

A CASE STUDY EXAMINING THE IMPACT OF FACTOR SCREENING FOR NEURAL NETWORK METAMODELS

Scott L. Rosen
Samar K. Guharay

The MITRE Corporation
McLean, VA 22012, USA

ABSTRACT

Metamodeling of large-scale simulations consisting of a large number of input parameters can be very challenging. Neural Networks have shown great promise in fitting these large-scale simulations even without performing factor screening. However, factor screening is an effective method for logically reducing the dimensionality of an input space and thus enabling more feasible metamodel calibration. Applying factor screening methods before calibrating Neural Network metamodels or any metamodel can have both positive and negative effects. The critical assumption for factor screening under investigation involves the prevalence of two-way interactions that contain a variable without a significant main effect by itself. In a simulation with a large parameter space, the prevalence of two-way interactions and their contribution to the total variability in the model output is far from transparent. Important questions therefore arise regarding factor screening and Neural Network metamodels: (a) is this a process worth doing with today's more powerful computing processors, which provide a larger library of runs to do metamodeling; and (b), does erroneously screening these buried interaction terms critically impact the level of metamodel fidelity that one can achieve. In this paper we examine these questions through the construction of a case study on a large-scale simulation. This study projects regional homelessness levels per county of interest based on a large array of budget decisions and resource allocations that expand out to hundreds of input parameters.

1 INTRODUCTION

Analysts routinely use simulation modeling to evaluate real-world systems. As the complexity of real-world systems increases, the modeling techniques employed to meaningfully capture their detailed characteristics also increase in complexity and time of execution. When model-execution times impede on decision-making timeframes, analysts often make a key tradeoff involving precision versus timeliness by resorting to simulation metamodels, or models of simulation models, to expedite analysis. Metamodels are best understood as mathematical functions with the relationship $y = f(\mathbf{x}) = g(\mathbf{x}) + \varepsilon$, where y and \mathbf{x} are scalar output and vector valued inputs to the simulation model, respectively. Moreover, consider $f(\mathbf{x})$ to be an implicit function representing the mapping between input parameters of the simulation model and the output performance measures of the simulation. The simulation metamodel $g(\mathbf{x})$ is an approximation of $f(\mathbf{x})$ with some error term ε .

There is good body of literature on the topic of simulation metamodeling; see Barton and Meckesheimer (2006) for a recent review of techniques. A few key techniques are highlighted here, which have all been applied towards simulation metamodeling:

- (a) Response Surface techniques (Myers 1976; Box and Draper 1987; Myers et al. 2009),
- (b) Splines (Eubank 1988; Deboor 1978; Myers et al. 1996),
- (c) Radial Basis Functions (Shin et al. 2002; Dyn et al. 1986; Meghabghab 2001; Hussain et al. 2002),
- (d) Stochastic Kriging (Sacks et al. 1989; Kleijnen 2000; Staum 2009; Kleijnen 2009; Ankenman et al. 2010),
- (e) Neural Networks (Lippman 1987; Fonseca 2003; Al-Hindi 2004), Inductive Learning (Michalski 1983), and
- (f) Genetic Programming (Koza 1992).

In the metamodeling literature, a metamodeling technique is often defined by one attribute, which can cause confusion. Radial basis functions serve as an example of this. Radial basis functions are a type of basis function that are used within a metamodel technique. However, they do not fully define the metamodel technique itself or fully classify it into a unique family as the basis functions can be expanded or mapped together in a variety of ways. A comprehensive definition of a metamodel technique must entail several factors including the form of the underlying basis functions, the structure for which they are integrated together, and the fitting strategy for achieving the metamodel with the smallest error.

When addressing large scale simulations, basis functions of complex shapes are needed along with a complex structure for piecing together these basis functions. The most intuitive method that provides this feature is the metamodel family of Neural Networks. In our past research on applying simulation metamodeling for cases with a high input parameter space, we have observed Neural Networks to yield very high fidelity and a superior goodness-of-fit to simulations over other metamodel families (Rosen et al. 2013). The flexibility of Neural Networks to repeatedly fit under a variety of network structures makes them suitable to handle a large-scale simulation consisting of many input parameters. Since a large parameter space is a realistic representation of a typical real-world scenario, research in metamodeling needs to encompass this type of problem. Such efforts with real-world scenarios will be complementary to the study of new methods that often deal with very limited number of input parameters.

