements in the masking layer and the wafer locations of these measurements cannot be directly controlled, but they improve model predictions when included as inputs. The other inputs can be manipulated directly. Figure 51 is an overview of the model.

These input variables were chosen by examining the dataset available and using the expertise of process engineers. Process recipes are fairly unchanging in a manufacturing environment, where a single recipe may be used for months. When a recipe does change, a small subset of all possible etch-inputs is changed. Model input variables were chosen from

---

<sup>3</sup>The Charged Species Tuning Unit allows control over the distribution of the plasma ions across the chamber and affords some control over the uniformity of the etch across the wafer. The CSTU settings are set points for current in an inner and outer ring.

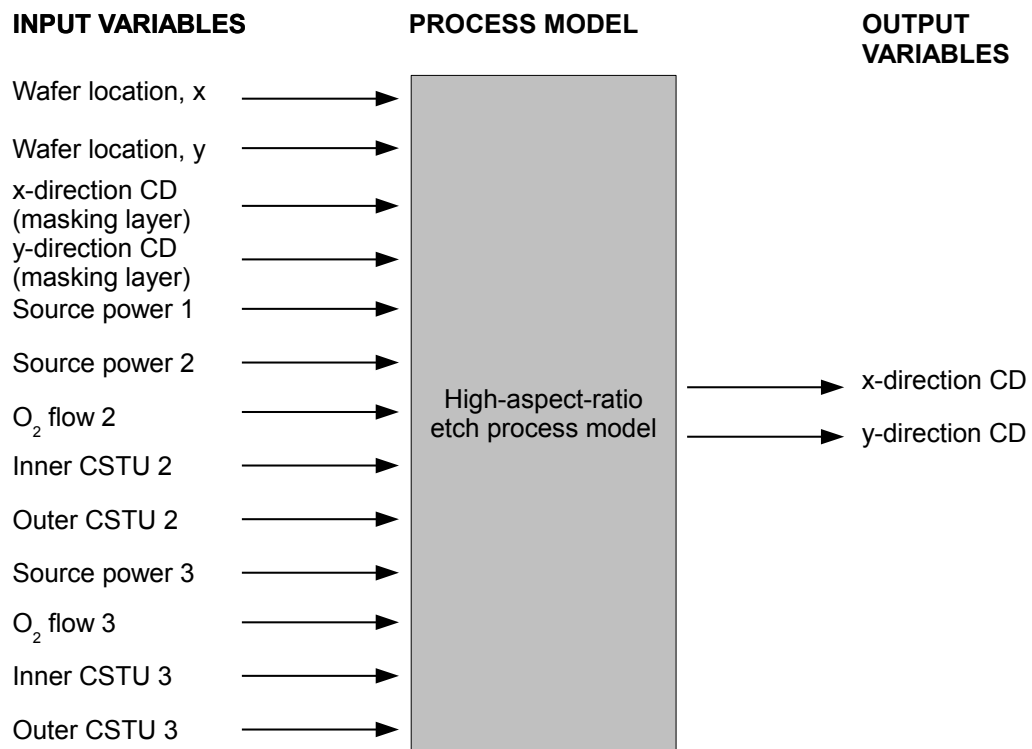


Figure 51: Overview of the model of the high-aspect-ratio etch. Numbers following the input description indicate etch step.

this small subset using the expertise of the process engineers. There are other heuristics for choosing model input variables as outlined in [35, 56]. The process of selecting input variables for a model is commonly referred to as feature selection or attribute selection.

### **6.2.3 Process recipe optimization with Genetic Algorithms**

Given a trained model, a forward pass (input presented to the input layer and its effects propagated) through the network is a simulation of the plasma etch. The output of a forward pass is the model prediction for the results of the etch. One can study the plasma etch response, assuming an accurate model, by running many simulations at different points in the input space. One way to automate this process is a genetic algorithm. A genetic algorithm is a global search heuristic. It was used to search the input space to the model to find the optimal combination of inputs (process recipe), *i.e.* one that produces a desired output.

Genetic algorithms are inspired by natural evolution and borrow its nomenclature. Any point in the input or solution space is represented by a binary string, and is labeled an individual (see Figure 52). An initial population consists of randomly chosen individuals, see Figure 53. Each individual is evaluated for fitness, which is how close its corresponding network output is to the desired output in terms of mean squared error. Fitter individuals are selected to breed the next generation of solutions. This involves crossover and mutation. Crossover is the operation in which two individuals or binary strings exchange some portion of their data, see Figure 54. Mutation is the operation in which the bits in an individual's data are flipped based on some low probability, see Figure 55. This process, of finding the fit individuals or solutions and letting them have greater representation in subsequent generations of solutions, continues until an acceptable solution is found or a maximum number of generations is reached. The aim is for the population of solutions to evolve to a global optimal solution [54]. A flowchart for the overall process is presented in Figure 56.

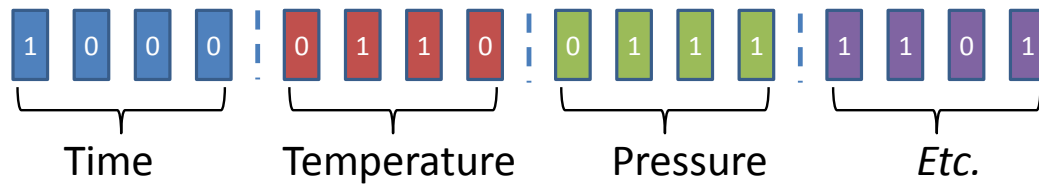


Figure 52: Example of a point in the input space encoded as a binary string. Points like these are labeled “individuals.”

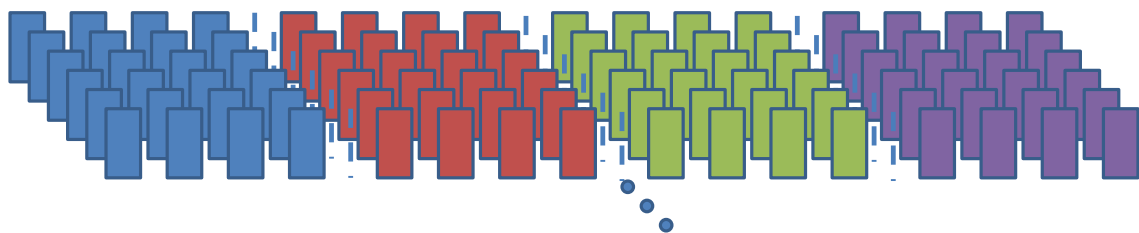


Figure 53: A collection of possible solutions, encoded as binary strings, is labeled a population.

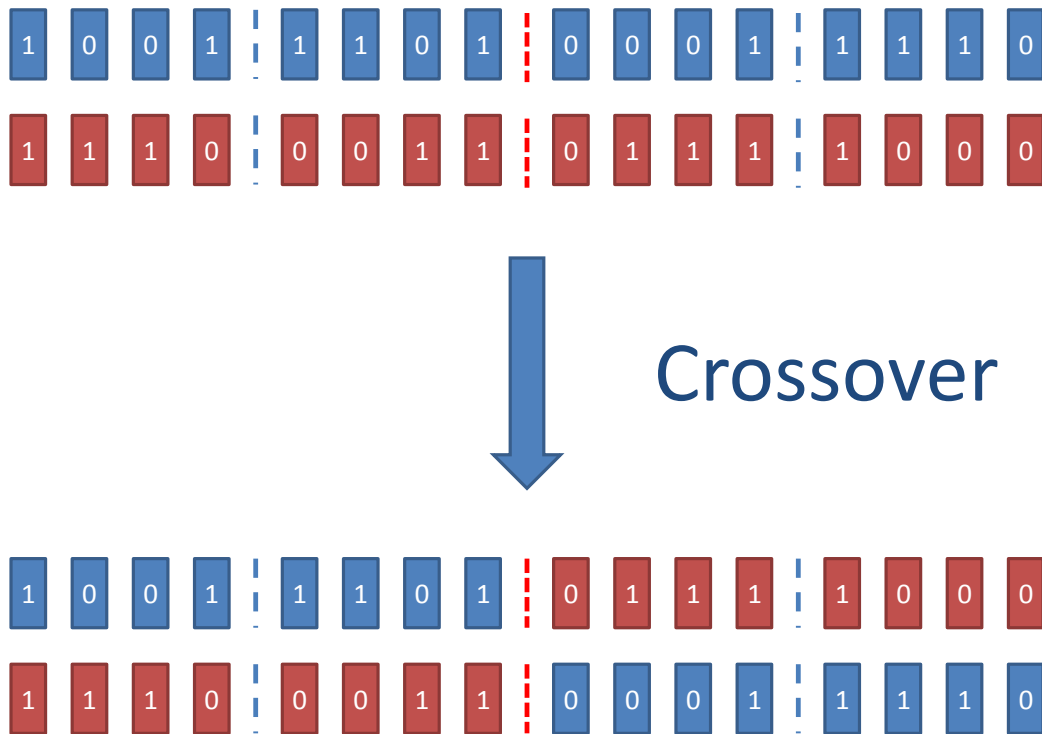


Figure 54: Illustration of crossover. The two individuals exchange their data to the right of the arrow.

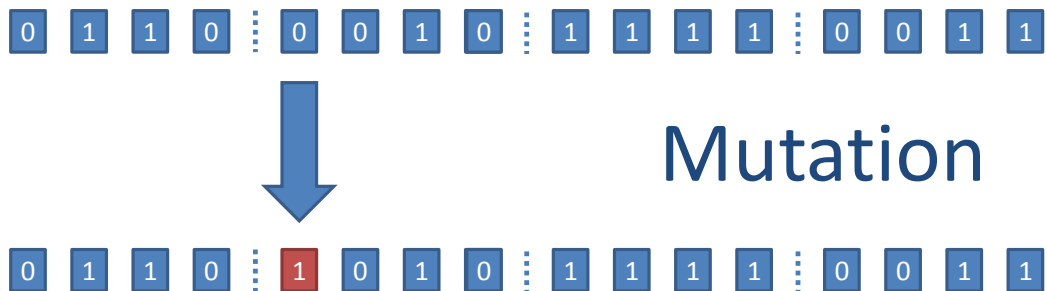


Figure 55: An example of mutation where one bit is “flipped.”



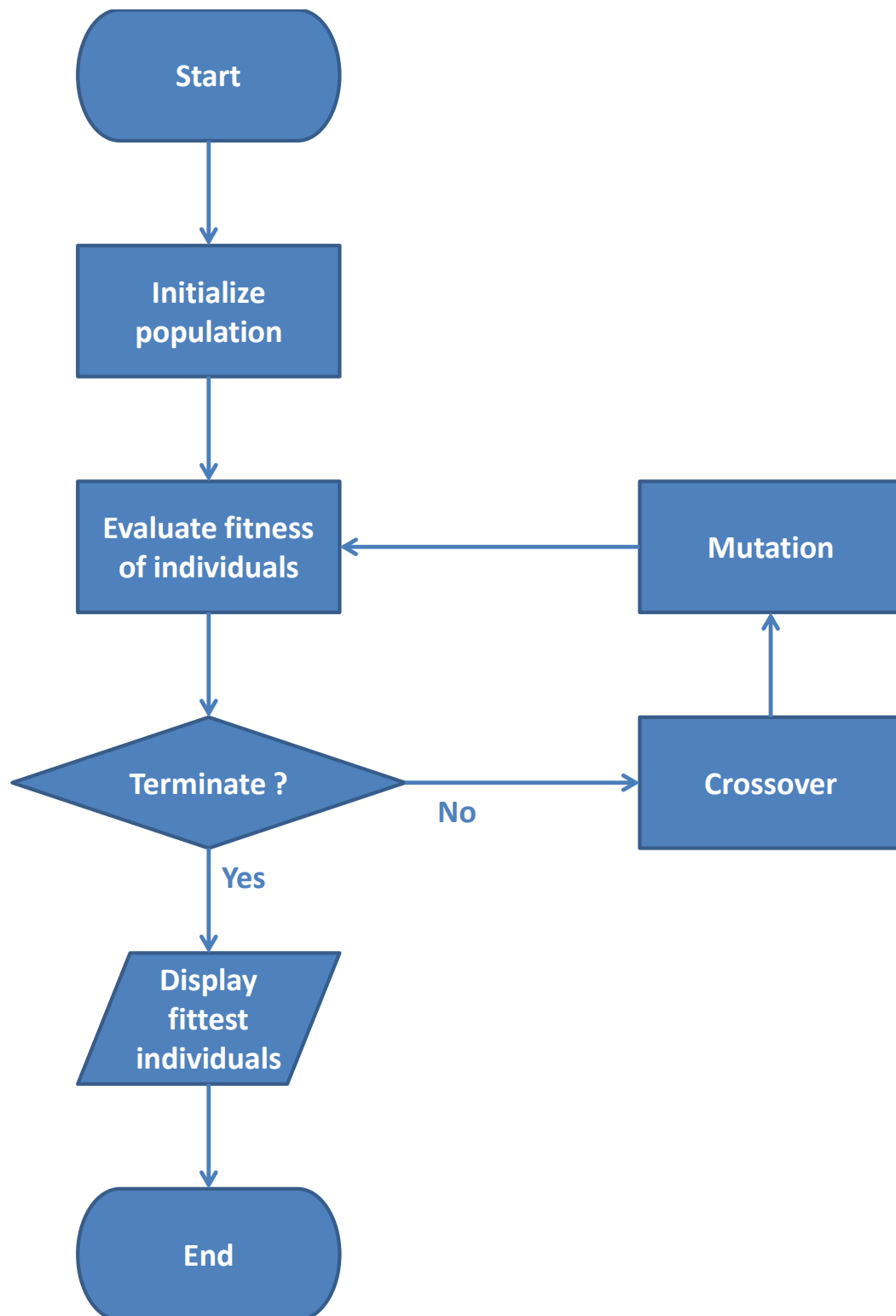


Figure 56: Flowchart of a genetic algorithm.

#### **6.2.4 Dataset**

The model was made using measured *in-situ* and post-process data from 431 days of manufacturing. These data came from about 2900 wafers in 723 different lots. The fidelity of a neural network model is tested on data that has not been used to train it. Therefore 20% of the data was randomly chosen and separated from the larger dataset for testing.

In contrast to the research and development setting for semiconductor products, process recipes change infrequently in a manufacturing setting. Thus the measured data for the input variables may not cover the entire input space possible, but only a subset of this. The genetic algorithm searches this subspace for the ideal process recipe because neural network models are better at interpolating than extrapolating [48]. The assumption is therefore that an optimal process recipe exists within the input space circumscribed by the measured data.

#### **6.2.5 Verification of the experimental recipe on the production line**

Running the genetic algorithm on the neural network process model returns several process recipes ranked in order of fitness. Recipes whose output has smaller deviation from the target output are ranked higher. The best of these recipes (experimental recipe) was chosen and used on the actual production line.

The experimental recipe was run on 16 wafers. These 16 experimental wafers were contained in two different lots, with eight wafers in each lot. The nominal (or “control”) recipe was run on 28 wafers in those same two lots, 17 wafers in one lot and 11 in the other.

### **6.3 Results**

#### **6.3.1 Modeling results**

Model fidelity was evaluated by comparing the response of the actual process to that of the neural network model, given the same input from the test data set. The test data set provides the output response of the actual process in the form of post-process measurements. The neural network model response is obtained by passing the values of the input variables in

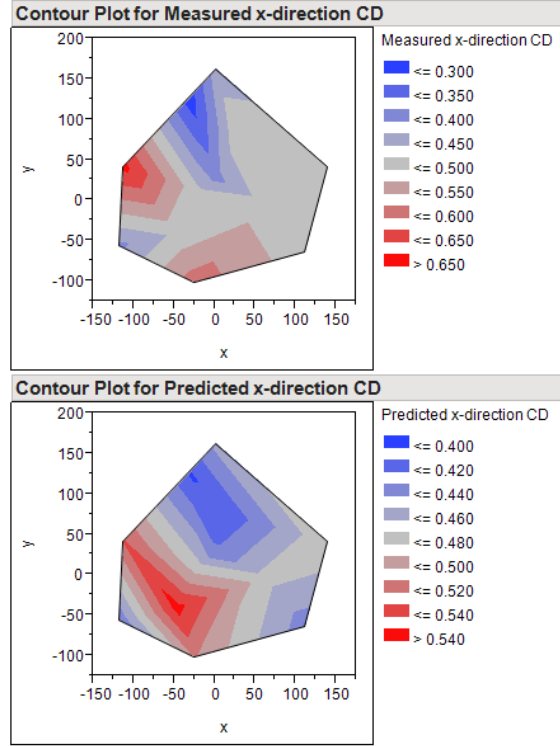


Figure 57: Contour plots using measured and predicted data for the x-direction CD.

the test data set as inputs to the model. Model output is labeled predicted data. A percentage prediction error was calculated for each point in the test data set and then averaged. The average prediction errors for the x-direction CD and y-direction CD were 2.6% and 3.8%, respectively.

Contour plots of normalized measured and predicted data were used to evaluate model performance. They provide visual corroboration that the model captured the behavior of the process - etch uniformity across the wafer in particular. The contour plot for measured and predicted data for x-direction CD is shown in Figure 57. A contour plot of prediction error for x-direction CD, as seen in Figure 58, shows how prediction error was distributed across the wafer and the range of these errors. Similarly, a contour plot for measured and predicted data for y-direction CD is shown in Figure 59, and a contour plot of prediction error for y-direction CD appears in Figure 60.

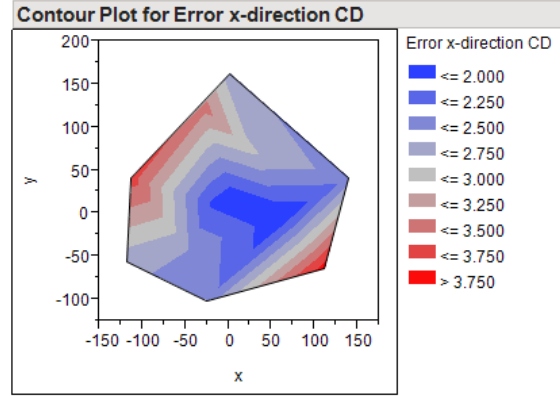


Figure 58: Contour plot of the percentage prediction error for x-direction CD.

### 6.3.2 Verification of experimental recipe

An experimental etch recipe was generated using genetic algorithms. The experimental recipe was evaluated by comparing the post-process metrology and yield of the wafers running the experimental recipe to those of the wafers running the control recipe. (Due to the proprietary nature of the process and data, we are unable to publish the details of the experimental recipe).

There was an improvement in average x-direction CD, which was 0.2 nm above target for the experimental group contrasted with 0.7 nm above target for the control group. There was also an improvement in average y-direction CD, which was 0.2 nm above target for the experimental group contrasted with 0.4 nm below target for the experimental group.

