2010 11th Int. Conf. Control, Automation, Robotics and Vision
Singapore, 7-10th December 2010

Predicting Amount of Saleable Products Using Neural Network Metamodels of Casthouses

Abbas Khosravi, Saeid Nahavandi, Doug Creighton, Bruce Gunn
CAST Cooperative Research Centre
Centre for Intelligent Systems Research (CISR)
Deakin University, Geelong, Vic, 3117, Australia
Email: {akhos, saeid.nahavandi, douglas.creighton, bruce.gunn}@deakin.edu.au

***Abstract*—This study aims at developing abstract metamodels for approximating highly nonlinear relationships within a metal casting plant. Metal casting product quality nonlinearly depends on many controllable and uncontrollable factors. For improving the productivity of the system, it is vital for operation planners to predict in advance the amount of high quality products. Neural networks metamodels are developed and applied in this study for predicting the amount of saleable products. Training of metamodels is done using the Levenberg-Marquardt and Bayesian learning methods. Statistical measures are calculated for the developed metamodels over a grid of neural network structures. Demonstrated results indicate that Bayesian-based neural network metamodels outperform the Levenberg-Marquardt-based metamodels in terms of both prediction accuracy and robustness to the metamodel complexity. In contrast, the latter metamodels are computationally less expensive and generate the results more quickly.**

I. Introduction

Complex systems like manufacturing enterprises are composed of hundreds of interconnected autonomous and non-autonomous components. The level of complexity and non-linearity in these systems is much beyond what physical and mathematical modeling principles can cope with. The prevalence of uncertainty in the operation of these systems also significantly degrades the validity and reliability of analytical models. These issues warrant application of simulation as a powerful technique for modeling and analysis of complex systems. The main power of simulation models relies on the fact that they can model real world systems to a high level of detail. Flexibility of simulation models also allows conducting what-if analysis scenarios making them a reliable decision supporting tool. There are numerous studies reporting successful implementation of simulation models for describing complex systems, including manufacturing enterprises [1], [2], [3], [4] and baggage handling system [5], [6]. The majority of simulation models are developed using discrete event simulation modeling techniques equipped with 3D graphical visualization.

Although simulation models are the most accurate and reliable tool for analysis of complex systems, they suffer from a couple of restricting issues. Simulation models throughout their lifecycle are expert intensive. A high level of expertise is usually required for development, maintenance and operation of these models [7]. Furthermore, these models are computationally expensive and massively time-consuming. This indicates that one can only use them in the design stage and their application for real time operational planning and scheduling is highly limited [8]. The final point is that a high level of detailed information and data are required for modeling different aspects of these systems. Any piece of inaccurate information may degrade performance of the developed simulation models.

Metamodels have been introduced in literature to overcome these problems [9]. A metamodel is an abstract or auxiliary model of a complex system approximating linear/nonlinear relationship between the response and the design variables. The relationship can be looked for in a local and/or the whole region of interest. Usually, a space-filling sampling method is applied for evenly locating sample points within the whole design domain. Then the collected samples are applied for developing metamodels using a variety of modeling techniques including regression models [10], [11], splines [9], kriging [12], and Neural Networks (NNs) [13], [14], [8], [15], [16]. The regression model is the most frequently used metamodel in the scientific and practical literature. This is mainly due to its functional simplicity and ease of development. The availability of statistical packages supporting the regression models also contributes to this popularity. This popularity, however, does not imply that these metamodels are the best in terms of prediction accuracy and reliability. Comparative and reviewing studies indicate that NNs are the superior candidate for construction of metamodels in many domains of science and engineering [17]. This is mainly due to their excellent learning capability and being universal approximator [18]. This discussion well justifies preferring NNs to other techniques for developing metamodels.

The main purpose of this study is to develop a NN metamodel for a real world metal casting plant. The quality of casting products reflected in the quantity of saleable product is the metamodeling target. Product quality has nonlinear relationships with many internal and external factors. Many of these relationships are unknown to the system operators and even system designers. Presence of uncertainty in the operation of this system also makes decision making highly difficult and prone to mistake, occasionally resulting in catastrophic consequences in terms of product quality. Also the concept of NN metamodeling has been introduced around two decades, it has not been applied for approximating nonlinear dependencies inside a casthouse. The NN metamodel will be developed to predict the product quality based on the available information. As this prediction can be done before starting the casting process, the obtained results can be used for operational planning, either off-line or on-line.

