



## Simulation metamodelling with neural networks: An experimental investigation

Ihsan Sabuncuoglu & Souheyl Touhami

To cite this article: Ihsan Sabuncuoglu & Souheyl Touhami (2002) Simulation metamodelling with neural networks: An experimental investigation, International Journal of Production Research, 40:11, 2483-2505, DOI: [10.1080/00207540210135596](https://doi.org/10.1080/00207540210135596)

To link to this article: <http://dx.doi.org/10.1080/00207540210135596>



Published online: 14 Nov 2010.



Submit your article to this journal [↗](#)



Article views: 73



View related articles [↗](#)



Citing articles: 34 View citing articles [↗](#)

## Simulation metamodeling with neural networks: an experimental investigation

IHSAN SABUNCUOGLU†\* and SOUHEYLA TOUHAMI‡

Artificial neural networks are often proposed as an alternative approach for formalizing various quantitative and qualitative aspects of complex systems. This paper examines the robustness of using neural networks as a simulation metamodel to estimate manufacturing system performances. Simulation models of a job shop system are developed for various configurations to train neural network metamodels. Extensive computational tests are carried out with the proposed models at various factor levels (study horizon, system load, initial system status, stochasticity, system size and error assessment methods) to see the metamodel accuracy. The results indicate that simulation metamodels with neural networks can be effectively used to estimate the system performances.

### 1. Introduction

Simulation has been widely accepted by the scientific community and practitioners as a flexible tool in modelling and analysis of complex systems. It reduces the cost, time and risks associated with the implementations of new designs. However, due to its lengthy computational requirements and the trail-and-error nature of the development process, simulation may not be always the first preferred tool in solving real-life problems. This issue is particularly important for problems where the solution space is very large and when the time available for decision-making is too limited for extensive analysis to be performed (i.e. on-line applications). Harmonosky and Robohn (1995) stated that CPU requirements appeared to be a major obstacle for the on-line applications of simulation. Time considerations are still an important issue, even with the off-line use of simulation due to the increasing complexity of the modern production systems. In this paper, it is argued that the use of simulation metamodels may help alleviate these problems.

A simulation metamodel is a simpler model of the real system. The simulation model is an abstraction of the real system in which a selected subset of inputs is considered. The effect of the excluded inputs is represented in the model in the form of the randomness to which the system is subject. As illustrated in figure 1, a metamodel is a further abstraction of the simulation model. Simulation is used to generate data sets, which in turn are used to build the metamodel. A simulation metamodel with neural networks is a neural network whose training is provided by a simulation model. In general, a metamodel takes a fewer number of inputs

---

Revision received November 2001.

† Bilkent University, Department of Industrial Engineering, Faculty of Engineering, 06533 Ankara, Turkey.

‡ Concordia University, Department of Decision Sciences & MIS, John Molson School of Business, 1455 de Maisonneuve Blvd. W. GM209-11, Montreal, Québec, H3G 1M8, Canada.

\* To whom correspondence should be addressed. e-mail: [sabun@bilkent.edu.tr](mailto:sabun@bilkent.edu.tr)

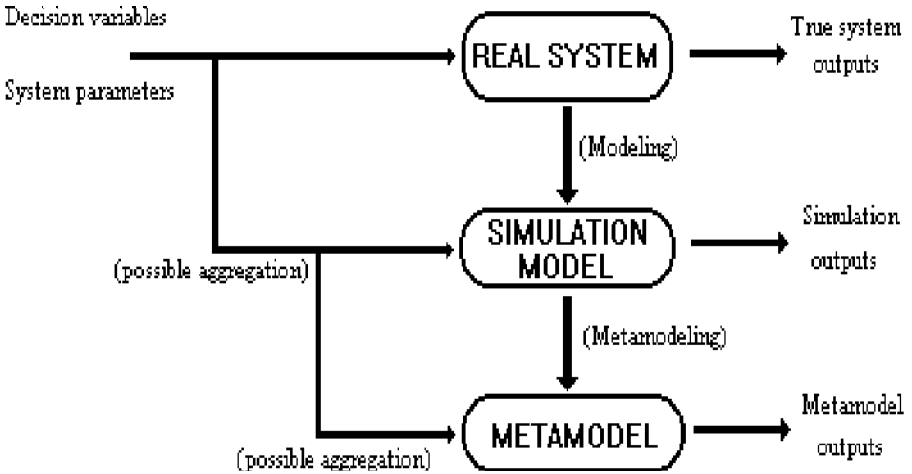


Figure 1. Metamodelling concept.

and is usually simpler than the simulation model. These savings result in lower running times, but at the expense of a reduction of the accuracy of the metamodel with respect to the original system.

Research in metamodelling is maturing. Since the late 1980s, the literature reveals a resurgence of interest in metamodelling in general and specifically in neural network-based metamodels. In this study, we develop and test simulation metamodels with back propagation artificial neural networks for the purpose of estimating system performance measures in job-shop scheduling environments. Extensive computational experiments are also carried out to identify the critical factors that affect the estimation process and accuracy of metamodels. The results provide important insights into the effective use of the simulation metamodels with neural networks.

The paper is organized as follows. Section 2 gives the research background and motivation behind the study. The details of the proposed study and the experimental settings are given in Section 3. The results are presented in Section 4, where the performances of the metamodels are also discussed. Concluding remarks and future research directions are given in Section 5.

## 2. Research Background

The metamodels are used to reduce the computer costs (memory and time) of simulation while making use of its potential of estimating performance of complex systems. Blanning (1975) is probably among the first scientists to propose the use of metamodels to alleviate the problems of simulation. The application of metamodels for manufacturing systems' management has increase since then. As pointed out by Yu and Popplewell (1994), the increases in publications for metamodelling of manufacturing systems indicates that the technique is of value in manufacturing systems' design and analysis. The related bibliography of metamodeling is as follows.

Friedman and Pressman (1988) explored the advantages of metamodelling. These included model simplification, enhanced exploration and interpretation of the model, generalization to other models of the same type, sensitivity analysis, answering inverse questions and a better understanding of the studied system and the

interrelationships of system variables. Barton (1992) provided a review on general-purpose mathematical approximations to simulation input–output functions. According to him, one of the major issues in the design of the mathematical approximation is the choice of a functional form for the output function. The metamodelling candidate approaches include Taguchi models, generalized linear models, radial basis functions, Kernel methods, spatial correlation models, frequency domain approximations and robust regression methods. Barton concluded that while some approaches cannot provide a global fit to smooth response functions of arbitrary shape, the others are computationally intensive and in some cases estimation problems are numerically ill-conditioned. Despite these potential drawbacks, these approaches produced satisfactory results. See McHaney and Douglas (1990) and Watson *et al.* (1994) for sample success stories with regression-based metamodelling. A methodology for fitting and validating regression-based metamodelling were discussed in detail by Kleijnen and Sargent (2000). There are also metamodelling developed by using rule-based expert systems (Pierreval 1995).

The use of neural networks is another approach for metamodelling that has recently emerged. This study aims at investigating the robustness of neural network-based simulation metamodelling. Specifically, it employs back propagation neural networks that belong to the supervised learning category. The major distinguishing feature of back propagation neural networks is *learning* the underlying mappings between the input and output variables from examples (Dayhoff 1990). In a traditional computer program, the programmer specifies every step in advance. The neural network, in contrast, would by itself build the mapping describing the input–output relationship and no programming is required. This is achieved through the learning process. Another important feature of neural networks is *generalization*. Although learning is based only on a limited set of examples, when it comes to applying the neural network model, the network should be able to extend its knowledge to outside this set of examples.

Neural networks have a wide range of applications. Zhang and Huang (1995) discussed the applications of neural networks in general. Burke and Ignazio (1992) reviewed the application of neural networks in operations research. Sabuncuoglu (1998) presented the theory and applications of neural networks in production scheduling. It appears that the interest in neural networks that mainly started from 1987 corresponds to the same time for which a resurgence of interest was noted for the use of metamodelling in manufacturing environments.

Even though neural networks have many success stories, one must recognize that metamodelling with neural networks has some shortcomings (Madey *et al.* 1990). First, constructing a neural network is time consuming since the process requires generating a training set, and empirically selecting an appropriate architecture. Second, the accuracy of the network outputs depends on the regularity of the behaviour of the system under study (by regularity, we mean that the system is subject to the same set of exogenous and uncontrollable factors). This implies that the time horizon of the study must be carefully selected. Third, the validity of the results also depends on the degree of aggregation selected for the input data. Aggregation of data is needed to reduce the size of the neural network and the effort required at generating the examples. This would have a negative impact on the precision of the neural network results. The disadvantages mentioned so far are common to most metamodelling techniques. Another more specific problem related to metamodelling with neural networks is the difficulty of making interpretations and analysis of the

input–output relationship. As mentioned above, the neural network generates its own rules but does not provide them explicitly to the user. To get an insight into the input–output relationship, one needs to analyse the weights of the connections between the processing units. This is not an easy task, and it is time consuming. Thus, providing a formal method to analyse the neural network may strengthen its value as a metamodelling approach. A brief summary of neural network applications to metamodelling is as follows.