There was an improvement in yield (number of die yielding per wafer) for the experimental group of about 0.52%. This was not statistically significant. However, there were statistically significant changes in certain yield metrics that are affected by the high-aspect-ratio etch process that was optimized. Two of these metrics reflected failures occurring in individual high-aspect-ratio holes, and therefore a failure of the etch to penetrate to the underlying layer. These metrics were reported as the number of die with the defect. The third metric reflected failures occurring with two adjacent holes, and therefore a failure in etch anisotropy because the two holes have merged somewhere along their profile. This

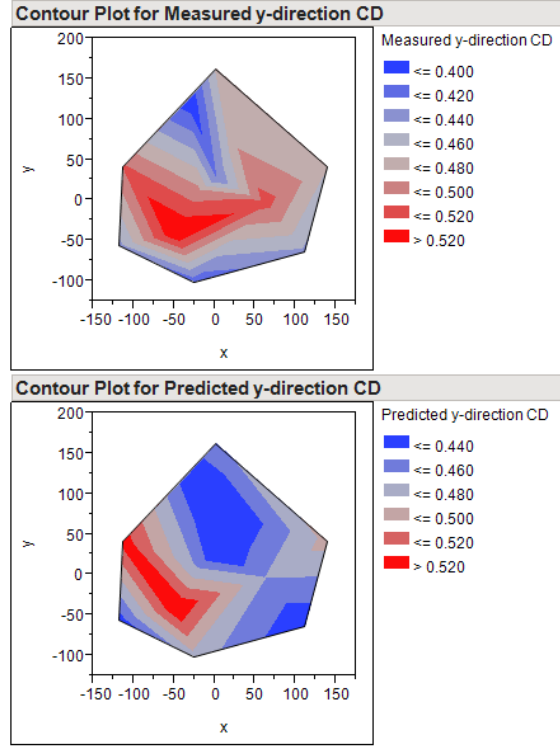


Figure 59: Contour plots using measured and predicted data for the y-direction CD.

metric was reported as the total number of fails per wafer. There was a statistically significant improvement in all three metrics. The p-values for the single-hole failure metrics were 0.0157 and 0.0488. The p-value for the two adjacent holes failure metric was 0.0455 [57].

## 6.4 Discussion

The average model prediction errors of 2.6% and 3.8% for x-direction CD and y-direction CD are reasonably low, and are evidence that the model has captured process behavior. These errors are comparable to other studies, see [35, 58, 59, 60, 61, 62], that have employed neural network to model fabrication processes. The similarity in the contour maps for measured and predicted output variables for the test data set provide further evidence of modeling accuracy. This shows the model can accurately predict CD uniformity across the wafer. Thus the neural network has modeled the high-aspect-ratio etch satisfactorily.

The experimental recipe resulted in improvements in the post-process metrology, thus bringing the CDs for the outlying chambers closer to target. Additionally, the recipe

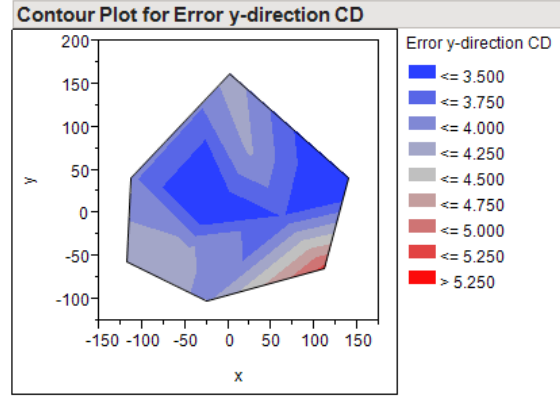


Figure 60: Contour plot of the percentage prediction error for y-direction CD.

brought statistically significant improvements to yield metrics that are affected by the high-aspect-ratio etch. Thus, the genetic algorithm optimization process performed satisfactorily.

The improvement in the CDs measured post-process was small. The product had been in high volume production for enough time that the high-aspect-ratio etch had become tightly controlled in terms of post-process metrology. This makes it difficult to make improvements to the recipe such that process improvements are statistically significant. The method would be more suited to optimize post-process metrology at earlier stages in the development or manufacturing of a semiconductor product.

The new recipe did make statistically significant improvements to yield metrics known to be affected by the etch process. This reveals an opportunity to use the modeling and optimization methodology for yield metrics. The method would be more suited to optimize yield metrics in the later stages of the manufacturing lifecycle of a semiconductor product. This is because yield metrics are more clearly assignable to upstream processes at these later stages. Reference [63] investigated modeling yield metrics from controllable inputs to an upstream process.

The chamber matching method might provide most benefit somewhere in the middle of the two manufacturing stages mentioned above. A product in the ramp up stage would have room for improvement in the post-process metrology, and enough yield that defects

are assignable to upstream processes. Applying the neural network modeling and genetic algorithm optimization method with a product in its ramping stage is explored in Chapter 4.

## **6.5 Summary**

In this study, we modeled a high-aspect-ratio etch process with artificial neural networks and found an optimal process recipe for this etch using genetic algorithms. The aim was to reduce chamber-to-chamber variation in the post-process metrology for the etch. This was achieved. There were improvements in yield metrics affected by the etch as well. The results could be stronger for a fabrication process that was not as mature. Both modeling and optimization methods are automatable and work on a time scale acceptable for run-to-run control, and could form the groundwork for an automatic controller.

## **CHAPTER 7**

### **COMPARISON TO LINEAR REGRESSION MODELS**

#### **7.1 Introduction**

It is worth verifying that neural network models of the fabrication processes have greater prediction accuracy than more established modeling methods. There is an inertia in the uptake of new methods within most manufacturing. This is a result of the opportunity costs associated with the effort of introducing something new. Additionally, there are certain trade-offs to using neural networks. Their black-box nature is commonly cited as the reason for their slow uptake within industry [64]. Using neural networks has also been cited as being more computationally intensive than more traditional statistical modeling methods, but practically, with modern computational speeds, this is not an issue. Empirically demonstrating the superior prediction abilities of neural networks might reduce the unease about their use within the industry.

There have already been several comparisons of the predictions abilities of neural networks to those of other modeling methods [17, 37]. Those results however, are not necessarily generalizable to the various models made in this study. Therefore this section will focus on comparing the predictive performance of linear regression models to those of neural network models, for the processes of concern.

#### **7.2 Method**

Linear regression models were made for the two hardmask etches, and the high-aspect-ratio etch. These models were created using the same training data that was used to train the neural network models. The prediction abilities of the regression models were tested using the same testing data that was used to test the neural network models. This allowed an apples-to-apples comparison of the prediction abilities, as judged by the prediction error, of the two types of models.



The inputs and outputs to linear regression models for hardmask etch 1, hardmask etch 2, and the high-aspect-ratio etch were chosen to be the same as those for the neural network models already created.

The regression models were created in the R software environment [65].

### 7.2.1 Linear regression

Linear regression is an approach to data modeling where the dependent variable ( $y$ ) is modeled as a linear combination of the explanatory or dependent variables ( $x_1, x_2, \dots, x_p$ ). A linear regression model might look like

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon_i, \quad i = 1 \dots n \quad (5)$$

where  $\beta_1 \dots \beta_p$  are the parameters to be estimated,  $n$  is the number of observations in the data set to be modeled, and  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$  is a random error term. The equation in matrix form is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (6)$$

The most commonly used method to fit such a model to a data set is least squares estimation, whereby the parameters ( $\beta_1 \dots \beta_p$ ) are estimated by minimizing the sum of the squared residuals. Residuals are the deviations of the model *estimated* values from the *observed* values.

The least squared estimation method results in a closed-form estimate for  $\boldsymbol{\beta}$ ,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}. \quad (7)$$

### 7.2.2 Hardmask etch 1

The linear regression model for hardmask etch 1 was fitted to the dataset that was used to train the neural network model for hardmask etch 1. This was done so that a fair comparison could be made between the two modeling methods. The linear regression model for

hardmask etch 1 took the form:

$$\begin{bmatrix} \mathbf{x\text{-direction CD (hardmask)}} \end{bmatrix} = \begin{bmatrix} \mathbf{1^T} \\ \mathbf{Wafer\ location, x^T} \\ \mathbf{Wafer\ location, y^T} \\ \mathbf{x\text{-direction CD (photoresist)^T} \\ \mathbf{Inner\ chuck\ temperature^T} \\ \mathbf{Outer\ chuck\ temperature^T} \\ \mathbf{DARC\ etch\ time^T} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} + \epsilon. \quad (8)$$

After having fitted the linear regression model to the training data set, it was tested using the same testing data set used with the neural network model of hardmask etch 1. The prediction errors for the 2 models were then compared

### 7.2.3 Hardmask etch 2

The linear regression model for hardmask etch 2 was fitted and tested following a similar procedure to that for hardmask etch 1. There were two outputs for hardmask etch 2 however, and thus two models were created to predict each separately. The linear regression models for hardmask etch 2 can be represented in matrix form as:

$$\begin{bmatrix} \mathbf{x\text{-direction CD (hardmask)^T} \\ \mathbf{y\text{-direction CD (hardmask)^T} \end{bmatrix} = \begin{bmatrix} \mathbf{1^T} \\ \mathbf{Wafer\ location, x^T} \\ \mathbf{Wafer\ location, y^T} \\ \mathbf{y\text{-direction CD (photoresist)^T} \\ \mathbf{x\text{-direction CD (hardmask-in)^T} \\ \mathbf{BARC\ etch\ time^T} \\ \mathbf{DARC\ etch\ time^T} \\ \mathbf{CH_2F_2\ flow^T} \\ \mathbf{CH_4/HBr\ concentration^T} \end{bmatrix} \begin{bmatrix} \beta_{0x} & \beta_{0y} \\ \beta_{1x} & \beta_{1y} \\ \beta_{2x} & \beta_{2y} \\ \beta_{3x} & \beta_{3y} \\ \beta_{4x} & \beta_{4y} \\ \beta_{5x} & \beta_{5y} \\ \beta_{6x} & \beta_{6y} \\ \beta_{7x} & \beta_{7y} \\ \beta_{8x} & \beta_{8y} \end{bmatrix} + \epsilon. \quad (9)$$

### 7.2.4 High-aspect-ratio etch

The high-aspect-ratio etch, like hardmask etch 2, has two outputs. The linear regression models for the high-aspect-ratio etch can be represented in matrix form as:

$$\begin{bmatrix} \text{x-direction CD (oxide)}^T \\ \text{y-direction CD (oxide)}^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{1}^T \\ \text{Wafer location, } \mathbf{x}^T \\ \text{Wafer location, } \mathbf{y}^T \\ \text{y-direction CD (hardmask)}^T \\ \text{x-direction CD (hardmask)}^T \\ \text{Etch step 1 gas flow}^T \\ \text{Etch step 1 time}^T \\ \text{Etch step 1 gas flow}^T \\ \text{Etch step 1 time}^T \\ \text{Etch step 1 gas flow}^T \\ \text{Etch step 1 time}^T \end{bmatrix}^T \begin{bmatrix} \beta_{0x} & \beta_{0y} \\ \beta_{1x} & \beta_{1y} \\ \beta_{2x} & \beta_{2y} \\ \beta_{3x} & \beta_{3y} \\ \beta_{4x} & \beta_{4y} \\ \beta_{5x} & \beta_{5y} \\ \beta_{6x} & \beta_{6y} \\ \beta_{7x} & \beta_{7y} \\ \beta_{8x} & \beta_{8y} \\ \beta_{9x} & \beta_{9y} \\ \beta_{10x} & \beta_{10y} \end{bmatrix} + \epsilon. \quad (10)$$

## 7.3 Results

This section presents the regression summaries, plots for regression diagnostics, and comparisons to neural networks for each of the regression models. The regression summaries include the estimated values for the model parameters  $(\beta_0, \beta_2, \dots, \beta_p)$ , the statistical significance of these estimates, and measures of the goodness of fit for the model. The plots for regression diagnostics provide a visual way to evaluate the goodness of fit of the models. They are commonly used plots of the residuals. Lastly, the comparison of the regression models to their neural network counterparts was made by both comparing the individual prediction errors, and the average prediction error for each model.

### 7.3.1 Hardmask etch 1

#### 7.3.1.1 Regression summary

The regression summary for hardmask etch 1 can be seen below.

```

> summary(train_results)

Call:
lm(formula = xCDhrdmsk ~ ., data = train)

Residuals:
    Min       1Q   Median       3Q      Max
-45.143 -10.496   0.258  10.846  47.984

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.174e+03   3.991e+02   2.942  0.00348 **
x            -7.534e-09   1.210e-08  -0.623  0.53385
y            -2.130e-08   1.147e-08  -1.856  0.06426 .
xCDphoto      9.203e-01   3.980e-02  23.121 < 2e-16 ***
Innertemp    -1.233e+01   4.421e+00  -2.788  0.00558 **
Outertemp    -1.466e+01   5.496e+00  -2.668  0.00797 **
DARct                NA          NA      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.28 on 362 degrees of freedom
Multiple R-squared:  0.6236,    Adjusted R-squared:  0.6184
F-statistic: 120 on 5 and 362 DF,  p-value: < 2.2e-16

```

The Estimate column lists the fitted  $\beta$  values, or the model parameters for hardmask etch

1. Substituting these values for the  $\beta$ 's gives a regression model that looks like:

$$\widehat{\text{x-direction CD}} = \begin{bmatrix} 1.174e+03 \\ -7.534e-09 \\ -2.130e-08 \\ 9.203e-01 \\ -1.233e+01 \\ -1.466e+01 \end{bmatrix}^T \begin{bmatrix} 1 \\ \text{Wafer location, x} \\ \text{Wafer location, y} \\ \text{x-direction CD (photoresist)} \\ \text{Inner chuck temperature} \\ \text{Outer chuck temperature} \\ \text{DARC etch time} \end{bmatrix}. \quad (11)$$

The F-statistic of 120 with (5, 362) degrees of freedom is the result of an overall F-test for regression where,

$$\begin{aligned} H_0 : \beta_0 = \beta_1, \dots, \beta_p = 0 \\ H_1 : \beta_j \neq 0, \text{ for some } j \in (1, 2, \dots, p). \end{aligned} \tag{12}$$

A p-value of less than  $2.2 \times 10^{-16}$  is small enough to reject the null hypothesis with  $\alpha = 0.01$ . Thus at least one of the parameters must be greater than 0, which means that at least one of the input variables has an effect on the output variable (x-direction CD in the hardmask).

### 7.3.1.2 Regression diagnostics

There are several assumptions made when creating linear regression models. One of these assumptions is the homogeneity of variance of *error* (homoscedasticity). This means that the standard deviation of the observed values of the dependent variable from some true value for that variable, is constant across the range of values of the independent variables. Or stated another way, there is some true function, from which our observed values for the dependent variable have a constant standard deviation. Another assumption is that error is normally distributed with a mean of zero. A third assumption is that the relationship between the dependent variable and the independent variables is linear. The validity of these assumptions can be tested by examining the *residuals*.

The *residuals*, or the deviations of the observed values of the dependent variable from the model predicted values of that variable, provide an estimate for the *errors*. The residuals are analyzed graphically. Figure 61 provides some graphs commonly used for regression diagnostics, and was created using the data from the model for hardmask etch 1.

The “Residuals vs Fitted” graph plots the residuals for each of the data points *versus* the fitted values for the dependent variable, in this case the x-direction CD in the hardmask. This graph allows one to check visually that the residuals are distributed around zero. One would expect also, that this graph could be used to check the assumption of homoscedasticity. If this assumption is true one would expect the residuals to form a rectangular cloud

about the horizontal axis. Residuals are not a perfect proxy for error however. Linear regression tends to fit the endpoints better than the points closer to the center of the domain. Therefore the residuals need to be “standardized” or “studentized”, so that their distribution approximates that of the errors. The “Scale-Location” graph uses the standardized residuals and is used to judge homoscedasticity.

The “Normal Q-Q” plot is used to check the assumption that the random errors are normally distributed. The “Q” stands for quantile. A Q-Q plot is used to compare two distributions by plotting the quantiles of one distribution against the quantiles of the other. If the distributions are similar then the plotted quantiles should fall on the diagonal. In Figure 61, the quantiles for the standardized residuals are plotted against the same quantiles for the standard normal distribution. The observations for the model for hardmask etch 1 lie on the diagonal. Thus it can be concluded that the standardized residuals are normally distributed and therefore the assumption that error is normally distributed is valid.

The “Residuals vs Leverage” plot is used to detect outlying data points that have a strong effect on the regression model. Leverage is a measure of the influence that an observation has on the least squares line. If a point with high leverage is far from the general trend of the data (outlier), then the cause for the outlying nature of such a data point should be investigated, and one must consider removing this data point from the analysis. If this cause is something like a measurement error then the point should be removed from the analysis because of the large influence it has on the results. The three points with high leverage that are outliers are labeled “293”, “295”, “299” in the figure. As seen from the solid red line, that remains relatively flat, the effects of these outliers cancel each other out and no points need to be removed.

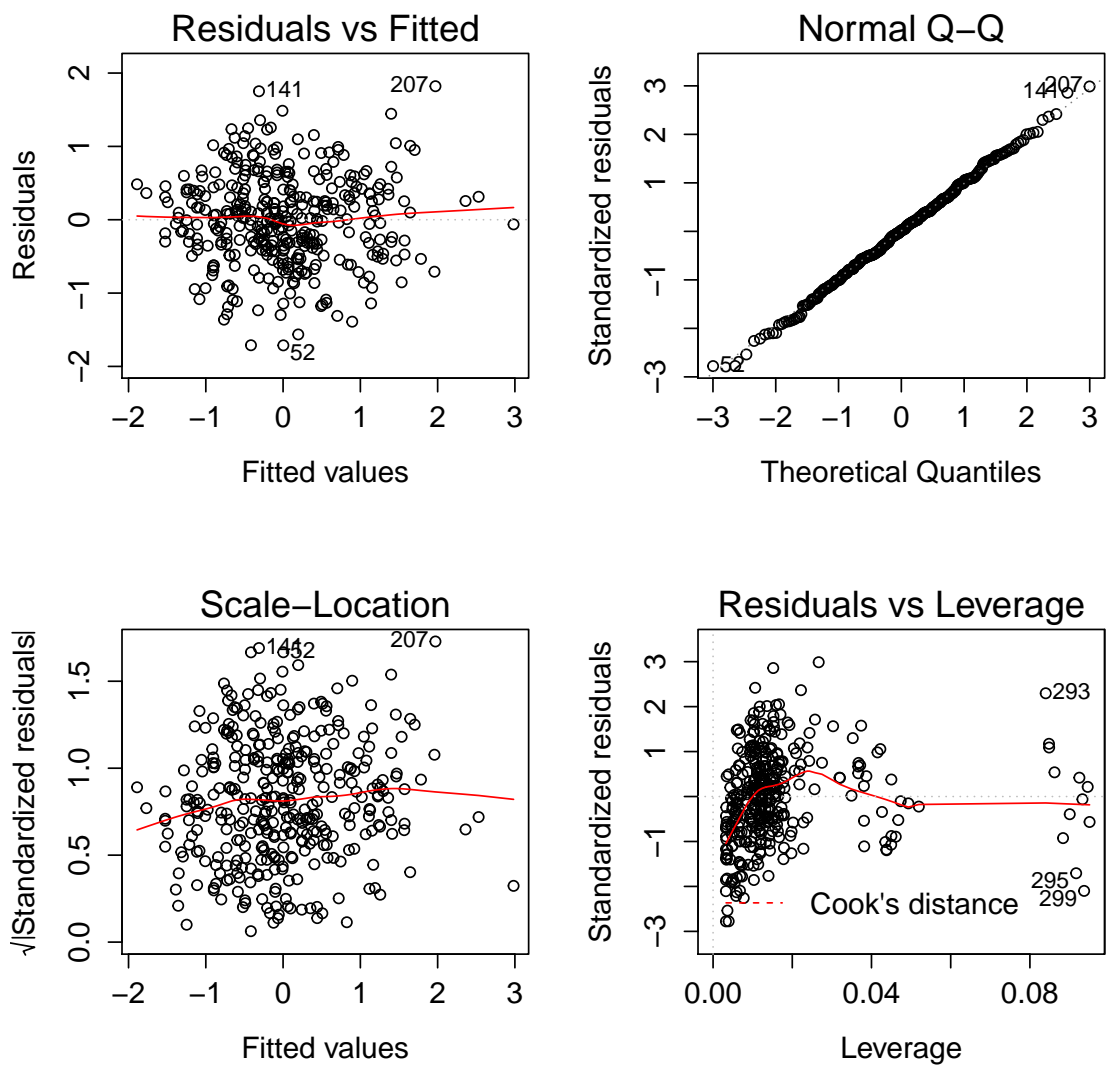


Figure 61: Residual plots for diagnosis of regression for hardmask etch 1

Table 2: Comparison of the average prediction errors for the linear regression and neural network models.