It is well known that NN prediction performance highly depends on the network structure (number of hidden layers and number of neurons per hidden layer) and the training process (initialization and the training algorithm). Two famous NN learning techniques are applied over a grid of NNs with different structures in the described experiments. The experiments are repeated a few times to eliminate effects of random initialization [18].

The rest of this paper is organized as follows: Section II introduces the metal casting process. Purpose of analysis and metamodeling steps are discussed in Section III. Experimental results are demonstrated in Section IV. Finally, Section V concludes the paper with a summary and some remarks for further study in this domain.

II. Metal Casting Plant

The underlying system in this study is a casthouse as part of a metal smelter facility. The casthouse produces a large range of alloys in different forms. Production complexity is high due to the presence of many influential factors on the product quality. The product scheduling and crucible routing are among the most challenging issues within the operation of the whole system. The most important components within the underlying casthouse are furnaces, casting stations, launders and filters, crucibles, and crucible transfer (start and end points).

The operation of the system is that molten metal is delivered from the pot-lines to the casthouse. The molten metal is delivered to the furnaces in the casthouse via the use of cylindrical crucibles. The crucibles follow fairly well defined paths in their travel from the pot-lines to the casthouse. Once the metal has been poured into the furnaces, the crucibles return to the pot-lines via return paths. Each furnace is governed by its own separate process logic. When the furnace is below the maximum capacity, the furnace status is that of furnace filling. When a crucible arrives at the pouring platform, the pouring sequence is started. In the beginning, the furnaces start at a low temperature, and have the temperature raised significantly with the addition of each crucible of metal from the pot-lines. This is due to the high energy content in the crucible, which is generally hotter than the furnace baring a significant delay, relative to the energy in the furnace. As the metal level in the furnace increases, the energy level in the furnace also increases, and so the temperature rises due to the pour of crucible metal decreases in magnitude. Once the furnace has the required amount of metal, the crucible deliveries to that furnace cease, and the waiting period starts. During this period, the furnace waits until both the casting temperature is reached and the casting station is ready to receive metal. The furnaces communicate with the casting station through the use of software switches. Once casting has been completed, the furnace status resets to furnace filling. The cyclic process begins again, with the furnace waiting for the arrival of crucibles from the pot-lines.

Upon completion of the casting process, the product quality is precisely examined. High quality casting products are delivered to the market based on customers' orders and the production schedule. For each casting process, the quantity of saleable products is reported in percentage varying between 0% (total loss of casting product) and 100% (perfect casting). This percentage is an indication of how well the casting process has been planned, proceeded, and completed.

Like many other manufacturing enterprises, there are several factors influencing the quality of the casting products. As many of these design factors are unknown or totally uncontrollable for operators, the level of uncertainty in operation of a casthouse is high. These uncertainties result in low quality of products, even if all production tasks are completed based on the operating manuals.

III. Metamodeling Procedure

The general metamodeling procedure, regardless of the metamodel type, can be summarized in 6 steps [19]:

1) Determining the goal of the metamodel;
2) Defining the ranges for the input variables;
3) Developing the experimental design;
4) Building a simulation model;
5) Developing the metamodel; and
6) Validating the metamodel.

A more detailed description of these steps can be also found in [20]. The main purpose of step 3 and 4 is to collect datasets useful for training and validating the developed metamodel. A Design of Experiments (DOE) accomplished in step 3 is fundamental to the successful development of a metamodel. The considered scenarios in design of experiments are applied to the detailed simulation model for data collection (step 4). The main difficulty is that in many real world applications, such as the casthouse in this study, these two steps can not be fully completed. There are often many financial and operational issues significantly restricting the number of samples one cat get from the system. Besides, lack of accurate information makes development of highly detailed simulation models impossible. An alternative solution is that metamodels are developed only based on the available datasets taken from the real system. The developed metamodels can be latter revised and updated in real time whenever new observations become available. According to these, metamodeling of the underlying casthouse is carried out as follows:

A. Metamodeling Goal

The purpose of metamodeling is to develop an abstract model of the casthouse for predicting the quantity of saleable products. The considered target is continuous ranging between 0% to 100%. Following steps defined in [20], metamodels are developed to approximate nonlinear relationships between this target and independent variables selected in the next step. The developed metamodel can be used for other purposes as well, including sensitivity analysis, optimization, conducting what-if studies, and operational planning.

