การสร้างโมเดลเพื่อทำนายจากเครือข่ายใยประสาทแบบสุ่ม สำหรับการผลิตเฮชจีเอ

ประสิทธิ์ชัย บุญเสริม [1] และ ธีรณี อจลากุล [2]
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี แขวงบางมด เขตทุ่งครุ กรุงเทพฯ 10140

บทคัดย่อ

ในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดร์ฟ ผลิตภัณฑ์แต่ละรุ่นมีคุณสมบัติที่แตกต่างกัน โดยคุณสมบัติของแต่ละผลิตภัณฑ์จะขึ้นอยู่กับความต้องการของลูกค้า การกำหนดคุณสมบัติของฮาร์ดดิสก์และปัญหาที่เกิดจากเครื่องจักรในกระบวนการผลิตมีผลต่อจำนวนผลผลิตอย่างมาก เพื่อให้ได้ผลผลิตที่เพิ่มขึ้นวิศวกรจำเป็นต้องหาสาเหตุที่แท้จริงของปัญหา โดยงานวิจัยนี้มุ่งเน้นในส่วนของการออกแบบระบบจำลองการทำนายเปอร์เซ็นต์ผลผลิตเป็นสำคัญ ซึ่งการทำนายนั้นจะเกิดขึ้นหลังกระบวนการวิเคราะห์สาเหตุและแก้ปัญหาเพื่อสนับสนุนกระบวนการดังกล่าว การทำนายผลจะยืนยันถึงแนวทางการแก้ปัญหาที่ผ่านการวิเคราะห์ ว่าสามารถนำไปปรับปรุงสายการผลิตได้หรือไม่ ในเบื้องต้นทางคณะวิจัยได้ทำการพิสูจน์สมมติฐานโดยการเขียนโปรแกรมเพื่อทำการวิเคราะห์เครือข่ายใยประสาทแบบสุ่ม โดยข้อมูลที่ใช้ในการคำนวนและทดสอบสมการถดถอยพหุคูณเป็นข้อมูลที่ได้มาจากกระบวนผลผลิตฮาร์ดดิสก์ไดร์ฟ การทดสอบดังกล่าวมีขึ้นเพื่อทดสอบความแม่นยำของตัวแบบทำนาย จากการพิสูจน์ดังกล่าวสามารถสรุปได้ว่า ทฤษฎีการวิเคราะห์การถดถอยพหุคูณ ให้ผลการทำนายที่มีความคลาดเคลื่อนสูงและข้อมูลที่ใช้มีความไม่เป็นเส้นตรง ดังนั้น คณะวิจัยจึงเลือกใช้ทฤษฎีเครือข่ายใยประสาทแบบสุ่มในการสร้างตัวแบบทำนาย ข้อมูลอินพุตของตัวแบบทำนายดังกล่าวเป็นข้อมูลชุดเดียวกับข้อมูลที่ใช้ในการพิสูจน์ทฤษฎีการวิเคราะห์การถดถอยพหุคูณ จากการทดสอบตัวแบบที่สร้างจากทฤษฎีเครือข่ายใยประสาทแบบสุ่ม พบว่า ตัวแบบดังกล่าวสามารถใช้กับระบบข้อมูลที่มีความซับซ้อนสูง รวมทั้งข้อมูลมีคุณสมบัติไม่เป็นเส้นตรงได้ ทฤษฎีดังกล่าวได้มีการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมในการเรียนรู้ เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมนั้นสามารถจัดการกับข้อมูลที่มีคุณสมบัติแบบไม่เป็นเส้นตรงและยังสามารถจัดการกับปัญหาความสัมพันธ์แบบสุ่มของข้อมูลอินพุตได้ดี จากการทดสอบพบว่า ตัวแบบทำนายที่สร้างจากทฤษฎีเครือข่ายใยประสาทแบบสุ่มโดยใช้ขั้นตอนวิธีเชิงพันธุกรรมในการเรียนรู้นั้น ให้ผลการทำนายที่มีความคลาดเคลื่อนน้อยมาก จึงสรุปได้ว่า ตัวแบบการทำนายที่สร้างบนพื้นฐานทฤษฎีเครือข่ายใยประสาทเหมาะสมที่จะใช้เป็นระบบจำลองการทำนายในอุตสหกรรมการผลิตฮาร์ดดิสก์ไดร์ฟ