Chryssolouris *et al.* (1990) used a neural network metamodel to reduce the computational efforts required in the long trial process associated with using simulation alone for the design of a manufacturing system. Chryssolouris *et al.* (1991) determined the parameters of the operational policy required to achieve some given levels system performances for their neural network metamodel. Mollaghasemi (1998) developed a neural network-based metamodel for the system design problem. Hurriion (1992) employed a neural network to estimate confidence intervals for the performance of an inventory depot. Hurriion (1997) combined the neural metamodel with simulation to speed up the search process in simulation. In a recent study, Hurriion (1998) proposed the visual interactive metasimulation using neural network.

Pierreval (1993) employed neural networks to rank dispatching rules in a stochastic flow shop. In the follow-up study, Pierreval (1996) investigated the ability of neural networks to estimate mean machine utilization of a deterministic small-sized problem. Philipoom *et al.* (1994) employed the neural networks as a simulation metamodel to assign due dates for jobs based on system characteristics and system status when jobs enter system. Badiru and Sieger (1998) proposed neural networks as a simulation metamodel in economic analysis of risky projects. Kilmer *et al.* (1999) use supervised neural networks as a metamodelling technique for discrete-event, stochastic simulation. The authors translated an  $(s, S)$  inventory simulation into a metamodel and estimated the expected total and its variance. They formed confidence intervals from both metamodels and the simulation model. The results indicate that the neural network metamodel is quite competitive in accuracy when compared with the simulation itself. They also showed that one neural network metamodels are trained, they can operate nearly in real-time. Huang *et al.* (1999) employed neural networks for the performance prediction system to help in identifying abnormalities in the system. Finally, Aussem and Hill (2000) proposed the use of neural networks as a metamodelling technique to reduce the computational burden of stochastic simulations. The proposed metamodel was successfully used to predict the propagation of the green alga in the Mediterranean.

To the best of our knowledge, there is not much work reported to compare the different metamodelling approaches. Among a few studies, Philipoom *et al.* (1994) compared the regression models and neural network models on the task of selected due date assignment rules. Their results showed a great success for neural networks compared with the regression models. Hurriion and Birgil (1999) compared two forms of experimental design methods (full factorial designs and random designs) used for the development of regression and neural network simulation metamodels. Their results showed that the neural network metamodel outperform conventional regression metamodels, especially when data sets based on randomized simulation experimental designs are used to produce the metamodels rather than data sets from similar-sized full-factorial experimental designs.

Raaymakers and Weijters (2000) used neural networks to estimate makespan in batch-processing industries. The results of their simulation experiments indicated

that the estimation quality of the neural network models is significantly better than the quality of regression models. There are also opposite results in the literature. For example, Fishwick (1989) found that a neural network model performs poorly when compared with a linear regression model and a surface response model applied to a ballistic system when measuring the horizontal distance covered by a projectile. We believe that both neural networks and regression models are good scientific tools in dealing with estimation problems. They have their own different characteristics that should be studied further in various problem domains. There even may be areas in which both can be used in a complementary fashion to enhance the solution quality. This requires further investigation by researchers. Unfortunately, this is not within the scope of this paper.

In all the papers discussed above, neural network-based metamodelling achieves reasonably good results. Several of the experiments report a percentage error of  $\leq 6$ . In these studies, the researchers showed that neural networks are very promising tools for predicting system measures. However, all these case studies deal with systems of reduced complexities or ones of a deterministic nature and do not allow us to generalize on the estimation capabilities of neural networks. In this study, in fact, we investigate how such factors affect the precision of the neural metamodelling. The objective of this experimental investigation is to determine system characteristics for which the use of neural metamodelling can be expected to produce an acceptable level of accuracy.

### 3. Proposed study

This study considers a job shop manufacturing system. Simulation models and neural metamodelling are developed for the purpose of estimating system performance measures under various system configurations. Three sets of experiments are considered. The first is conducted to estimate the mean machine utilization for a system in steady-state. This is referred to as estimating long-term mean machine utilization. The second set of experiments is similar to the first, except that long-term mean job tardiness is estimated. In these two sets of experiments, the following factors are examined.

- System complexity: simple versus complex.
- Stochasticity: deterministic system, system with stochastic job interarrival times only, system with stochastic job processing times only, or system with both variables being stochastic.
- System load: low, medium or high demand.
- Metamodel error assessment criteria: mean average deviation (MAD) versus percentage error (%error).
- Degradation in precision on test data set relative to the training data set.

The third set of experiments focuses on the short-term (or transient) state performance of the system. It is important to distinguish between long- and short-term system performances. The analysis of the transient state of a system is especially relevant if the system is terminating or if there is no steady-state behaviour due to continuous changes and interruptions in the system operation. In such cases, the time horizon of the analysis is too short for the system to reach an equilibrium state. In this set of experiments, the objective of the metamodelling is to estimate mean job tardiness. In addition to the factors listed above, this set of experiments also exam-

ines the effect of initial system status. This is an important factor because system performance is significantly affected by the starting conditions for short time horizons.

### 3.1. *Experimental settings*

This study considers a job shop system with two levels of complexity: simple and complex. The simple system is the one used by Pierreval (1996). It consists of three job types, four machines and three free transporters. The complex system is an extension of the simple one. It consists of three additional job types and three additional machines. Each job type has a fixed routing in the system with fixed travel distances between machines. Jobs wait in queues according a waiting discipline: shortest processing time (SPT), earliest due date (EDD) or modified operation due date (MODD). At their arrival to the system, jobs are assigned due dates according to the total work content method (Sabuncuoglu and Hommertzhaim 1995). This method assigns a flow allowance for each job equal to the total processing time multiplied by a tightness factor.

When the objective is to measure the long-term steady-state system performance of the system, statistics are cleared for an initial transient period. In the case where the objective is to measure the short-term performance, statistics are collected for a given time period without deleting any data points. In the deterministic system configuration, all jobs of the same job type enter the system at constant interarrival times and stay on the machines for constant processing time. In the stochastic case, the interarrival times are sampled from an exponential distribution. The mean of the exponential distribution is equal to the value of the constant interarrival time used under the deterministic configuration. Similarly, the mean of the exponential processing time distribution is set equal that of the deterministic case.

For each system configuration, both the training and test data sets required for the neural networks are obtained by running the simulation models with different levels of the input variables: the due date tightness factor, the average job interarrival time and the queue waiting discipline. The values for these inputs are randomly generated from a uniform distribution. For instance, if we consider the complex system under general demand conditions, for each example in the data set, the mean interarrival time of each job type is randomly generated from the interval [20..100]. In addition, the due date tightness factor is randomly generated from within [2..9], and the queuing rule is randomly selected in the set {SPT, EDD, MODD}. For a complete list of the input ranges used for the different input variables, see <http://alcor.concordia.ca/~souheyl/MThesis>, where a copy of the thesis, from which this paper is extracted, is available.

For each input combination, the simulation model is then run and the system performance (mean machine utilization, mean job tardiness and mean job flow time) is recorded. For each experiment, the simulation model was run 1000 times: 70% of the simulation output data are used for training data set and 30% are used for the test data set.

