ECONSTOR
Make Your Publications Visible.

A Service of
ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Nara Samattapapong; Nitin Afzulpurkar

Article

A production throughput forecasting system in an automated hard disk drive test operation using GRNN

Journal of Industrial Engineering and Management (JIEM)

Provided in Cooperation with:
The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa (ESEIAAT), Universitat Politècnica de Catalunya (UPC)

Suggested Citation: Nara Samattapapong; Nitin Afzulpurkar (2016) : A production throughput forecasting system in an automated hard disk drive test operation using GRNN, Journal of Industrial Engineering and Management (JIEM), ISSN 2013-0953, OmniaScience, Barcelona, Vol. 9, Iss. 2, pp. 330-358,
https://doi.org/10.3926/jiem.1464

This Version is available at:
https://hdl.handle.net/10419/188773

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

https://creativecommons.org/licenses/by-nc/3.0/

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2016 – 9(2): 330-358 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
http://dx.doi.org/10.3926/jiem.1464

A production Throughput Forecasting System in an Automated Hard Disk Drive Test Operation Using GRNN

Nara Samattapapong[1], Nitin Afzulpurkar[2]

[1]*Industrial Systems Engineering, Asian Institute of Technology (AIT) (Thailand)*
[2]*Visiting Faculty in Mechatronics Field of Study, Industrial Systems Engineering Department, School of Engineering and Technology, Asian Institute of Technology (AIT) (Thailand)*

nara@sut.ac.th, afzulpurkar.n@gmail.com

Received: April 2015
Accepted: March 2016

Abstract:

Purpose: The goal of this paper is to develop a pragmatic system of a production throughput forecasting system for an automated test operation in a hard drive manufacturing plant. The accurate forecasting result is necessary for the management team to response to any changes in the production processes and the resources allocations.

Design/methodology/approach: In this study, we design a production throughput forecasting system in an automated test operation in hard drive manufacturing plant. The proposed system consists of three main stages. In the first stage, a mutual information method was adopted for selecting the relevant inputs into the forecasting model. In the second stage, a generalized regression neural network (GRNN) was implemented in the forecasting model development phase. Finally, forecasting accuracy was improved by searching the optimal smoothing parameter which selected from comparisons result among three optimization algorithms: particle swarm optimization (PSO), unrestricted search optimization (USO) and interval halving optimization (IHO).

***Findings*:** The experimental result shows that (1) the developed production throughput forecasting system using GRNN is able to provide forecasted results close to actual values, and to projected the future trends of production throughput in an automated hard disk drive test

operation; (2) IHO algorithm performed as appropriate optimization method better than the other two algorithms. (3) Compared with current forecasting system in manufacturing, the results show that the proposed system's performance is superior to the current system in prediction accuracy and suitable for real-world application.

Originality/value: The production throughput volume is a key performance index of hard disk drive manufacturing systems that need to be forecast. The production throughput forecasting result is useful information for management team to respond to any changes in production processes and resources allocation. However, a practical forecasting system for production throughput has not been described in detail yet. The experiments were conducted on a real data set from the final testing operation of hard disk drive manufacturing factory by using Visual Basic Application on Microsoft Excel© to develop preliminary forecasting system for testing and verification process. The experimental result shows that the proposed model is superior to the performance of the current forecasting system.

Keywords**:** forecasting system, general regression neural network, production throughput, interval halving, hard disk drive manufacturing

1. Introduction

Recently, hard disk drive (HDD) manufacturing systems have become more complex because all manufacturers have implemented fully or partially automated equipment in most production areas. The most complicated area in an HDD manufacturing system is automated test operation, which includes the final quality inspection processes for over 20 testing lines. Each testing line consists of more than 10 automated testing machines, each containing over 2,000 slots for testing individual disk drives. All the testing machines are able to test different hard drives simultaneously using the specific test criteria for each product, which include over 200 categories. Considering these complexities, the development of a monitoring and controlling system for this operation is very difficult and challenging for academic research. In addition to the problem of complexity, the production volume of any automated test operation directly affects the ability to ship the product on schedule.

Generally, delivery time was scheduled by estimating the production throughput of the HDD automated test operation, which is based on a deterministic estimation or derived from the average value of a production parameter. Many parameters are characterized as random values, including the required test time, which varies according to the individual HDD capacity and quality. Furthermore, mixed loading

with different types of HDD configurations present in a single automated testing line is a major cause of bottleneck events inside an automated testing machine. With this inherent complexity in the automated test operation and daily uncertainties in the HDD manufacturing environment, it is difficult to estimate or approximate the throughput of an automated testing machine using traditional methods or even advanced analytical methods (Sukthomya & Tannock, 2005; Azizi, Ali & Ping, 2012). Consequently, the estimation of production throughput for highly complex manufacturing systems has become an important issue.

Several forecasting models have been developed to predict or estimate different production performance indices (Stoop & Bertrand, 1997; Huang, 1999; Sivakumar & Chong, 2001; Shanthikumar, Ding & Zhang, 2007; Hillberg, Sengupta & Til, 2009; Pradhan & Damodaran, 2009). Only a few forecasting models have been developed for predicting production throughput, and none of these address the forecasting of production throughput in HDD manufacturing. A subset of the reviewed papers deals with other production performance index predictions, such as yield rate, cycle-time, flow time, overall equipment effectiveness, production time, and completion time.

Forecasting systems for production throughput have been tested for small production system using an approximate method and analytical formula (Baker & Powell, 1995; Popova & Wilson, 2000; Blumenfeld & Li, 2005). However, current production systems have become larger and more complicated with co-utilization of resources and various uncertainties. As a result, the use of an analytical formula and an approximate method has become difficult, and this approach is inadequate for generating accurate forecasts.

Perkinson, McLarty, Gyurcsik and Cavin (1994), Sivakumar and Chong (2001), Backus, Janakiram, Mowzoon, Runger and Bhargava (2006), Shanthikumar et al. (2007), Azizi, Ali, Ping and Mohammadzadeh (2011) and Azizi, Ali, Ping and Mohammadzadeh (2012) have also studied other forecasting models such as simulation models, queuing models, regression, Bayesian methods, data mining, and neural network models. The main disadvantage of these models is consumption of resources, especially in the data collection process before the model is built, and in the computational time, which may require several hours to run, even on a powerful computer. Another problem is the difficulty in modifying the model when the conditions and the assumptions of the actual system are changed.

Although these models can be used as prediction models for handling both linear and nonlinear relationships, the development process is rather difficult and impractical for the current research problem. Additionally, the accuracy of this prediction model fits only the specified data set. Thus, this approach results in overfitting and an inability to generalize for another input data set (Huang, 1999). Furthermore, the calculation time of the forecasting model is also important and should focus on current research. (Sun, Lang, Wang & Liu, 2014)

These previous forecasting models can also only provide precision for the data set used during the model development. Unfortunately, these models are impractical to manage real production due to rapid changes in production data and other variables in current manufacturing lines, as mentioned previously. Therefore, it is not feasible to use these historical forecasting models because none of them are able to produce the required precision in the acceptable range over time. Moreover, the development of a new forecasting model may require a considerable amount of time and may not produce the forecasted result for a decision making process in time to adjust production plan in response to changes. (Van Til, Hillberg & Sengupta, 2003; Chien, Hsiao, Meng, Hong & Wang, 2005; Li, Fang, Liu & Juang, 2012; Chien, Hsu & Hsiao, 2012)

In this work, production throughput is defined as the number of HDDs tested per hour. Production throughput of test operation in an HDD manufacturing line directly affects on-time delivery. Production throughput also measures the production capacity of each product and is the most important index for controlling and monitoring shop floor operations in HDD manufacturing, as mentioned earlier.

Therefore, this article proposes a forecasting system for production throughput in an automated HDD manufacturing test operation which can be practically implemented. The proposed forecasting system includes three main stages: the input variable selection stage, the forecasting model construction stage, and the forecasting accuracy improvement stage. The proposed system will forecast production throughput using historical shop floor data taken from the actual database servers of an HDD manufacturing factory.

The remainder of this paper is arranged as follows. In Section 2, the proposed forecasting system and the theoretical background of the selected forecasting methodology are presented. Section 3 explains the details of the forecasting system development process. Section 4 presents the results of a comparative study among three selected optimization methods for improving of forecasting accuracy which are conducted on the actual data from a real HDD manufacturing, to demonstrate the pros and cons of the selected methods and to illustrate the applicability and accuracy of each approach. Section 5 presents discussion and conclusions for further study and for practical applications.