The focus of this paper is on the application of metamodeling of a large-scale simulation. More specifically, we consider the problem of factor screening for Neural Network metamodels and examine the validity of some of the inherent assumptions. The objective is to gain insight into the consequences in metamodel fitting capability when the assumptions may not be valid. A case study is performed to study metamodel fitting of large scale simulations with and without factor screening methods. The simulation model used in this case study evaluates regional homelessness for a given area of interest.

Section 2 provides more background on Neural Network metamodeling and also provides background on techniques for factor screening in simulation experiments. A description of the scenario / simulation used for the case study is given in Section 3. Section 4 presents a case study involving the fitting of a Neural Network metamodel with and without factor screening. Conclusions and summary of the main results are presented in Section 5.

2 NEURAL NETWORK METAMODELS AND FACTOR SCREENING

The two topics of Neural Networks and factor screening have not been intertwined much before in the literature. In this section an overview of Neural Network metamodels and the topic of factor screening is provided. Factor screening has the inherent capability to save the amount of simulation runs required for fitting a Neural Network, especially for a large number of input parameters. The key research question needing further investigation is when to do factor screening with Neural Network metamodels and wheth-

er or not the screening methods hinder a Neural Network fitting ability. More attention needs to be given to this issue by the simulation research community since Neural Networks are such a powerful method for doing simulation metamodeling.

2.1 Neural Network Metamodeling for Large Scale Simulations

We have applied Neural Networks for the metamodeling of large scale simulations in several case studies; one of the recent applications pertain to the configuration of a standoff detection system (Rosen et al. 2013). In these cases Neural Networks achieved the best goodness of fit over all of the other metamodel families tested, which included Response Surface methods, Stochastic Kriging, and Genetic Algorithms. We have also successfully applied Neural Networks to a complex metamodeling problem involving time series inputs and outputs (Rosen et al. 2012).

Neural Networks provide a complex integration of basis functions, embedded in neurons, that is emblematic of the structure of neurons in the human brain (Al-Hindi 2004, Fonseca 2003). The basis functions in Neural Networks are connected through complex mappings coupled with varying weight coefficients; Figure 1 shows a generic illustration.

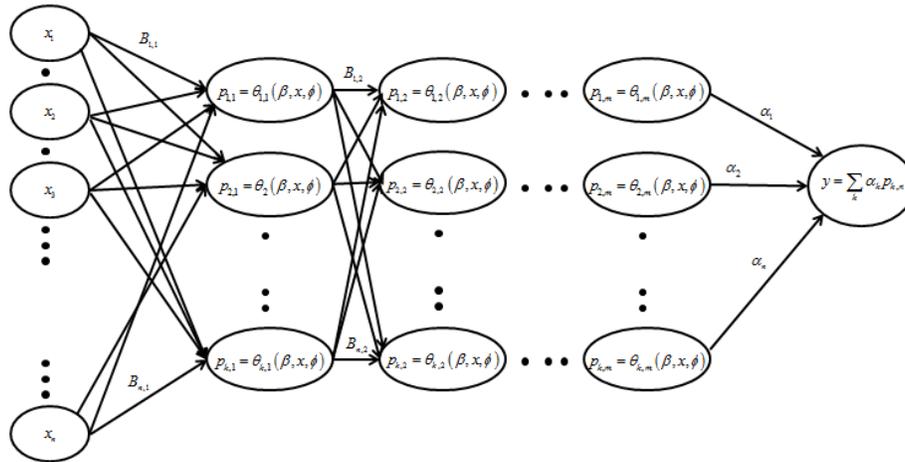


Figure 1: Neural network for simulation metamodeling with multiple layers.

Figure 1 presents a graph of a multiple layer, feed-forward Neural Network structure for a vector of simulation inputs x and performance measure of interest y . In this network structure, m is the number of hidden nodes and α and β represent the connection weights of the network. The input layer is the column of nodes on the left-hand side of the network. Each node pertains to a single input parameter of the simulation. The hidden layer is contained in the middle column of nodes, which constitute a transformation from some subset of the input nodes in the network through weight terms $\beta_{i,j}$ with the first subscript referring to the hidden layer of the Neural Network and the second subscript referring to the node within the hidden layer.