The neural network architecture is varied according to the problem under consideration. Figure 2 shows the general architecture used. The inputs presented to the network at the input layer are the average job interarrival time, the due date tightness factor and the queue waiting discipline. The outputs of the network would be the system performance measure of interest. The number of hidden layers and the number of nodes at each layer are varied empirically according to the problem at

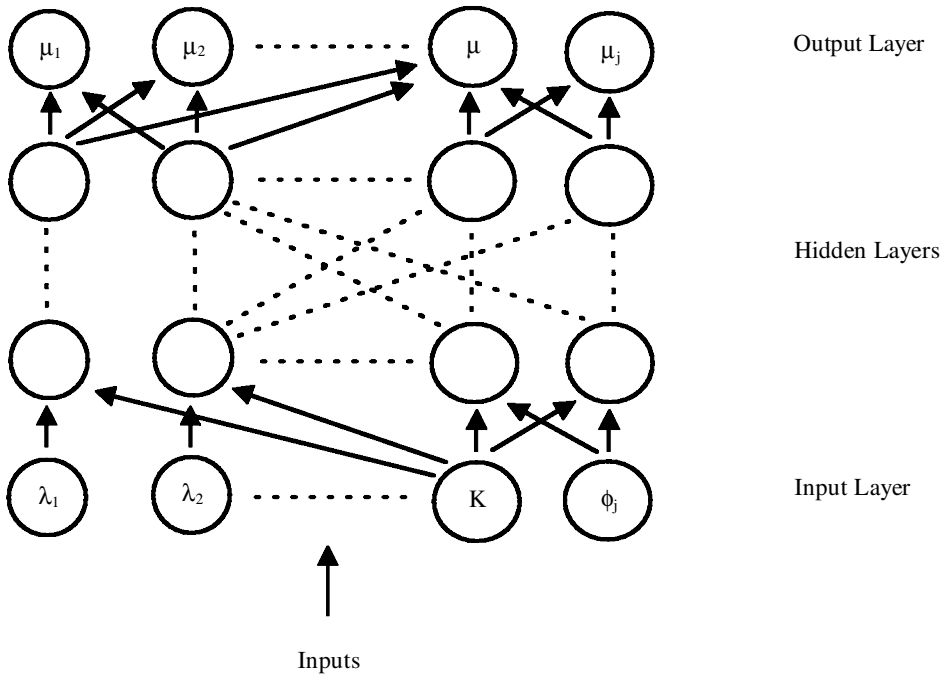


Figure 2. General neural network architecture.

hand. Note that too many nodes in hidden layers cause the network to memorize rather than to learn. By the same token, too few nodes result in no learning. Hence, the size of the hidden layer is determined as a result of pilot runs. For example, for estimating the mean job tardiness for a simple deterministic system, three network architectures were tested and the best result was considered. These architectures were 5-6-7-6-3, 5-8-9-8-3 and 5-55-3 (where the first digit represents the number of nodes in the input layer, the last digit represents the number nodes in the output layer and the digits in between represent the number of nodes per hidden layer). For the complex system configuration, the network sizes were increased to 8-12-14-12-6, 8-15-19-15-6 and 8-45-6.

In the experiments, three network architectures were implemented and the best one was selected for comparison. The back propagation networks operate with a learning coefficient and a momentum term, which are both empirically set. These coefficients remain constant throughout the training session, which has a length of 500 000 input-output combinations (the available data set is presented a multiple number of times). For simplicity, no bias has been introduced to the networks.

An essential aspect of metamodelling is to evaluate the quality of the estimate of the system performance measure (produced by the neural network metamodel) as compared with the true value (produced by the simulation models). For estimating mean machine utilization, two criteria are used: mean absolute deviation (MAD) and percentage error (%error). MAD consists of computing the absolute difference between each estimate and the true value and these differences are then averaged across all the output nodes and all the examples in the data set. The percentage error approach is the classic one where the absolute difference is divided by the true value.



For estimating mean job tardiness, the classic percentage error approach cannot be used (since tardiness can have a zero value). For this reason, another error assessment criterion is employed in the experiments. We divide the difference by the flow allowance (as assigned when setting the due dates). This will be denoted as FA.

### 4. Results of the experiments

#### 4.1. Estimating long-term machine utilization

The first set of experiments is designed to measure metamodel accuracy in estimating long-term machine utilization. The results are depicted in figures 3–6. Figures 3 and 4 display the errors obtained on the training set for a system with medium demand level. Figures 5 and 6 show the increase in the error levels of the metamodels when the test data sets are used, i.e. graphs plot the error on the test data set minus error on training data set.

In general, the neural network metamodels perform well to estimate mean machine utilization (note that utilization is measure on a scale of 1). It appears

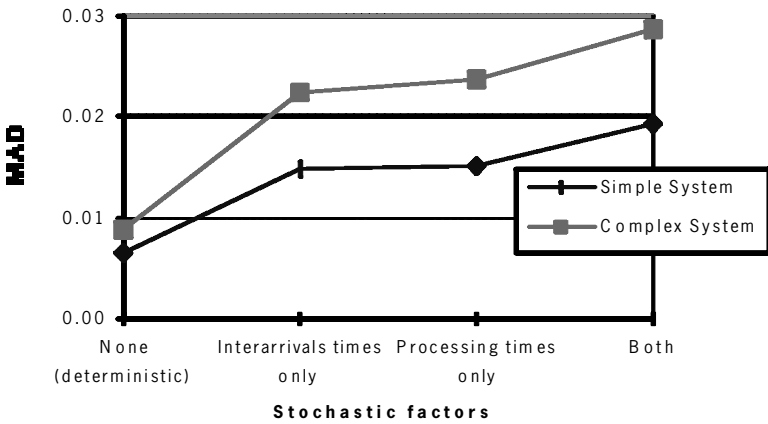


Figure 3. Effect of stochasticity and complexity (training data).

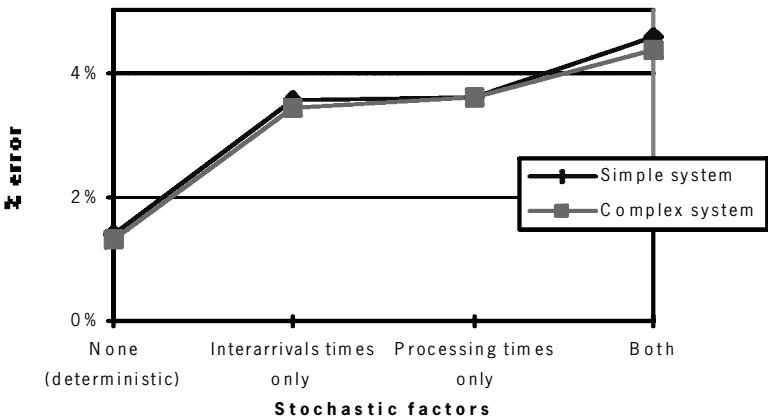


Figure 4. Effect of stochasticity and complexity (training data).

Downloaded by [Bilkent University] at 03:01 13 November 2017

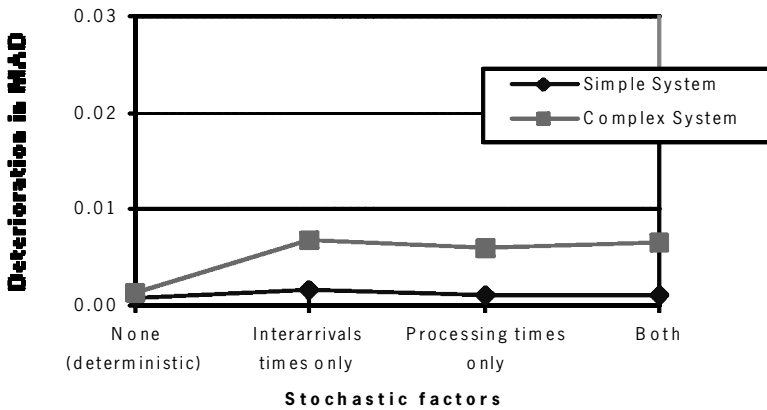


Figure 5. Effect of stochasticity and complexity (error on test data – error on training data).

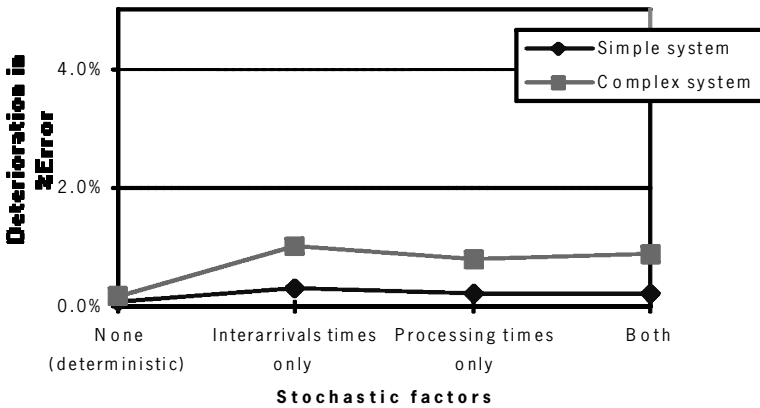


Figure 6. Effect of stochasticity and complexity (error on test data – error on training data).

that the metamodel accuracy decreases as system complexity increases both on the training and test data sets. It is also noted that the accuracy decreases as a result of adding stochasticity to the system. An interesting result is that the stochastic factors examined in this study yield similar error levels as illustrated by the almost flat middle portion in figures 3–6. When both variables (interarrival times and processing time) are stochastic, the error increases relative to when only one of them is stochastic (figure 3 and 4). However, the deterioration in metamodel accuracy is almost the same when either one of the variables stochastic (figures 5 and 6).