2. Methodology

As described previously, we have identified a model for developing a forecasting system for the production performance index, in which only production throughput is emphasized as a measure that can be feasibly used in a real factory. Therefore, the aim of this research is to develop a forecasting system with three important characteristics. First, the system must be able to decide on and select the input data

that are most viable to the forecasted result. Second, the system itself must be able to automatically select the format or parameters of the forecasting model. Lastly, in the future, if the system outputs a highly imprecise forecast, it will be able to immediately and automatically adjust the forecasting model.

To accomplish the above objectives, the proposed approach in this research will develop a forecasting system by following three main stages: the input variable selection stage, the forecasting model construction stage, and the forecasting accuracy improvement stage, as shown in Figure 1.

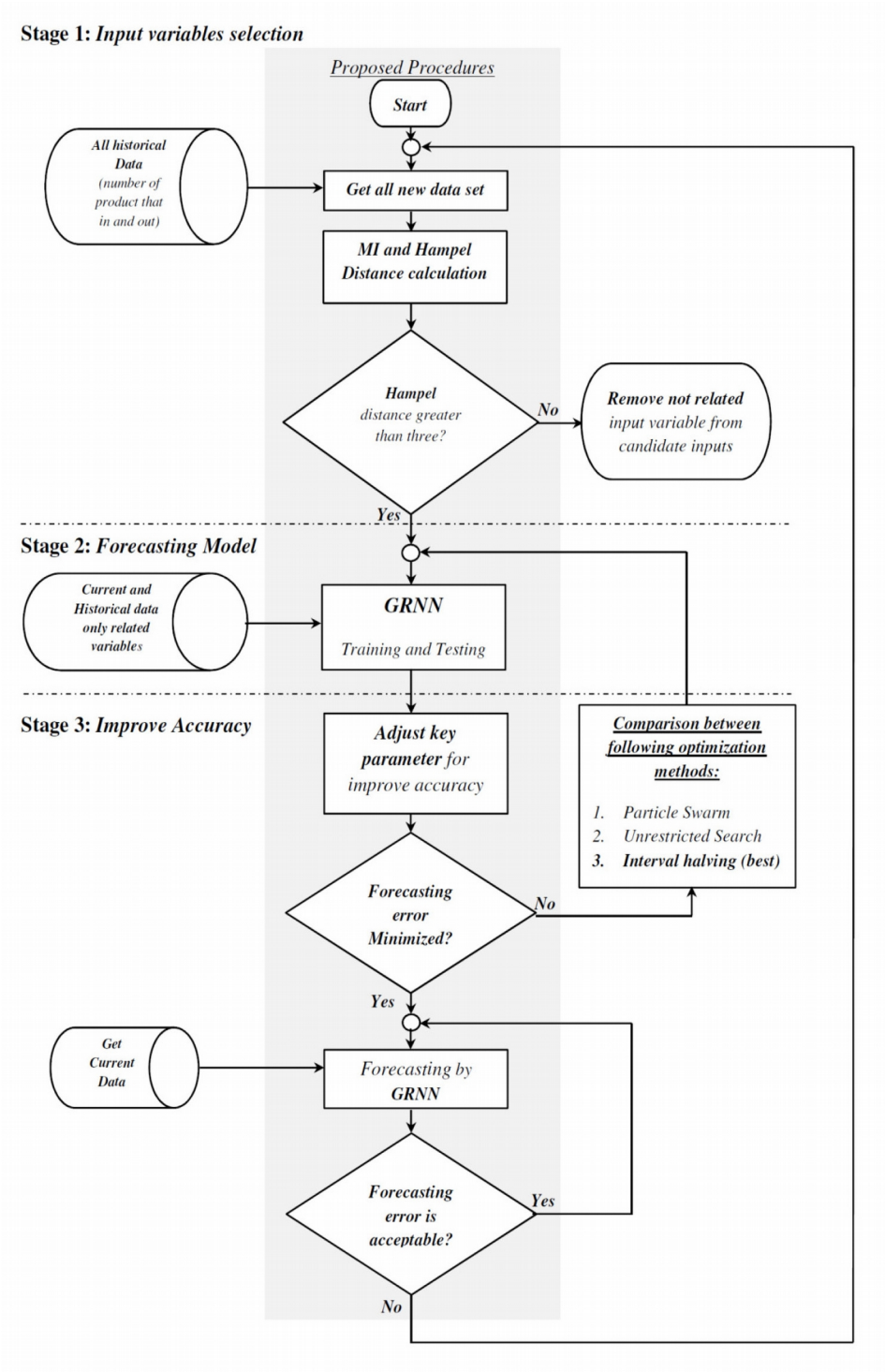


Figure 1. Proposed procedure for development of a forecasting system

As shown in Figure 1, all historical data from a factory database are collected by the system in the first stage. These data include the number of products that enter and exit the process during various time periods (e.g., by hour, shift, and day). Next, each of the input variables enters the screening process, which is carried out using the mutual information (MI) method.

The MI method (Shannon, 1948) was chosen as an appropriate method for input variable selection in this study because it contains the required characteristics. MI is able to select variables that have either linear or non-linear correlation, and therefore, determination of the correlation before selection is not necessary. This method is also able to screen all data sets quickly without requiring pre-processing of inputs. The applied theory and calculation procedures are elaborated in the next section.

2.1. Mutual Information (MI)

The mutual information method does not require the development of a forecasting model before use and can be considered a model-free method (unlike a model-based method, which requires the model to be developed beforehand). The model-free method is simpler because it requires less time to screen the data because there is no need to run a model. MI has been developed based on the principles of information theory and the notion of entropy proposed by Shannon (1948). The mutual information equation for bivariate data is shown below.

$$MI = \frac{1}{N}\sum_{i=1}^{N} \ln\left[\frac{P_{x,y}(x_i, y_i)}{P_x(x_i), P_y(y_i)}\right] \tag{1}$$

where x_i and y_i are the bivariate sample pair, N is the sample size, $P_{x,y}$ (x_i, y_i) is the join probability density at the sample point, and P_x (x_i) and P_y (y_i) are the univariate marginal probability densities at the sample point, respectively. Equation (1) shows that probability density is necessary to determine the MI score. As a result, Gaussian kernel density estimation was chosen as the appropriate and applicable method because it is more stable and better suited for calculation efficiency. The equation for kernel density for the multivariate density function is shown below.

$$\hat{F}_x(x) = \frac{1}{N(2\lambda)^d}\sum_{i=1}^{N}\prod_{j=1}^{d} e^{(-|x_j - x_{ij}|/\lambda)} \tag{2}$$

which can be simplified into the following equation.

$$\hat{F}_x(x) = \frac{1}{N(2\lambda)^d}\sum_{i=1}^{N} exp\left[-\frac{1}{\lambda}\sum_{j=1}^{d}|x_j - x_{ij}|\right] \tag{3}$$

where $\widehat{F}_x(x)$ is the multivariate kernel density estimation, d is the dimension of the variable, N is the sample size, and λ is a smoothing parameter that can be estimated by the following equation.

$$\lambda = \left(\frac{4}{d+2}\right)^{1/(d+4)} N^{(-1/(d+4))} \tag{4}$$

Then, substitution of all values in Equation (1) is used to obtain the MI score for each variable. This MI score is then applied in the next step of the input variable selection method. The principle of Hampel distance is applied in this work using the following calculation principle (May, Maier, Dandy & Fernando, 2008). Hampel distance is defined as follows.

$$Hampel\ Distance = \frac{(MI_i - MI_{0.5})}{s} \tag{5}$$

where s represents the modified Z-score scale and is defined as $s = 1.4826 * median\{|MI_i - MI_{0.5}|\}$, and $MI_{0.5}$ = the median value of $\{MI_i\}$. After obtaining the Hampel distance for each variable, those variables that have a Hampel distance greater than three are selected because they are considered to be crucial for the forecasting output.

This approach to detecting data outliers was proposed by Fernando, Maier and Dandy (2009). The MI value can be calculated from the statistical relationship between the input and output variables as described in Equations (1) - (4).

Next, the Hampel distance is calculated using Equation (5). If an input variable has a Hampel distance greater than three, it will be removed from the data set.

In the second stage, all the data for the selected variables are used to train and test the forecasting model using the neural network approach. Actually, in the literature relating to the application of the neural networks in forecasting models (Chtioui, Panigrahi & Francl, 1999; Cigizoglu & Alp, 2006; Firat & Gungor, 2009; Turan & Yurdusev, 2009), only two architectural formats are considered to be appropriate for the development of a forecasting model. One is the multilayer perceptron (MLP) format, which is a time-lag feed-forward neural network (TLFN), and the other is the generalized regression neural network (GRNN) format.