	PERCENTAGE ERROR			
	Linear regression		Neural network	
	x-direction CD	y-direction CD	x-direction CD	y-direction CD
Hardmask etch 1	2.48 %		<b>1.27 %</b>	
Hardmask etch 2	4.92 %	11.36 %	<b>4.42 %</b>	<b>3.06%</b>
High-aspect-ratio etch	<b>3.15 %</b>	4.54 %	3.50 %	<b>4.41%</b>

### 7.3.1.3 Comparison to neural network predictions

The prediction errors for the linear regression model is compared to those for the neural network models in Figure 62. The errors for each model are calculated using the formula

$$\text{Percentage prediction error} = \left| \frac{\text{Observed value} - \text{Model predicted value}}{\text{Observed value}} \right| * 100\%. \quad (13)$$

The x-axis is the percentage prediction error. The pink curve represents the density of the distribution for prediction errors of the linear regression model. The blue curve represents the density distribution of the prediction errors for the neural net model. The blue curve is more concentrated towards the left of the graph than the pink one, indicating that the prediction errors for the neural network models tended to be smaller than those for the linear model. The figure was created with the ggplot2 package [66].

The average prediction errors for the linear regression models and the neural network models for each of the fabrication processes are presented in Table 2. The neural network model has a smaller average prediction error (1.27 %) than the linear regression model (2.48 %) for hardmask etch 1.

### 7.3.2 Hardmask etch 2

Hardmask etch 2 has two outputs. Two linear regression models were created, one to predict the values of each of the process outputs.



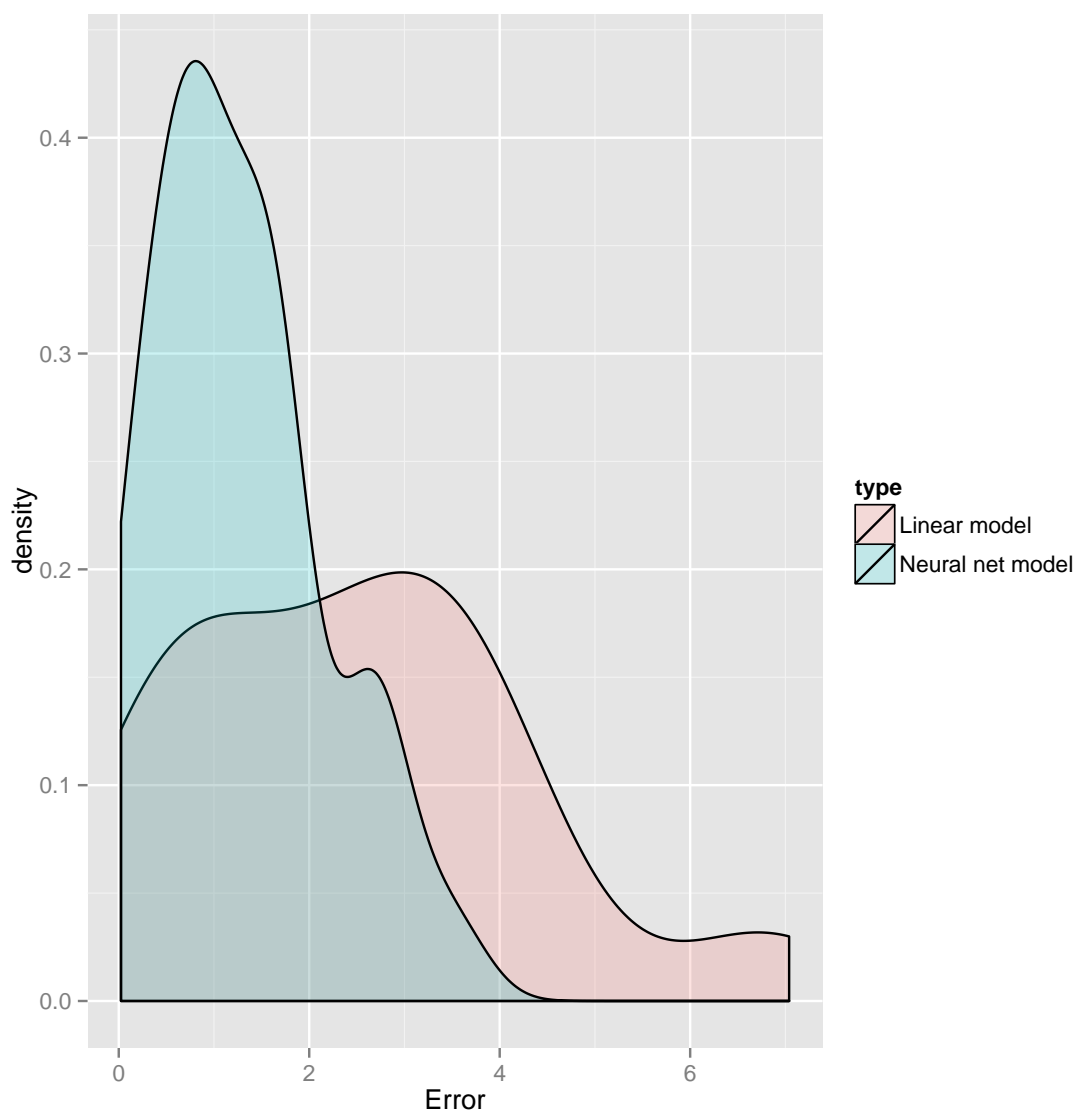


Figure 62: Comparison of the residuals for regression and neural network models of hard-mask etch 1.

### 7.3.2.1 Regression summaries

The regression summary for the model created to predict values of x-direction CD after hardmask etch 2 is presented below.

```
> summary(model_CD1)

Call:
lm(formula = xCDhrdmskout ~ ., data = train_CD1)

Residuals:
    Min       1Q   Median       3Q      Max
-126.570  -16.935    0.634   15.761  101.838

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.595e+02   7.097e+01   3.656 0.000291 ***
x             2.264e-08   1.988e-08   1.139 0.255377
y             2.066e-08   1.993e-08   1.037 0.300435
yCDphoto     1.920e-01   4.778e-02   4.017 7.06e-05 ***
xCDhrdmskin  5.120e-01   4.672e-02  10.957 < 2e-16 ***
BARCt        -4.224e+00   1.504e+00  -2.809 0.005218 **
DARCt        -2.390e+01   1.010e+00 -23.653 < 2e-16 ***
HBr          1.058e+01   5.666e-01  18.679 < 2e-16 ***
CH2F2        9.208e+00   8.479e-01  10.860 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29.41 on 394 degrees of freedom
Multiple R-squared:  0.657,    Adjusted R-squared:  0.6501
F-statistic: 94.35 on 8 and 394 DF,  p-value: < 2.2e-16
```

All of the inputs, except for the x-direction CD and y-direction CD, are significant at a level of  $\alpha = 0.01$ .

The regression summary for the model created to predict values of y-direction CD after hardmask etch 2 is presented below.

```
> summary(model_CD2)

Call:
lm(formula = yCDhrdmsk ~ ., data = train_CD2)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-98.208 -12.998  -0.384   14.993   70.397

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.185e+02  5.518e+01   5.772 1.59e-08 ***
x              1.423e-08  1.546e-08    0.921 0.357726
y             -5.409e-08  1.549e-08   -3.491 0.000535 ***
yCDphoto      3.722e-01  3.715e-02  10.019 < 2e-16 ***
xCDhrdmskin   4.050e-01  3.633e-02  11.149 < 2e-16 ***
BARCt         -1.386e+01  1.169e+00 -11.856 < 2e-16 ***
DARCt         -7.233e+00  7.855e-01  -9.208 < 2e-16 ***
HBr           6.648e+00  4.406e-01  15.089 < 2e-16 ***
CH2F2         4.251e+00  6.593e-01   6.448 3.32e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.87 on 394 degrees of freedom
Multiple R-squared:  0.6457,    Adjusted R-squared:  0.6385
F-statistic: 89.74 on 8 and 394 DF,  p-value: < 2.2e-16

```

All of the inputs, except for the x-direction CD, are significant at a level of  $\alpha = 0.001$ .

### 7.3.2.2 Regression diagnostics

The regression diagnostics for the model made to predict x-direction CD after hardmask etch 2 are presented in Figure 63. There data is more concentrated near the higher end of the range of fitted values, as can be seen in the “Residuals vs Fitted” graphs and the “Scale-Location” graphs. The deviations from the dashed line in the “Normal Q-Q” graph suggest that the distribution of residuals has wider tails than a normal distribution. The “Scale-Location” graph shows that the residuals are larger for fitted values at the lower end of the scale. This violates the assumption of homoscedasticity and is worth investigating.

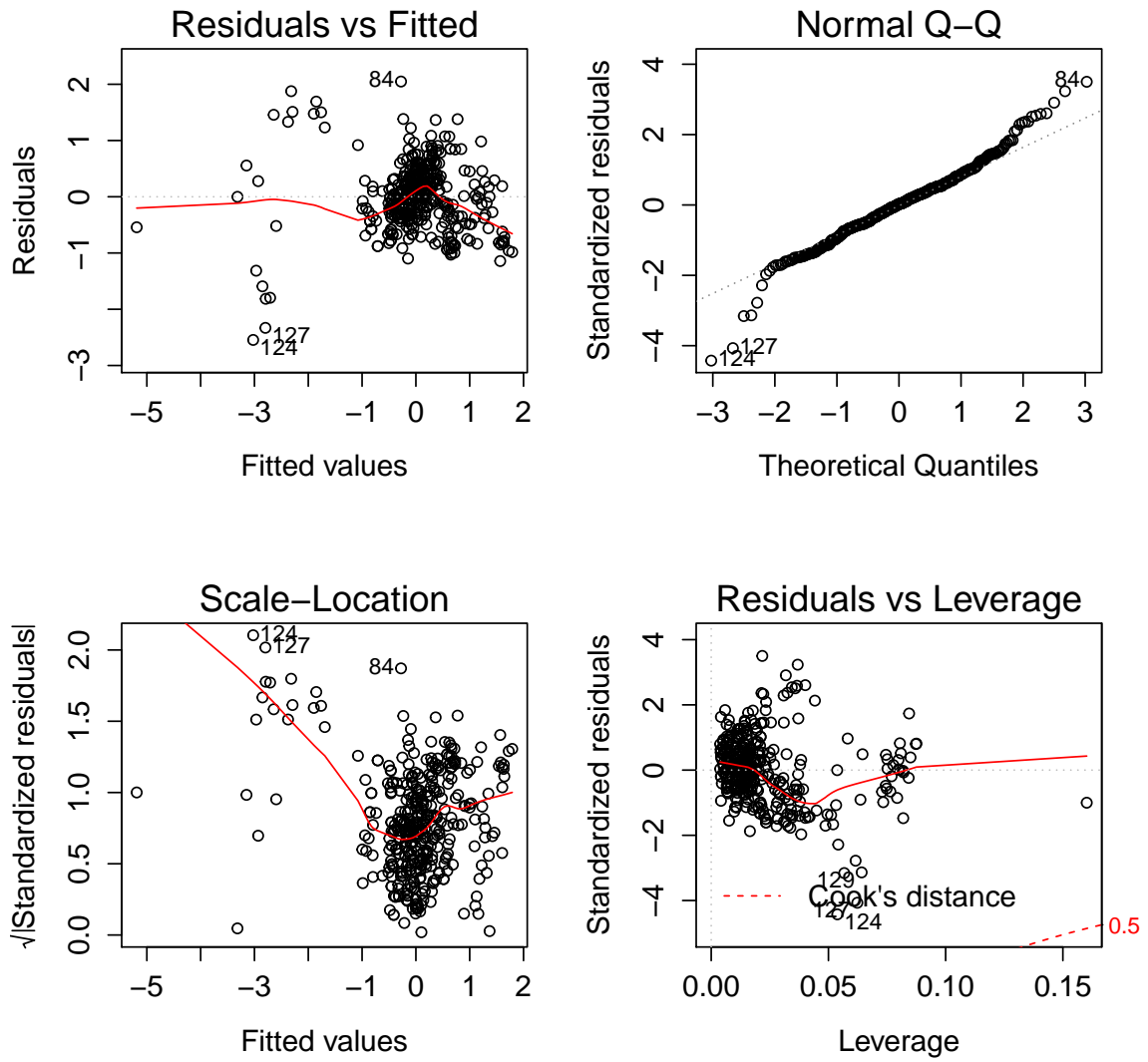


Figure 63: Residual plots for diagnosis of regression model for x-direction CD for hard-mask etch 2.

Figure 64. shows the distribution of the x-direction CDs in the training dataset for the model. The data from index value 122 to 132, the points in red, fall outside the range of the remaining data. The deviation is the result of mis-processing, probably over-etch, of a wafer in the training dataset. The reason for this is not evident in the training data. Perhaps there was an issue with the process chamber. This explains why the “Scale-Location” graph in Figure 63 looks like it does. One can consider removing this mis-processed wafer’s data

from the training dataset before building the linear regression model.

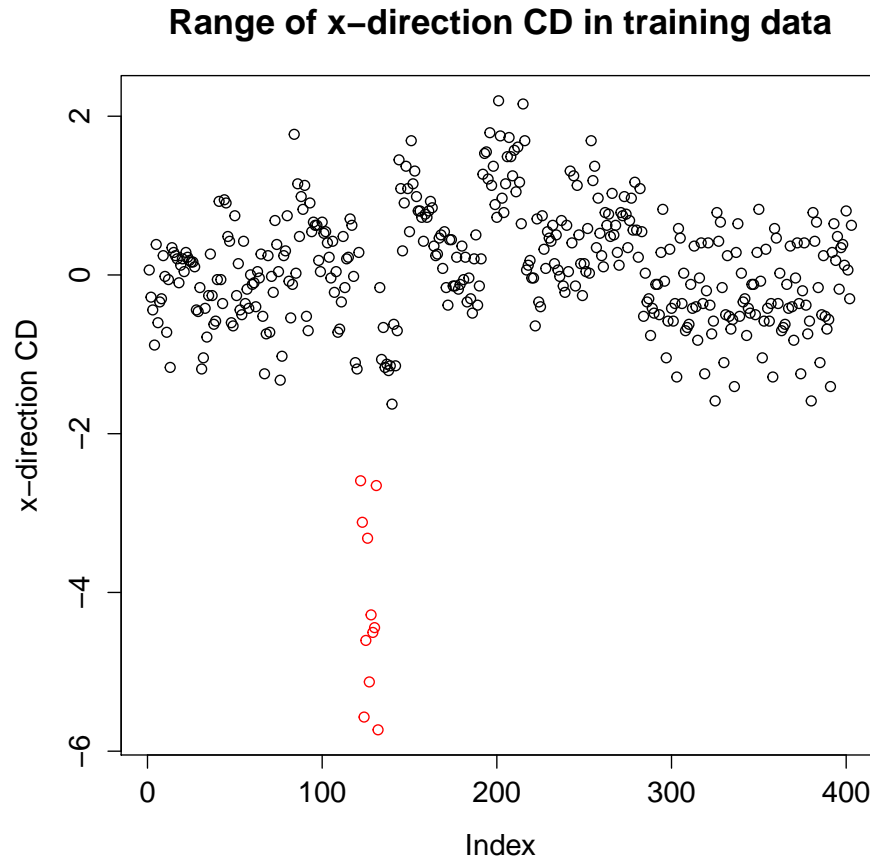


Figure 64: Distribution of the values of x-direction CD in the training dataset.

The regression diagnostics for the model made to predict y-direction CD after hard-mask etch 2 are presented in Figure 65. This set of regression diagnostics is immediately perceived as more favorable than those for x-direction CD. The “Residuals vs Fitted” graph is a rectangular cloud around the Residuals = 0 line, and the data in the “Normal Q-Q” plot remains mostly on the normal line.

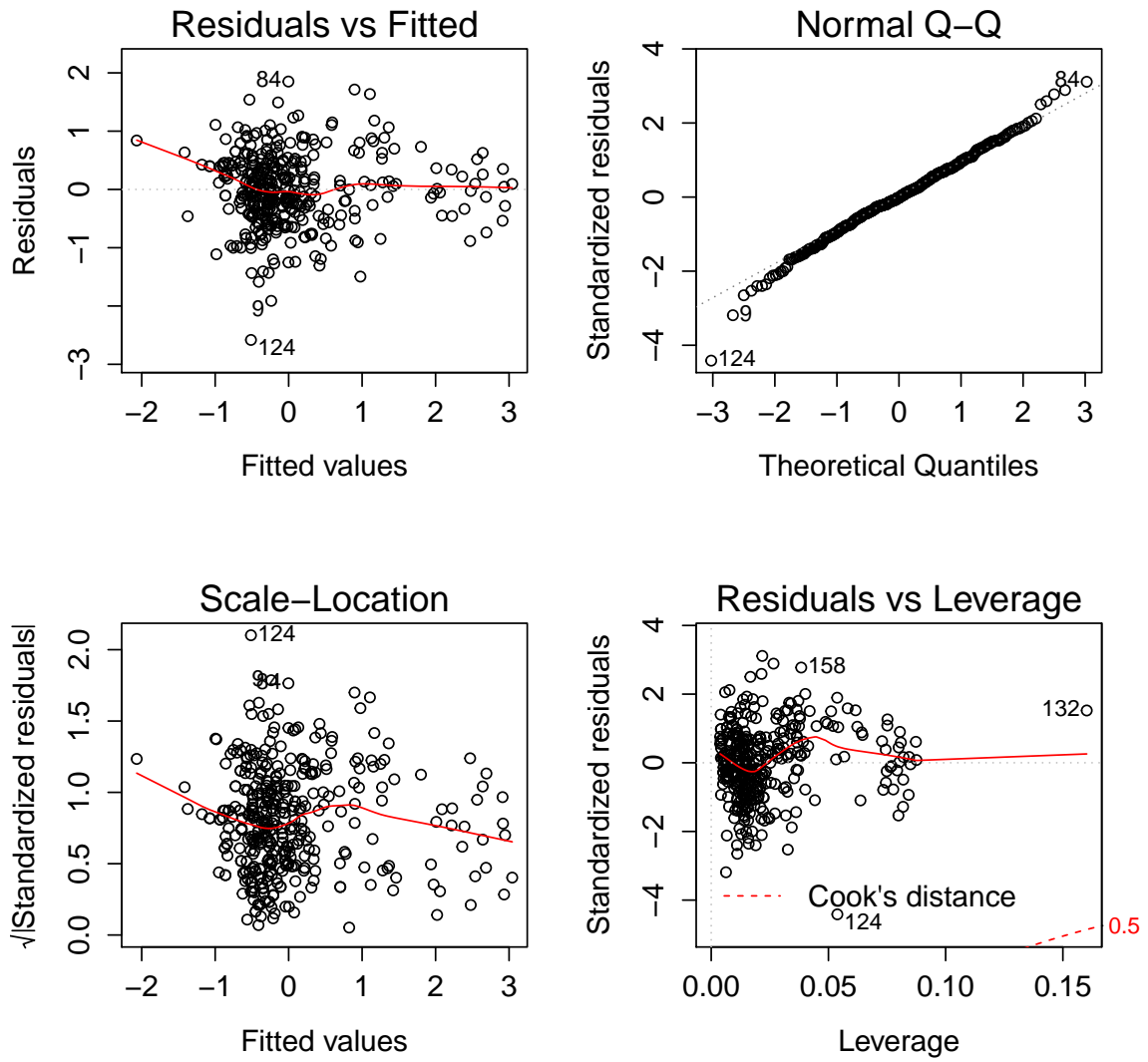


Figure 65: Residual plots for diagnosis of regression model for y-direction CD for hard-mask etch 2.