We also examine the effect of variation in the demand levels on metamodel accuracy. The complex deterministic system is considered with three levels of demand. Tables 1a and b shows the error achieved by the metamodels for both error assessment criteria. It appears that increasing demand levels result in reduced metamodel accuracy both on the training and test data sets. However, this deterioration in metamodel accuracy is very small compared with the deterioration due to the added system complexity or stochasticity.

The ability of the metamodel to estimate machine utilization when inputs are outside the range of the training data set is interesting to observe. At this stage, the

MOD no.	Training set no.	Test set no.	No. of job types	No of mach	Interarrival time		Processing time range	Due date tightness factor	Scheduling rule
					Nature	Range			
1	1	1	6	7	deterministic	[20..100]	deterministic	[2..9]	SPT, MODD or EDD
2	2	2	6	7	deterministic	[20..40]	deterministic	[2..9]	SPT, MODD or EDD
3	3	3	6	7	deterministic	[40..70]	deterministic	[2..9]	SPT, MODD or EDD
4	4	4	6	7	deterministic	[70..100]	deterministic	[2..9]	SPT, MODD or EDD
5	5	5	6	7	deterministic	[20..100]	deterministic	[2..9]	SPT, MODD or EDD
*machine utilization < 98%									
6	6	6	6	7	stochastic (exponential)	mean in [40..100]	deterministic	[2..9]	SPT, MODD or EDD
7	7	7	6	7	deterministic	[40..100]	stochastic (exponential)	[2..9]	SPT, MODD or EDD
8	8	8	6	7	stochastic (exponential)	mean in [40..100]	stochastic (exponential)	[2..9]	SPT, MODD or EDD

Table 1(a). Effect of demand on the system.

Demand level	Training data		Deterioration on test data	
	MAD	% Error	MAD	% Error
High	0.0036	0.4	0.0008	0.1
Medium	0.0027	0.4	0.0002	0.0
Low	0.0009	0.5	0.0010	0.0

Table 1(b). Effect of demand on the system.

simple deterministic system was considered. Table 2 shows the metamodel accuracy on three test data sets. The first set (first row) corresponds to data set with inputs generated from the same range as the training data set (job interarrival times generated from interval [10,85]). The second data set corresponds to input generated from a range adjacent to the training range (job interarrival times generated from [85,100]). Similarly, the third data set corresponds to inputs generated from a range adjacent to the range of the second data set (job interarrival times are generated from [100,120]). These ranges correspond to testing a neural metamodel built for a system with low demand levels and that is required to extrapolate on higher demand levels. As table 2 shows the metamodel accuracy has a slow degradation outside the training range. This indicates that neural metamodels are robust to changes in operating conditions.

Set	MAD	Mean % error
1	0.006	1.4
2	0.04	3.8
3	0.06	5.3

Table 2. Generalization outside the experiment range.

We also observe both error assessment criteria produced consistent results in the experiments. In other words, the different models would rank similarly for each criterion. This consistency is a good point as it confirms the conclusions made so far. In short, the results indicated that neural network metamodelling designed to estimate long-term mean machine utilization are quite successful and can handle system stochasticity and complexity.

#### 4.2. Estimating long-term job tardiness

This section examines the capability of neural networks to estimate long-term job tardiness at two levels of system complexity. MAD and FA are used to assess network accuracy. Recall that due date related criteria are commonly used in practice to measure the system performances and the mean tardiness is the popular one in the literature.

The results are plotted in figures 7–14. It appears that neural network metamodelling are better at estimating long-term machine utilization than estimating long-term

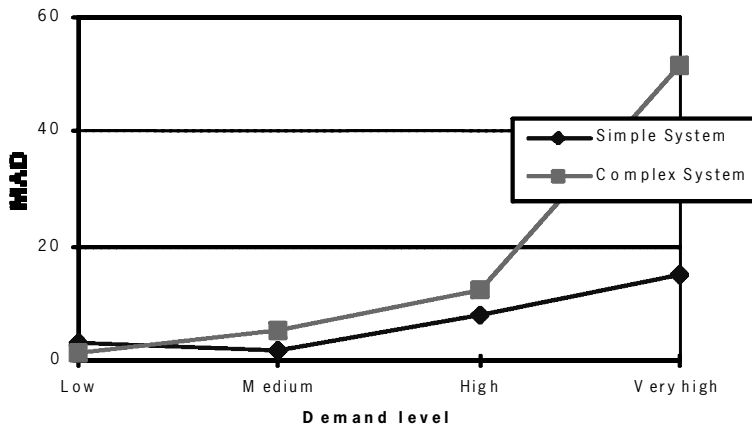


Figure 7. Effect of complexity and demand (training data).

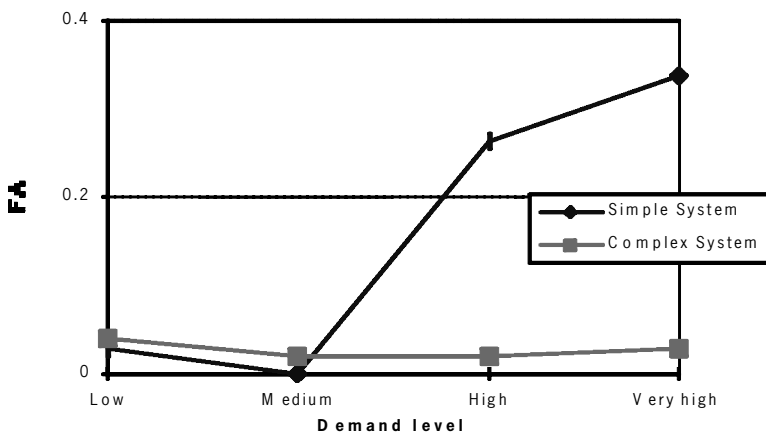


Figure 8. Effect of complexity and demand (training data).

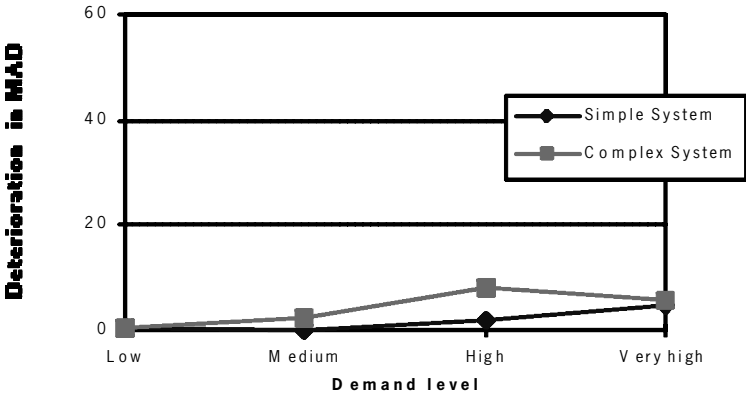


Figure 9. Effect of complexity and demand (error on test data – error on training data).

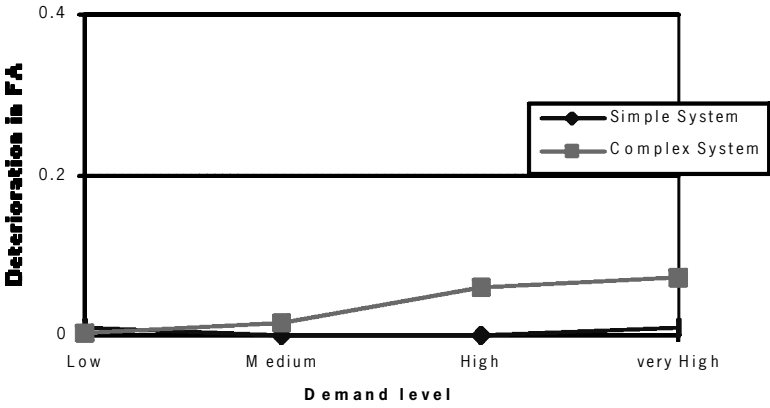


Figure 10. Effect of complexity and demand (error on test data – error on training data).