คำสำคัญ : เครือข่ายใยประสาทแบบสุ่ม / การถดถอยพหุคูณ / เทคนิคสำหรับทำนายผลการผลิต

[1] *นักศึกษาปริญญาเอก, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์*

[2] *รองศาสตราจารย์, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์*

The SNN-Based Predictive Model for HGA Manufacturing

Prasitchai Boonserm [1] and Tiranee Achalakul [2]

King Mongkut's University of Technology Thonburi, Bang Mod, Thung Khru, Bangkok 10140

Abstract

Product types in the hard disk drive (HDD) industry have different specifications depending on the customer orders. These specifications along with the machine parameters have a direct impact on the production yield. The problems on the manufacturing line are called the “root cause”. By accurately identifying the root cause, the engineers can suggest yield improvement solutions. Our research focus on a design of the effective prediction technique required at the end of the analysis steps in order to validate the suggested solution by simulation. Initially, we developed a “C” program based on a multiple regression technique and tested it’s validity toward prediction accuracy. From the experimental results, we could conclude that the multiple regression method (up to 10 polynomial degrees) did not produce a sufficiently good result. The method gave a high error rate for fault prediction. The Hard Gimbal Assembly (HGA) yield prediction is proved to be non-linear. Therefore, we adapted the Stochastic Neural Networks (SNNs) for use with the yield prediction. The inputs of our model consist of several machine parameters and specification attributes. Our version of SNNs can approximate a complex non-linear system. The genetic algorithm is used as a learning algorithm instead of the backpropagation method in order to handle the non-linear and stochastic relationships between input parameters. Our prediction model can then be used to validate and revise the yield improvement plan. The output of the prediction model is the yield rate. From SNNs' results, we can conclude that our initial version of SNNs gave a favorable prediction results with very low error rates. The model can thus be used as a simulation tool for yield improvement without having to actually implement the solution on the production line.

Keywords : Stochastic Neural Networks / Multiple Regression /Yield Prediction Technique

[1] *Ph.D Student, Department of Computer Engineering, Faculty of Engineering.*

[2] *Associate Professor, Department of Computer Engineering, Faculty of Engineering.*

1. Introduction

In disk drive manufacturing, every product type has a different product specification depending on customer demands. These specifications have a direct impact to the production yield. Products with higher specification are generally harder to be manufactured and can cause the yield degradation. Moreover, machine quality in both production and testing lines, as well as human errors can also degrade yield. In order to improve yield, the “root cause” of the problem in manufacturing must be analyzed.

Currently, most disk drive companies have their own tools for yield improvement analysis. However, the analysis processes lack automated tools and depend heavily on the expertise of the engineers and the specialists. In order to reduce the man power and analysis time needed, the automate yield analysis framework should be designed and implemented.

This paper focuses on the prediction technique required at the end of the analysis steps. After “root causes” that include machine parameters and product specifications are identified, the suggestive plan for yield improvement can be drawn from the tool set. Our prediction model is used to validate and revise the yield improvement plan. The output of the prediction model is the yield rate and the inputs are many possible root causes of the manufacturing problem. Our work can be used as a prediction tool for yield improvement without having to actually implement the solution on the production line.

In general, the linear regression approach has been commonly used for data mining applications in manufacturing plants [1-4]. However, many applications, such as yield prediction and optimization, may have inputs that are random in nature. The problems can be unpredictable and non-linear [5]. Such applications require more complicated models. In those cases, the Neural Network-based models are often explored to cope with the stochastic components [6].

The characteristics of Hard Gimbal Assembly (HGA) data are dynamic because machine parameters and specification attributes may change on a daily basis. A set of inputs may be paired with all possible output yields. The probabilities of occurrences can also be varied. We thus suspected that the simple linear regression method may not yield a good result [7]. In order to validate that assumption, we experimented with a multiple regression (MR) model of different polynomial degrees to study the nature of the manufacturing data as well as to discover whether the model can fit the data. After finding out a flaw in adopting the regression technique, we design an algorithm and implement a program for SNN-based predictive model. Our prediction model and the accuracy analysis are presented in this paper.