B. Metamodel Inputs

The quantity of saleable products depends on many internal and external factors. Statistical analysis shows that variation of the considered target has a meaningful relationship with 11 out of 25 independent variables. Some of the variables are related to the casting facility, such as casting station and pouring location. Others are specific to the product and its features, including type, alloy, group, and profile. As the majority of these variables are categorical, they are converted to numerical values.

C. Metamodel Type and Structure

As discussed in Section I, NNs are a reasonable choice for developing metamodels. A two layer feedforward NN is considered for metamodeling the underlying casthouse based on the considered inputs. Although NNs with one hidden layer are universal approximator, in practice it is often found that NNs with more hidden layers perform better in term of prediction accuracy and generalization power. Fig. 1 demonstrates the considered NN metamodel in this study. The transfer functions in the two hidden layers are tangent sigmoid capable of extracting and describing nonlinear relationships. The transfer function in the output layer is a linear mapping of the second layer tangent sigmoids via the weights to generate the model output. This feature of having both nonlinearity and linearity in the model makes the considered model very versatile.

D. Metamodel Training

NN parameters can be adjusted using a variety of methods and based on minimization of different cost functions. In this study, two famous methods are applied: the Levenberg-Marquardt algorithm, and the Bayesian technique [18]. Both methods have obtained excellent reputation in the NN community. The former one has been applied more frequently due to its easily understandable mechanism. The later method interprets NN parameters with probabilistic distribution. The main features of the Bayesian technique is that it provides a measure of the effectiveness of NN parameters. This measure can be used as an indication of the appropriate size of NN models. Furthermore, the Bayesian learning algorithm well avoids the over-fitting problem, a common problem when using other NN training methods [18].

E. Metamodel Validation and Examination

Performance of the developed metamodel can be measured using some statistical measures. Among them, coefficient of determination (R^2) is the best as it provides a measure of target variation captured by the metamodel. R^2 is calculated as follows,

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y_i})^2} \tag{1}$$

where y_i, $\hat{y}_i$, and $\overline{y_i}$ are the i-th target, the i-th prediction, and the target mean, respectively. Prediction models with an R^2 sufficiently close to one are the best. Other measures, such as mean square error, should always be interpreted based on the range of targets to avoid misleading conclusion. Test samples are applied for examining the metamodel performance upon completion of the training stage.

TABLE I
STATISTICAL CHARACTERISTICS OF R^2 FOR ALL CONDUCTED EXPERIMENTS OVER THE GRID OF NNS

Training Method	Min	Max	Mean	S.D.
Levenberg-Marquardt	2.21	60.86	40.67	10.91
Bayesian	24.94	77.29	69.48	7.03

IV. EXPERIMENTS AND RESULTS

For the underlying problem, a training samples is a 12 labeled pair (X_i, y_i), where X_i represents the i-th set of inputs shown in Fig. 1 and y_i is the corresponding target. The weight decay cost function is applied for NN training. The regularizing factor is set to be 0.9. The training set accounts for 80% of the total available observations. The rest of the samples (20%) are used of testing the performance of developed metamodel. All samples are preprocessed to have zero mean and unit variance. This is done to give all variables an equal significance when training metamodels.

The performance of NN metamodels and their generalization power highly depends on their initial parameters and structure. To avoid any subjective judgment about NN performance, each NN metamodel is trained two times and the averaged results are reported. To examine the effects of network complexity on the metamodel performance, a grid of different structures is developed through changing number of neurons between 1 to 20 in each layer ($n_1 \in [1, 20]$, and $n_2 \in [1, 20]$). Coefficient of determination (R^2) is calculated and averaged for the test samples. Results are reported for both the Levenberg-Marquardt and Bayesian techniques.