Both of these approaches have advantages and limitations. Considering the main objective of this research, the TLFN is more complicated in terms of programming and calculation methods and is also highly time-consuming for identifying the appropriate architectural format for the forecast. In contrast, the GRNN contains a general architectural format that requires less time for developing the forecasting model from all available input data. However, certain parameters are required to improve the precision of the forecast. The details of the GRNN are discussed in the next section.

2.2. Generalized Regression Neural Network (GRNN)

The GRNN was first proposed in a study by Specht (1991) and can be considered as a type of probabilistic neural network that is different from other types of neural networks. The GRNN does not require an iterative training procedure because it can formulate the forecasting model from a large amount of input data in a single run. The error in the forecasting results is also consistent. The architectural structure of the GRNN consists of four layers: the input layer, the pattern layer, the summation layer, and the output layer, as shown in Figure 2.

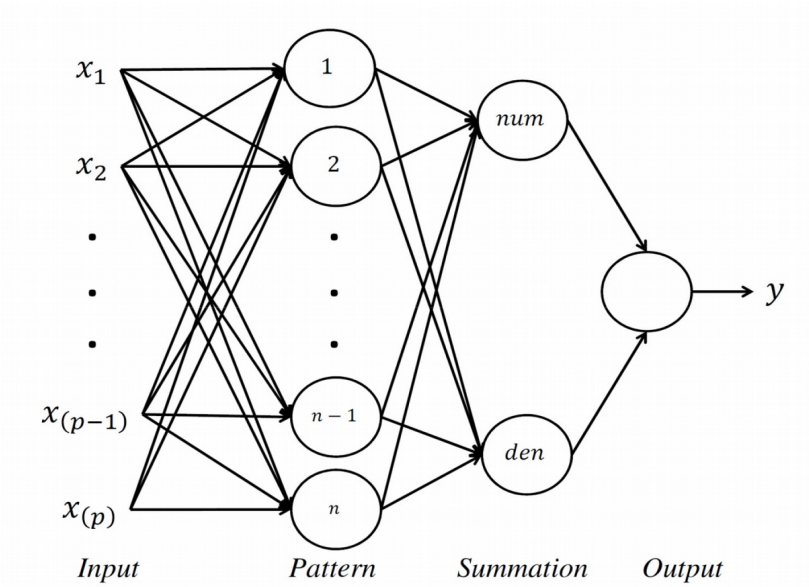


Figure 2. GRNN architecture

The input layer is a layer of the input parameters obtained from the selection method and is linked with the pattern layer. The pattern layer is a layer of the forecasting model that is used for testing and consists of the output data collected in the past. The pattern layer is linked to the summation layer in which the numerator and denominator of the forecasting equation are calculated. This calculated result is the Y value in the output layer, which is the last layer, as shown in Equation (6).

$$\hat{Y}(x) = \frac{\sum_{i=1}^{N} Y_i exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{N} exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \tag{6}$$

where σ is the smoothing parameter (sigma- weight), N is the sample size, and $D_i^2 = (x - x_i)^T(x - x_i)$ is the Euclidean distance. A randomized initial value for the smoothing parameter was adopted in this stage, as shown in Equation (6). The forecasting accuracy is improved in the third stage and the last stage. In the

last stage, the smoothing parameter is adjusted to reduce the forecasting error using an optimization method. As mentioned, the smoothing parameter (σ) must be optimized to ensure maximum precision from the forecasting model.

2.3. Optimization Methods for GRNN

We have studied three optimization methods selected based on their initial characteristics, which are primarily consistent with the research objectives: simple, less time-consuming, and more practical that mean should move towards the best solution with requires only the common control parameter. These three methods are: 1) particle swarm optimization, 2) the unrestricted search method, and 3) the interval halving method. The calculation procedures for each method are explained in detail by Rao (2009) and are summarized below.

2.3.1. Particle Swarm Optimization

Particle swarm optimization (PSO) mimics the behavior of a colony or swarm of insects (e.g., ants or bees) or a flock of birds. For example, each particle represents a bird in a flock or a bee in a swarm. Each particle in a swarm behaves using its own knowledge as well as the group intelligence. If one particle finds a good path to food, the rest of the swarm will follow that path instantaneously, even if their locations in the swarm are far away. The general concepts of PSO are listed below.

1. Each particle is located initially at random.
2. Each particle is assumed to have two characteristics, a position and a velocity.
3. Each particle travels throughout the design space and remembers the best position.
4. The particles inform each other of good positions and adjust their individual positions and velocities to follow the best position.

2.3.2. Unrestricted Search Method

The unrestricted search method uses the concept of the range of an optimum solution that is unknown in the most pragmatic optimization problem. As a result, the search steps must be carried out with no restrictions on the values of the variables to find the optimal solution. The simplified steps of the unrestricted search method are described below.

1. Initial estimate point = x_1
2. Find $f_1 = f(x_1)$
3. Set a step size = s and an accelerated step size = a; find $x_2 = x_1 + (a \times s)$
4. Find $f_2 = f(x_2)$
5. If $f_2 < f_1$ then $x_i = x_1 + (i - 1)s$. This process continues until $f(x_i)$ increases; after that, $x_{(i-1)}$ or x_i can be taken as the optimum.
6. If $f_2 > f_1$ then the search direction will be reversed by $x_i = x_1 - (i - 1)(a \times s)$. this process continues until $f(x_i)$ increases; after that, $x_{(i-1)}$ or x_i can be taken as the optimum.

2.3.3. Interval Halving Method

In the interval halving method, approximately one-half of the range of the current interval will be deleted in every iteration of the process. The simplified procedure can be described as follows:

1. Divide the initial interval of the restricted range $L_0 = [a,b]$ into four equal parts and label the middle point as x_0 and the quarter-interval points as x_1 and x_2.
2. Calculate the evaluation function of $f_1 = f(x_1)$, $f_0 = f(x_0)$ and $f_2 = f(x_2)$.
3. Delete one half of the interval depending on f_1, f_0 and f_2 values.
4. Stop when $(b\text{-}a)$ is less than or equal to the target value.

Iteration of the optimization process in the last stage will continue until the forecasting error has reached a minimum point. At this instant, the forecasting model will promptly forecast the output. While the forecasting model is in use, a periodic check process will operate to determine whether the forecasting error is in the acceptable range. The forecasting model will continue to run while the forecasting error is in range. Otherwise, the forecasting process will be halted and will return the first procedure of Stage 1.

The proposed forecasting system differs from previous forecasting systems in the following ways.

- The forecasting system using GRNN is suitable for real world application, especially in HDD manufacturing environment, unlike the other throughput forecasting systems that use another forecasting model in their applications.
- The optimization algorithm using the interval halving method is simple and applicable, and only requires a short run time to find optimal parameter.

- The proposed system is developed using real manufacturing data and is based on useable conditions so that it can be used as a real-time monitoring system.
- The model can self-formulate and adjust (via diagnostics) to the required accuracy threshold within a short time period and does not need to go off-line to construct new model architectures.

3. Development of a Production Throughput Forecasting System

To improve the applicability and performance of the proposed procedure, the forecasting system was developed using the Visual Basic Application (VBA) programming in Excel because of the ease of importing data from the factory database and the simplicity of programming all algorithms using the cell operation format. Furthermore, VBA is still widely used for production planning system in the factory for several reasons including the ability to perform rapid prototyping and especially in familiarity of production planner (McKay & Black, 2007), so new production planning software will not be required to implement this system. The forecasting system can be separated into three modules, as described in the following subsections.

3.1. Input Selection Module

The input selection module was developed to calculate the MI score and the Hampel distance for the input variable selection process. A screen capture of the module output is shown in Figure 3. The highlighted areas are the crucial input variables that have Hampel distances greater than three. The details of the algorithm in this module are shown below.