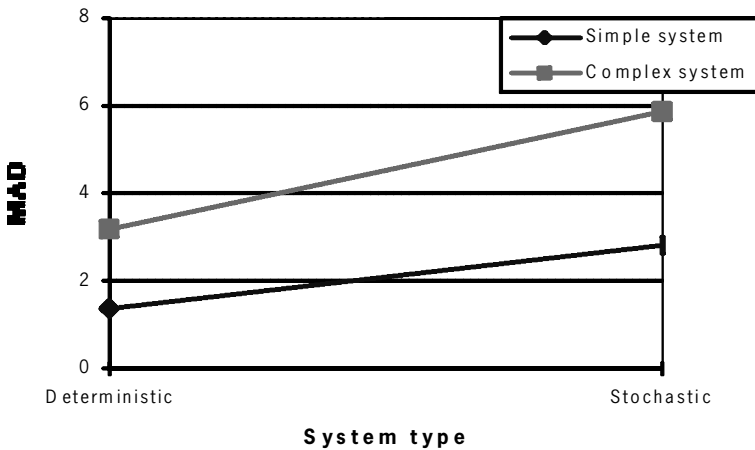


Figure 11. Effect of stochasticity and complexity (training data).

Downloaded by [Bilkent University] at 03:01 13 November 2017

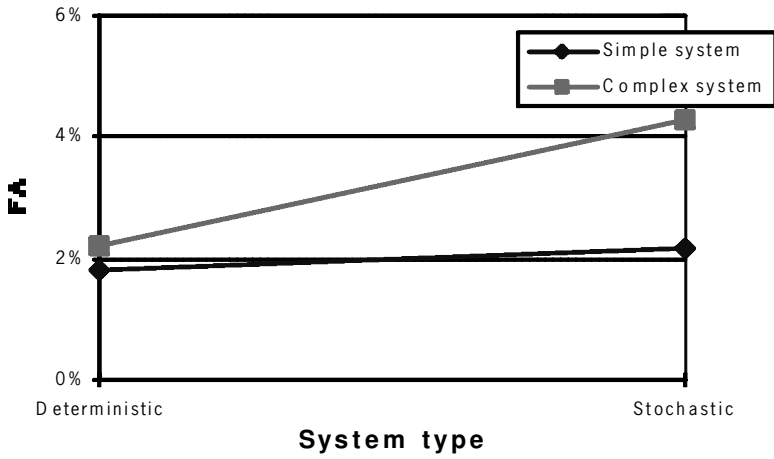


Figure 12. Effect of stochasticity and complexity (training data).

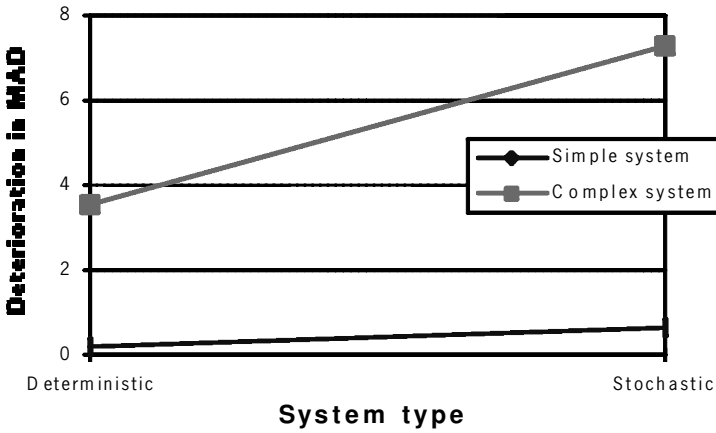


Figure 13. Effect of stochasticity (error on test data – error on training data).

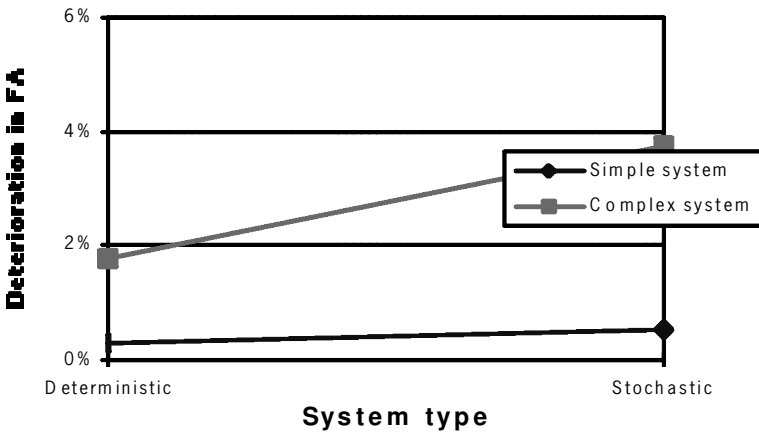


Figure 14. Effect of stochasticity (error on test data – error on training data).

job tardiness. According to the FA error assessment criteria, the job tardiness is within 10% of flow allowance (for a system in steady-state). This could be considered as acceptable since there is still room to improve metamodel accuracy through further fine-tuning of the metamodel parameters (i.e. by changing network architectures and algorithms' learning and momentum coefficients).

The first point of interest was to examine the effect of variations in demand levels on metamodel accuracy. Figures 7 and 8 show the error levels achieved by the metamodels for four demand levels on the training data set (based on a deterministic system). Figures 9 and 10 show the degradation in error levels when metamodels are evaluated based on the test data sets. Low demand levels correspond to a data set where the mean of the distribution of the interarrival time is large compared with the case with medium, high or very high demand. Originally, the means of the distributions of the interarrival times were randomly generated from within a large interval. To study the effect of different demand levels, this interval was divided into four non-overlapping subintervals, corresponding to low, medium, high and very high demand levels.

These figures illustrate a non-linear deterioration in metamodel accuracy and generalization capability with the most significant deterioration occurring in the very high demand case. The impact of increasing demand levels is most significant for the complex system configuration. The very high demand case is the case where the training and test sets contain examples for which the system is not in a steady-state. In such a situation, long queues build up and several machines become bottlenecks. As a comparison, in the high demand case, the maximum machine utilization in the examples included in the data sets is 98%. This indicates that metamodels are accurate enough as long as the system is in steady-state. However, as illustrated by figures 9 and 10, the increased demand levels do not have such a significant impact on amplifying the error level on the test data sets.

Figures 11 and 12 show the effect of stochastic interarrival times and processing times on metamodel accuracy, and figures 13 and 14 show the effect of stochasticity on the error deterioration on test sets. We observe that adding stochastic elements to the model reduces metamodel accuracy and amplifies error deterioration. This impact is further amplified by having more complex system configuration.

Figures 15 and 16 depict the impact of stochasticity on metamodel accuracy in more detail. Figures 17 and 18 show the effect of different stochastic variables on degradation in a metamodel error on the test data set. When processing times are stochastic, the metamodels have achieved slightly higher error levels than when interarrival times are stochastic. When both variables are stochastic, metamodel accuracy decreases sharply. Once again, system complexity amplifies error levels significantly.

#### 4.2.1. *Machine breakdowns*

The systems modelled so far are not subject to machine breakdowns. Hence, one may question the robustness of the metamodels with respect to random breakdown interruptions (i.e. can a metamodel trained on a system with a 100% efficiency, generalize to examples from the same system but subject to machine breakdowns?).

To perform this test, we reduce the machine efficiency to 92%. Tables 3 and 4 display the results in terms of the level of error achieved by the metamodels on the test data set both simple and complex systems and for the two error assessment criteria.

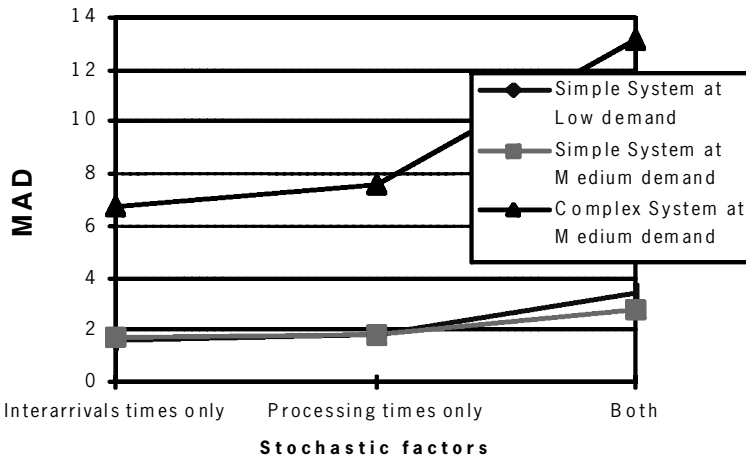


Figure 15. Effect of stochasticity, demand and complexity (training data).