Fig. 2 and Fig. 3 show the filled contours of the averaged R^2 over the defined grid of NNs trained using the Levenberg-Marquardt and Bayesian techniques, respectively. The horizontal and vertical axes indicate the number of neurons in the first and second hidden layers of NN metamodels. Statistical characteristics for these two experiments have been reported in Table I.

The obtained results can be analyzed and discussed from two standpoints: the prediction accuracy and the metamodel complexity effects.

A. Prediction Accuracy

According to demonstrated graphs and statistics shown in Table I, the best results in terms of R^2 are respectively 60.86% and 77.29% for the Levenberg-Marquardt and Bayesian techniques. Obviously, metamodels trained based on the Bayesian technique outperform the Levenberg-Marquardt-based metamodels. Therefore, if the prediction accuracy is the main concern, one should vote in favor of the Bayesian-based metamodels.

The provided measures in Table I also indicate that the Bayesian-based NN metamodels generate on average more accurate results over the grid of networks. The R^2 mean

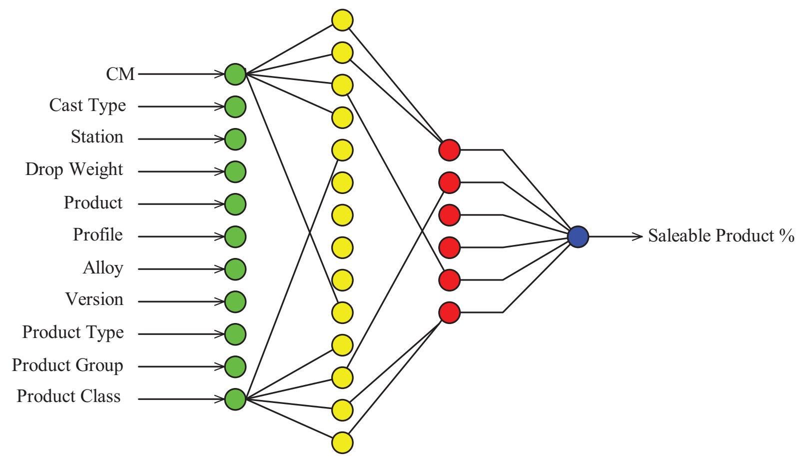

Fig. 1. The two layer NN metamodel structure with the assigned inputs

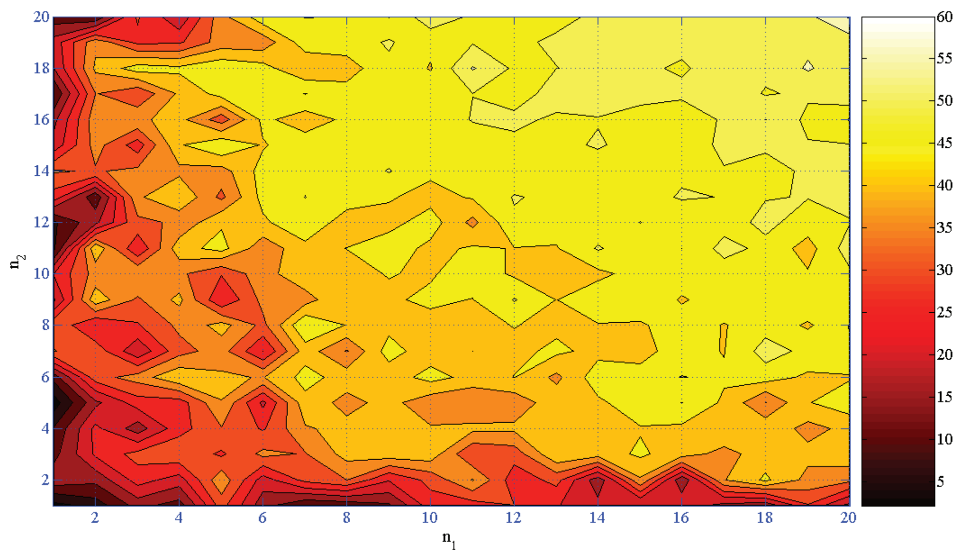

Fig. 2. Averaged coefficient of determination on a grid composed of n_1 and n_2 for NN metamodels trained using the Levenberg-Marquardt method.

for these metamodels is around 29% higher than the mean of R^2 for the Levenberg-Marquardt-based metamodels. The higher mean with a smaller standard deviation reflects better generalization power of these metamodels.