Within each node in the hidden layer is a threshold transfer function $p_k = \theta_k(\beta, x, \phi)$. It is common (Lippman 1987) to apply a sigmoid form here controlled by a threshold ϕ , but there are many other basis functions to consider, such as radial basis functions, or more generic polynomial functions. The network then maps each of the function outputs p_k to a single-valued numerical output y , which is targeting the

value of the simulation model under the input parameters x . Training or calibrating the network involves the use of sampled simulation runs and solving for the coefficients α and β that minimize the error between the simulation response and the Neural Network response.

2.2 Factor Screening

Since Bettonvil (1990) factor screening has gained more recognition as an important, preliminary step in performing simulation-based analysis, metamodeling, or optimization. The screening problem is more relevant for simulation experiments than real-world experiments as it is easier to control combinations of parameters in a simulated setting. The common traditional approaches for screening experimental parameters involve Factorial and Fractional Factorial designs (Montgomery 2008) and Group Screening designs (Watson 1961), which involve dividing parameters into groups and then performing a fractional factorial experiment with the parameters in that group. Screening methods applied to simulation models adapt these traditional approaches. Methods include Latin hypercube sampling for first-order metamodel generation, Sequential Bifurcation, Controlled Sequential Bifurcation, and Cheng's method and its many variations.

Latin Hypercube sampling (Mckay et al. 1979) can be used in factor screening by building a linear first-order metamodel from the sample simulation responses of the generated Latin Hypercube design. Based on the level of the estimated coefficients, factors can be recognized as important or unimportant. The advantage of the Latin Hypercube sampling approach is that it can be performed through the execution of a single batch of runs. For simulations requiring complicated scenario configurations or embedded cluster computing environments, this can be one of the primary reasons for choosing this approach. The critical assumption for Latin Hypercube designs is that the design matrix has orthogonal columns; this enables uncorrelated estimates of the metamodel first order polynomial coefficients. See Sanchez (2005) for Latin Hypercube design tables.

Sequential Bifurcation (Bettonvil 1996) is a group screening method, and this is known as a supersaturated experimental design in that the total number of experimental design points used is less than the total number of factors being estimated. Sequential Bifurcation (SB) is well-known for its efficiency especially when comparing to first-order polynomial metamodel generation via Latin Hypercube sampling. The basic idea is to start with a single group containing all to be screened factors. The analyst tests for a significant effect on the response due to variation of each factor in the group. If there is a significant effect of the group, the group is then split into two subgroups. Then the significant effect of the two subgroups is tested and if the group is found to be significant, it is split into two further subgroups. If a subgroup is found to not have a significant effect on the response, those factors contained in the subgroup are then found to be insignificant and can be screened out for consideration when doing metamodel calibration. This iterative process of splitting of subgroups continues until all factors are classified as significant or insignificant. There are some assumptions that can be questioned when applying Sequential Bifurcation. One of these assumptions is that a first-order polynomial is an approximation with negligible error. Through a fold-over design to achieve resolution IV design (main effects not confounded with two way interactions) adaptation to Sequential Bifurcation, Bettonvil relaxed this assumption and included a polynomial with interaction effects. However, there still exists an assumption that states that if a factor is unimportant, then every interaction effect containing that factor is marked as unimportant as well. This is questionable, and it is this assumption that we will focus on during the study within this paper.

Controlled Sequential Bifurcation (Wan et al. 2003) is an extension of Sequential Bifurcation and it accounts for the uncertainty present with this method while applying it to a stochastic situation. This kind of situation is commonly faced in the simulation metamodeling problem. The Controlled Sequential Bifurcation incorporates a two-phase hypothesis testing approach in the Sequential Bifurcation procedure in order to measure the Type I error involved in classifying unimportant factors as important as well as determining the power involved for determining critical factors. Cheng's method (1997) is another extension of Sequential Bifurcation to address the stochastic nature of simulation models. Similar assumptions

involving the adequacy of a first-order polynomial are still present in Controlled Sequential Bifurcation and Cheng's method.