### 7.3.2.3 Comparison to neural network predictions

The neural network models had better prediction accuracy than the linear regression models as evidenced in Figure 66. For both x-direction and y-direction CDs, the blue curve is higher than the pink on the left of the graph, indicating that percentage prediction error tended to be smaller for the neural network models.

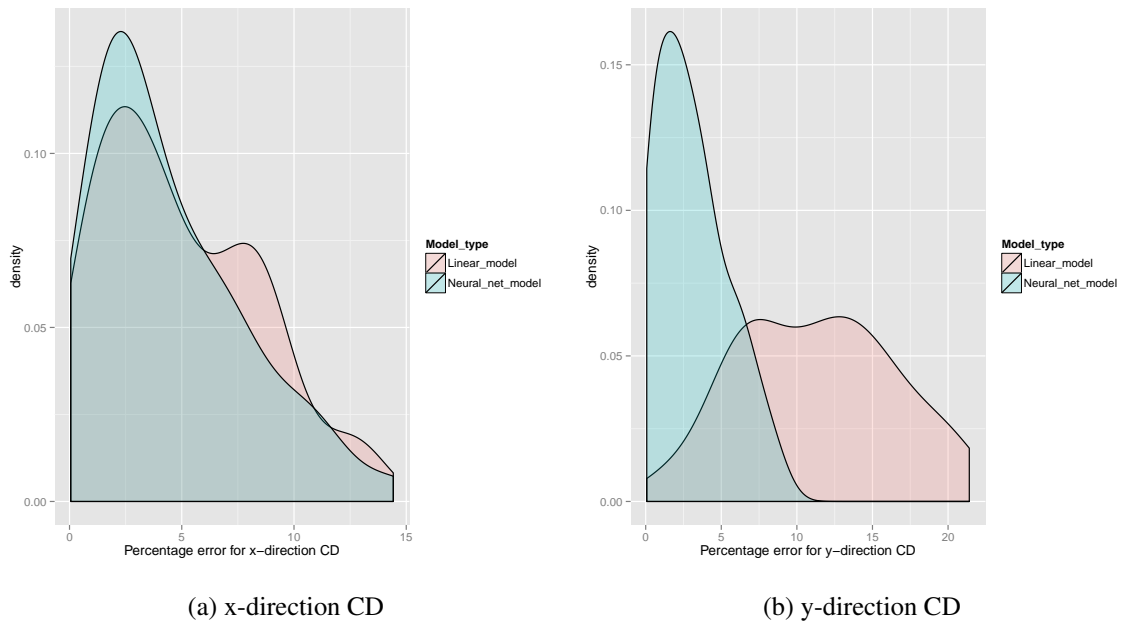


Figure 66: Comparison of the distributions of the residuals for regression and neural network models for the hardmask etch 2

### 7.3.3 High-aspect-ratio etch

The high-aspect-ratio etch also has two outputs, x-direction CD and y-direction CD. Separate linear regression models were created to predict each output.

#### 7.3.3.1 Regression summaries

The regression summary for the model created to predict values of x-direction CD after high-aspect-ratio etch is presented below.

```
> summary(lm11)

Call:
lm(formula = xCDox ~ ., data = tr11)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0071186 -0.0011628 -0.0000062  0.0011437  0.0080069

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.00000000  0.00000000  0.0000000  1.0000000
xCDox         0.00000000  0.00000000  0.0000000  1.0000000
```

```

(Intercept)  3.514e-02  6.106e-03  5.756 8.80e-09 ***
x            1.039e-06  2.190e-07  4.746 2.10e-06 ***
y            1.425e-06  2.125e-07  6.703 2.13e-11 ***
chamber      -1.736e-05  3.412e-06 -5.087 3.70e-07 ***
xCDhrdmsk    1.551e-01  7.113e-03 21.800 < 2e-16 ***
yCDhrdmsk    5.511e-02  4.868e-03 11.320 < 2e-16 ***
ME102        -1.362e-04  2.903e-05 -4.693 2.72e-06 ***
X2sourceavg  -9.451e-05  3.161e-05 -2.990 0.002791 **
X3sourceavg   1.075e-04  3.091e-05  3.478 0.000508 ***
X3sourcestd  -2.309e-04  9.361e-05 -2.466 0.013669 *
X2maginavg   -5.326e-03  1.615e-03 -3.299 0.000974 ***
X3maginavg    5.224e-03  1.752e-03  2.981 0.002875 **
X2maginstd   -4.758e-03  9.209e-04 -5.166 2.42e-07 ***
X3maginstd   -1.203e-02  5.278e-03 -2.279 0.022709 *
X2magoutavg   1.818e-03  2.080e-03  0.874 0.382285
X3magoutavg  -1.880e-04  2.038e-03 -0.092 0.926498
X1bias1std   -3.112e-06  2.785e-06 -1.117 0.263894
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.001767 on 13171 degrees of freedom
Multiple R-squared: 0.1124, Adjusted R-squared: 0.1113
F-statistic: 104.2 on 16 and 13171 DF, p-value: < 2.2e-16

```

The regression summary for the model created to predict values of y-direction CD after high-aspect-ratio etch is presented below.

```

> summary(lm12)

Call:
lm(formula = yCDox ~ ., data = tr12)

Residuals:
    Min       1Q   Median       3Q      Max
-0.0170555 -0.0018830  0.0000115  0.0018757  0.0138306

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.482e-02  1.013e-02   1.463 0.143586
x            1.386e-06  3.634e-07   3.815 0.000137 ***
y            1.393e-06  3.528e-07   3.950 7.86e-05 ***
chamber      -2.435e-05  5.664e-06  -4.298 1.73e-05 ***
xCDhrdmsk    1.820e-01  1.181e-02  15.417 < 2e-16 ***
yCDhrdmsk    1.735e-01  8.081e-03  21.470 < 2e-16 ***

```



```

ME102      -3.196e-05  4.818e-05  -0.663  0.507162
X2sourceavg -1.552e-04  5.246e-05  -2.959  0.003094 **
X3sourceavg  1.911e-04  5.131e-05   3.725  0.000196 ***
X3sourcestd  2.533e-04  1.554e-04   1.630  0.103061
X2maginavg  -7.977e-03  2.680e-03  -2.976  0.002924 **
X3maginavg   8.812e-03  2.909e-03   3.030  0.002452 **
X2maginstd  -4.394e-03  1.529e-03  -2.874  0.004055 **
X3maginstd  -2.649e-02  8.760e-03  -3.023  0.002504 **
X2magoutavg  8.860e-04  3.453e-03   0.257  0.797524
X3magoutavg -5.752e-04  3.383e-03  -0.170  0.865012
X1bias1std  -6.091e-06  4.623e-06  -1.318  0.187685
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.002934 on 13171 degrees of freedom
Multiple R-squared:  0.1094,    Adjusted R-squared:  0.1083
F-statistic: 101.1 on 16 and 13171 DF,  p-value: < 2.2e-16

```

### 7.3.3.2 Regression diagnostics

Figure 67 (x-direction) and Figure 68 (y-direction) present the regression diagnostics for the high-aspect-ratio etch models. There was more data available to create the high-aspect-ratio models as is evident by how dark the clouds of data points are. For both models the residuals are normally distributed about 0, satisfying the assumptions of the regression analysis.

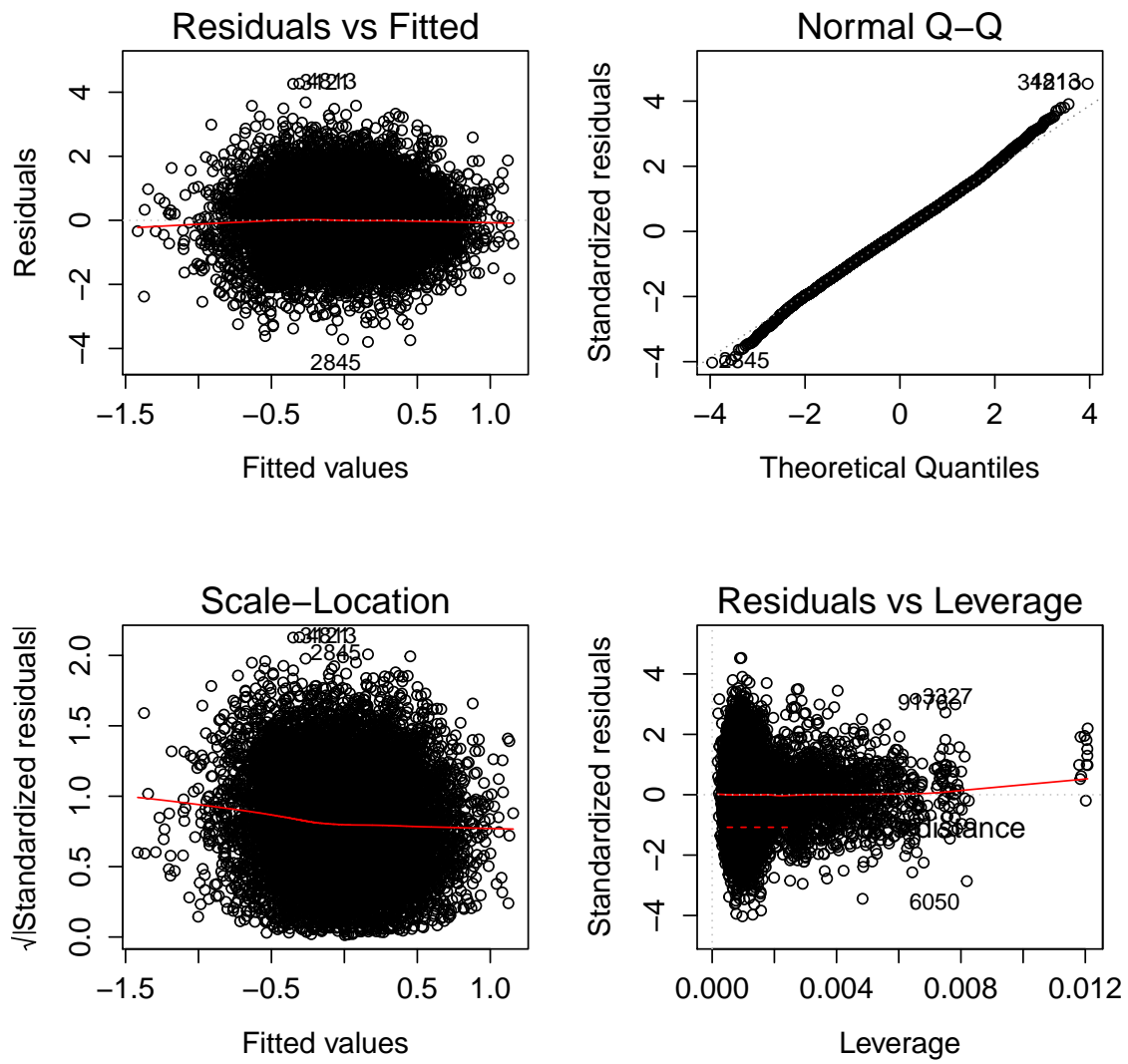


Figure 67: Residual plots for diagnosis of regression model for x-direction CD for high-aspect-ratio etch.

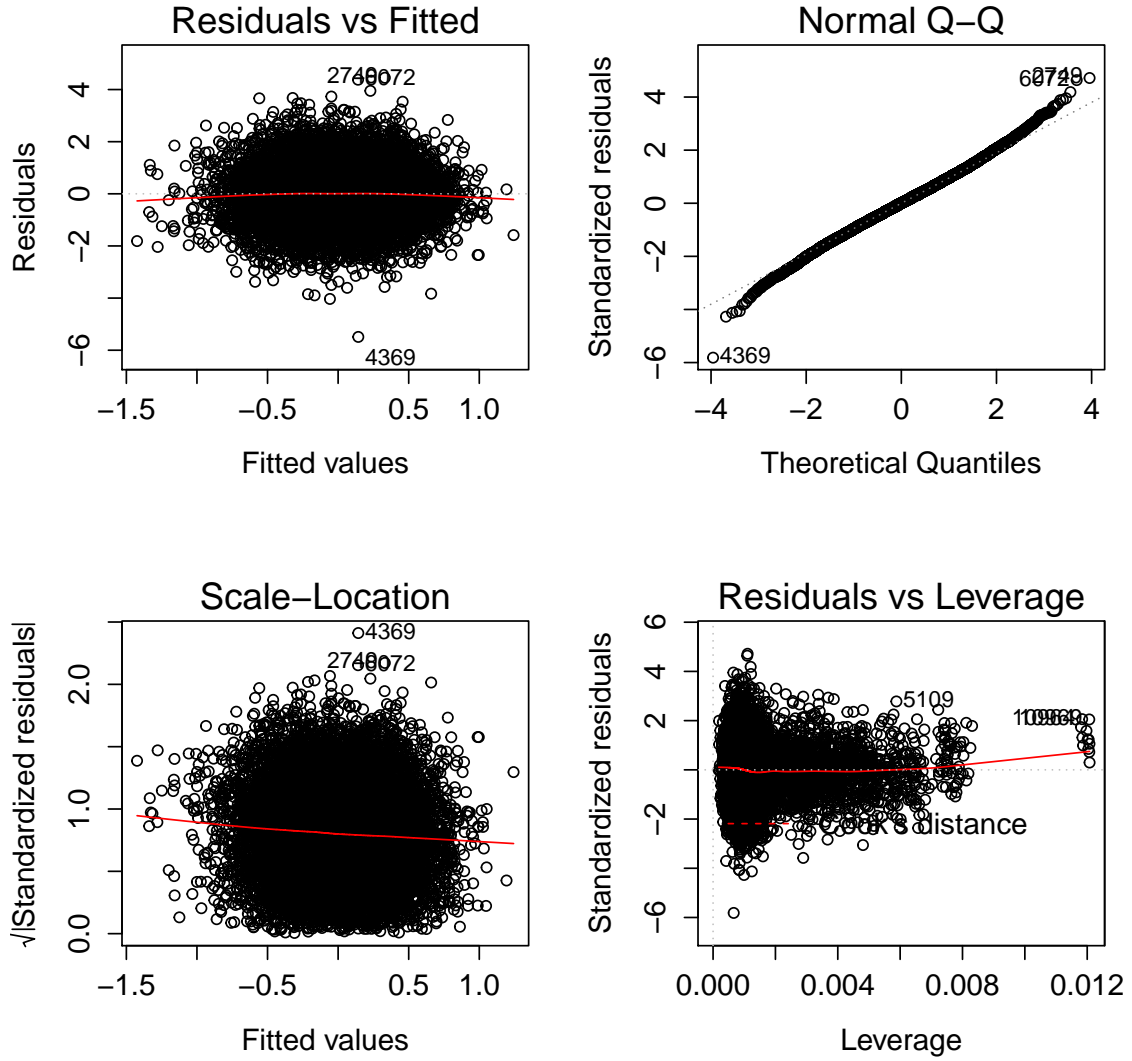


Figure 68: Residual plots for diagnosis of regression model for y-direction CD for high-aspect-ratio etch.

### 7.3.3.3 Comparison to neural network predictions

A comparison of the prediction accuracy of the models to those of neural networks is presented in Figure 69. Once again the pink and blue curves represent the percentage error of the linear regression models, and neural network models respectively. The graph for Figure 69 (a) shows the one result where the linear regression model error was smaller.

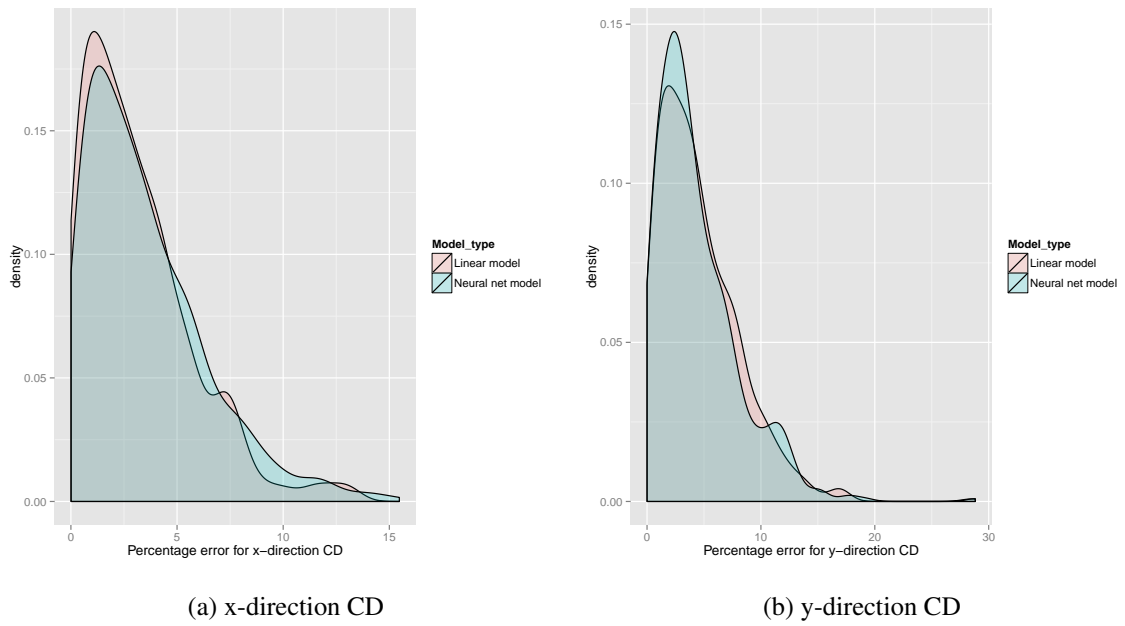


Figure 69: Comparison of the distributions of the percentage prediction error for regression and neural network models for the high-aspect-ratio etch

A contour plot for the distribution of x-direction CDs across the wafer can be seen in Figure 70.