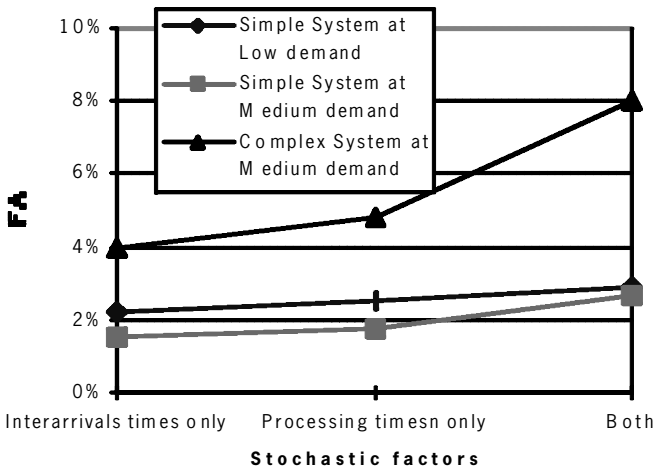


Figure 16. Effect of stochasticity, demand and complexity (training data).

It appears that the robustness of the metamodells to machine breakdown is highly dependent on system complexity and the demand on the system and to a lesser extent on stochasticity. The most significant deterioration is observed at the highest demand and at the complex system. Analysing the data for this situation indicates that machine breakdowns cause large queues to form, which in turn does not allow the system to reach its steady-state equilibrium.

#### 4.3. Estimating short-term job tardiness

This section investigates the potential use of neural network to estimate short-term performance of the system. In this type of situation, the analyst wants to estimate the system performance for a certain period given well-defined initial conditions, e.g. given the number of jobs and machines with current load, to estimate job



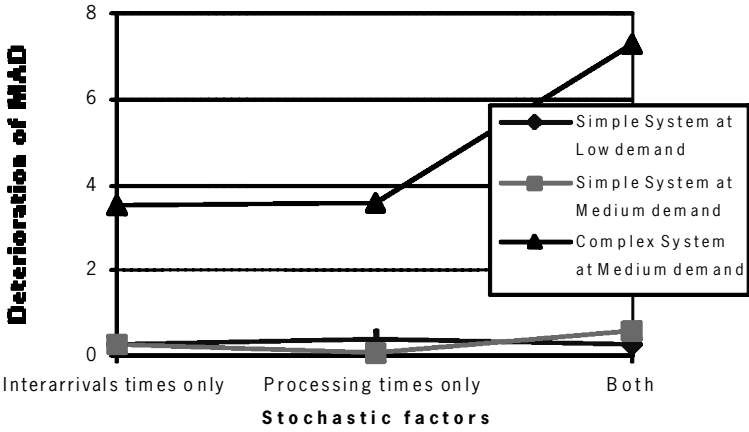


Figure 17. Effect of stochasticity, demand and complexity (error on test data – error on training data).

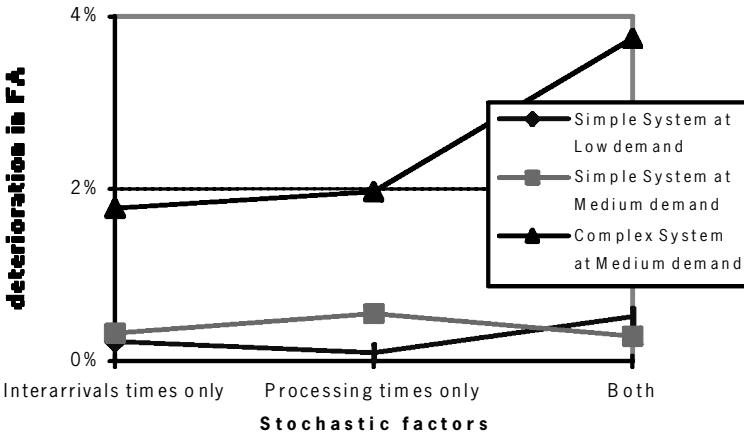


Figure 18. Effect of stochasticity, demand and complexity (error on test data – error on training data).

		Deterministic system			Stochastic system	
		Low demand	Medium demand	High demand	Low demand	Medium demand
MAD approach	with breakdowns	3.3	0.8	104.8	6.1	7.3
	without breakdowns	3.0	1.6	15.0	3.6	3.1
FA approach	with breakdowns (%)	5.7	0.4	45.7	9.2	3.5
	without breakdowns (%)	4.0	2.1	2.7	2.7	2.9

Table 3. Simple system: robustness to machine breakdowns.

Downloaded by [Bilkent University] at 03:01 13 November 2017

		Deterministic system			Stochastic system	
		Low demand	Medium demand	High demand	Low demand	Medium demand
MAD approach	with breakdowns	3.3	36.3	277.0	9.0	55.6
	without breakdowns	1.3	5.4	51.8	13.2	13.2
FA approach	with breakdowns (%)	2.9	21.2	178.8	7.0	27.5
	without breakdowns	1.2	3.7	46.1	8.0	8.0

Table 4. Complex system: robustness to machine breakdowns.

completions or job tardiness. Since the initial system status is now a relevant factor in the system, new nodes are added to the input layer of the neural networks. The information about initial system status is incorporated into the model in terms of the work in progress inventory of each job type waiting before each machine in the system.

The results of experiments are presented in figures 19–22. The curve titled ‘generalization’ in these figures represents the differences between metamodel errors on test and training sets. As expected, stochasticity and system complexity act negatively on the metamodel performance and its generalization capability. The results for the complex stochastic system configuration are not shown because the errors are very high. It appears that system complexity has a more negative effect on the system performance than stochasticity. This significant increase in error level in stochastic systems indicated that neural networks might not be suitable for estimating short-term performance of the systems when the level of variation and stochastic elements are high.

The effect of demand on system performance is displayed figures 21 and 22. It appears that the higher the demand, the better is the metamodel accuracy and generalization capability. This result was unexpected since for estimating long-term job-tardiness, we found that increasing demand (or load on the system) resulted in a decreased accuracy of the metamodels. In this set of experiments, increasing the

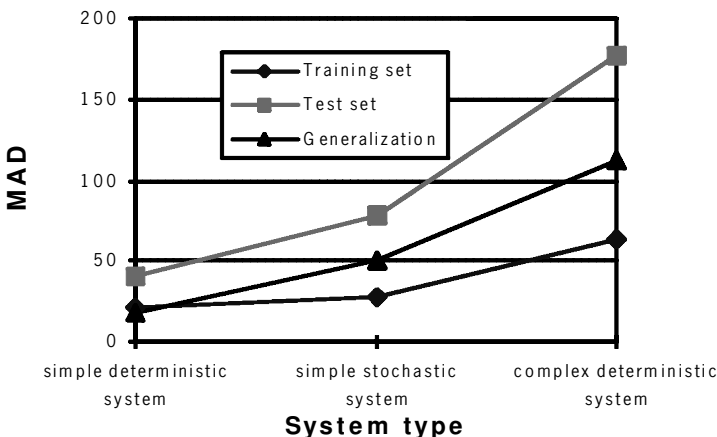


Figure 19. Effect of stochasticity and complexity.

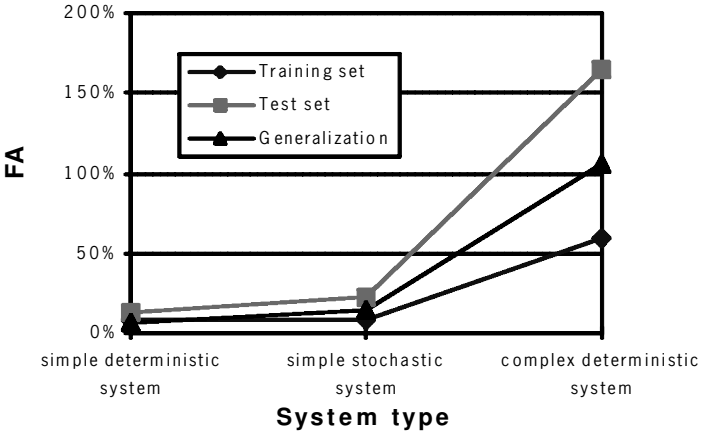


Figure 20. Effect of stochasticity and complexity.

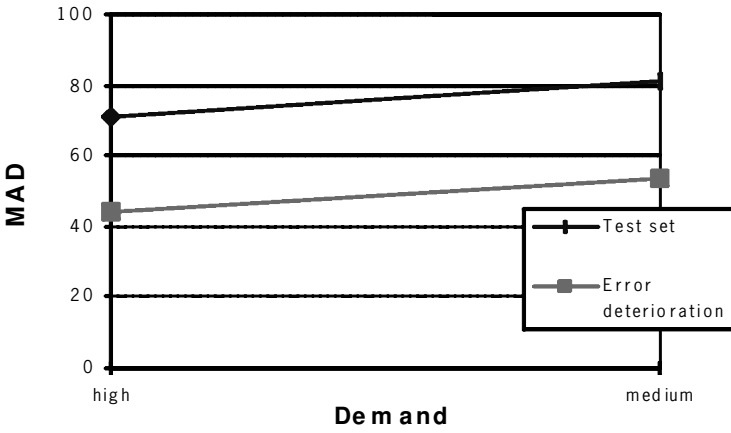


Figure 21. Effect of demand level.