The assumptions in these factor screening methods can be drastic and yield significant errors because the analyst can unknowingly remove strong two-way interaction effects; other higher-order interactions are not discussed here. In large-scale simulations with very high-dimensional input parameter spaces, these interactions may be more prevalent than a subject matter expert is initially led to believe. So, the question investigated by this case study is expected to shed light on performing factor screening. There is a tradeoff in the benefits of screening. Screening can reduce the total number of simulation runs for metamodel fitting at the potential cost of a loss in fidelity.

3 LARGE-SCALE TESTING SIMULATION

We describe here a large-scale simulation consisting of 191 input parameters. Even with subject matter expert examination, it is very difficult to screen the role of any of these parameters out of the model. The large input space and unknown number of significant factors and interactions makes it an ideal test-bed for this case study on factor screening for Neural Network metamodels. We briefly describe the modeling of the scenario in Section 3.1, and in Section 3.2 we provide an overview of the input parameters that were involved in the model.

3.1 Description of the Large Scale Simulation

The simulation model under examination evaluates regional homelessness for a given area of interest. Homelessness is a very complicated state that involves environmental factors, physical and mental health conditions, personal and public support chains, econometric impacts, and above all, homelessness support program capabilities. The objective function related to this model determines how to minimize the amount of people residing in the street. This naturally involves the right combination of resources and capacities to shelter people, to assist people back into contributing to the community, and to find permanent housing. This simulation model assists decision makers in estimating the impact of budget decisions for programs and for resource levels on homelessness levels in order to find the best tradeoff between the two.

Due to model size and complexity, it takes a significant amount of time to perform a batch of runs, with a single replication taking 15 minutes. More concerning is the setup time for the model. It takes the analyst a considerable amount of time to setup the model to evaluate a single scenario. If harnessed successfully, metamodeling not only alleviates lengthy run times of the simulation model, but it also allows for easier scenario configuration; this is another benefit of metamodeling that is often overlooked, especially in the context of very high-dimensional input parameters.

3.2 Characterization of Model Inputs

As stated earlier, there are 191 input parameters for the user to consider. Table 1 below lists the main groups of input parameters for the user to consider. The majority of parameters pertained to resources for housing services that enabled a greater capacity and effectiveness for sheltering homelessness and budget levels of various programs. Parameters on budgets impact the quality of the resources and services provided while parameters on specific services reflect the density or amount throughout the area of interest.

| Parameter Grouping | Number of Parameters |
|-------------------------|----------------------|
| Housing Services | 81 |
| Budget Levels by year | 45 |
| Support Services | 33 |
| Domiciliary Care | 12 |
| Economic forecasts | 8 |
| Healthcare benefits | 6 |
| Work Finding Assistance | 6 |

Table 1: Breakdown of input parameters in large scale simulation.

4 NEURAL NETWORK METAMODELING WITH AND WITHOUT FACTOR SCREENING

The focus of this study is to examine the merit of performing factor screening before fitting a Neural Network metamodel of a large-scale simulation. Benefit in this experiment is measured by the goodness of fit of the metamodel. The screening technique applied was Latin Hypercube sampling to perform estimation of first-order coefficients as discussed in Section 2. Due to the limited availability of the computing environment, Sequential Bifurcation was not performed. Specifically speaking, the limitations are due to the interactive requirements for scenario configuration of this large scale simulation, requirement of repeated batches of runs and repeated reservations of the equipment. However, both the Latin Hypercube sampling and Sequential Bifurcation technique make the same assumption which is, if including an input parameter does not cause a significant main effect, then this parameter is not part of any significant two way or higher interactions in the metamodel. Therefore, the lessons learned from this study can possibly be applied to Sequential Bifurcation as well.

In this experiment 382 runs are reserved to perform the factor screening such that main effects maintain orthogonality with each other. After the factor screening phase, the number of input parameters was reduced from 191 to 42. Neural Network metamodels are then fitted under varying levels of simulation design points; at 1000, 2000, 3000, 4000, and 5000 runs. A Latin Hypercube design is also used for the fitting of the metamodels after factor screening. In these Latin Hypercube designs the number of divisions is dependent on the computing resource constraints and the varying level of simulation design points that were utilized. Each of these designs had significantly more runs enabling complex order terms and interactions to be estimated. In Section 4.1, we provide the Neural Network metamodel results when fitting to all 191 input parameters as well as the results under strictly the set of 42 screened parameters. Section 4.2 discusses error at different amounts of metamodel fitting data for both cases.