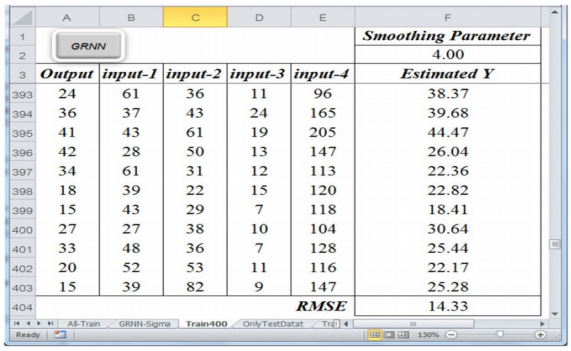
	A	B	C	D	E	F	G	H
1								
2		*Sepecific Value for Calculation*						RUN
3		*No. of Variable*		*Lamda d = 1*		*Lamda d = 2*		
4		*504*		*0.305141755*		*0.35448259*		
5		*Lag*	*Input*	*Output*	*Calculation for Xi*			
6		*t*	*X(t)*	*Y(t)*	*Candidates*	*MI*	*dj*	*Hampel*
7		1	2	0	X(t-26)	1.9204	0.8291	4.5297
8		2	0	0	X(t-25)	1.8348	0.7434	4.0618
9		3	0	0	X(t-27)	1.8284	0.7371	4.0272
10		4	0	0	X(t-28)	1.7525	0.6612	3.6122
11		5	0	0	X(t-24)	1.7331	0.6418	3.5063
12		6	39	0	X(t-29)	1.6902	0.5989	3.2721
13		7	361	0	X(t-23)	1.6777	0.5864	3.2037
14		8	199	0	X(t-22)	1.6247	0.5334	2.9140
15		9	36	0	X(t-30)	1.6223	0.5309	2.9007
16		10	7	0	X(t-31)	1.5651	0.4738	2.5885
17		11	20	0	X(t-21)	1.5596	0.4682	2.5582
18		12	17	0	X(t-32)	1.5501	0.4588	2.5065
19		13	0	0	X(t-33)	1.5452	0.4538	2.4796

Figure 3. Output screen capture of the input selection module

1. Count the number in the data set
2. For each input variable x
3. Calculate smoothing parameter (λ) using Equation (4)
4. Estimate multivariate kernel density ($\hat{f}_x(x)$) using Equation (3)
5. Calculate the *MI* score using Equation (1)
6. Calculate the *Hampel distance* using Equation (5)
7. If (Hampel distance > 3), then
8. Select that input variable for the forecasting module
9. End if
10. Repeat Steps 3-9 for all input variable x values

3.2. Forecasting Module

The forecasting module developed to train and test the GRNN algorithm was implemented to forecast the output with a randomly selected initial smoothing parameter as shown in the screen capture of Figure 4. The details of the algorithm in this module are shown below.



GRNN

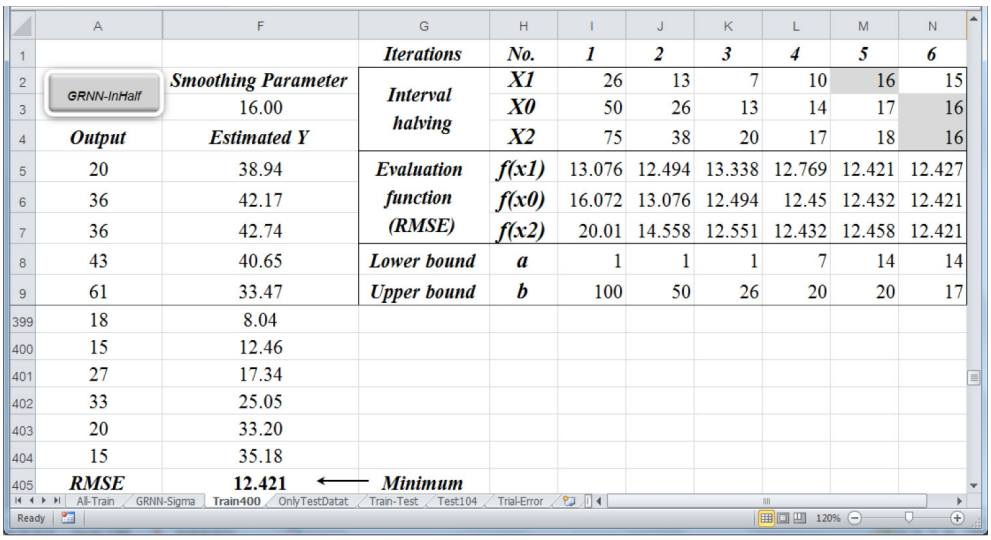
	Output	input-1	input-2	input-3	input-4	Estimated Y
Smoothing Parameter						4.00
393	24	61	36	11	96	38.37
394	36	37	43	24	165	39.68
395	41	43	61	19	205	44.47
396	42	28	50	13	147	26.04
397	34	61	31	12	113	22.36
398	18	39	22	15	120	22.82
399	15	43	29	7	118	18.41
400	27	27	38	10	104	30.64
401	33	48	36	7	128	25.44
402	20	52	53	11	116	22.17
403	15	39	82	9	147	25.28
404					RMSE	14.33

Figure 4. Output screen capture of the forecasting module using GRNN calculation

1. Choose the initial random smoothing parameter (σ)
2. For all input data in the set (number of the collected data set)
3. For all input variables from the 1st module
4. Calculate the Euclidean distance $D_i^2 = (x - x_i)^T (x - x_i)$
5. Next input variable
6. Sum up for only part of the numerator in Equation (6)
7. Next data set
8. Use final summation result of step 6 to estimate $\hat{Y}(x)$ using Equation (6)

3.3. Accuracy Improvement Module

In the accuracy improvement module for the comparative studies, three optimization algorithms were developed to find the ideal smoothing parameters to minimize forecasting error: particle swarm optimization, the unrestricted search method, and the interval halving method. Of the three, the interval halving method showed the best result (the interval value is the smoothing parameter), so only the screen capture of the interval halving method is shown in Figure 5. The highlighted cells shown in column N, rows 3 and 4, represent the optimum points of the algorithm.



	A	F	G	H	I	J	K	L	M	N
1			Iterations	No.	1	2	3	4	5	6
2	GRNN-InHalf	Smoothing Parameter	Interval halving	X1	26	13	7	10	16	15
3		16.00		X0	50	26	13	14	17	16
4	Output	Estimated Y		X2	75	38	20	17	18	16
5	20	38.94	Evaluation function (RMSE)	f(x1)	13.076	12.494	13.338	12.769	12.421	12.427
6	36	42.17		f(x0)	16.072	13.076	12.494	12.45	12.432	12.421
7	36	42.74		f(x2)	20.01	14.558	12.551	12.432	12.458	12.421
8	43	40.65	Lower bound	a	1	1	1	7	14	14
9	61	33.47	Upper bound	b	100	50	26	20	20	17
399	18	8.04								
400	15	12.46								
401	27	17.34								
402	33	25.05								
403	20	33.20								
404	15	35.18								
405	RMSE	12.421 ←	Minimum							

Figure 5. Output screen capture of the accuracy improvement module using the interval halving method

Since the interval halving method showed the best results in the comparative study section, only the details of this method's algorithm are show below.

1. Divide the initial interval boundary into three values between 1 and 100
2. For all three initial boundary values
3. Calculate the evaluate function (RMSE value)
4. Store each RMSE for comparison
5. Next boundary values
6. Compare each RMSE of the three boundary values
7. Set a new outer boundary that is smaller than the initial value from the comparison
8. If (*at least two of three boundaries are equal*), then
9. Set the optimal value as the boundary point
10. Else
11. Go back to Step 2
12. End if

4. Evaluation of Forecasting System

In this section, we describe the data used for the study, and report the results from all three optimization algorithms in the forecasting models. Finally, we evaluate the results using the three criteria, and comparisons are made among them.

4.1. Data Description

The data used to develop the forecasting model in this research were obtained from a hard disk drive manufacturing facility in Thailand. Only data from the automatic test operation were collected, because the test process is considered to be the crucial step for product quality testing before sending the unit to packaging and shipping.

The testing process operation is quite complex because it uses an automatic testing machine system. Additionally, the transfer of products in this process requires a production conveyor and autonomous

robots. The testing machine is also used for testing other products, which complicates the situation even further.

In addition to the complexity of this automatic testing system, the nature of this industry is such that at least 100 products of various types and storage capacities must be tested, all require different testing periods and conditions. As a result, operations with different testing periods have resulted in increasingly complex product testing procedures, thereby rendering operational management even more difficult.

This management difficulty also includes the need to predict the production throughput rate, or the testing throughput rate, as stated in this research. This performance index is critical and affects the product shipping date, which directly impacts customer satisfaction. Therefore we have selected data from a production testing unit in this hard drive manufacturing facility to develop the forecasting model.

The data being used to develop this forecasting model are taken from the overall data recorded by the factory's data management system. The system collects data on a number of products that are fed into various process lines for testing and the number of tested products. The tested products are classified as 'Pass' or 'Fail', and the data are collected hourly, as shown in the sample data in Table 1. The first column on the left side of the table shows the time period in which the data were collected, and the subsequent columns show the number of hard drives fed into the testing unit and the number of hard drives that were tested, respectively, from the first process through the last.