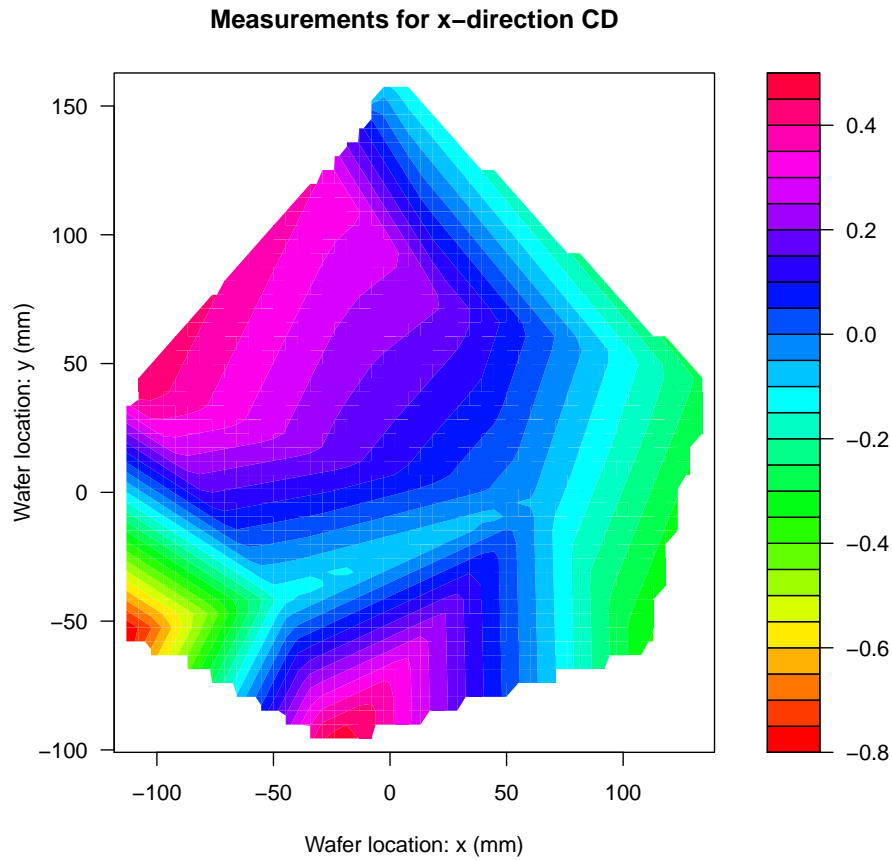


Figure 70: Contour plot of x-direction CDs measured after the high-aspect-ratio etch

Contour plots for the x-direction CDs predicted by the linear regression model and the neural network model can be seen in Figure 71. In comparing these plots of predicted distributions to that of the measured distribution in Figure 70., it is clear that the neural network model presents a closer visual match. Thus although the average prediction error is higher for the neural network predictions, it has better captured the way x-direction CD is distributed across the wafer.

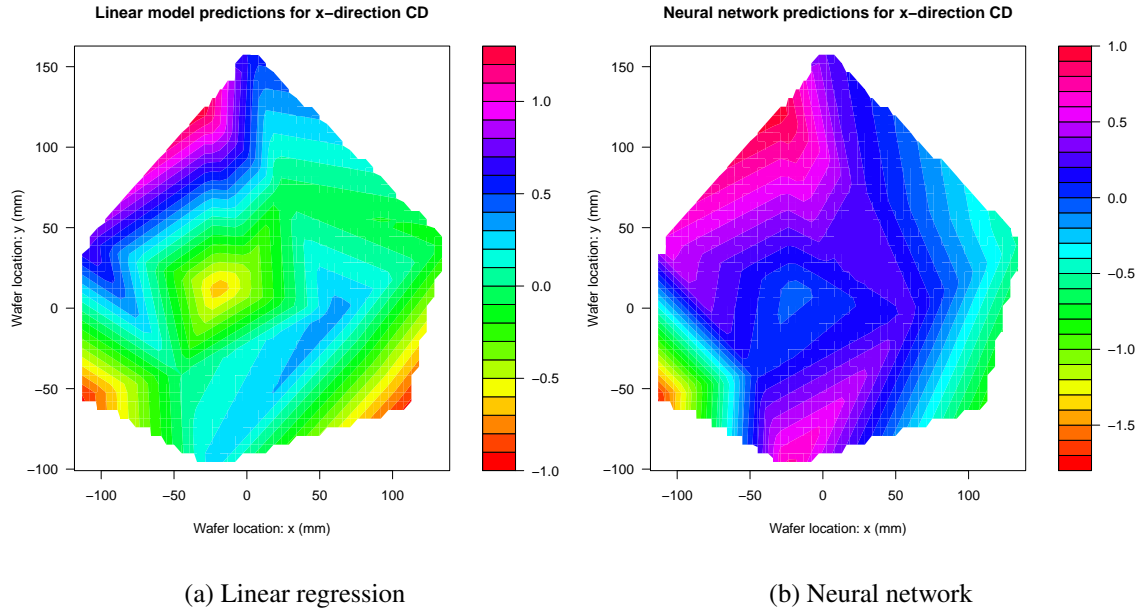


Figure 71: Contour plots for the x-direction CDs after the hardmask etch as predicted by the linear regression and neural network models.

## 7.4 Discussion

The neural network models performed better, in terms of average percentage prediction error, than the linear regression models. The exception was the model for x-direction CD after the high-aspect-ratio etch.

It might be possible to bring the linear regression models performance up to the level of the neural network's with better feature selection and a thorough investigation of interactions between input variables. With neural networks, however, the creations of sophisticated non-linear models is relatively simple.

## CHAPTER 8

### COMPARISON TO SUPPORT-VECTOR-REGRESSION MODELS

#### 8.1 Introduction

Support vector machines (SVM) are a set of models and algorithms that can be used for both classification and regression. They were first developed by Vapnik in 1963 [67], and attracted greater interest within the research community in the 1990s. They are considered by many to be among the best supervised learning algorithms [68]. Thus it is a useful exercise to compare the performance of the neural network models created in previous chapters to that of SVMs created using the same data.

The impetus for their creation was the problem of the bias-variance trade-off [69]. Prediction error can be decomposed into a noise component, a bias component, and a variance component. The best predictor performance is achieved, for a given training dataset, by balancing the bias and variance components of the prediction error. This is achieved by balancing the prediction accuracy on the training data and the “capacity” of the predictor.

The concept of “capacity” is central to the study of SVMs. Capacity is the ability of a learning machine to learn any training dataset without error. For a given dataset, using a predictor with a large “capacity” reduces the bias component of error because the predictor is more able to capture the complexities of the function from which the training data is derived. But this also increases the variance component of the error as model predictions become more dependent on the training dataset (over-fitting). Conversely, a predictor with a small capacity will increase the bias component of error, and reduce the variance component possibly capturing too little of the peculiarities of the data set (under-fitting).

##### 8.1.1 Support vector regression

Although originally developed for classification tasks, with a particular focus on OCR (optical character recognition) [70], the support vector algorithm has been modified to perform

regression.

Suppose training data looks like:  $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset \mathbf{X} \times \mathbb{R}$ , where  $\mathbf{X} \in \mathbb{R}^d$ . Thus the input is a vector of dimension  $d$  and the output is a scalar, and there are  $l$  observations. In the simple linear case the function to be parameterized is

$$f(x) = \langle w, x \rangle + b, \text{ with } w \in \mathbf{X} \text{ and } b \in \mathbb{R}. \quad (14)$$

The objective/cost function needs to balance the bias and variance components of the error. This is achieved in two ways. The first is the use of an  $\epsilon$  tube. Any prediction that is less than a distance of  $\epsilon$  away from the respective measured value does not affect the parameterization of the model, thus preventing overfitting. The second is by minimizing the norm of  $w$ . Thus the parameterization problem can be posed as

$$\text{minimize} \quad \frac{1}{2} \|w\|^2 \quad (15)$$

$$\text{subject to} \quad |y_i - \langle w, x_i \rangle - b| \leq \epsilon, \quad (16)$$

based on the assumption such a function exists. Slack variables  $\xi, \xi^*$  were introduced to the algorithm in [71], to extend its application to problems where this assumption may not be true. The parameterization problem then becomes

$$\text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=0}^l (\xi_i + \xi_i^*) \quad (17)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ y_i - \langle w, x_i \rangle - b \geq \epsilon - \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (18)$$

where  $C$  is a user manipulated variable to control the bias-variance trade-off.

This is a quadratic programming problem, whereby a quadratic function has several linear constraints, and for which there exists a foundation of mathematical theory to provide a solution [70].

Support vector regression can be further extended to solve nonlinear problems using the ‘kernel trick.’ Kernel functions are used so that the algorithm can operate in very



high dimensional, possibly infinite dimensional, spaces without having to actually map the training vectors to that space. The RBF kernel was used in this work:

$$e^{-\gamma|x-x'|^2} \quad (19)$$

where  $\gamma$  is another user manipulated parameter.

## 8.2 Method

Support vector regression models were made for the two hardmask etches and the high-aspect-ratio etch. These models were created using the same training data that was used to train the neural network models. The prediction abilities of the support vector regression models were tested using the same testing data that was used to test the neural network models. This allowed an apples-to-apples comparison of the prediction abilities, as judged by the prediction error, of the two types of models.

The inputs and outputs to the support vector regression models for hardmask etch 1, hardmask etch 2, and the high-aspect-ratio etch were chosen to be the same as those for the neural network models already created. The models were created and trained using the Python library scikit-learn [72]. Hyper-parameters  $C$  and  $\gamma$  are selected via a grid search.

## 8.3 Results

### 8.3.1 Hardmask etch 1

The graph of residuals *versus* fitted values is presented in Figure 72. The residuals appear to be normally distributed around zero across the range of fitted values. Therefore this does not negate the assumption of homoscedasticity.

Contour plots for measured and support-vector-machine predicted data are presented in Figure 73. One can see that the contour plots are very similar. There is an area of low CD, represented by blue, on the left half of the wafer and an area of high CD in red on the lower right part of the wafer.

The contour plot for absolute percentage prediction error is presented in Figure 74. It

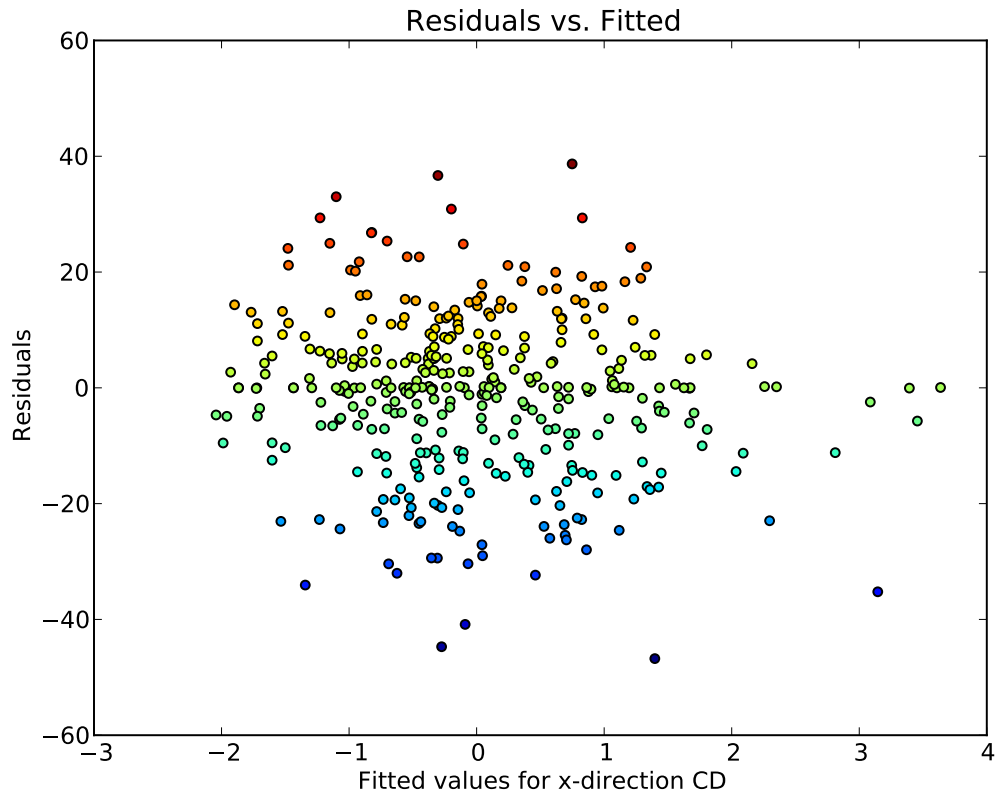


Figure 72: Plot of the residuals *versus* the fitted values for the dependent variable, for each of the data points in the training dataset.

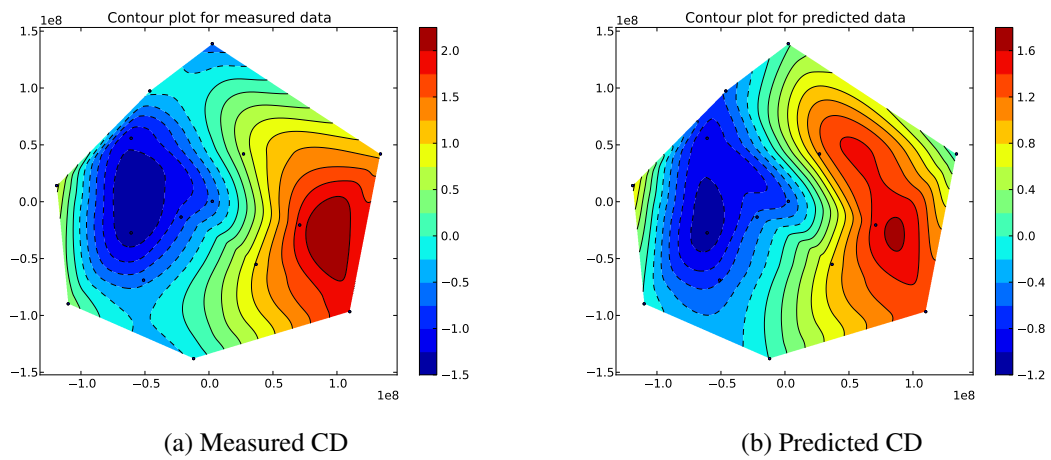


Figure 73: Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm.

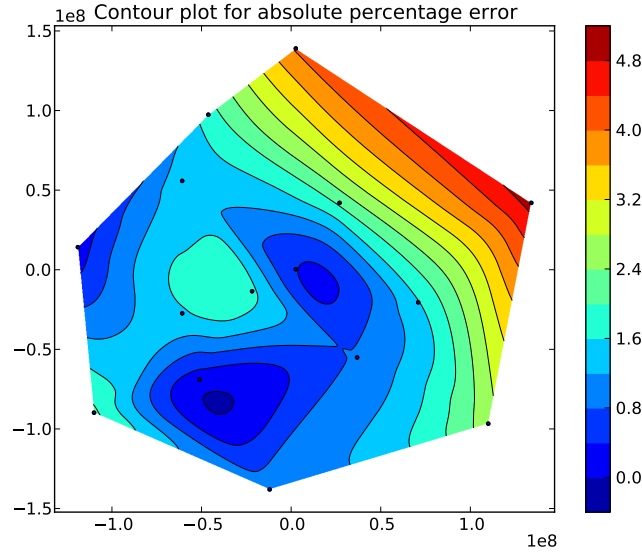


Figure 74: Contour plot for absolute percentage error of the support-vector-regression predictions.

can be seen that most predictions had errors of less than 5%. Figure 75 presents a plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for hardmask etch 1. The pink curve represents neural network error and the blue the support vector regression error. It can be seen that in this instance the neural network outperformed the support vector machine as its prediction error is more concentrated near zero.

### 8.3.2 Hardmask etch 2

The graph of residuals *versus* fitted values for x-direction CD for hardmask etch 2 is presented in Figure 76. The model did a good job of predicting the many outliers on the lower end of the range of fitted values. The variance of the residuals does seem to be slightly larger when normalized x-direction CD is 0. This might indicate a slight bias in the model. This can be corrected by increasing the  $\gamma$  hyper-parameter [73].

Contour plots for measured and support-vector-machine predicted data for x-direction CD are presented in Figure 77. Again, it can be seen that the contour plots are very similar, with areas of high CD at the center and bottom of the wafer, and the lower CDs spread

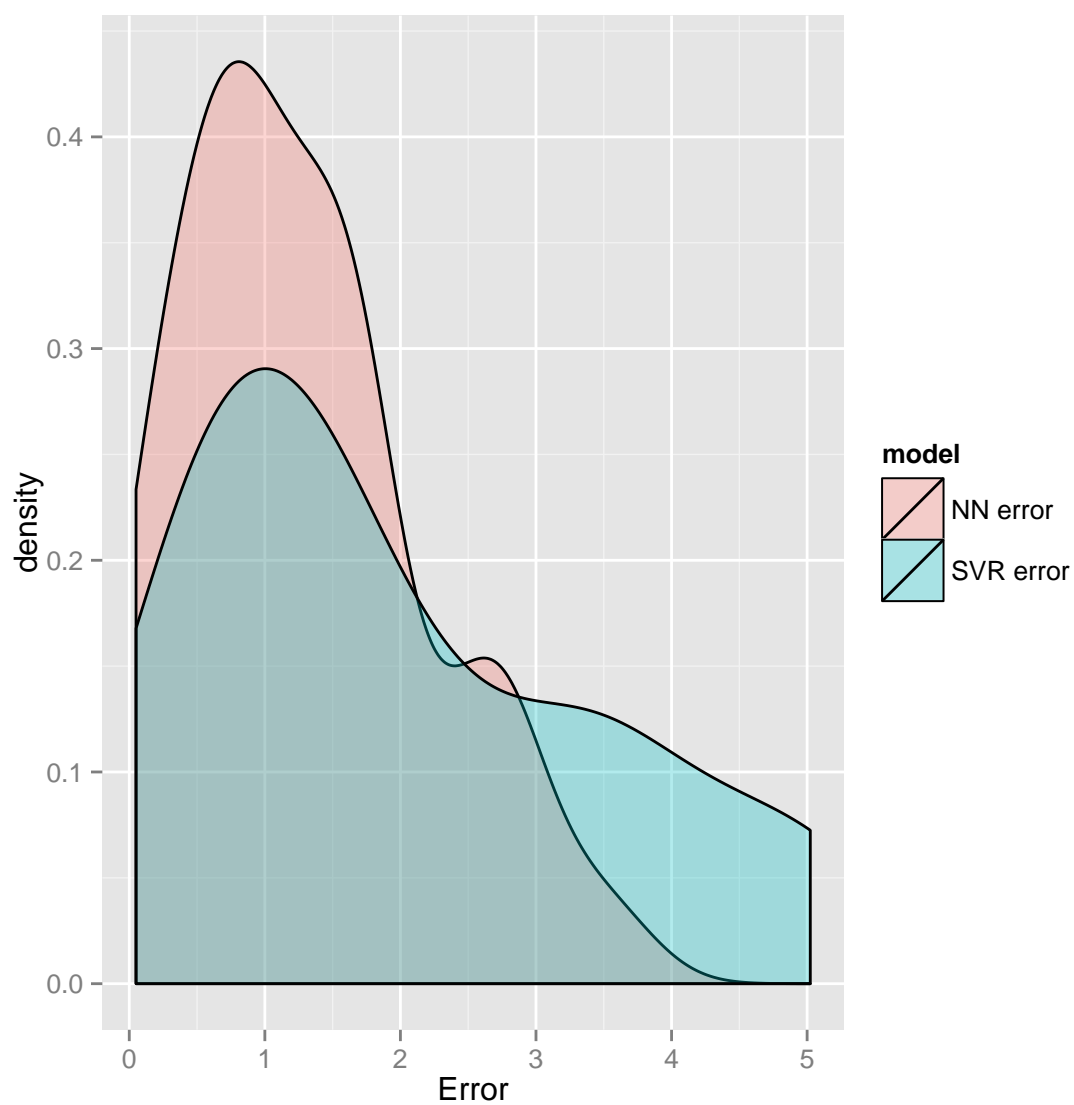


Figure 75: Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for hardmask etch 1.