The Neural Network employed was a multiple hidden layer, feed forward Network. The number of hidden layers was kept variable. Each node contained a threshold function of sigmoid form and a pre-specified threshold value. For each metamodel calibration, multiple Neural Networks were fit under varying numbers of hidden layers, threshold values and network structures and the best fitting Neural Network was retained. The fitting procedure applied was the backpropagation method, which has shown (Hect_Nielsen 1989) to have success in training Neural Networks. This fitting procedure was kept constant when fitting to all 191 input parameters and the screened set of 42 input parameters.

4.1 Metamodeling Results for the Complete Set of Design Points

The results are first presented by providing regression plots of metamodel versus actual simulation values for the fitting case involving all 5000 simulations. The performance measure of interest targeted by these metamodels is the number of homeless people served, which can imply either permanent housing or temporary shelter with food and other resources. These regression plots are first provided on the data used to

train the model and then with an extra set of data reserved specifically for validation. A uniform random generation of design points across the design space is performed to obtain data for validation. The regression plot in Figure 2 is shown below for the fitting case involving all 191 input parameters.

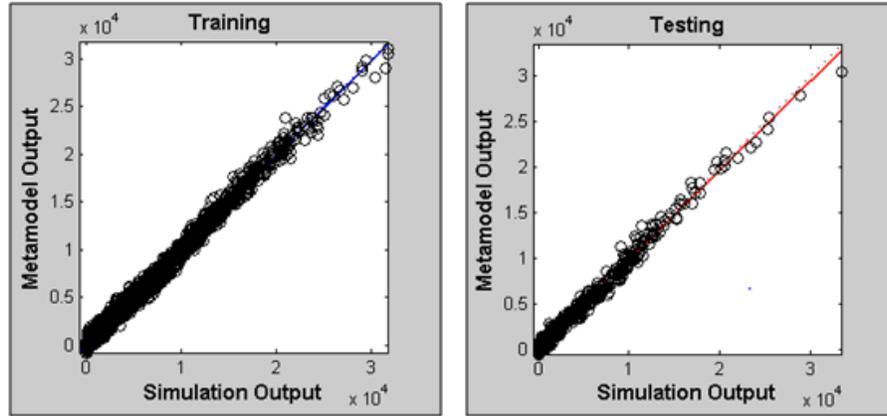


Figure 2: Metamodel vs. simulation regression plots for unscreened input space; left plot is on data used to train the model, right plot is on data not used to train the model.

The metamodel versus simulation regression plots for the fitting case involving only the screened data are provided in Figure 3. It can be seen by examination of the regression / goodness of fit plots that the fit is marginally better for the screened data.

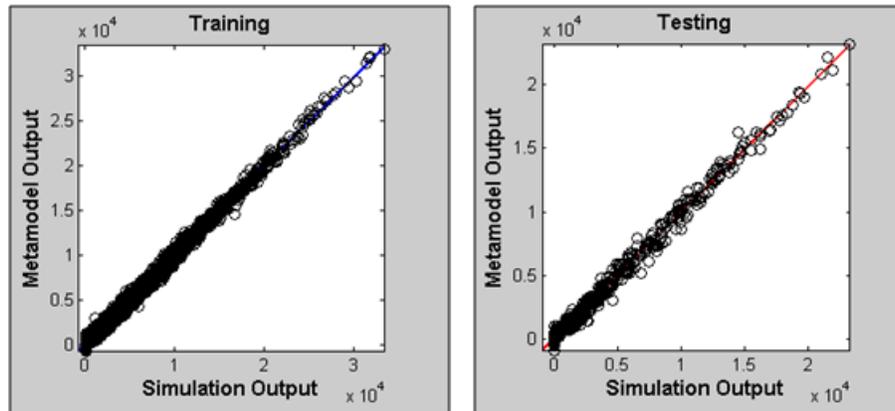


Figure 3: Metamodel vs. simulation regression plots for screened input space; left plot is on data used to train the model, right plot is on data not used to train the model.