Time	1st Testing Process			2nd Testing Process			3rd Testing Process		
	Input	Out Pass	Out Fail	Input	Out Pass	Out Fail	Input	Out Pass	Out Fail
12-1PM	62	43	3	31	65	2	44	59	1
1-2PM	65	60	2	28	65	3	65	57	3
2-3PM	113	61	3	22	64	1	72	17	5
3-4PM	233	110	6	23	51	0	51	28	9
4-5PM	180	54	0	24	42	1	48	43	4
5-6PM	47	53	1	12	47	1	48	38	2
6-7PM	218	50	2	3	36	1	54	28	2
7-8PM	165	18	6	0	34	0	44	19	7
8-9PM	105	4	7	1	27	2	45	35	5
9-10PM	78	0	3	22	22	1	36	32	2

Table 1. Real input data set sampled from manufacturing facility

In addition, the sample data from one product show the relationship between the drive input numbers and the drive output (pass) numbers as depicted in Figure 6.

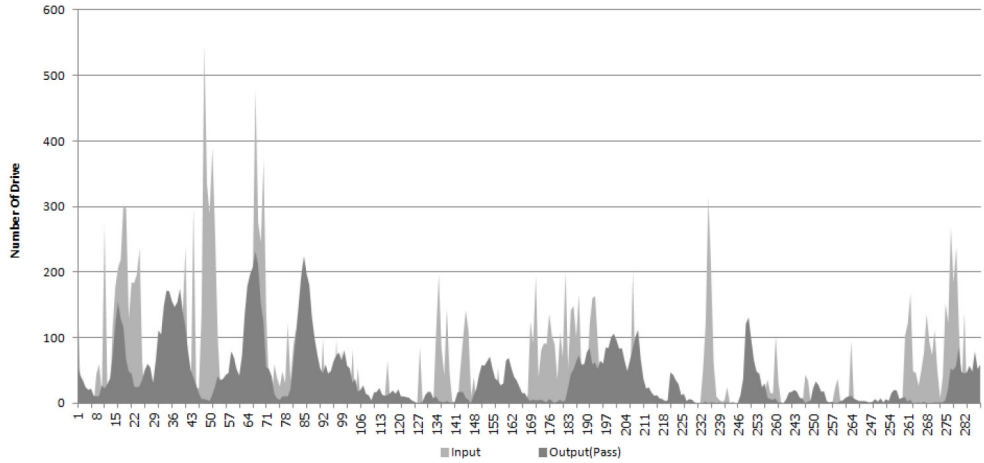


Figure 6. Drive input numbers and output (pass) numbers for each hour

In developing the forecasting model, the main purpose was to minimize the complexity of these processes in an automatic testing machine by using data from various products that can be tested by this single testing machine. Unfortunately, these products require different numbers of testing processes, ranging from two processes up to five processes. As a result, the total testing time for each product is not equal.

To minimize the complications inherent in developing a forecasting model specifically for each product, we assumed that all automatic testing machines in the testing department were a "black box" with each individual function at each time point not completely understood. Based on this assumption, only the data from the number of products fed into the testing department at a particular hour will remain as input data for the forecasting system, as shown in Figure 7.

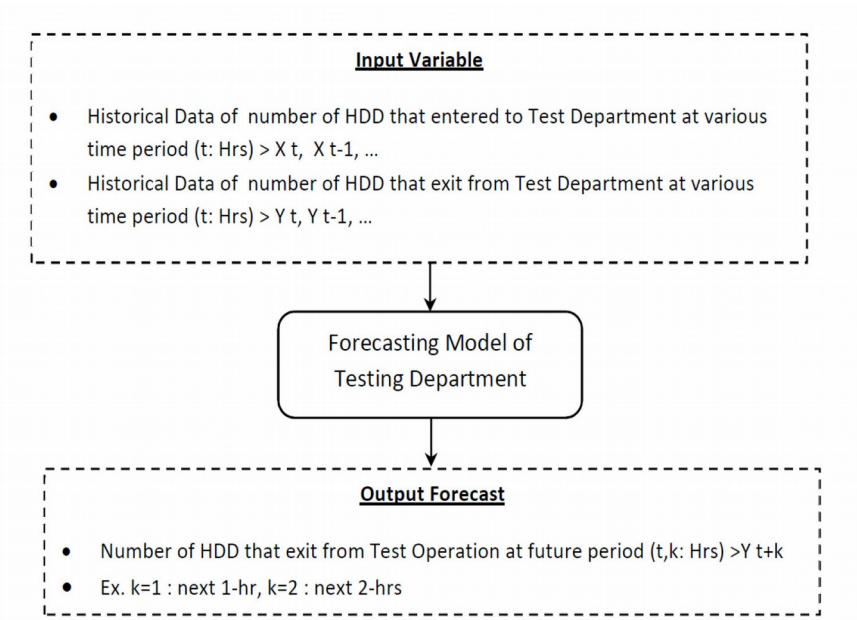


Figure 7. Input and output variables for the forecasting model

This simplifying assumption reduces the complexity of the forecasting system by minimizing the amount of primary input data required for the system and also generalizes the forecasting system. Generalization allows the testing of the forecasting ability of the system with all products, thereby ensuring that this developed system will be applicable for real manufacturing situations, which differs from forecasting models developed in the past. Applicability of the forecasting model is the main goal of this research.

4.2. A Comparative Study of the Three Optimization methods

Comparison of the results obtained from the three methods for finding an optimal smoothing parameter shows that the most appropriate method is the interval halving method. The numbers and characteristics of the optimized values obtained by each method are shown in Table 2 through Table 4.

4.2.1. Particle Swarm Optimization (PSO)

The particle swarm optimization algorithm was adopted to find the best smoothing parameter for minimizing forecasting error. The step-by-step details of the PSO algorithm for the comparison study are presented below and the results from the PSO algorithm are shown in Table 2.

Step 1: Choose the number of particles = 4 (to see swarm behavior)

Step 2: Set each initial particle value randomly (between 1 and 100)

Step 3: Calculate the evaluation function values for each particle

Step 4: Set the initial velocity of each particle = 0 (Table 2, cells B15:B18)

Step 5: Find P_{best} and G_{best} (Table 2, rows 20 to 23, in bold)

Step 6: Add one to the iteration number

Step 7: Find the velocities of the particles

Step 8: Find the new values for each particle

Step 9: Repeat Step 3 through 8 until all particle values converge

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
2		**Iteration**																	
3		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
4		**No. of Particles = 4**																	
5		2	**14.27**	26.54	27.59	18.78	3.53	-6.55	19.74	41.10	37.70	3.88	**-15.78**	-35.43	-40.01	-13.80	11.34	-8.38	
6		25	25.00	21.89	**14.87**	7.85	11.53	**16.74**	21.95	20.20	**15.24**	10.28	11.73	-2.93	-12.90	-7.21	2.01	7.55	
7		50	36.66	**16.84**	-2.98	-8.94	22.98	51.27	22.57	-13.88	-21.83	20.44	56.83	48.07	5.64	-43.69	-69.25	20.73	
8		100	59.99	6.75	-38.68	-42.73	34.36	102.13	20.12	-65.14	-71.49	40.72	122.58	116.98	26.41	-95.41	-164.51	42.24	
9		**Evaluate Values**																	
10		15.12	**12.44**	13.14	13.26	12.49	14.51	13.46	12.54	14.95	14.52	14.37	**12.42**	14.23	14.81	12.46	12.62	13.03	
11		12.97	**12.97**	**12.68**	**12.43**	13.14	12.60	**12.43**	12.69	12.56	**12.42**	12.73	12.58	14.75	12.50	13.29	15.11	13.21	
12		16.07	**14.39**	**12.43**	14.73	12.93	12.77	16.24	12.74	12.45	12.68	12.58	16.99	15.83	13.74	15.28	18.97	12.60	
13		24.65	**17.46**	**13.41**	14.64	15.15	14.09	25.01	**12.56**	18.27	19.37	14.90	27.91	27.21	13.12	23.85	31.44	15.09	
14		**Velocities**																	
15		0	12.27	12.27	1.05	-8.81	-15.26	-10.07	26.29	21.36	-3.40	-33.83	-19.65	-19.65	-4.58	26.21	25.14	-19.72	
16		0	0.00	-3.11	-7.02	-7.02	3.67	5.21	5.21	-1.75	-4.96	-4.96	1.45	-14.66	-9.97	5.69	9.22	5.53	
17		0	-13.34	-19.82	-19.82	-5.96	31.92	28.29	-28.70	-36.44	-7.95	42.27	36.39	-8.76	-42.43	-49.33	-25.56	89.98	
18		0	-40.01	-53.25	-45.42	-4.05	77.09	67.77	-82.01	-85.26	-6.36	112.21	81.86	-5.59	-90.58	-121.82	-69.10	206.75	
19		**Pbest**	**(Best position of each particle)**																
20		2	**14.27**	14.27	14.27	14.27	14.27	14.27	14.27	14.27	14.27	14.27	**-15.78**	-15.78	-15.78	-15.78	-15.78	-15.78	
21		**25**	**25.00**	**21.89**	**14.87**	14.87	14.87	**16.74**	16.74	16.74	**15.24**	15.24	15.24	15.24	15.24	15.24	15.24	15.24	
22		50	**36.66**	**16.84**	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	16.84	
23		100	**59.99**	**6.75**	6.75	6.75	6.75	6.75	**20.12**	20.12	20.12	20.12	20.12	20.12	20.12	20.12	20.12	20.12	
24		**Gbest**	**(Best position of all particle)**																
25		25	**14.27**	**16.84**	**14.87**	14.87	14.87	**16.74**	16.74	16.74	**15.24**	15.24	**-15.78**	-15.78	-15.78	-15.78	-15.78	-15.78	