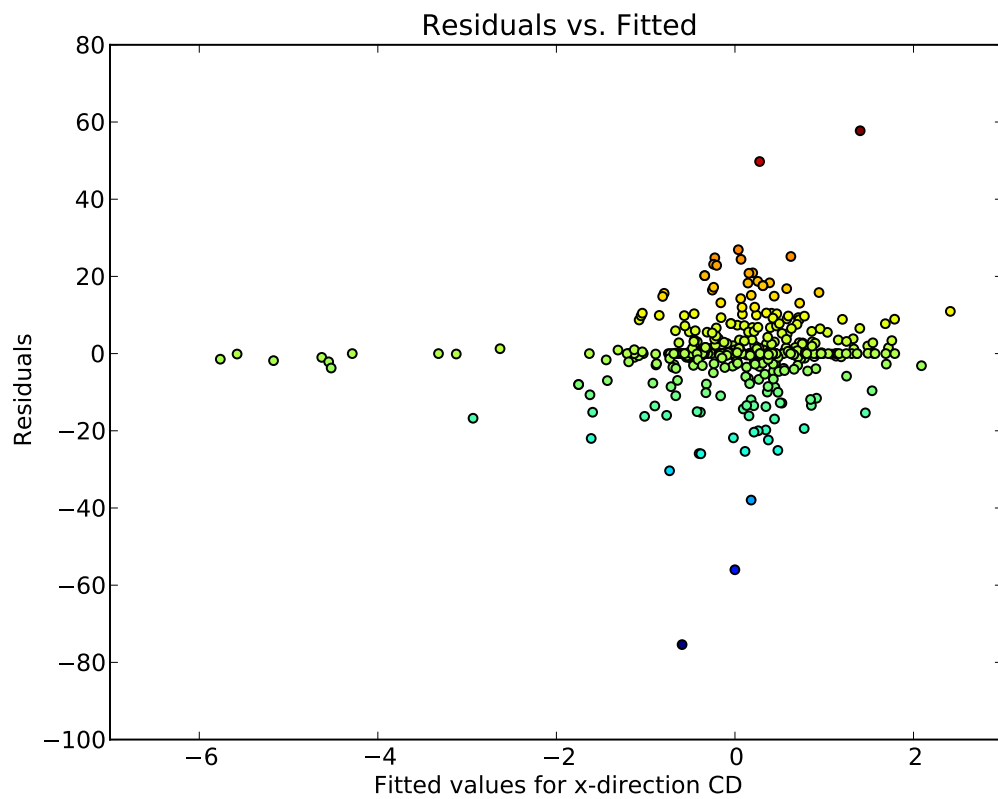


Figure 76: Plot of the residuals *versus* the fitted values for the dependent variable, for each of the data points in the training dataset.

around them. This indicates that the support vector machine was able to accurately model the uniformity of CD across wafer.

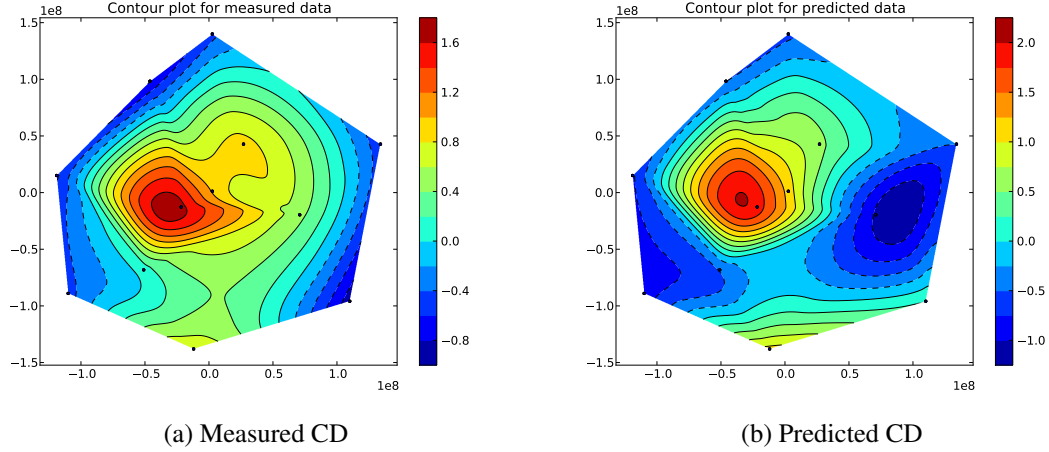


Figure 77: Comparison of measured (left) and predicted (right) output for x-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm.

The contour plot for absolute percentage prediction error for x-direction CD in Figure 78. The error is below 10% across the wafer and below 5% for about three-fourths of the wafer.

Figure 79 presents a plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for hardmask etch 2. The pink curve represents neural network error and the blue the support vector regression error. It can be seen that in this instance the support vector machine outperformed the neural network as its prediction error is more concentrated near zero.

The graph of residuals *versus* fitted values for y-direction CD is presented in Figure 80. The variance of the residuals seems higher for fitted values near 1.

Contour plots for measured and support-vector-machine predicted data for y-direction CD are presented in Figure 81. Again it can be seen that the contour plots are very similar, with areas of high CD at the center and bottom of the wafer, and the lower CDs spread around them. This indicates that the support vector machine was able to accurately model

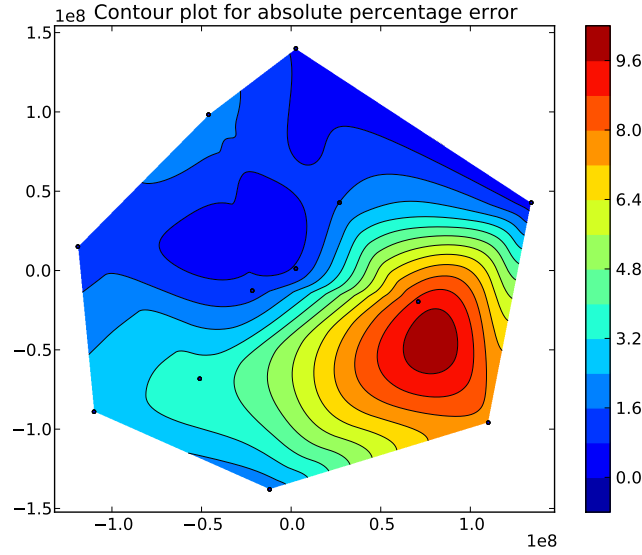


Figure 78: Contour plot for absolute percentage error of the support-vector-regression predictions.

the uniformity of CD across wafer.

The contour plot for absolute percentage prediction error for y-direction CD is presented in Figure 82. The prediction errors remained below 4 % across the wafer. Figure 83 presents a plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for hardmask etch 2. The pink curve represents neural network error, and the blue the support vector regression error. It can be seen that in this instance too, the support vector machine outperformed the neural network as its prediction error is more concentrated near zero.

Table 3 provides a comparison of the mean absolute prediction error for the neural network and support vector regression models created.

### 8.3.3 High-aspect-ratio etch

The graph of residuals *versus* fitted values for x-direction CD for high-aspect-ratio etch is presented in Figure 84. The variance of the residuals remains constant through the range of fitted values, as is witnessed by rectangular cloud of observations.

Contour plots for measured and support-vector-machine predicted data for x-direction

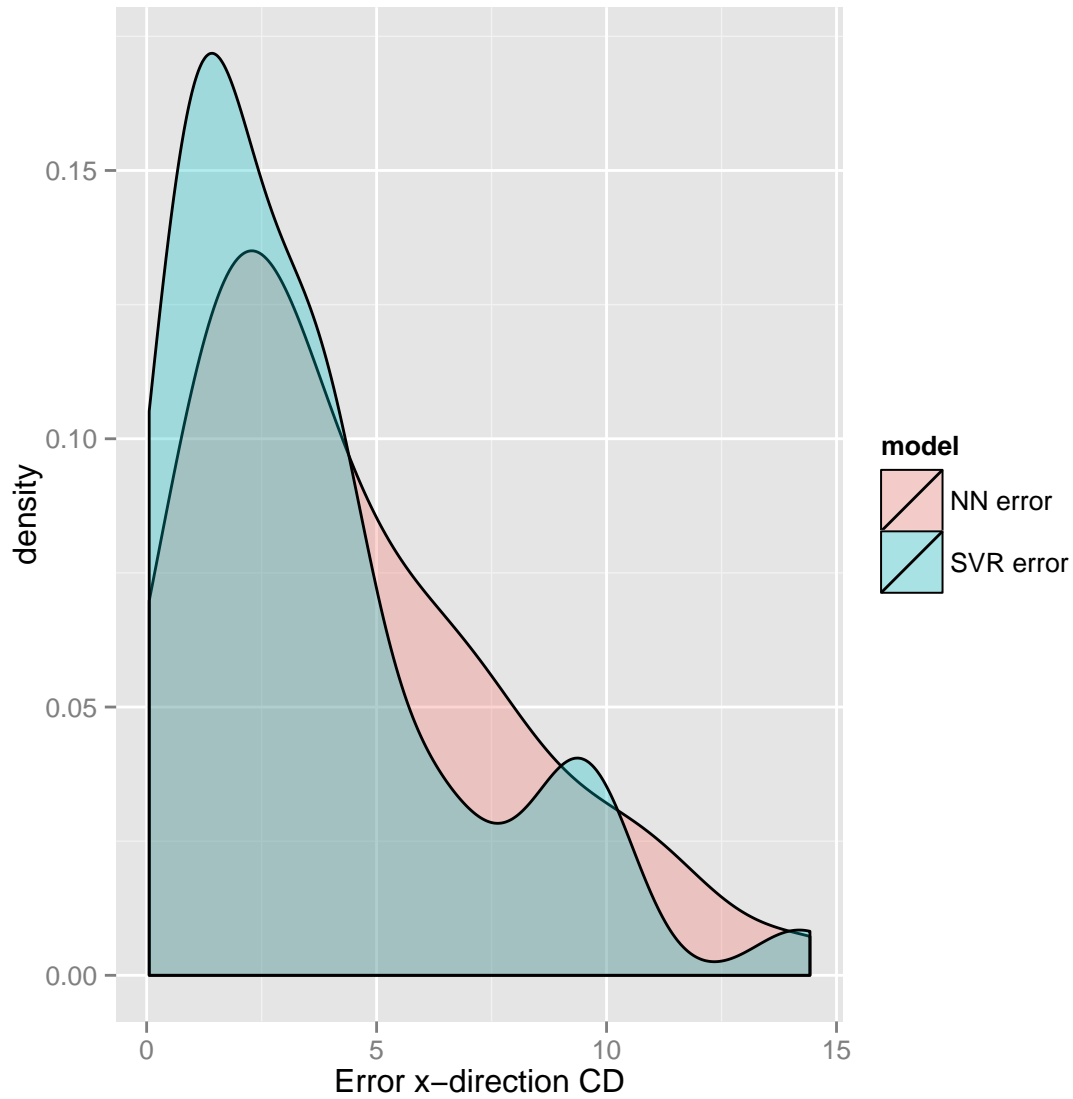


Figure 79: Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for hardmask etch 2.

Table 3: Comparison of the mean absolute percentage prediction errors for neural network and support vector regression models.

	Neural network	Support vector regression
Hardmask etch 1	<b>1.2787%</b>	1.9904%
Hardmask etch 2 x-CD	4.4235%	<b>3.5695%</b>
Hardmask etch 2 y-CD	3.0641%	<b>2.2015%</b>
HAR etch x-CD	3.5016%	<b>3.0628%</b>
HAR etch y-CD	4.4060%	<b>4.3850%</b>



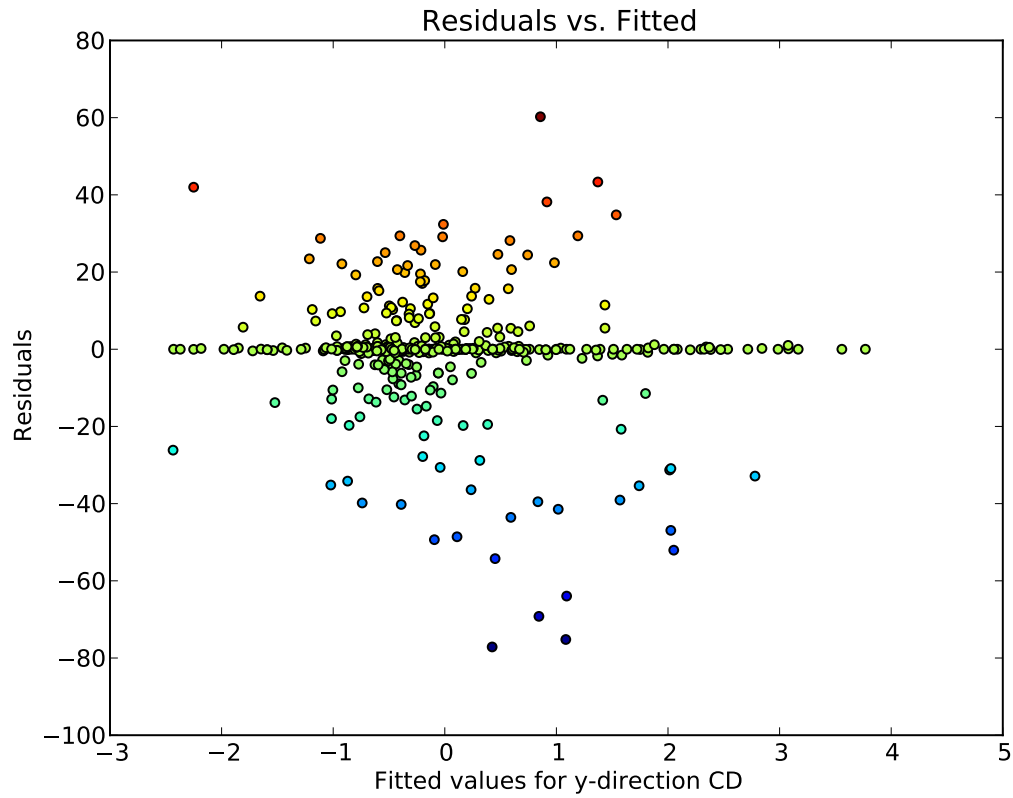


Figure 80: Plot of the residuals *versus* the fitted values for the dependent variable, for each of the data points in the training dataset.

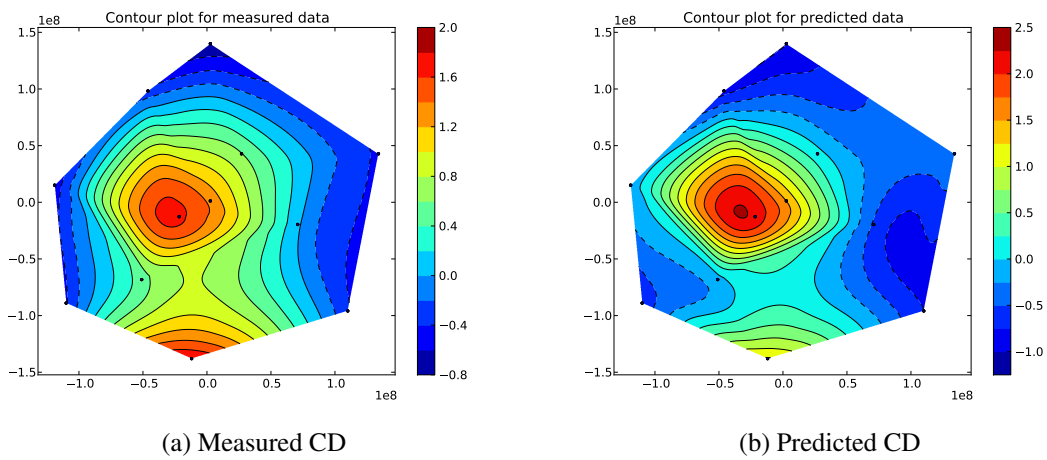


Figure 81: Comparison of measured (left) and predicted (right) output for y-direction CD in the hardmask. The x-axis and y-axis represent wafer location in mm.

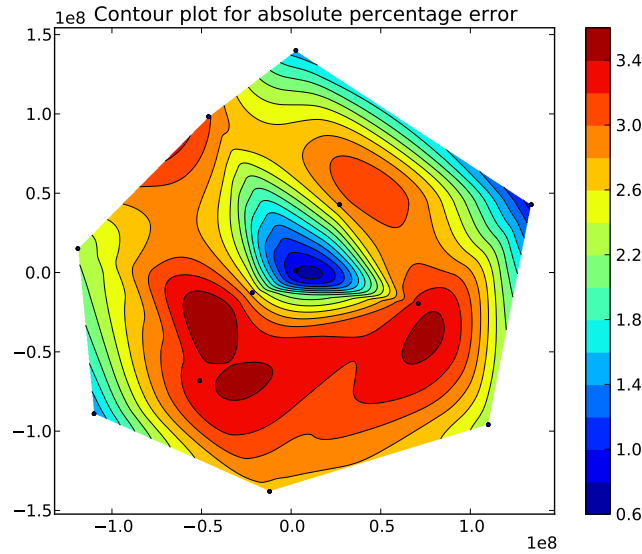


Figure 82: Contour plot for absolute percentage error of the support-vector-regression predictions.

CD for the high-aspect-ratio etch are presented in Figure 85. The plots are not as similar in this instance as they were for the models presented earlier in this chapter.

Figure 86 presents a plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for the high-aspect-ratio etch. The pink curve represents neural network error and the blue the support vector regression error. It can be seen that in this instance too, the support vector machine outperformed the neural network as its prediction error is more concentrated near zero.

The graph of residuals *versus* fitted values for y-direction CD for high-aspect-ratio etch is presented in Figure 87. The variance of the residuals remains constant through the range of fitted values, as is witnessed by rectangular cloud of observations.