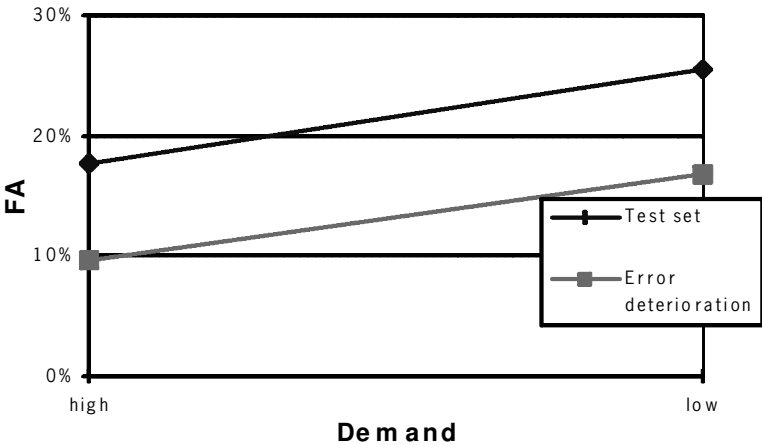


Figure 22. Effect of demand level.

demand resulted in a higher accuracy level. The interpretation made sense at the time. In estimating short-term job tardiness, the system is not in steady-state, but there is a degree of how far it is from its steady-state. The interpretation of these results is that at high demand, the system is nearer to its steady-state than with lower demand. The idea is that because of the stability of the system, steady-state performance is easier to predict. We saw in the previous experiments (long-term) that the more the system is overloaded, the worse the accuracy. Similarly, in this case (short-term), the more the system is under-loaded, the worse its accuracy. We can view demand as on a line: in the middle, we have the range of the demand where the system is in steady-state. In this interval, the metamodel accuracy is good. The more we move the right or to the left of this interval, the worse is the metamodel accuracy. The previous experiments showed that the quicker the system can tend towards its steady-state equilibrium, the better is the metamodel accuracy. However, these experiments assumed that all machine queues are empty initially.

Next, we investigated the effect of the initial system status for variable demand on system, by fixing the number of jobs (from each job type) in each queue. The results are displayed for the cases where demand level is high, medium or both combined (referred to as general). As seen in figures 23 and 24, increasing the initial number of jobs in the queues results in improving the results until some point then no improvement is observed. The points at which the error levels level off correspond approximately to the average number of jobs in the queue when the system is at steady-state. Even if this initial number exceeds the average queue size at steady-state, metamodel performance does not change. The same effect of the initial system status exists when we split the range of the interarrival times to low demand and to high demand.

The effect of initial system status on metamodel generalization capability is shown in figures 25 and 26. As can be seen, the deterioration in error levels does not have a specific pattern.

The purpose of using neural metamodels to estimate short-term tardiness was to investigate their possible use in a real-time decision. One situation frequently encountered in the real-life scheduling practices is to select the best scheduling rule

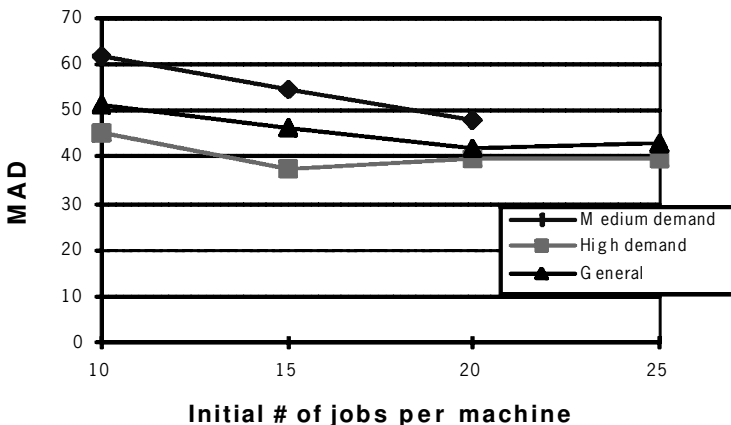


Figure 23. Effect of initial system status (trained data).

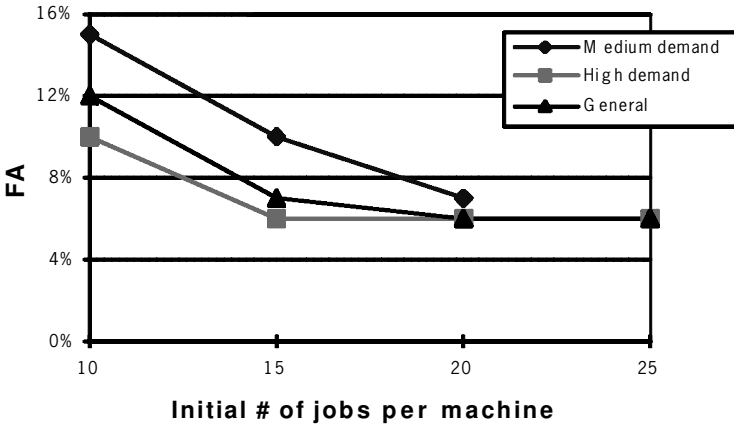


Figure 24. Effect of initial system status (trained data).

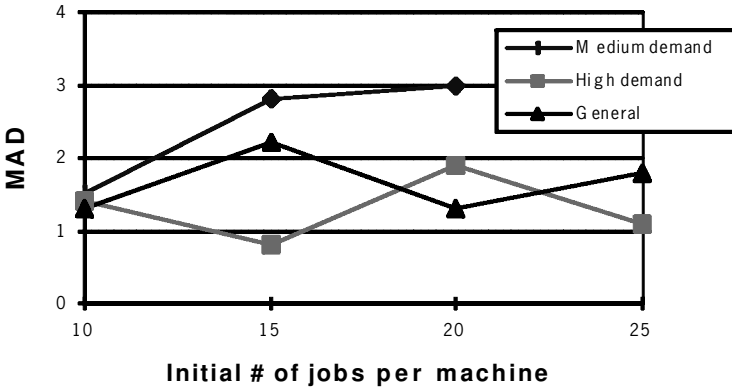


Figure 25. Effect of initial system status (error on test data – error on training data).

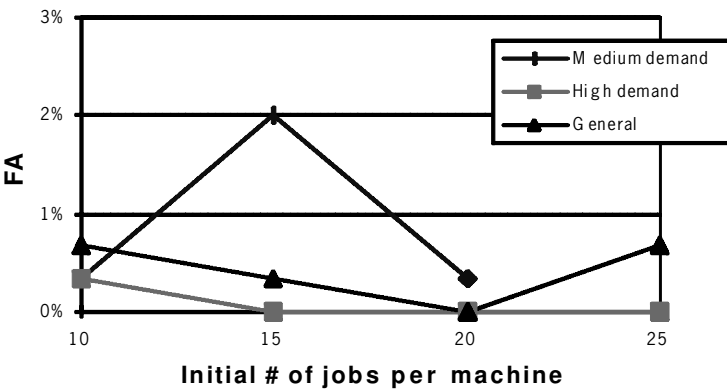


Figure 26. Effect of initial system status (error on test data – error on training data).

or dispatching policy based on the information about the current state of system and objectives. For that reason, we conducted another sets of experiments to see if one can make good decisions via neural networks. In other words, we wanted to know if, based on the metamodel, we would select the same dispatching rule, as we would do if we base the decision on the simulation model. This test is applied on three general models, corresponding to deterministic simple system stochastic simple system, and deterministic complex system. In selecting the dispatching rule (based on the simulation models or on the neural metamodel), several alternatives are available. If the job types have different weights, then we may select the dispatching rule that minimizes the tardiness of the job type with highest weight. This may be the case when one job type is very profitable or very strategic to the business. Another alternative would be to select the rule that produces the minimum overall average tardiness (across all the job types).

Table 5 shows the results of these experiments. Entries represent the percentage of the example in the test data set for which the correct decision was made by the metamodel. The columns correspond to different decision criteria. For example, the first column corresponds to the case where the objective is to select the dispatching rule that best minimizes tardiness for job type 1 (and similarly for columns 2–6). The last column corresponds to the case where the objective is to select the dispatching rule that best minimized the overall tardiness.