A multivariate linear regression model was also developed to see how much linear the relationship between the independent variables and the quantity of saleable product is. The calculated R^2 for this metamodel is negative. This explicitly points out that the relationships in the metal casting facility are highly nonlinear.

B. Metamodel Complexity Effects

It is also important to study the metamodel complexity effects on its performance. Fig. 4 displays the NN complexity (number of NN parameters) as a function of number of neurons in the first and second hidden layers. This graph is based on the NN configuration and architecture shown in Fig. 1. A visual comparison of this graph with results demonstrated in Fig. 2 and Fig. 3 reveals many similarities. R^2 is low with rapid fluctuations for small to medium sized NN metamodels. This explicitly means that the considered structure for the NN metamodel can not completely identify the nonlinear relationship amongst the dependent and independents

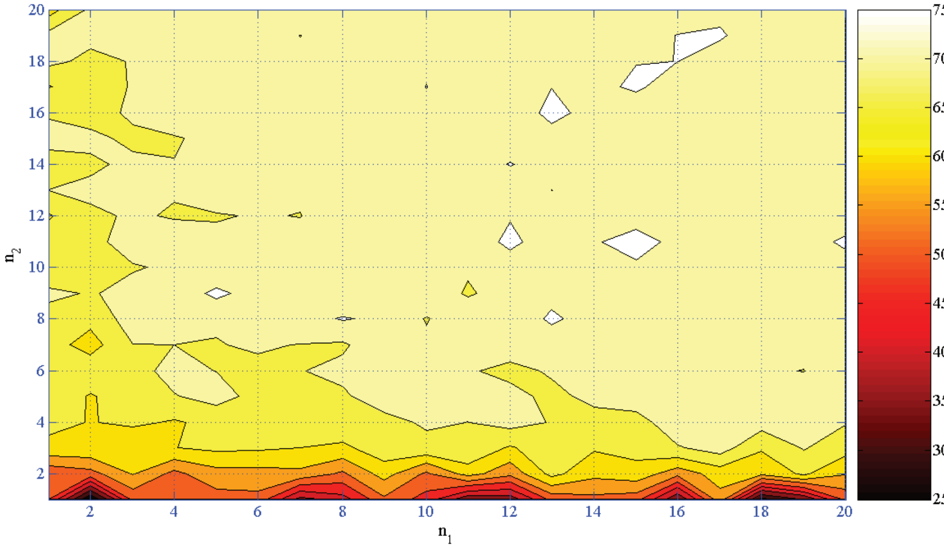

Fig. 3. Averaged coefficient of determination on a grid composed of n_1 and n_2 for NN metamodels trained using the Bayesian method.

variables. This behavior is more common for NN metamodels developed using the Levenberg-Marquardt technique. While the Bayesian-based metamodels with similar complexities have the same prediction performance, the Levenberg-Marquardt-based metamodels highly differ in terms of their performance. This can also be attributed to the high sensitivity of the Levenberg-Marquardt-based NN metamodels to their initial parameters.

As networks become bigger, R^2 gradually settles and becomes stable on large regions. Settlement of R^2 with high values in these regions is an indication of its insensitivity to the network structure. Therefore, NN metamodels in the borders of these regions are suitable candidates with satisfactory prediction performance and the minimum complexity.

It is also useful to calculate the correlation coefficient between the matrix of averaged R^2 and the network complexity. This measure provides an indication of the network performance dependency on its structure. While coefficient of correlation for the NN metamodels trained by the Levenberg-Marquardt algorithm is 68%, it is 56% for the Bayesian-based NN metamodels. These correlations again emphasize direct relationship between the metamodel performance and its structure.