The rest of the paper is organized as follows. Section 2 presents the related research in SNNs. Several Stochastic Neural Network-Based approaches are discussed and compared. Section 3 describes our initial experiments with the multiple regression methods using the actual HGA manufacturing data. The discussions and concluding remarks in multiple regression methods are also provided. Section 4 describes our general design framework of SNNs for automate yield analysis with the emphasis on yield prediction model. Then the experiments and results of Stochastic Neural Networks are discussed in section 5. Finally, in section 6, conclusions and future work are discussed.

2. Related Research

Neural Network-Based approaches have been widely used in data mining applications, such as manufacturing yield prediction, optimization, and plant design. Many efforts have been put into applying neural networks (NNs) [8] to create decision support systems. Numerous examples were presented in recent literature. For example, in the study of Baba and Yanjun [9], a new model was created by combining NNs with Genetic Algorithm (GA). The model was then used to predict the highest and the lowest future values for Japan's stock exchanges. However, many applications have inputs that are random in nature. A regular ANNs may not be able to accurately model such behavior. Thus, Stochastic Neural Networks (SNNs) [10-13] was introduced. SNNs are able to cope with the stochastic components. Models in SNNs are possible to be trained by the backpropagation method, which can be extended with stochastic resonance features.

SNNs are generally used for minimizing the redundancy between elements in the output layer using a learning algorithm, which is extended from the unsupervised learning principle of Barlow [14]. The algorithm was designed based on the probabilistic non-linear neurons and can be used for a network with recurrences. The learning algorithm interprets a weighted combination of Hebbian and anti-Hebbian rule [15]. This approach performs a non-linear and a factorial feature extraction, by maximizing the mutual information between sensory inputs and output[5].

The study of Tanaka [16] extended the SNN model to handle non-linear systems with missing data. The stochastic model was used to estimate outputs even with missing elements in the input vectors within a condition estimation framework. The Expectation-Maximization (EM) algorithm was used to estimate the model parameters [16]. The study of Polvichai et al. [11] used SNNs with genetic algorithm to learn a large data set consisting of probabilistic relationship between configurations, environments and performance metrics. The optimal configuration, leading to a better quality solution, in new scenarios was derived.

SNNs were used most in prediction systems. Kamitsuji and Shibata [13] have developed an efficient training algorithm to maximize the likelihood of a neural network. This algorithm allowed the SNNs to be applied to a practical problem, such as the prediction of stock price index. Chryssolouris [17], on the other hand, he used neural networks to identify the important relationship of decision criteria. In his work, NNs are used to establish adequate weights of the criteria for the decision-making process. The proposed idea was capable of determining suitable criteria weights for the entire sequence of multiple-criteria decisions. His work, however, was better suited to the complex applications involving chains of decisions.

Polvichai and Khosla [18] used a genetic algorithm to create a dynamic neural network. In order to solve the non-deterministic tasks, they presented a new concept; a dynamic network that mapped all possible input and output patterns with different probabilities of occurrences.

In addition, NN models can be used in manufacturing applications. An example was presented in [6]. In their work, the NN-based algorithm was used to predict the wafer yield in Integrated Circuit (IC) manufacturing. As the wafer size was increased, the clustering phenomenon of the defects became increasingly apparent. The fuzzy ART network for the clustering analysis was thus used to adjust the number of wafer defects and the Poisson distribution was used to predict the wafer yield.

In this work, we attempt to predict the HGA yield in the disk drive manufacturing line. The problem is non-linear and stochastic as machine parameters and specification attributes may change on a daily basis. A set of inputs may be matched with all possible output yields. The probabilities of occurrences can also be varied. Thus, we choose to explore the concept presented in [6, 18]. The detailed framework of our adapted algorithm will be presented in section 4 following the discussion on initial experiments of multiple regression.

3. Initial Experiments with Multiple Regression

In order to investigate whether a simple regression model was sufficient in our problem domain, we have developed a custom "C" program to automatically fit data into curves of various polynomial degrees. The input parameters are the learning data set, the testing data set, a selected threshold, and the maximum polynomial degree. The learning and testing data sets are obtained from actual HGA manufacturing data records (instances).