Contour plots for measured and support-vector-machine predicted data for y-direction CD for the high-aspect-ratio etch are presented in Figure 88. Again these plots are not as similar as for the hardmask etches. There are some similarities between measured and predicted data however. In both contour plots the lower CDs run along the edge of the

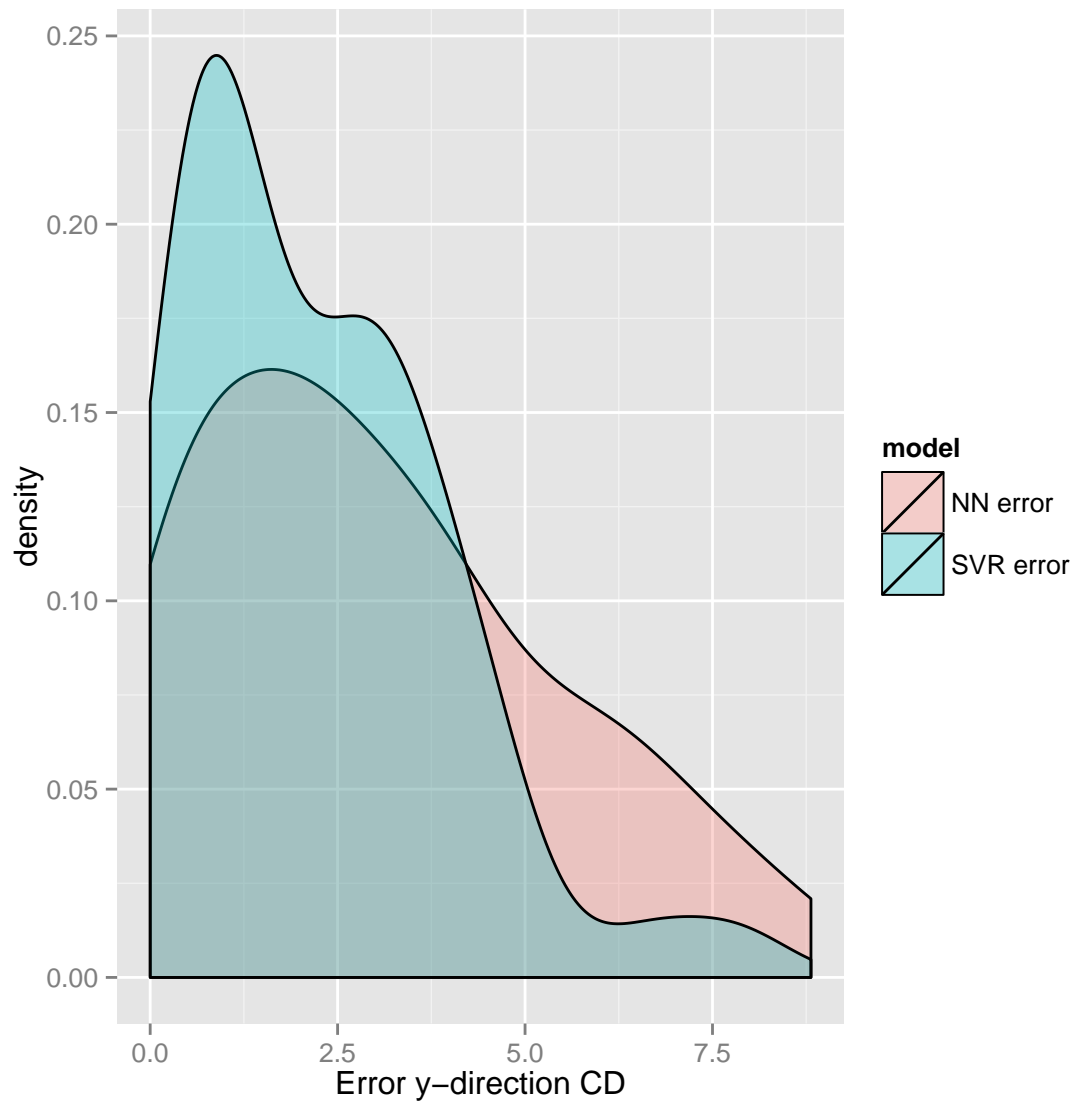


Figure 83: Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for hardmask etch 2.

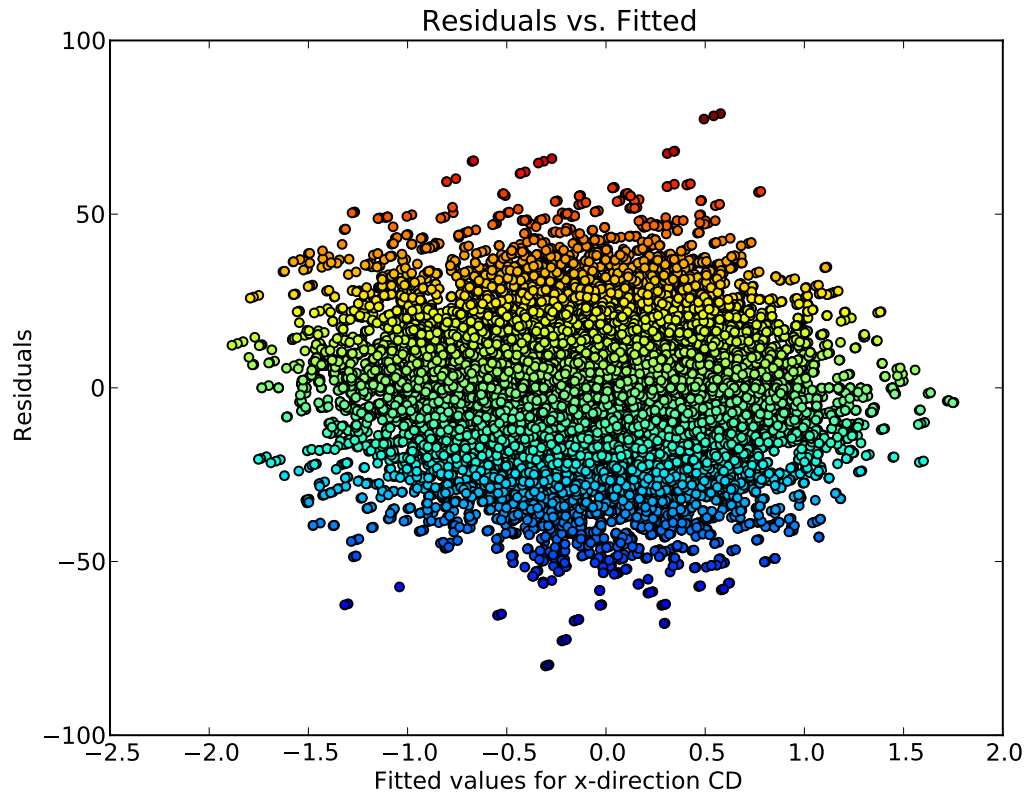


Figure 84: Plot of the residuals *versus* the fitted values for the dependent variable, for each of the data points in the training dataset.

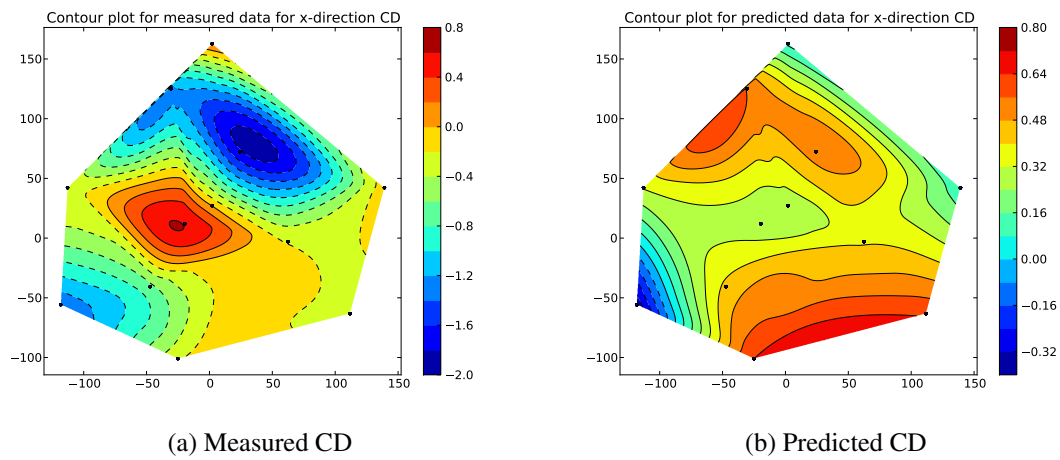


Figure 85: Comparison of measured (left) and predicted (right) output for x-direction CD in the oxide layer. The x-axis and y-axis represent wafer location in mm.

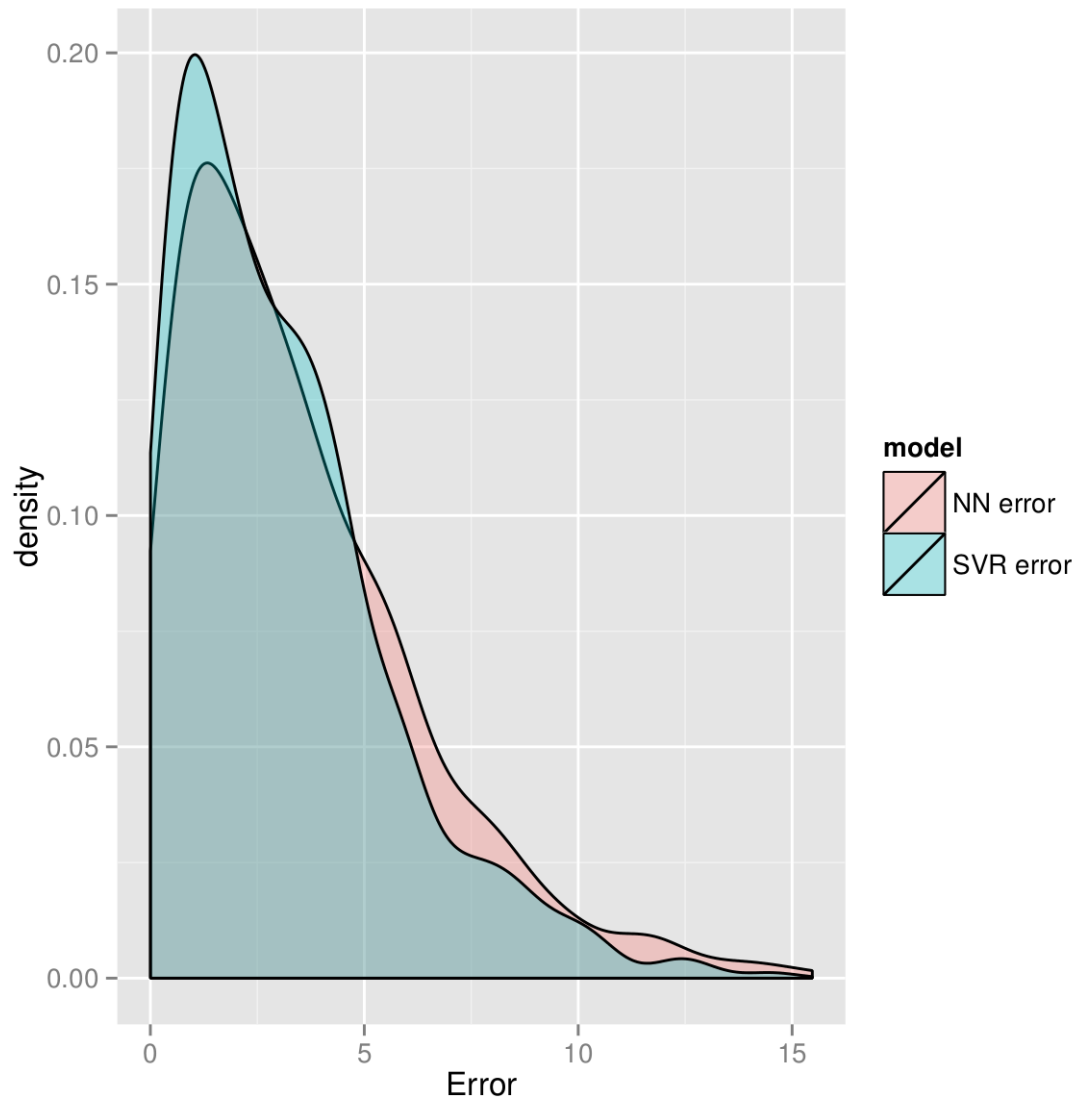


Figure 86: Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for x-direction CD for the high-aspect-ratio etch.

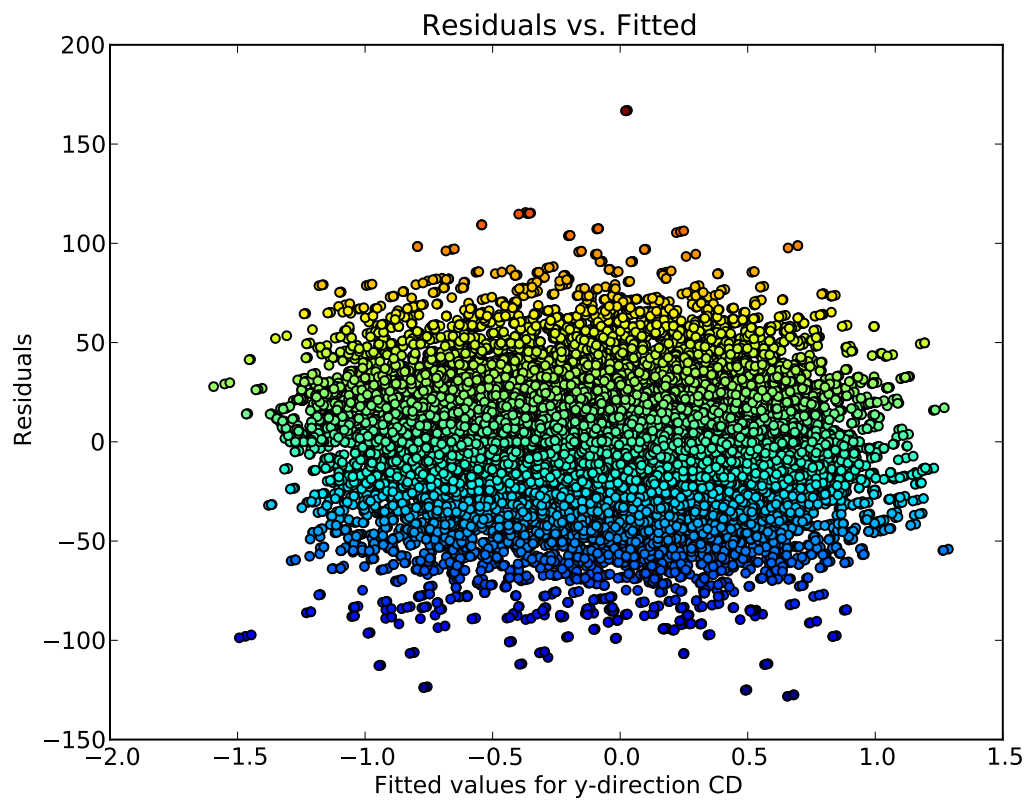


Figure 87: Plot of the residuals *versus* the fitted values for the dependent variable, for each of the data points in the training dataset.

wafer and there is an area of lower CD in the center of the wafer.

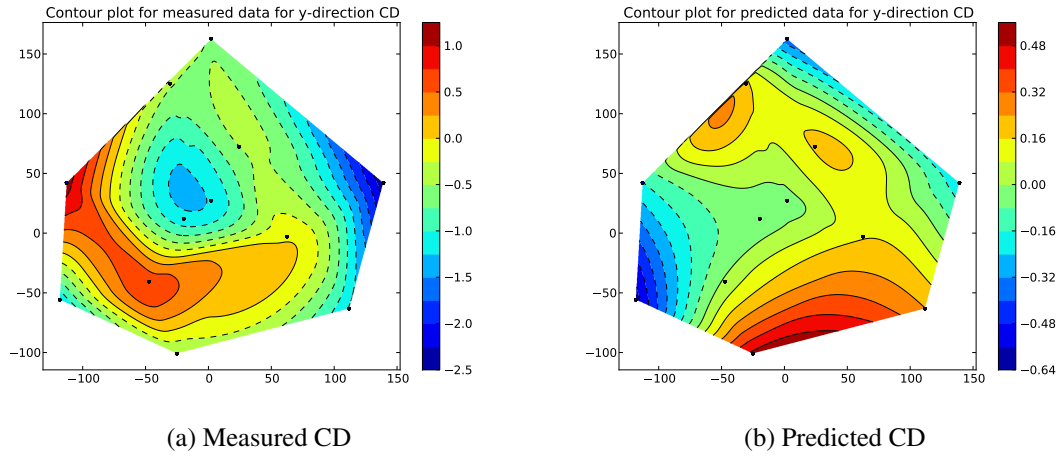


Figure 88: Comparison of measured (left) and predicted (right) output for y-direction CD in the oxide layer. The x-axis and y-axis represent wafer location in mm.

Figure 89 presents a plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for the high-aspect-ratio etch. The pink curve represents neural network error and the blue the support vector regression error. It is ambiguous which model performed better because the neural network error is more concentrated near zero but the mean absolute percentage error for the neural network is higher.

## 8.4 Discussion & conclusion

The support vector machine models outperformed the neural network models in terms of MAPE for four cases out of five. This is not conclusive evidence that support vector machines are better than neural networks to model semiconductor fabrication processes. There were differences between the uniformities of measured and SVR predicted CDs for the high-aspect-ratio etch. It is subjective, but there is more similarity in the uniformities of the neural network predictions, Figures 57 and 59, and the measured data for this process than there is for the support vector regression predicted data, Figures 85 and 88.

Both methods involved some level of trial and error in picking training parameters

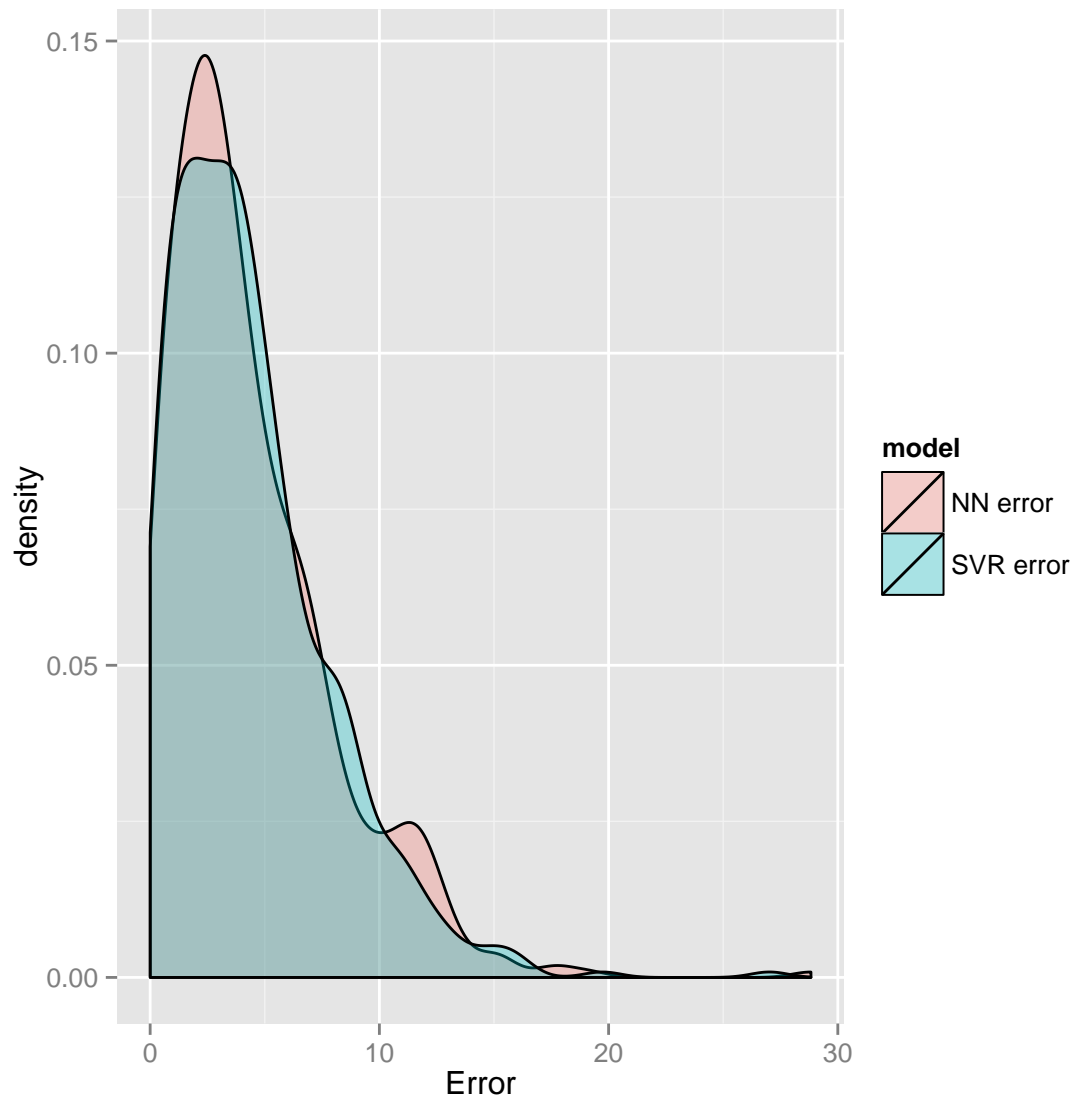


Figure 89: Plot of the densities of absolute percentage prediction errors for both the neural network model and the support vector regression model created for y-direction CD for the high-aspect-ratio etch.



for the models. While generally training times were comparable, in certain instances the SVMs did not converge to a solution in a reasonable amount of time. This can happen for large datasets, as noted by the authors of the implementation used in this study [74]. Although, training of SVMs always converges to a global minimum, unlike training for neural networks [69].