If the Bayesian learning method is applied for training NN metamodels, R^2 is more robust against the network structure and complexity. Stable regions are wider with smoother borders. High hills and deep valleys are less frequent, indicating that this technique highly regularizes the effects of network complexity. These features may encourage one to apply this technique rather than the Levenberg-Marquardt technique for achieving the best results.

The best R^2 in all conducted experiments is 77.29%. The imperfectness of R^2 is attributable to any or a combination of the following factors:

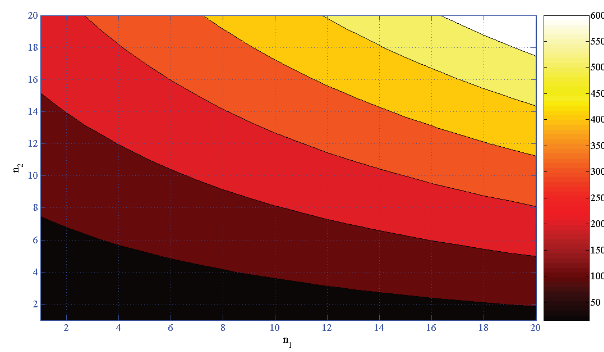

Fig. 4. Profile of metamodel complexity as a function of number of neurons in its hidden layers.

- Insufficient number of training samples: the NN metamodel has 11 inputs. As some of these inputs have many states, it is highly likely that some regions in the input space have remained unexplored. Therefore, performance of the NN metamodel in those regions is low.
- Lack of important explanatory variables: The developed metamodels have been setup mainly based on the product features including alloy, type, and group. There has been no information about casting conditions, waiting periods for crucibles, filling procedure, and other process relevant variables. Inclusion of any of these inputs in the model can enhance its prediction accuracy. At this study, lack of this information has been interpreted as uncertainty in the data set.

Although the training procedure of NN metamodels can be done offline, it is interesting to compare the elapsed time for developing NN metamodels using the two training algorithms. Experiments show that the computational requirement of the Bayesian technique is significantly higher than the Levenberg-Marquardt algorithm. In average, the NN training period using the Bayesian technique is 28.62 times longer than the NN training time using the Levenberg-Marquardt method. As NNs become bigger and more complex, the training time significantly increases for the Bayesian learning technique. Despite this, the computational burden is ignorable after the training stage and both types of metamodels quickly generate the results in real time applications.

The developed metamodels can be used as decision-aiding tools in planning and completion of the following tasks:

- Guiding to cast some alloys in specific stations according to the prediction results;
- Making decision about each product features and its relationship with the product quality; and
- Optimization of the operating schedule through conducting some if-then analysis using the developed metamodels.

V. Conclusion

A neural network metamodel for a metal casting facility was implemented in this paper. The purpose of metamodeling was to predict the quantity of saleable products before starting the casting process. In the experiments, a grid of neural network structures was developed and applied for examining the complexity effects on prediction performance. To eliminate the effects of random initializations of neural network parameters, all experiments were repeated a few times. The Levenberg-Marquardt and Bayesian learning techniques were applied for training neural networks. Calculated statistical measures for prediction results showed that prediction accuracy of neural network metamodels developed using the Bayesian method is better than prediction accuracy of the Levenberg-Marquardt-based neural network metamodels. The former metamodels were also less sensitive to the network structure. In contrast, the Levenberg-Marquardt technique was computationally less expensive. Imperfectness of prediction results is mainly attributable to a high level of uncertainty present in the operation of the underlying casthouse.

The developed metamodel can be applied for real time operational planning and production scheduling. The system operators can apply metamodels for examining a wide variety of scenarios in a short time. This makes possible optimizing short and long term production schedules and improving the overall performance and productivity.

Acknowledgment

The authors would like to acknowledge the support of the CAST CRC, established and supported by the Australian Government's Cooperative Research Centres Programme.