Our program can calculate the regression coefficients based on sets of input parameters and the learning data set. Once the model is formulated, the testing data set is used to evaluate the regression model. Each polynomial degree has its regression equation. Our program automatically generates all possible regression equation terms using the Brute force technique. Finally, the program produces a set of confusion matrix for each regression equation.

The dataset used in our experiments are the top 5 defect attributes which affect the Hard Gimbal Assembly Electrical Test (HGA ET), which is one of the QA parameters in HGA manufacturing. The objective of the regression model is to predict whether the HGA ET test on products will pass or fail. The data records obtained for our initial experiment were from 7 consecutive manufacturing days. We used 5,012 records of data which is sampled. The "PASS" class has 4,283 instances and the "FAIL" class has 729 instances. This particular dataset gave a yield of 85.45 percents and the percentage of the HGA ET failure is equal to 14.55.

Ten percents of the dataset (501 instances) selected randomly were used as the testing set and the remaining records (4,511 instances) were the learning set used for least square fitting of the model. The least-square parameters (regression coefficients) were then estimated using pseudo inverse. Following this, the equation that construct from those coefficients was used as the initial prediction model in our experiments. Then, different polynomial degrees were experimented with.

In the predictive analysis, we used a table of confusion (also known as a confusion matrix). The true positive value refers the number of the right prediction items in the "PASS" class. The false positive refers to the number of the wrong prediction items in the "PASS" class. False negative items are the number of wrong prediction in the "FAIL" class, while true negative refers to the right prediction in the "FAIL" class.

In our experiments, items were deposited in the PASS class if the $\hat{y}$ value was more than or equal to 0.6. Otherwise the items were considered a part of the FAIL class. The value of 0.6 was the threshold, which gave the maximum accuracy percentage, obtained empirically.

From the initial experiment, the total accuracy calculated was 93.6 percent, which looked satisfactory. However, from our data set, the number of items in the PASS class was a lot larger than that of the FAIL class. The correct prediction of the PASS items could thus dominate the total accuracy. The accuracy of the FAIL class prediction was only 63.2 percent. In a new manufacturing line where unpredictability of the fail rate can be assumed, the 36.7 prediction error rate will not be acceptable. In other words, the prediction model should weight the importance of the true negative and true positive equally and the high prediction rate should be observed for both the PASS and FAIL classes. In conclusion, the regression model failed to predict the true behavior in the manufacturing line and the higher total accuracy did not imply the better prediction rate in the FAIL class. Fig. 1 shows the scatter plot of the prediction results.

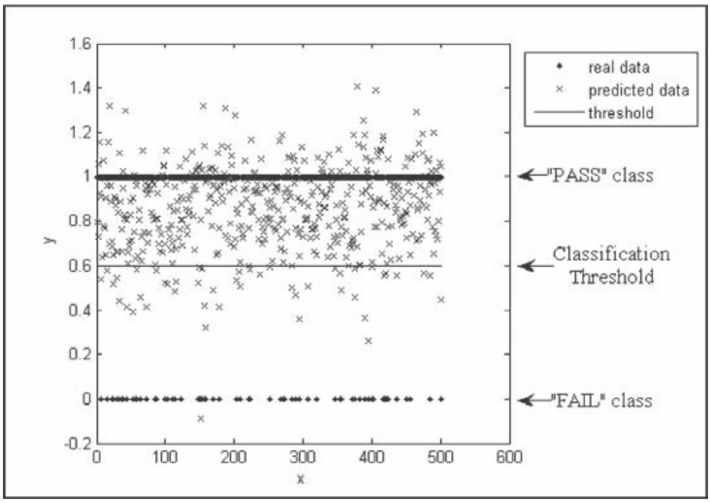
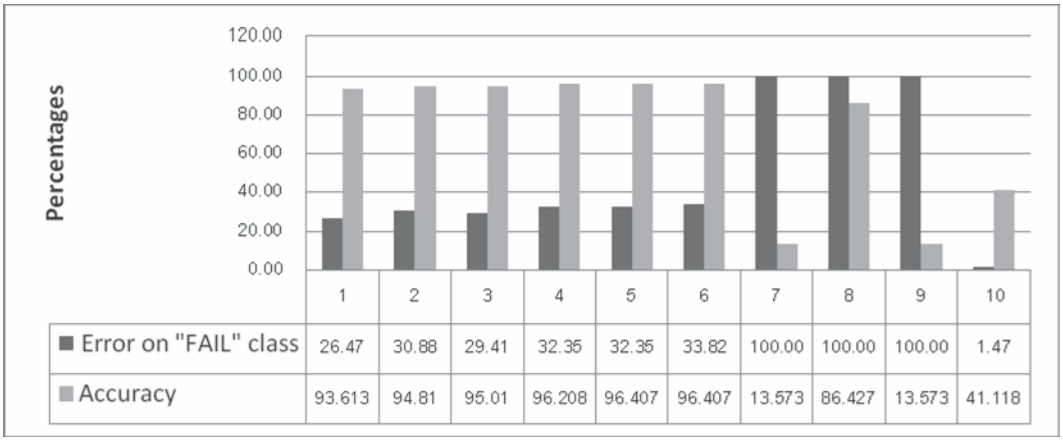