One can conclude that support vector machines show promise as a modeling method for semiconductor fabrication data, and delivered performance comparable to neural networks. There are also some situations where they might be preferable. In [75] they conclude that SVM's performance increases relative to that for neural networks as the size of the dataset decreases. Therefore SVMs might be a better choice for modeling from smaller datasets.

## **CHAPTER 9**

### **CONCLUSION**

The aims of this research were to present and explain the use of artificial neural networks and other machine learning algorithms to model and optimize the complex manufacturing processes used by a semiconductor manufacturer. Another goal was to do so across their various fabrication environments and stages of manufacturing, using actual industrial data that exists in their databases.

A fabrication process like plasma etch may have dozens of controllable inputs (gas flows, pressures, temperatures, voltages, currents), and require a level of precision measured in nanometers. At the same time, such a process may not be fully understood by the manufacturer because the mathematic or scientific theory are not yet existent. Manufacturers arrive at a recipe (a list of settings for the dozens of process inputs) through tweaking and experimentation. Given the complexities being dealt with, one can see that process development requires extensive time and effort on the part of the manufacturer. Given the cost of materials used and the cost of the manufacturing equipment process development is expensive as well. Given the rate of obsolescence for semiconductor products, manufacturers are very keen to shorten their process development times. How does a manufacturer increase actionable information about a process, decrease the development time for that same process, and do so without spending more resources on its development?

One way to do this is too extract more value from the resources available. A large, growing, and largely underutilized resource for manufacturers is the fab database. Advances in machine learning provide manufacturers a toolbox to make better use of this resource. A notable aspect of this research is the use of actual manufacturing datasets, as opposed to experimentally generated ones, that existed in the fab databases as a result of the normal operation of the facility.

The literature provides examples of the use of neural networks to perform process characterization, recipe generation, real-time process control, and virtual metrology. However, the literature has not examined how the usefulness of neural networks might change over the different stages of production for a semiconductor product, given how the characteristics of the datasets change over these different stages. Furthermore, there is little investigation of how these activities affect yield, as this is often impossible when studying a process that is not part of a set of manufacturing steps for a semiconductor product. Therefore two other notable aspects of this research are that it does consider the changing characteristics of the datasets across fabs and manufacturing stages, and that it does investigate how the use of machine learning algorithms to model and optimize upstream processes affect yield metrics.

Three processes were modeled in a R&D fab. The first two created a hardmask that is used during the third: a high-aspect-ratio etch. Their combined effect is to create an array of high-aspect-ratio cylindrical holes in an oxide layer. The separation between holes in the x and y direction were the dimensions of interest and to be controlled. Neural networks were able to accurately model each of these processes, with mean absolute prediction errors (MAPE) remaining below 5% for all models. Given the low yields and the difficulty of attributing yield issues to upstream processes that are characteristic of R&D fabs, the neural network models to predict yield metrics from upstream process input did not perform as well. A sensitivity analysis provided insight about the effects of varying individual process inputs at an important point in the input space, like the process recipe of record.

The high-aspect-ratio etch was modeled during the ramping stage. A variety of feature selection algorithms were used to search for the best model inputs. This included the first use of the minimum redundancy maximum relevance (mRMR) algorithm with semiconductor manufacturing data to the author's knowledge. Stepwise regression produced the dataset that resulted in the models with the lowest MAPE of 0.6473%. The coefficient of determination was 0.5270. Ideally this would be higher.

A genetic algorithm was used to generate a new process recipe. The genetic algorithm converged very quickly and definitively to a solution, which was impressive given the dimensionality of the solution space and the heuristic nature of the algorithm.

Neural network models were also made for a more mature high-aspect-ratio etch process in the manufacturing fab. There was a lot more product and hence a lot more data to model with. However, there was less variation in the process settings which was also reflected in the modeling dataset. It was still possible to build accurate models of the process, with MAPE remaining below 5%.

Another result of a mature process and product was that yield issues were ascribable to upstream processes. This allowed one to find dataset where there was a causal link between the inputs to the high-aspect-ratio etch and various yield metrics. Thus a models of the relationship between process inputs and yield could be built. These had larger MAPES than the models of post-process metrology. Perhaps this is to be expected given the distance between the process and final yield measurements. However the model captured the across-wafer distribution of the yield metrics very well.

A chamber matching exercise was performed as a real world test for these successes in neural network modeling and recipe generation with genetic algorithms. Manufacturers typically operate several reaction chambers in parallel to achieve their desired product volumes. There will be performance variation across chambers even if they are identical (manufacturer, model, configuration) and running the same recipe. This is undesirable and can affect product performance and reliability. A neural network was used to model the etch process of a reaction chamber whose performance was outlying. Then a genetic algorithm was used to search for a recipe that brought performance back in line. This recipe was then tested on an actual manufacturing line, and it improved both post-process and yield metrics for the outlying chamber. Both modeling and optimization methods can be automated and work on a time scale acceptable for run-to-run control.

Lastly, the performances of the neural network models were compared to those of more

traditional and more modern modeling methods. The more former was linear regression and the result of this first comparison was clear. The neural networks outperformed linear regression.

The comparison to support vector regression is more ambiguous, however. SVR outperformed the neural networks in terms of MAPE for four out of the five models created. However, the neural networks were better able to model the CD distributions for the high-aspect-ratio etch.

Thus this work has demonstrated that neural networks can be used to model the plasma etch process across a manufacturer's different fabs and stages of manufacturing using pre-existing data (no added costs for experimentation), and can bring improvements to the etch process and yield when used in conjunction with genetic algorithms.

## **9.1 Future work**

Performing this research brought to the attention many avenues of inquiry that are worthy of future research. Some of these are presented in this section.

One would be modeling how the inputs to an upstream process affect yield, especially in an R&D fab where yields are low. This is even more true when the effects are not reflected in the post-process metrology. Could these machine learning tools unearth information about effects a process has after several subsequent processes have been performed?

Investigating what metrology could be added or adjusted for the high-aspect-ratio etch during the ramping stage so that the dataset could produce models with higher coefficients of determination is also interesting. Or more generally, an investigation of how machine learning could be used to determine the types and amounts of metrology to perform could be fruitful. It is costly to store the amounts of data generated by metrology, and the types of information collected for a process are not constant. Could the models help in deciding what information to collect?

Incorporating the models into an automatic controller for a process, and comparing

performance to what is currently employed would be an area interesting to manufacturers.

Performing a quantitative analysis of the time and cost savings, and the revenue increases that could be achieved if the modeling and optimization methods are incorporated into the manufacturing line is also very interesting. Comparing this to the investment that would be needed perform such an incorporation would provide information about the monetary value of these tools to a manufacturer.

And lastly, SVR showed promise as a modeling method with semiconductor fabrication data. Further exploring their use with such data, and comparing it to the use of neural networks, would be an interesting area for future work.

## REFERENCES

- [1] G. Moore *et al.*, “Cramming more components onto integrated circuits,” *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.
- [2] G. E. Moore, “Progress in digital integrated electronics,” in *Electron Devices Meeting, 1975 International*, vol. 21, pp. 11–13, IEEE, 1975.
- [3] “International technology roadmap for semiconductors.”
- [4] I. Insights, “The mcclean report,” *IC Insights Inc., Arizona*, 2013.
- [5] C. Weber, “Yield learning and the sources of profitability in semiconductor manufacturing and process development,” *Semiconductor Manufacturing, IEEE Transactions on*, vol. 17, no. 4, pp. 590–596, 2004.
- [6] J. Moyne, E. Del Castillo, and A. Hurwitz, *Run-to-run control in semiconductor manufacturing*. CRC, 2000.
- [7] M. Appleyard, N. Hatch, and D. Mowery, “Managing the development and transfer of process technologies in the semiconductor manufacturing industry,” *The nature and dynamics of organizational capabilities*, pp. 183–207, 2000.
- [8] R. Goodwin, R. Miller, E. Tuv, A. Borisov, M. Janakiram, and S. Louchheim, “Advancements and applications of statistical learning/data mining in semiconductor manufacturing,” *Intel Technology Journal*, vol. 08, no. 04, pp. 325–336, 2004.
- [9] N. Hatch and D. Mowery, “Process innovation and learning by doing in semiconductor manufacturing,” *Management Science*, vol. 44, no. 11-Part-1, pp. 1461–1477, 1998.
- [10] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [11] E. Rietman and E. Lory, “Use of neural networks in modeling semiconductor manufacturing processes: An example for plasma etch modeling,” *Semiconductor Manufacturing, IEEE Transactions on*, vol. 6, no. 4, pp. 343–347, 1993.
- [12] F. D. Palma, G. D. Nicolao, G. Miraglia, E. Pasquinetti, and F. Piccinini, “Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing,” *Pattern recognition letters*, vol. 26, no. 12, pp. 1857–1865, 2005.
- [13] C.-T. Su, T. Yang, and C.-M. Ke, “A neural-network approach for semiconductor wafer post-sawing inspection,” *Semiconductor Manufacturing, IEEE Transactions on*, vol. 15, no. 2, pp. 260–266, 2002.

- [14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [15] S. Haykin, *Neural networks: a comprehensive foundation*. Macmillan, 1994.
- [16] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [17] C. Himmel and G. May, "Advantages of plasma etch modeling using neural networks over statistical techniques," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 6, no. 2, pp. 103–111, 1993.
- [18] T. Edgar, S. Butler, W. Campbell, C. Pfeiffer, C. Bode, S. Hwang, K. Balakrishnan, and J. Hahn, "Automatic control in microelectronics manufacturing: Practices, challenges, and possibilities," *Automatica*, vol. 36, no. 11, pp. 1567–1603, 2000.
- [19] J. Ringwood, S. Lynn, G. Bacelli, B. Ma, E. Ragnoli, and S. McLoone, "Estimation and control in semiconductor etch: Practice and possibilities," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 23, no. 1, pp. 87–98, 2010.
- [20] S. Han and G. May, "Modeling the plasma enhanced chemical vapor deposition process using neural networks and genetic algorithms," in *Tools with Artificial Intelligence, 1994. Proceedings., Sixth International Conference on*, pp. 760–763, IEEE, 1994.
- [21] S. Han and G. May, "Using neural network process models to perform pecvd silicon dioxide recipe synthesis via genetic algorithms," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 10, no. 2, pp. 279–287, 1997.
- [22] E. Rietman and R. Frye, "A genetic algorithm for low variance control in semiconductor device manufacturing: Some early results," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 9, no. 2, pp. 223–229, 1996.
- [23] J. Ahn, T. Kang, W. Lim, S. Han, H. Kim, and S. Hong, "Reptor: An intelligent hybrid neural network based recipe generator for semiconductor process modeling and characterization," in *New Trends in Information and Service Science, 2009. NISS'09. International Conference on*, pp. 1118–1123, IEEE, 2009.
- [24] H. Bae, T. Jeon, S. Kim, H. Kim, D. Kim, S. Han, and G. May, "Optimization of silicon solar cell fabrication based on neural network and genetic programming modeling," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 2, pp. 161–169, 2010.
- [25] S. Hong, G. May, and D. Park, "Neural network modeling of reactive ion etching using optical emission spectroscopy data," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 16, no. 4, pp. 598–608, 2003.



- [26] G. Triplett, G. May, and A. Brown, "Modeling electron mobility in mbe-grown inas/al<sub>s</sub>b thin films for hemt applications using neural networks," *Solid-State Electronics*, vol. 46, no. 10, pp. 1519–1524, 2002.
- [27] S. Hong and G. May, "Neural network-based real-time malfunction diagnosis of reactive ion etching using in situ metrology data," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 17, no. 3, pp. 408–421, 2004.
- [28] E. Rietman and S. Patel, "A production demonstration of wafer-to-wafer plasma gate etch control by adaptive real-time computation of the over-etch time from in situ process signals," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 8, no. 3, pp. 304–308, 1995.
- [29] J. Card, D. Sniderman, and C. Klimasauskas, "Dynamic neural control for a plasma etch process," *Neural Networks, IEEE Transactions on*, vol. 8, no. 4, pp. 883–901, 1997.
- [30] C. Davis and G. May, "Neural network control of variable-frequency microwave processing of polymer dielectric curing," *Electronics Packaging Manufacturing, IEEE Transactions on*, vol. 31, no. 2, pp. 104–113, 2008.
- [31] B. Widrow and M. Bilello, "Adaptive inverse control," in *Intelligent Control, 1993., Proceedings of the 1993 IEEE International Symposium on*, pp. 1–6, IEEE, 1993.
- [32] D. Stokes and G. May, "Indirect adaptive control of reactive ion etching using neural networks," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 5, pp. 650–657, 2001.
- [33] R. Setia and G. May, "Run-to-run failure detection and diagnosis using neural networks and dempster-shafer theory: an application to excimer laser ablation," *Electronics Packaging Manufacturing, IEEE Transactions on*, vol. 29, no. 1, pp. 42–49, 2006.
- [34] R. Setia and G. May, "In-line failure detection and diagnosis of excimer laser-based microvia fabrication using computational intelligence," *Journal of Laser Applications*, vol. 18, p. 258, 2006.
- [35] S. Lynn, J. Ringwood, E. Ragnoli, S. McLoone, and N. MacGearailty, "Virtual metrology for plasma etch using tool variables," in *Advanced Semiconductor Manufacturing Conference, 2009. ASMC'09. IEEE/SEMI*, pp. 143–148, IEEE, 2009.
- [36] S. Lynn, N. MacGearailty, and J. Ringwood, "Real-time virtual metrology and control for plasma etch," *Journal of Process Control*, 2012.
- [37] S. Lynn, J. Ringwood, and N. MacGearailty, "Global and local virtual metrology models for a plasma etch process," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 25, no. 1, pp. 94–103, 2012.

- [38] S. Han, M. Ceiler, S. Bidstrup, P. Kohl, and G. May, "Modeling the properties of pecvd silicon dioxide films using optimized back-propagation neural networks," *Components, Packaging, and Manufacturing Technology, Part A, IEEE Transactions on*, vol. 17, no. 2, pp. 174–182, 1994.
- [39] G. H. John, R. Kohavi, K. Pfleger, *et al.*, "Irrelevant features and the subset selection problem.," in *ICML*, vol. 94, pp. 121–129, 1994.
- [40] R. May, G. Dandy, and H. Maier, "Review of input variable selection methods for artificial neural networks," *Artificial neural networksmethodological advances and biomedical applications*, pp. 19–44, 2011.
- [41] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial intelligence*, vol. 97, no. 1, pp. 245–271, 1997.
- [42] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, 1998.
- [43] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [44] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of bioinformatics and computational biology*, vol. 3, no. 02, pp. 185–205, 2005.
- [45] T. S. Kim and G. S. May, "Sequential modeling of via formation in photosensitive dielectric materials for mcm-d applications," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 12, no. 3, pp. 345–352, 1999.
- [46] C.-T. Su and T.-L. Chiang, "Optimizing the ic wire bonding process using a neural networks/genetic algorithms approach," *Journal of Intelligent Manufacturing*, vol. 14, no. 2, pp. 229–238, 2003.
- [47] W. Sukthomya and J. Tannock, "The training of neural networks to model manufacturing processes," *Journal of Intelligent Manufacturing*, vol. 16, no. 1, pp. 39–51, 2005.
- [48] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.
- [49] C. Davis, S. Hong, R. Setia, R. Pratap, T. Brown, B. Ku, G. Triplett, and G. May, "An object-oriented neural network simulator for semiconductor manufacturing applications," in *The 8th world multi-conference on systemics cybernetics informatics*, vol. 5, pp. 365–370, 2004.
- [50] G. May and C. Spanos, *Fundamentals of semiconductor manufacturing and process control*. Wiley Online Library, 2006.

- [51] M. Higashide, K. Nishina, H. Kawamura, and N. Ishii, "Statistical process control for semiconductor manufacturing processes," *Frontiers in Statistical Quality Control* 9, vol. 9, p. 71, 2010.
- [52] D. Boning and J. Chung, "Statistical metrology: Understanding spatial variation in semiconductor manufacturing," in *Microelectronic Manufacturing 1996*, pp. 16–26, International Society for Optics and Photonics, 1996.
- [53] G. S. May, "Manufacturing ics the neural way," *Spectrum, IEEE*, vol. 31, no. 9, pp. 47–51, 1994.
- [54] J. Frenzel, "Genetic algorithms," *Potentials, IEEE*, vol. 12, no. 3, pp. 21–24, 1993.
- [55] J. Card, "A study in dynamic neural control of semiconductor fabrication processes," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 13, no. 3, pp. 359–365, 2000.
- [56] E. Ragnoli, S. McLoone, S. Lynn, J. Ringwood, and N. Macgearailt, "Identifying key process characteristics and predicting etch rate from high-dimension datasets," in *Advanced Semiconductor Manufacturing Conference, 2009. ASMC'09. IEEE/SEMI*, pp. 106–111, IEEE, 2009.
- [57] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability and statistics for engineers and scientists*, vol. 5. Macmillan New York, 1972.
- [58] T. Kim and G. May, "Optimization of via formation in photosensitive dielectric layers using neural networks and genetic algorithms," *Electronics Packaging Manufacturing, IEEE Transactions on*, vol. 22, no. 2, pp. 128–136, 1999.
- [59] Z. Mevawalla, G. May, and M. Kiehlbauch, "Neural networks for advanced process control," in *Advanced Semiconductor Manufacturing Conference (ASMC), 2010 IEEE/SEMI*, pp. 137–142, IEEE, 2010.
- [60] Z. Mevawalla, G. May, and M. Kiehlbauch, "Neural network modeling for advanced process control using production data," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 24, no. 2, pp. 182–189, 2011.
- [61] R. Setia and G. May, "Modeling and optimization of via formation in dielectrics by laser ablation using neural networks and genetic algorithms," *Electronics Packaging Manufacturing, IEEE Transactions on*, vol. 27, no. 2, pp. 133–144, 2004.
- [62] E. Wornyo, G. May, and K. Gall, "Modeling and optimization of the deposition of shape memory polymers for information storage applications," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 22, no. 3, pp. 409–416, 2009.
- [63] Z. Mevawalla, G. May, M. Honjo, and M. Kiehlbauch, "Neural network modeling of fabrication yield using manufacturing data," in *Advanced Semiconductor Manufacturing Conference (ASMC), 2011 22nd Annual IEEE/SEMI*, pp. 1–6, IEEE, 2011.

- [64] W. F. Clocksin, “Artificial intelligence and the future,” *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1809, pp. 1721–1748, 2003.
- [65] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [66] H. Wickham, *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.
- [67] V. Vapnik, “Pattern recognition using generalized portrait method,” *Automation and remote control*, vol. 24, pp. 774–780, 1963.
- [68] A. Ng, “Support vector machines.” Stanford University Lecture, CS229.
- [69] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
- [70] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [71] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [73] G. Valentini and T. G. Dietterich, “Bias-variance analysis of support vector machines for the development of svm-based ensemble methods,” *The Journal of Machine Learning Research*, vol. 5, pp. 725–775, 2004.
- [74] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [75] K.-S. Shin, T. S. Lee, and H.-j. Kim, “An application of support vector machines in bankruptcy prediction model,” *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.