References

[1] P. R. Moore, J. Pu, H. C. Ng, C. B. Wong, S. K. Chong, X. Chen, J. Adolfsson, P. Olofsgard, and J. O. Lundgren, "Virtual engineering: an integrated approach to agile manufacturing machinery design and control," *Mechatronics*, vol. 13, no. 10, pp. 1105–1121, Dec. 2003.

[2] M. Bal and M. Hashemipour, "Virtual factory approach for implementation of holonic control in industrial applications: A case study in die-casting industry," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 570–581, Jun. 2009.

[3] J. H. Weber-Jahnke and J. Stier, "Virtual prototyping of automated manufacturing systems with geometry-driven petri nets," *Computer-Aided Design*, vol. In Press, Corrected Proof, pp. –, 2009.

[4] W. Dangelmaier, M. Fischer, J. Gausemeier, M. Grafe, C. Matysczok, and B. Mueck, "Virtual and augmented reality support for discrete manufacturing system simulation," *Computers in Industry*, vol. 56, no. 4, pp. 371–383, May 2005.

[5] V. Le, D. Creighton, and S. Nahavandi, "Simulation-based input loading condition optimisation of airport baggage handling systems," in *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, 2007, pp. 574–579.

[6] M. Johnstone, D. Creighton, and S. Nahavandi, "Enabling industrial scale simulation / emulation models," in *Proceedings of the 2007 Winter Simulation Conference*, 2007, pp. 1028–1034.

[7] A. Law and W. Kelton, *Simulation Modeling and Analysis*. 2nd Edition, McGraw-Hill, Singapore, 2000.

[8] D. J. Fonseca, D. O. Navaresse, and G. P. Moynihan, "Simulation metamodeling through artificial neural networks," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 3, pp. 177–183, Apr. 2003.

[9] R. R. Barton and M. Meckesheimer, "Chapter 18 metamodel-based simulation optimization," in *Simulation*, S. G. Henderson and B. L. Nelson, Eds. Elsevier, 2006, vol. Volume 13, pp. 535–574.

[10] R. W. McHaney and D. E. Douglas, "Multivariate regression metamodel: A dss application in industry," *Decision Support Systems*, vol. 19, no. 1, pp. 43–52, Jan. 1997.

[11] S. Durieux and H. Pierreval, "Regression metamodeling for the design of automated manufacturing system composed of parallel machines sharing a material handling resource," *International Journal of Production Economics*, vol. 89, no. 1, pp. 21–30, May 2004.

[12] J. P. Kleijnen, "Kriging metamodeling in simulation: A review," *European Journal of Operational Research*, vol. 192, no. 3, pp. 707–716, Feb. 2009.

[13] A. B. Badiru and D. B. Sieger, "Neural network as a simulation metamodel in economic analysis of risky projects," *European Journal of Operational Research*, vol. 105, no. 1, pp. 130–142, Feb. 1998.

[14] R. A. Kilmer, A. E. Smith, and L. J. Shuman, "Computing confidence intervals for stochastic simulation using neural network metamodels," *Computers & Industrial Engineering*, vol. 36, no. 2, pp. 391–407, Apr. 1999.

[15] F. M. Alam, K. R. McNaught, and T. J. Ringrose, "A comparison of experimental designs in the development of a neural network simulation metamodel," *Simulation Modelling Practice and Theory*, vol. 12, no. 7-8, pp. 559–578, Nov. 2004.

[16] Y. Kuo, T. Yang, B. A. Peters, and I. Chang, "Simulation metamodel development using uniform design and neural networks for automated material handling systems in semiconductor wafer fabrication," *Simulation Modelling Practice and Theory*, vol. 15, no. 8, pp. 1002–1015, Sep. 2007.

[17] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Systems with Applications*, vol. 36, no. 1, pp. 2–17, Jan. 2009.

[18] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University, Press, Oxford, 1995.

[19] C. N. Madu, "Simulation in manufacturing: A regression metamodel approach," *Computers & Industrial Engineering*, vol. 18, no. 3, pp. 381–389, 1990.

[20] J. P. C. Kleijnen and R. G. Sargent, "A methodology for fitting and validating metamodels in simulation," *European Journal of Operational Research*, vol. 120, no. 1, pp. 14–29, Jan. 2000.