Fig. 1 Prediction results

The y-axis represented the HGA ET values. The PASS and FAIL classes were marked at 1.0 and 0.0 respectively.



	1	2	3	4	5	6	7	8	9	10
■ Error on "FAIL" class	26.47	30.88	29.41	32.35	32.35	33.82	100.00	100.00	100.00	1.47
■ Accuracy	93.613	94.81	95.01	96.208	96.407	96.407	13.573	86.427	13.573	41.118

Fig. 2 The prediction results with various polynomial degrees.

The threshold of 0.6 was used to classify the results. The plot showed that the prediction values were scattered along the y-axis with no clear distinction between pass and fail items. The results implied that the model was not good enough for use in yield prediction even with the empirical obtained threshold.

We then extended our experiments to cover higher polynomial degrees (from 1 to 10). Fig. 2 showed the percentage of the total accuracy and the wrong prediction on FAIL class. The X-axis represented various polynomial degrees. When the maximum polynomial degree used was 10, the error rate of the FAIL class prediction was low (1.47 percents), however, the overall accuracy also dropped to 41.118 percent. There was no pattern or relation between the FAIL class error rate and the overall accuracy. These results implied that using multiple polynomial degrees did not improve the prediction accuracy.

From a set of experiments discussed, we can see that using the multiple regression method on our dataset produced poor results. The method gave a high error rate for fault prediction. Various polynomial degrees (1 to 10) were also explored and the results were not favorable. We can conclude that the HGA yield prediction is a non-linear problem as suspected.

4. A Design Framework of SNNs for Automatic Yield Prediction

The general design framework for an automatic yield prediction is described in this section. The HGA data retrieved from a company's data warehouse are used as inputs for the prediction model. These data are generated from a combination of many machine parameters and specification attributes. Examples of the machine parameters include the value of resistance, amplitude, and frequency. The HGA specification examples given by customers include write resistance and write width.

From our initial research, we discover that the output of the predictive model can be different, though the inputs are the same. There is also a high variation for the output values. Thus, the output should be presented in the form of the probabilities of occurrences. The yield prediction method of such complex data is non-deterministic and non-linear making SNN-based algorithm a method of choice.

There are three key ideas in our SNN-based algorithm design as described below:

```
Generate aninitial weight population
generation = 0
do
    Sub-sampling the HGA data to be the new training data
    Start creating the new generation of genes
    Calculate weight sets
    Evaluate weight sets
    Rank weight sets
    generation = generation + 1
until errors between generation are below a pre-defined threshold
return
```

Fig. 3 Learning step.

Key idea #1: extra input nodes are added to the input layer. The extra input nodes are fed with random signals, which are uniformly varied between 0.0 and 1.0. The random signal values are mapped to the possible HDD yield rate represented by machines parameter and customer specifications. Each iteration uses a new set of random signals. All nodes in the network use the sigmoid units.