In the previous experiments, we observed that stochasticity and system complexity have negative effects on the accuracy of the metamodel. However, the results in this section give the opposite indications. The difference lies on the how the metamodel is evaluated. While in the previous experiments, the metamodels were evaluated on their capacity to forecast the performance measures of the manufacturing system, in this section the metamodels were evaluated on their capacity to rank the performance of different queuing rules, which would allow us to select the best queuing rule.

If the selection of the queuing rule is based on the overall average tardiness, the metamodel success rate on the complex deterministic system ranks better than on the simple stochastic system, which in turn ranks better than on the simple deterministic system. The reason for this contradiction may be that complexity (whether due to stochasticity or to system configuration) acts as to increase the difference between the performances of the different dispatching rules. In other words, the added complexity makes it more difficult to forecast the value of the performance measures. However, under these same conditions of added complexity, it is easier to differentiate between the performances of the different queuing rules. Thus, it may be easier

System	Job type						Average
	1	2	3	4	5	6	
Simple, stochastic	96	84	49	n.a.	n.a.	n.a.	91
Simple, deterministic	81	100	79	n.a.	n.a.	n.a.	79
Complex, deterministic	100	100	71	97	98	100	100

Data are percentages.

n.a., not applicable.

Table 5. Testing capability to select operational policies.

for the metamodel to compare different queuing rules rather than provide precise measures of the performance of each rule. This implies that using neural metamodel is likely to produce good decisions as the system gets more complex. In addition, the test results show that making the decisions based on different job types leads to different metamodel performances. It appears that the metamodel performance is related to the job type characteristics. The results also indicated that different criteria lead to opposite interpretations or different decisions. Hence, the choice of the criteria should depend on the objective of the study.

## 5. Concluding remarks and future research directions

In this paper, we have conducted an initial investigation of the potential use of neural networks as simulation metamodels for estimating manufacturing system performances. The experiments indicated that application to real-life problems is not straightforward. We also observed that performance measures, the study horizon, the system load, the initial system status, stochasticity, complexity and error assessment methods can affect metamodel accuracy in different ways. For example, experiments indicated that metamodel accuracy can decrease rapidly when estimating short-term job tardiness for terminating type systems in the context of stochastic or complex systems. However, the metamodels were successful in discriminating between dispatching policies in this same contexts. Therefore, the success of metamodeling with neural networks depends on the combination of the system characteristics and the error assessment criteria, as well as the purpose of simulation applications.

In conclusion, we believe that metamodeling with neural networks is a promising tool and can be of value for manufacturing systems' management. To investigate further the potential of this tool, one can examine the effect of system configuration (flow shop to job shop), system disturbances such as machine break down, applying different distributions in studying stochasticity, the size of data sets, or estimating other measures such as the standard deviation of the performance measures.

## References

- AUSSEM, A. and HILL, D., 2000, Neural-network metamodeling for the prediction of *Caulerpa taxifolia* development in the Mediterranean sea. *Neurocomputing*, **30**, 71–78.
- BADIRU, A. B. and SIEGER, D. B., 1998, Neural network as a simulation metamodel in economic analysis of risky projects. *European Journal of Operational Research*, **105**, 130–142.
- BARTON, R. R., 1992, Metamodeling for simulation input-output relations. *Proceedings of the 1992 Winter Simulation Conference*.
- BLANNING, R. W., 1975, The construction and implementation of metamodeling. *Simulation*, **24**, 177–184.
- BURKE, L. I. and IGNIZIO, J. P., 1992, Neural networks and operation research: an overview. *Computer Operations Research*, **19**, 179–189.
- CHRYSSOLOURIS, G., LEE, M. and DOMROESE, M., 1991, The use of neural networks in determining operational policies for manufacturing systems. *Journal of Manufacturing Systems*, **100**, 166–175.
- CHRYSSOLOURIS, G., LEE, M., PIERCE, J. and DOMROESE, M., 1990, Use of neural networks for the design of manufacturing systems. *Manufacturing Review*, **3**, 187–194.
- DAYHOFF, J. E., 1990, *Neural Network Architectures: an Introduction* (New York: Van Nostrand Reinhold).
- FISHWICK, P. A., 1989, Neural network models in simulation: a comparison with traditional modeling approaches. *Proceedings of the 1989 Winter Simulation Conference*, pp. 702–710.

- FREIDMAN, L. W. and PRESSMAN, I., 1988, The metamodeling in simulation analysis: can it be trusted? *Journal of the Operational Research Society*, **39**, 939–948.
- HARMONOSKY, C. M. and ROBOHN, S. F., 1995, Investigating the application potential of simulation for real time control decisions. *International Journal of Computer Integrated Manufacturing*, **8**, 126–132.
- HUANG, C. L., HUANG, Y. H., CHANG, T. Y., CHANG, S. H., CHUNG, C. H., HUANG, D. T. and LI, R. K., 1999, The construction of production performance prediction system for semiconductor manufacturing with artificial neural networks. *International Journal of Production Research*, **37**.
- HURRION, R. D., 1992, Using a neural network to enhance the decision making quality of a visual interactive simulation model. *Journal of Operational Research Society*, **43**, 333–341.
- HURRION, R. D., 1997, An example of simulation optimization using a neural network meta-model: finding the optimum number Kanbans in a manufacturing system. *Journal of the Operational Research Society*, **48**, 1105–1112.
- HURRION, R. D., 1998, Visual interactive meta-simulation using neural networks. *International Transactions Operational Research*, **5**, 261–271.
- HURRION, R. D. and BIRGIL, S., 1999, Comparison of factorial and random experimental design methods for the development of regression and neural network simulation meta-models. *Journal of the Operational Research Society*, **50**, 1018–1033.
- KILMER, R. A., SMITH, A. E. and SHUMAN, L. J., 1999, Computing confidence intervals for stochastic simulation using neural network neural network. *Computers and Industrial Engineering*, **36**, 391–407.
- KLEINEN, J. P. C. and SARGENT, R. G., 2000, A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, **120**, 14–29.
- MADEY, G. R., WEINRTH, J. and SHAH, V., 1992, Integration of neurocomputing and system simulation for modeling continuous improvement systems in manufacturing. *Journal of Intelligent Manufacturing*, **3**, 193–204.
- McHANEY, R. W. and DOUGLAS, D. E., 1997, Multivariate regression metamodel: a DSS application in industry. *Decision Support Systems*, **19**, 43–52.
- MOLLAGHASEMI, M., LECROY, K. and GEORGIPOULOS, M., 1998, Application of neural networks and simulation modeling in manufacturing system design. *Interfaces*, **28**, 100–114.
- PHILIPOOM, P. R., REES, L. P. and WIEGMANN, L., 1994, Using neural networks to determine internally set due date assignments for shop scheduling. *Decision Sciences*, **25**, 825–851.
- PIERREVAL, H., 1992, Rule based simulation metamodels. *European Journal of the Operational Research Society*, **61**, 6–17.
- PIERREVAL, H., 1993, *Neural network to select dynamic scheduling heuristics*. *Revue des Systèmes de Décision*, **2**, 173–190.
- PIERREVAL, H., 1996, A metamodeling approach based on neural networks. *International Journal of Computer Simulation*, **6**, 365–378.
- RAAYMAKERS, M. H. M. and WEIJTERS, A. J. M. M., 2000, Using regression models and neural network models for makespan estimation in batch processing. In van der Bosch and H. Weigand (Eds.), *Proceedings of the Twelfth Belgium–Netherlands Conference on AI*, pp. 141–148.
- SABUNCUOGLU, I., 1998, Scheduling with neural networks—a review of the literature and new research directions. *Production and Inventory Control*, **9**, 2–12.
- SABUNCUOGLU, I. and HOMMERTZHEM, D., 1995, Experimental investigation of an FMS due-date scheduling problem: an evaluation of due-date assignment rules. *International Journal of Computer Integrated Manufacturing*, **8**, 133–144.
- WATSON, E. F., CHAWDA, P. P., MCCARTHY, B., DREVNA, M. J. and SADOWSKI, R. P., 1998, A simulation metamodel for response-time planning. *Decision Sciences*, **29**, 217–241.
- YU, B. and POPLEWELL, K., 1994, Metamodeling in manufacturing: a review. *International Journal of Production Research*, **32**, 787–796.
- ZHANG, H. C. and HUANG, S. H., 1995, Applications of neural networks in manufacturing: a state of the art survey. *International Journal of Production Research*, **33**, 705–728.