The regression plots above depict the R value for the Neural Network metamodels. The R value for the Neural Network metamodel trained on the unscreened input space is 0.9945. The R value for the Neural Network metamodel trained on the screened input space is 0.9961. These R values were computed on data points not used to train the Neural Networks and the simulation model still required input from the variables screened from the model. The R values are very close, but it can be seen here that the screened data provides a slightly better fit when working with 5000 simulation runs. The root mean

squared error (RMSE) between the metamodels and the simulation model is 1.812% for the screened data and 2.289% for the unscreened data. This reveals in this case that the interaction effects of screened out terms from the model do not significantly limit the fidelity of the fitted metamodel.

This study shows that the impact of the factor screening assumption in two way interactions involving a screened variable does not lead to fitting of a metamodel with an increase in error. The removal of insignificant main effects did not reduce the goodness of fit of the metamodel under a large set of runs. In fact, the removal of insignificant main effects provides the extra degrees of freedom to derive more complex interaction terms in the metamodel.

4.2 Metamodeling Results under Different Quantities of Training Data

Here we study how the relative goodness of fit of the metamodel changes with respect to the number of simulation runs used to fit the metamodel. The plot below depicts the RMSE estimate under increasing levels of simulation design points: 1000, 2000, 3000, 4000, and 5000 simulation runs.

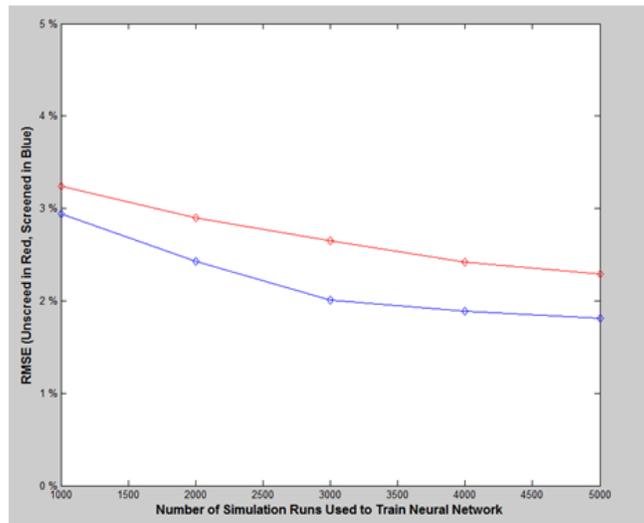


Figure 4: Neural network RMSE over varying number of simulation runs for fitting with the red line representing fitting done with the unscreened inputs and the blue line representing fitting done with the screened inputs.

The graph depicted in Figure 4 illustrates how the metamodel RMSE reduces as a function of the number of simulation runs used for fitting. The blue line represents fitting done with screened inputs and the red line represents the fitting done with the unscreened inputs. First the Neural Network metamodel trained on the screened data achieves a lower RMSE for each number of simulation runs used. For a batch of 1000 runs, the discrepancy in RMSE is not significantly larger than what it is for 5000 runs: the difference is 0.30% for 1000 runs and 0.42% for 5000 runs. This result shows that the interaction effects, for which there is concern about prematurely screening out, do not appear to be overly prevalent in this case. If these effects were significant, then the discrepancy between the RMSE values at the 5000 simulation run level would be smaller than it is at the 1000 run point. This is because at the 5000 level, there are adequate runs for the Neural Network to explore and integrate these interaction effects into the model.

An interesting finding emerges from this figure concerning the rate at which the RMSE decreases for both Neural Networks. For the Neural Network metamodel trained with screened data, the highest de-

crease in RMSE occurs between 1000 and 2000 runs with the RMSE starting to level off and reach its limit at around 3000 runs. For the Neural Network trained on unscreened data, the drop in RMSE seems to be linearly decreasing and it is unknown how and when the RMSE would likely start leveling off. This finding can have an important practical relevance in terms of the relationship among error, the number of runs, and role of factor screening.