Key idea #2: backpropogation is replaced with the genetic algorithm. The genetic algorithm is chosen because the relation between parameters and specifications are non-linear and stochastic. All weight sets of the SNN connections are randomly generated at the start. The new set of weights is generated repeatedly using three genetic operators: reproduction, crossover, and mutation until the best weight set of SNNs is found. Fig. 3 shows the pseudo-code of the learning method and the description is given below.

Line 1 of Fig. 3 (3.1) randomly generates an initial weight population for training SNNs. Then the machine parameters and specification attributes are sampled to be used as the new training data. In general, if the training and the testing set of SNNs are too large, a subset of data is randomly selected and used. The sub-sampling step is performed to avoid over-fitting due to the local minima problem (3.4). The new generation of chromosome is then created by performing crossover and mutation of the top 5 percentile best genes from the previous generation. The probability of 0.001 is used in this step (3.5). In step (3.6), the generated genes are used as weight sets. The sampled HGA data are then divided into 2 subsets, where the first subset is used in training and the last subset is used in testing. The average error is computed and used as the weight set (chromosome) error. In step (3.7), all weight sets are evaluated using chromosome error. The highest score will be assigned to the weight set which has the least error. The results are fitness values of all chromosomes. The values are then used to rank all weight sets. The top rank weight set is considered to have the best performance score. This ranking information is mainly used for creating a new generation (3.8). The learning procedure will be terminated when the errors between generations do not vary more than a pre-defined threshold.

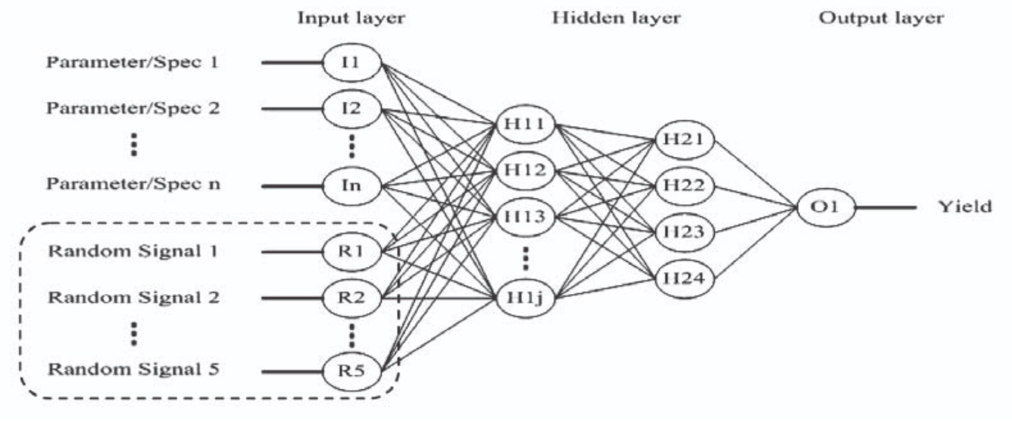


Fig. 4 The new stochastic neural network model.

Key idea #3: the output of the neural network model will be non-deterministic. The root causes of HGA manufacturing failure will be used as inputs of the model. The same set of inputs will be fed through the model 10,000 times with different random signals. Each time, the output will vary. In other words, SNNs will generate an uncertainty distribution of the output data. The distribution of possible yields can thus be generated more accurately. Fig. 4 describes our SNN model. Our SNN model consists of three layers: input, hidden and output layers. The input nodes are machine parameters and specifications. R1 though R5 are the extra randomly generated input signals with the values between 0.0 to 1.0. These random signal values are mapped to possible yield rates represented by HGA parameters and specifications. Different functions will be created for nodes in the hidden layers, which will be used to generate the manufacturing yield in the output nodes.

In our design, numbers of input nodes can vary depending on number of interested root causes of failure in a production each day. The prediction process in each day will start by comparing a set of inputs between previous and current day. If the lists of parameters are different drastically, the model must be trained again. The new set of weights in SNNs will be re-computed. The framework also has a mechanism to save the weight set for any particular input parameter list. Thus, re-learning process will only be required for a brand new parameters list.

5. Experiments and Results

Actual manufacturing data records were obtained for our experiment. The data used are from the manufacturing of the disk drive. We used the total of 185,000 sample records, which were retrieved from seven days of manufacturing.