Table 2. Result from the particle swarm optimization method

The criteria used to evaluate the results of PSO are shown in Table 2. These results show that even though the optimal value can be reach at the 3rd or 4th iteration, the particle values still do not converge, and therefore the stop condition cannot be met at this iteration. Therefore, the algorithm continues to run without the convergence of the particle values after the 7th iteration. Moreover, the G_{best} value (shown in row 25) did not reach the optimal value, continuing to ramp up and progress to a new and impossible optimal point, which demonstrates the main disadvantage of the PSO algorithm in our experiments.

4.2.2. Unrestricted Search Optimization (USO)

The unrestricted search optimization algorithm was implemented to find the smoothing parameter for minimizing forecasting error. This method can be categorized as two search methods, one that searches with a fixed step size and another that searches with an accelerated step size. In the proposed study, we applied USO with accelerated step size due to the additional computational work that would be required if a fixed step size ware adopted. The details of the USO algorithm for the comparison study in this section are presented step-by-step below, and the results are shown in Table 3.

Step 1: Set estimated values for the step size and the accelerated step size

Step 2: Set the initial X1 point (row 6 on Table 3) randomly

Step 3: Find the evaluation value of X1 (row 8)

Step 4: Calculate the X2 point (row 7)

Step 5: Find the evaluation value of X2 (row 9)

Step 6: Compare the evaluation values of X1 and X2

```
If (evaluation value of X2 < evaluation value of X1), then
    Do while (evaluation value of X2 < evaluation value of X1)
        Set X1 = X2
        Set evaluation value of X1 = evaluation value of X2
        Set X2 = X1 + (Accelerated step size * Step size)
        Find the evaluation value of X2
    Loop
Else if (evaluation value of X2 > evaluation value of X1) then
    Do while (evaluation value of X2 < evaluation value of X1)
        Set X1 = X2
        Set evaluation value of X1 = evaluation value of X2
        Set X2 = X1 - (Accelerated step size * Step size)
        Find the evaluation value of X2
    Loop
```

Step 7: Stop, if evaluation value of X2 increases

Step 8: Repeat Steps 1 through 7 for each run

The results of USO are shown in Table 3. Each run applies a different step size and an accelerated step size. The results show that the optimized point with the smallest evaluation value is 16. However, the unrestricted search method displayed inconsistencies in run 2 and run 10, which used the same step size and accelerated size but differed in their number of iterations.

Run No.	1	2	3	4	5	6	7	8	9	10
Step Size	2	1	1	1	1	1	1	1	1	1
Accelerated Size	1	1	2	1.7	1.5	1.4	1.2	1.1	1.1	1
Number of Iteration	28	38	2	3	4	4	5	6	6	71
X1	15	16	23	65	43	60	45	16	11	16
X2	13	15	7	1	1	1	1	37	1	15
Evaluation Value of X1	12.427	12.421	12.776	18.246	15.189	17.458	15.439	12.421	12.65	12.4211
Evaluation Value of X2	12.494	12.427	13.338	4.47E+59	4.47E+59	4.47E+59	4.47E+59	14.4312	4.47E+59	12.4268
Minimum Value	15	16	23	65	43	60	45	16	11	16

Table 3. Result from the unrestricted search optimization method

4.2.3. Interval Halving Optimization (IHO)

The interval halving optimization algorithm was applied to find the smoothing parameter that minimizes the forecasting error. As explained in Section 2, one half of the current interval is deleted at every stage. The details of the IHO algorithm for the comparison study in this section are presented in Section 3.3. The results from the IHO are shown in Table 4; it uses only six iterations, which means that the algorithm is able to reach the optimization point at 16 for minimization of the evaluation value. The result also remains the same even if the runs are repeated many times, demonstrating that the IHO algorithm performs with consistency.

Iterations	*No.*	*1*	*2*	*3*	*4*	*5*	*6*
Interval halving	*X1*	26	13	7	10	16	15
	X0	50	26	13	14	17	16
	X2	75	38	20	17	18	16
Evaluation function (RMSE)	*f(x1)*	13.076	12.494	13.338	12.769	12.421	12.427
	f(x0)	16.072	13.076	12.494	12.45	12.432	12.421
	f(x2)	20.01	14.558	12.551	12.432	12.458	12.421
Lower bound	*a*	1	1	1	7	14	14
Upper bound	*b*	100	50	26	20	20	17

Table 4. Result from the interval halving optimizations method

From the above results for each algorithm, comparing these three methods shows that the most appropriate method is the interval halving method, with comparisons for the three criteria shown in Table 5.

Criterions	Particle Swarm *(PSO)*	Unrestriced Search *(USO)*	Interval Halving *(IHO)*
1. Speed	Bad	Good	Very Good
2. Consistency	Good	Bad	Very Good
3. Practically	Bad	Good	Very Good

Table 5. Comparison of the results among the three of the optimization algorithms

After deciding to use the interval halving method to find the optimal smoothing parameter, the root mean square error (RMSE) was chosen as the evaluation value. The results prove that this approach is able to find an optimal smoothing parameter that gives the smallest forecasting error, as shown in Figure 8.

Smoothing Parameter	RMSE
100	24.65
50	16.07
25	12.97
22	12.69
21	12.62
20	12.55
19	12.50
18	12.46
17	12.43
16	12.42
15	12.43
14	12.45
13	12.49
12	12.56
11	12.65
10	12.77
5	13.96
2	15.12

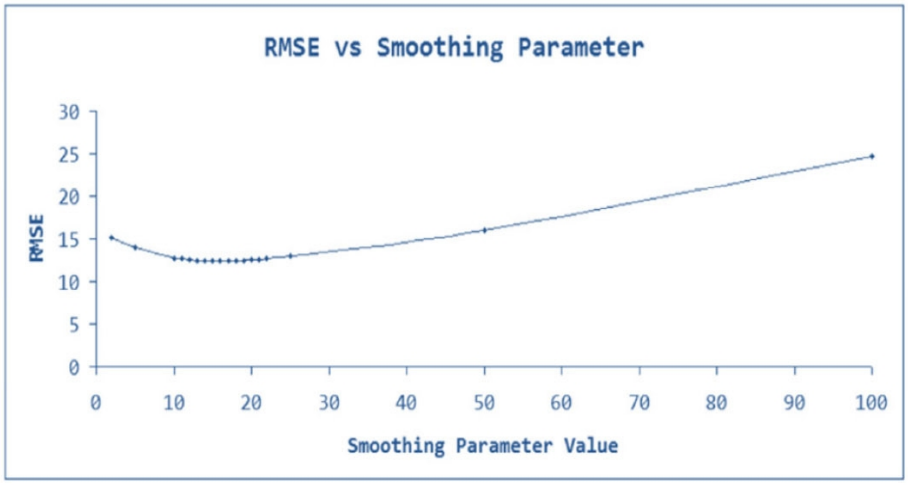


Figure 8. RMSE of the GRNN forecasting model using the IHO method

4.3. Evaluation of Forecasting Performance

The forecasting model and its procedure were developed based on 500 data sets obtained from the manufacture of a particular product in one HDD manufacturing facility in Thailand. The data sets were divided into two parts: 400 sets for training and 100 sets for testing and validation. The forecasting ability of this model is shown in Figure 9. To present the actual forecasting ability, the research team selected five methods to measure the forecasting error and compare the forecasted results with the results from traditional and currently used practical forecasting systems. These five metrics are defined as follows.