5 CONCLUSIONS

Neural Networks have shown to be an effective technique in metamodeling of large scale simulations. However, there are still unexplored areas as far as how to best fit Neural Network metamodels when doing simulation metamodeling for scenarios with a very large number of input parameters. The issue of factor screening and its role in Neural Network metamodeling is explored in this paper. From the case study performed it is observed that factor screening not only saves the amount of simulation runs required for fitting a Neural Network, but it also does not erroneously screen too many significant interaction effects from the model such that the fidelity of the metamodel is lowered. Furthermore, the extra degrees of freedom obtained from factor screening can be applied towards finding complex interactions amongst the significant main effects. In this case study this is shown to be beneficial to metamodel fidelity as RMSE was lower for the metamodel trained on the screened simulation inputs. Another important observation involved the rate of change of metamodeling error with respect to the number of simulation runs between the cases with and without factor screening; RMSE drops off at a faster rate initially and then it levels off for the screened case while it drops off almost linearly with number of simulation runs for the unscreened case.

More case studies and explorations are necessary to gain further insight into the role of factor screening; this paper raises questions and stimulates further research on performing factor screening with Neural Network metamodels. Because of its value as a good metamodeling technique, further studies are warranted by the simulation community. The details of our work will be presented at the conference.

ACKNOWLEDGMENTS

This work is supported by the MITRE Innovation Program. The authors are thankful to Ms. Marie Francesca, Dr. Kevin Cabana, Dr. Christopher Saunders, and Dr. Len Seligman for many insightful suggestions. Thanks are also due to Ms. Stephanie Glasser and Dr. Kel Burt for their help in executing large-scale M&S experiments.

REFERENCES

- Al-Hindi, H. 2004. "Approximation Of A Discrete Event Stochastic Simulation Using An Evolutionary Artificial Neural Network." *Engineering Science* 15(1): 3-16.
- Ankenman, B., B. Nelson, J. Staum. 2010. "Stochastic Kriging For Simulation Metamodeling." *Operations Research* 58(2): 371-382.
- Barton, R., and M. Meckesheimer. 2006. "Metamodel-Based Simulation Optimization," In *Handbook in OR & MS*, Vol. 13, edited by S.G. Henderson, 2006.
- Bettonvil, B. 1990. *Detection of Important Factors by Sequential Bifurcation*. Ph.D. thesis, Tilburg University.
- Bettonvil, B. and J.P.C. Kleijnen. 1996. "Searching For Important Factors In Simulation Models With Many Factors: Sequential Bifurcation." *European Journal of Operational Research* 96: 180-194.
- Box, G. and N. Draper. 1987. *Empirical Model-building and Response Surfaces*. New York: John Wiley and Sons.