In our experiment, we configured the SNNs to have 4 layers. The top 5 critical parameters and 1 random signal were used as input nodes in the first layer (input layer). The second and third layers were hidden layers with 8 nodes each. The last layer is the output layer with only one output node referring to the yield prediction result. The number of weight used in this model was 120, 48 were used in the first layer, 64, 64, and 8 in the second, third and fourth layers respectively. Each weight represents a gene and 120 genes represent a chromosome. All chromosomes can be permuted to create 115 million initial chromosomes populations.

During the learning process, chromosome values were randomly generated. From our empirical study, the appropriate values were between -40.0 to 40.0. 100,000 data records were then randomly selected to be used as the training data. In the learning step, the new generations of genes were created by crossover and mutation the top 5 percentile best genes from the previous generation. The algorithm proceeded as described in the previous section. The estimated errors between iterations were computed. The learning procedure was terminated when the errors between generations equal to or less than 5 percent. When the learning process terminated, we used the best gene as our SNNs weight set. The best prediction result from our SNNs showed a 0.61% error rate on a “FAIL” class. The worst case prediction produced a 2.57% error rate on a “FAIL” class. This is no error rate on a “PASS” class.

We also configured a network structure of ANNs with a standard backpropagation in order to be used as a baseline for comparison with our SNNs results. The same data sets were used.

The learning procedure was terminated when the errors between actual output and predictive result equal to or less than 5 percents. When the learning process was terminated, the best prediction result from ANNs showed a 6.72% error rate, while the worst case prediction produced a 28.41% error rate on a "FAIL" class. Therefore, we can claim that using our SNNs-based algorithm, the predictive performance could be improved by 10 times over a regular ANNs model.

Referring to the initial experiments with multiple regression, we can also see that the method gave a high error rate for fault prediction than SNNs (32.35% compare to 2.57% in worst case).

Table 1 The comparison of error rate on fail class.

Methods	% Error on "FAIL" class
Multiple Regression	26.47 – 32.35
ANNs	6.72 – 28.41
SNNs	0.61 – 2.57

From these results, we can thus conclude that our version of SNNs gave a favorable prediction results with very low error rates. However, the data records obtained for our initial experiments were from 7 consecutive manufacturing days. Low variation in environments, specifications, machine configurations were observed. The prediction can thus be performed accurately. In order to deploy our model on the actual manufacturing floor, several parameters still need fine tuning. Moreover, the execution of the training phase was still rather slow and a parallel computation model should be designed and applied in the future.

6. Conclusions and Future Work

In this paper, we described a version of SNNs-based algorithm with the implementation of the genetic algorithm to solve the problem related to manufacturing yield prediction. Our experimental data is the HGA manufacturing data retrieved from an actual data warehouse. In our design and implementation, the number of input nodes can vary depending on the number of root causes identified. Thus, the SNNs can change and a model re-training will be needed periodically. In addition, the random signal is used as extra inputs in order to generate stochastic signals. Each training iteration uses difference random signal set. The sigmoid function is used as an activation function.

We then compared the predictive accuracy among the multiple regression technique, the regular ANNs model, and our version of SNNs-based algorithm. The results showed that the SNNs algorithm outperformed both regressions and ANNs models by more than 10 times, when the error rate was used as a measurement index. The results verify that after the root causes of the yield degradation are identified and the yield improvement solution is suggested, the engineers may use our prediction model to evaluate the solution quality.

However, our version of SNNs is highly compute-intensive. In order to improve the execution time to the point where the program is practical for use on the manufacturing line, a parallel programming technique is explored. Currently, we are designing a data parallel algorithm that can be deployed on a shared memory, multi-core and multi-processor system. The thread programming model with open MP library will be adopted in our implementation plan. We believe that our implementation can benefit HDD manufacturers in the future.

7. Acknowledgement

This work is supported by the Industry/University Cooperative Research Center in Hard Disk Drive Advanced Manufacturing (I/U CRC in

HDD Advanced Manufacturing), King Mongkut's University of Technology Thonburi.

We would like to thank the Product Performance Engineering Department, Seagate Technology (Thailand) Ltd., for their consultations.