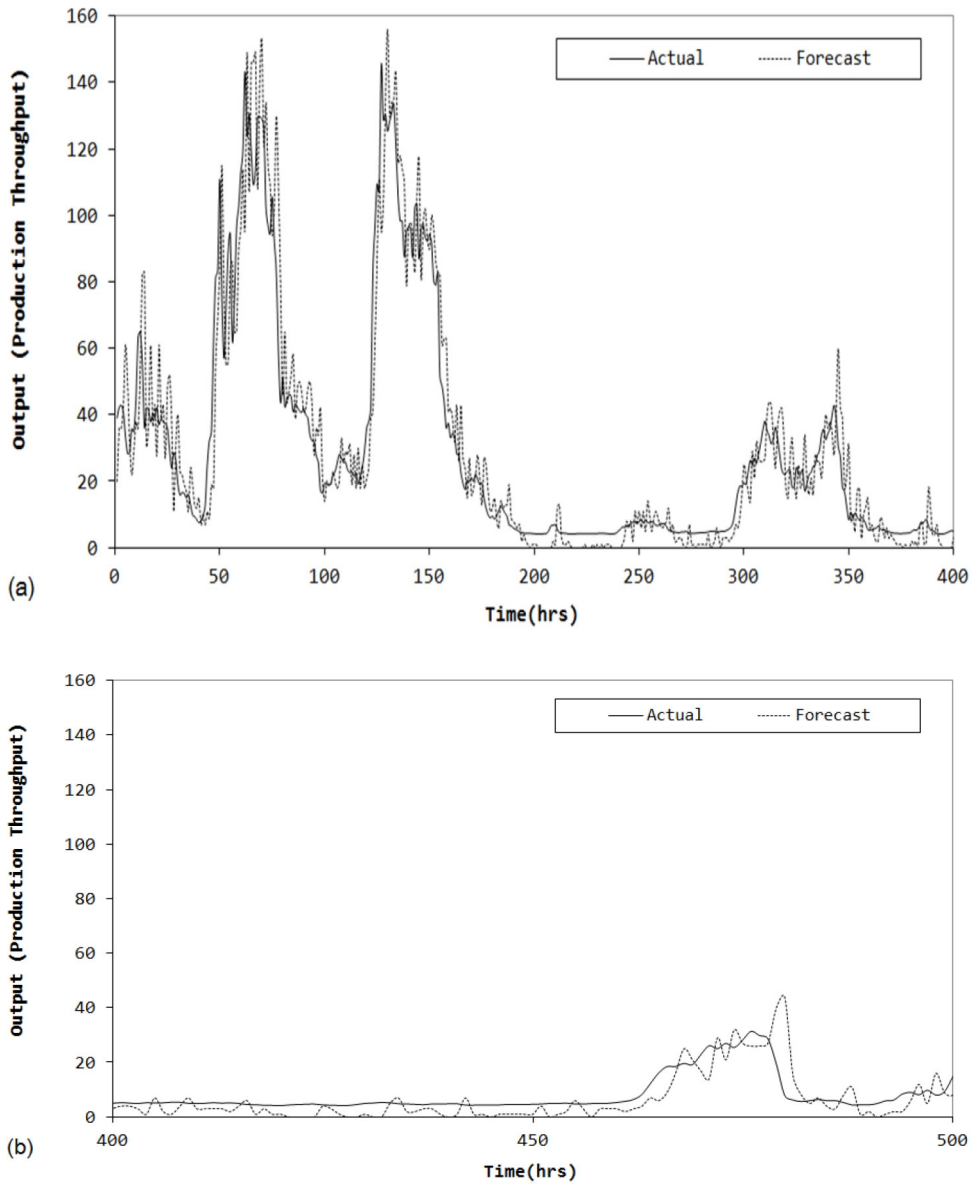


Figure 9. Production throughput forecasting using a IHO-GRNN: (a) training data set, (b) testing data set

Mean Absolute Deviation (MAD)

$$MAD = \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \hat{Y}_t\right| \tag{7}$$

Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2 \tag{8}$$

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2} \tag{9}$$

Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{Y_t} \tag{10}$$

Mean Percentage Error (MPE)

$$MPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left(Y_t - \hat{Y}_t\right)}{Y_t} \tag{11}$$

where Y_t is the actual value and $\hat{Y}_t$ is the forecasted value.

4.3.1. Current Forecasting System in Manufacturing

Currently, in hard disk drive manufacturing, none of the commercial software products for production throughput forecasting are used. The enterprise resource planning (ERP) system is well-known and is used in large-scale industrial manufacturing sites, but only a subset of the ERP modules are used, e.g., the material planning module, the customer services response module, and the financial and accounting module. The production planning and forecasting module is not useful because of difficulties in the compatibility of the modified ERP with each specific manufacturing process, production strategy, and policy for the facility.

This manufacturing case study used only one simple equation.

$$\hat{Y}_t = X_{t-h} \tag{12}$$

where x_{t-h} is the actual number of hard disk drives that entered to the testing machine in the period of hth hour, and h is the testing time of each hard disk drive configuration. That means the manufacturing just simple forecasting production throughput base on testing time only.

The current forecasting systems are shown in Figure 10. The forecast values do not match actual trends, especially between the 30th and 45th hours. The comparisons between the current and proposed systems show that the proposed forecasting system developed in this study produces better forecasting accuracy than the current system for all measurement criteria, as shown in Table 6.

Table 6 shows that the RMSE, MAPE, and MPE values for the proposed system are smaller than those of the current system by nearly a factor of ten. It also shows that the MAD and MSE measurement values are smaller than those of the current system.

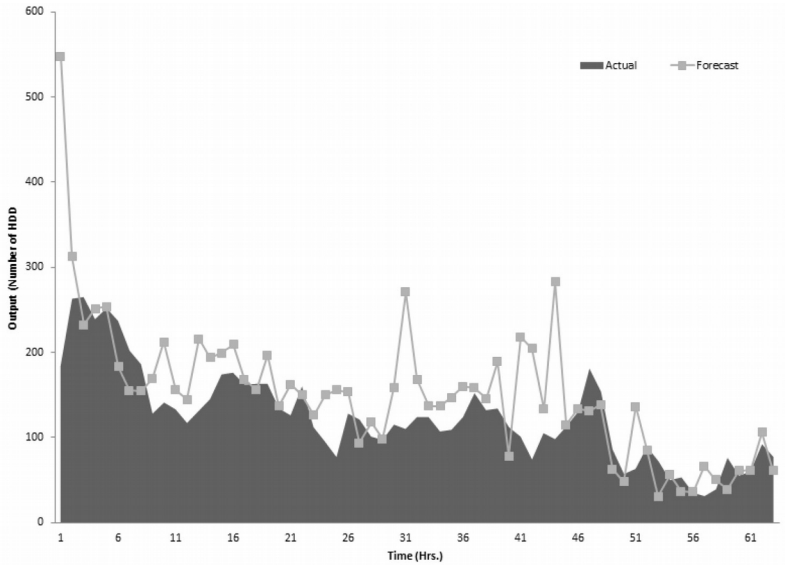


Figure 10. Current forecasting system in HDD manufacturing

Measurement Prediction Errors	Current System	GRNN Interval Halving
MAD	29.9	3.9
MSE	2692.1	33.2
RMSE	51.9	5.8
MAPE	10.5	1.1
MPE	-10.3	-0.9

Table 6. Comparison of the results among the three of the optimization algorithms

5. Conclusions

In this research, a forecasting system for production throughput was developed to create a system that is applicable to real manufacturing situations and can be objectively utilized. Real data from an actual factory were used to develop the system. The complexity of the production process, which uses advanced technology, further increases the difficulty of data collection and analysis. Compared with the existing method and the selected optimization tools used in this comparative study, the proposed system has following advantages:

- The mutual information method is used as the input variable selection method due to its simplicity and speed in working without corrections between data sets.
- The generalized regression neural network method is selected as the architectural format in this study because it takes less time to develop the forecasting model from all input data available.
- The interval halving method is selected as the most appropriate optimization method for the smoothing parameter due to its simplicity, minimal run time, and practicality.
- The principle of this developed forecasting system allows it to automatically modify and select the best input data by using the mutual information method, and to improve its forecasting precision by using a generalized regression neural network integrated with the interval halving method.

The comparative study results show that the developed forecasting system for production throughput is able to provide forecasted results close to actual values, and to project the future trends of production throughput. This information is critical for decision making in production planning and to manage unforeseen problems. This study provides compelling evidence that this system should be used for all forecasting steps. The reason for the seeming disparity is that this system differs from similar systems that have been studied previously.