- Cheng, R. 1997. "Searching for Important Factors: Sequential Bifurcation Under Uncertainty." In *Proceedings of the 1997 Winter Simulation Conference*, edited by S. Andradottir and K.J. Healy, 275-280. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Deboor, C. 1978. *A Practical Guide To Splines*. New York: Springer-Verlag,
- Dyn, N., D. Levin, and S. Rippa. 1986. "Numerical Procedures For Surface Fitting Of Scattered Data By Radial Functions." *SIAM Journal of Scientific and Statistical Computing* 7: 639-659.
- Eubank, R. 1988. *Spline Smoothing And Nonparametric Regression*. New York: Marcel Dekker.
- Hussain, M., R. Barton, and S. Joshi. 2002. "Metamodeling: Radial Basis Functions, Versus Polynomials." *European Journal of Operational Research* 138: 142-154.
- Fonseca, D., D. Navarrese, and G. Moynihan. 2003. "Simulation Metamodeling Through Artificial Neural Networks." *Engineering Applications of Artificial Intelligence* 16:177-183.
- Hect-Nielsen, R. 1989. "Theory Of The Backpropagation Neural Network." *Proceedings of the International Joint Conference on Neural Networks* 1: 593-605.
- Kleijnen, J. 2009. "Kriging Metamodeling In Simulation: A Review." *European Journal of Operational Research* 192 707-716.
- Koza, J. 1992. *Genetic Programming: On The Programming Of Computers By Means Of Natural Selection*. Boston: MIT Press.
- Lippman, R. 1987. "An Introduction To Computing With Neural Nets." *IEEE ASSP Magazine* 4-22.
- Mckay, M. R. Beckman, and W. Conover. 1979. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics* 21(2):239-245.
- Meghabghab, M. 2001. "Iterative Radial Basis Functions Neural Networks As Metamodels Of Stochastic Simulations Of The Quality Of Search Engines In The World Wide Web." *Information Processing and Management* 37: 571-591.
- Michalski, R. 1983. *A Theory And Methodology Of Inductive Learning, Machine Learning: An Artificial Intelligence Approach*. Palo Alto: TIOGA Publishing Co.
- Montgomery, D. 2008. *Design and Analysis of Experiments*. 7th ed. New York: Wiley.
- Myers, R. 1976. *Response surface methodology*. Boston: Allyn and Bacon.
- Myers, R., M. Montgomery, and C. Anderson-Cook. 2009. *Response Surface Methodology: Process And Product Optimization Using Designed Experiments*, 3rd ed. New York: Wiley.
- Myers, R., V. Ugru, X. Luo and R. Grandhi, 1996. "MIDAS For Pre- And Post-Processing Of ASTROS Unsteady Aerodynamic Flutter Models, 6th AIAA/NASA/ISSMO." In *Symposium of Multi-disciplinary Analysis and Optimization*.
- Rosen, S., C. Saunders, and S. Guharay. 2013. "A Structured Approach for Rapidly Mapping Multi-Level System Measures via Simulation Metamodeling." Submitted to *Systems Engineering*.
- Rosen, S., C. Saunders, and S. Guharay. 2012. "Metamodeling Of Time Series Inputs And Outputs." In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, R. Pasupathy, and J. Himmelspach, 2366-2377. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Rosen, S., C. Saunders, and S. Guharay. 2013. "Configuration of a Standoff Detection System via Rapid, Model Based Systems Engineering." In *Proceedings of the IEEE Systems of Systems Engineering Conference*.
- Sacks, J., W. Welch, T. Mitchell and H. Wynn. 1989. "Design And Analysis Of Computer Experiments." *Statistical Science* 4: 409-435.

- Sanchez, S. 2005. "Work Smarter, Not Harder. Guidelines for Designing Simulation Experiments. In *Proceedings of the 2005 Winter Simulation Conference*, edited by N. Steiger, M. Kuhl, 69–82. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Shin, M., R. Sargent and A. Goel. 2002. "Gaussian Radial Basis Functions For Simulation Metamodeling." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucesan and C. Chen, 483–488. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Staum, J. 2009. "Better Simulation Metamodeling: The Why, What, And How Of Stochastic Kriging." In *Proceedings of the 2009 Winter Simulation Conference*, edited by N. Rossetti, R. Hill, and B. Johansson, 119–133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Wan, H., B. Ankenman, and B. Nelson. 2003. "Controlled Sequential Bifurcation: A New Factor Screening Method for Discrete Event Simulation. In *Proceedings of the 2003 Winter Simulation Conference*, edited by P. Sanchez and S. Chick, 565–573. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Watson, G. 1961. A Study of the Group Screening Method. *Technometrics* 3(3): 371-388.

AUTHOR BIOGRAPHIES

SCOTT L. ROSEN is a Group Leader for the Probability and Statistics Group within the MITRE Corporation's Operations Research Department. He has spent the last seven years at MITRE applying Operations Research to some of the United States' most critical and challenging Systems Engineering problems. His research interests include Simulation Optimization, Simulation Metamodeling, Decision Analysis, and Quantitative Systems Engineering. He received his B.S. from Lehigh University in Industrial and Systems Engineering in 1998 and a M.S. and Ph.D. in Industrial Engineering and Operations Research from The Pennsylvania State University in 2000 and 2003, respectively. His email address is srosen@mitre.org.

SAMAR K. GUHARAY has been leading research in the area of interdisciplinary problems covering novel sensing technology, algorithms, and rapid systems engineering analysis. He received his Ph.D. in Physics in 1980 from the University of Calcutta, India. Dr. Guharay participated in many national and international research programs over more than thirty years of his professional career. He has been affiliated with universities in different capacities. In addition to leading research programs at MITRE he has a long background as PI/Co-PI of projects sponsored by multiple agencies including NSF, NIH, DoE, DoD, industries and abroad. His email address is sguharay@mitre.org.