8. References

1. Huang, L. and Chen, J.C., 2001, "A Multiple Regression Model to Predict In-process Surface Roughness in Turning Operation via Accelerometer", *Journal of Industrial Technology*, Vol. 17, No. 2, pp. 1-8.

2. Savage, M. and Chen, J.C., 2002, "Multiple Regression-Based Multilevel In-process Surface Roughness Recognition System in Milling Operations", *Journal of Technology Studies*, Vol. 28, No. 1, pp. 28-34.

3. Bielecki, J.V., 2003, "Estimating Engineering and Manufacturing Development Cost Risk Using Logistic and Multiple Regression", *M.S. Thesis, Department of Air Force, Air Force Institute of Technology*, Ohio, USA.

4. Wu, T. and Li, C., 2006, "A Multiple Regression Analysis on Safety Performance in University Laboratories", *World Transactions on Engineering and Technology Education*, Vol. 5, No. 3, pp. 449-452.

5. Deco, G. and Parra, L., 1997, "Non-linear Feature Extraction by Redundancy Reduction in an Unsupervised Stochastic Neural Network", *Journal of Neural Networks*, Vol. 10, No. 4, pp. 683-691.

6. Tong, L., Lee, W., and Su, C., 1997, "Using a Neural Network-based Approach to Predict the Wafer Yield in Integrated Circuit Manufacturing", *IEEE Transactions on Components Packaging and Manufacturing Technology*, Vol. 20, No. 4, pp. 288-294.

7. Berry, W. D. and Feldman, S., 1985, *Multiple Regression in Multiple Regression in Practice*, Sage Publications Inc.

8. Mitchell, T.M., 1997, *Artificial Neural Networks in Machine Learning*, McGraw-Hill Higher Education, pp. 81-127.

9. Baba, N. and Yanjun, Y., 2002, "Utilization of Neural Networks & Genetic Algorithms for Constructing a Reliable Decision Support System which Deals Nikkei-225", *Proceedings of 41st SICE Annual Conference*, Vol. 3, pp. 1603-1608.

10. Wong, E., 1991, "Stochastic Neural Networks", *Algorithmica*, Vol. 6, No. 1-6, pp. 446-478.

11. Polvichai, J., Scerri, P., and Lewis, M., 2008, "An Approach to Online Optimization of Heuristic Coordination Algorithms", *Proceeding of 7th International Joint Conference on Autonomous Agents and Multiagent System*, Vol. 2, pp. 623-630.

12. Turchetti, C., Conti, M., Crippa, P., and Orcioni, S., 1998, "On the Approximation of Stochastic Processes by Approximate Identity Neural Networks", *IEEE Transactions Neural Networks*, Vol. 9, No. 6, pp. 1069-1085.

13. Kamitsuji, S. and Shibata, R., 2003, "Effectiveness of Stochastic Neural Network for Prediction of Fall or Rise of TOPIX", *Asia-Pacific Financial Markets*, Vol. 10, No. 2-3, pp. 187-204.

14. Barlow, H.B., 1989, "Unsupervised Learning", *Neural Computation*, Vol. 1, No. 3, pp. 295-311.

15. Meyer-Base, A., Chen, Y., and McCullough, S., 2001, "Hebbian and Anti-Hebbian Learning for Independent Component Analysis", *Proceeding of International Joint Conference in Neural Networks*, Vol. 2, pp. 920 - 925.

16. Tanaka, M., 1996, "Identification of Nonlinear Systems with Missing Data Using Stochastic Neural Network", *Proceeding of 35th IEEE Decision and Control*, Vol. 1, pp. 933-934.

17. Chryssolouris, G., Lee, M., and Domroese,

M., 1991, “The Use of Neural Networks in Determining Operational Policies for Manufacturing Systems”, *Manufactuting Systems*, Vol. 10, No. 2, pp.166-175.

18. Polvichai, J. and Khosla, P., 2002, “An Evolutionary Behavior Programming System with Dynamic Networks for Mobile Robots in Dynamic Environments”, *IEEE/RSJ International Conference Intelligent Robots and System*, Vol. 1, pp. 978-983.