This study began by using the mutual information method for input selection. This method can be used for all kinds of input data. For the forecasting model, this study proves that GRNN with an interval halving method is appropriate for this kind of forecasting problem. This combination between GRNN and optimization with an interval halving method has not been studied previously.

The following recommendations are proposed regarding the model's applicability. The diagnostic ability of the proposed system relies on the established acceptable error range for the forecast. Once the error exceeds the acceptable range, the system will adjust its forecasted result by starting over from the first step of the input variables selection and retrieve new values for the forecasted result.

However, this self-adjustment capability must not be overly time-consuming. Consequently, when being applied to a real world situation, a high-speed data retrieval system and a large amount of computational work may be necessary to determine the forecasted result.

Acknowledgment

The authors acknowledge receiving the data set obtained in the comparative study and evaluation from information technology department from manufacturing site. Moreover, the authors also thank the engineering department for their valuable comments and suggestions that have led to a substantial improvement of the research. Lastly, the authors greatly acknowledge the financial support from Royal Thai Government with the Joint Industrial project grant with Asian Institute of Technology.

References

Azizi, A., Ali, A., & Ping, L. (2012). Production Throughput Modeling under Five Uncertain Variables Using Bayesian Inference. *World Academy of Science, Engineering and Technology, International Science Index 69, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 6(9), 1876-1883.

Azizi, A., Ali, A., Ping, L., & Mohammadzadeh, M. (2011). A Bayesian Autoregressive Integrated Moving Average Model for Estimating the Production Throughput under Uncertain Conditions: A Case Study. *International Journal for Advances in Computer Science*, 2(4), 5-10.

Azizi, A., Ali, A.Y.B., Ping, L.W., & Mohammadzadeh, M. (2012). Estimating and Modeling Uncertainties Affecting Production Throughput Using ARIMA-Multiple Linear Regression. In *Advanced Materials Research*, 488, 1263-1267. Trans Tech Publications. http://dx.doi.org/10.4028/www.scientific.net/AMR.488-489.1263

Backus, P., Janakiram, M., Mowzoon, S., Runger, G.C., & Bhargava, A. (2006). Factory cycle-time prediction with a data-mining approach. *Semiconductor Manufacturing, IEEE Transactions on*, 19(2), 252-258. http://dx.doi.org/10.1109/TSM.2006.873400

Baker, K.R., & Powell, S.G. (1995). A predictive model for the throughput of simple assembly systems. *European Journal of Operation Research*, 81, 336-345. http://dx.doi.org/10.1016/0377-2217(93)E0283-4

Blumenfeld, D.E., & Li, J. (2005). An analytical formula for throughput of a production line with identical stations and random failures. *Mathematical Problem in Engineering*, 3, 293-308. http://dx.doi.org/10.1155/MPE.2005.293

Chien, C.F., Hsiao, C.W., Meng, C., Hong, K.T., & Wang, S.T. (2005). Cycle time prediction and control based on production line status and manufacturing data mining. In *Semiconductor Manufacturing, 2005. ISSM 2005, IEEE International Symposium*, 327-330. http://dx.doi.org/10.1109/ISSM.2005.1513369

Chien, C.F., Hsu, C.Y., & Hsiao, C.W. (2012). Manufacturing intelligence to forecast and reduce semiconductor cycle time. *Journal of Intelligent Manufacturing*, 23(6), 2281-2294. http://dx.doi.org/10.1007/s10845-011-0572-y

Chtioui, Y., Panigrahi, S., & Francl, L. (1999). A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease. *Chemometrics and Intelligent Laboratory Systems*, 48(1), 47-58. http://dx.doi.org/10.1016/S0169-7439(99)00006-4

Cigizoglu, H.K., & Alp, M. (2006). Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software*, 37(2), 63-68. http://dx.doi.org/10.1016/j.advengsoft.2005.05.002

Fernando, T.M.K.G., Maier, H.R., & Dandy, G.C. (2009). Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. *Journal of Hydrology*, 367(3-4), 165-176. http://dx.doi.org/10.1016/j.jhydrol.2008.10.019

Firat, M., & Gungor, M. (2009). Generalized regression neural networks and feed forward neural networks for prediction of scour depth around bridge piers. *Advances in Engineering Software*, 40(8), 731-737. http://dx.doi.org/10.1016/j.advengsoft.2008.12.001

Hillberg, P.A., Sengupta, S., & Til, R.P.V. (2009). A comparative study of three predictive tools for forecasting a transfer line's throughput. *International Journal of Industrial Engineering*, 16(1), 32-40.

Huang, C.-L. (1999). The construction of production performance prediction system for semiconductor manufacturing with artificial neural networks. *International Journal of Production Research*, 37(6), 1387-1402. http://dx.doi.org/10.1080/002075499191319

Li, D.C., Fang, Y.H., Liu, C.W., & Juang, C.J. (2012). Using past manufacturing experience to assist building the yield forecast model for new manufacturing processes. *Journal of Intelligent Manufacturing*, 23(3), 857-868. http://dx.doi.org/10.1007/s10845-010-0442-z

May, R.J., Maier, H.R., Dandy, G.C., & Fernando, T.M.K.G. (2008). Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modelling and Software*, 23(10-11), 1312-1326. http://dx.doi.org/10.1016/j.envsoft.2008.03.007

McKay, K.N., & Black, G.W. (2007). The evolution of a production planning system: A 10-year case study. *Computers in Industry*, 58(8), 756-771. http://dx.doi.org/10.1016/j.compind.2007.02.002

Perkinson, T.L., McLarty, P.K., Gyurcsik, R.S., & Cavin III, R.K. (1994). Single-wafer cluster tool performance: An analysis of throughput. *Semiconductor Manufacturing, IEEE Transactions on*, 7(3), 369-373. http://dx.doi.org/10.1109/66.311340

Popova, E., & Wilson, J.G. (2000). Adaptive time dynamic model for production volume prediction. *International Journal of Production Research*, 38(13), 3111-3130. http://dx.doi.org/10.1080/00207540050117477

Pradhan, S., & Damodaran, P. (2009). Performance characterization of complex manufacturing systems with general distributions and job failures. *European Journal of Operational Research*, 197(2), 588-598. http://dx.doi.org/10.1016/j.ejor.2008.07.013

Rao, S.S., (2009). *Engineering Optimization: Theory and Practice.* 4th ed. Hoboken, New Jersey: John Wiley and Sons. http://dx.doi.org/10.1002/9780470549124

Shannon, C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379-423. http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x

Shanthikumar, J.G., Ding, S., & Zhang, M.T. (2007). Queuing theory for semiconductor manufacturing systems: A survey and open problems. *IEEE Transactions on Automation Science and Engineering*. 4(4), 513-522. http://dx.doi.org/10.1109/TASE.2007.906348

Sivakumar, A.I., & Chong, C.S. (2001). A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing. *Computers in Industry*, 45(1), 59-78. http://dx.doi.org/10.1016/S0166-3615(01)00081-1

Specht, D.F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6), 568-576. http://dx.doi.org/10.1109/72.97934

Stoop, P.P.M., & Bertrand, J.W.M. (1997). Performance prediction and diagnosis in two production departments. *Integrated Manufacturing Systems*, 8(2), 103-109. http://dx.doi.org/10.1108/09576069710165783

Sukthomya, W., & Tannock, J. (2005). The training of neural networks to model manufacturing processes. *Journal of Intelligent Manufacturing*, 16(1), 39-51. http://dx.doi.org/10.1007/s10845-005-4823-7

Sun, Y., Lang, M., Wang, D., & Liu, L. (2014). A PSO-GRNN model for railway freight volume prediction: empirical study from china. Journal of Industrial Engineering and Management, 7(2), 413-433. http://dx.doi.org/10.3926/jiem.1007

Van Til, R.P., Hillberg, P., & Sengupta, S., (2003). Real-Time Prediction of a Manufacturing System's Throughput. *Preceeding of The 3rd Annual Hawaii International Conference on Business*. June 18-21

Turan, M.T., & Yurdusev, M.A. (2009). River flow estimation from upstream flow records by artificial intelligence methods. *Journal of Hydrology*, 369, 71-77. http://dx.doi.org/10.1016/j.jhydrol.2009.02.004

Journal of Industrial Engineering and Management, 2016 (www.jiem.org)

Article's contents are provided on an Attribution-Non Commercial 3.0 Creative commons license. Readers are allowed to copy, distribute and communicate article's contents, provided the author's and Journal of Industrial Engineering and Management's names are included. It must not be used for commercial purposes. To see the complete license contents, please visit http://creativecommons.org/licenses/by-nc/3.0/.