J. Appl. Environ. Biol. Sci., 6(2S)17-31, 2016

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

The Superiority of Evolutionary Algorithms to Robustify BPNN Learning Algorithm

Saadi bin Ahmad Kamaruddin, Nor Azura Md. Ghani, and Norazan Mohamed Ramli

Computational and Theoretical Sciences Department, Kulliyyah of Science, International Islamic University Malaysia, Jalan Istana, Bandar Indera Mahkota, 25200 Kuantan, Pahang Darul Makmur, MALAYSIA
Center for Statistical Studies and Decision Sciences, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor Darul Ehsan, MALAYSIA

Received: January7, 2016
Accepted: March 2, 2016

ABSTRACT

ANNs have contributed tremendously towards time arrangement expectation coliseum, yet the vicinity of anomalies that as a rule happen in the time arrangement information may pollute the system preparing information. Hypothetically, the most widely recognized calculation to prepare the system is the backpropagation (BP) algorithm which is in light of the ordinary least squares (OLS) estimator as far as mean squared error (MSE). Be that as it may, this calculation is not absolutely vigorous in the vicinity of anomalies and may bring about bogus forecast of future qualities. Subsequently, in this paper, we actualize another calculation which exploits firefly calculation on the firefly algorithm on the least median of squares (FA-LMedS) estimator for manufactured neural system nonlinear autoregressive (BPNN-NAR) and simulated neural system nonlinear autoregressive moving normal (BPNN-NARMA) models to provide food the different degrees of peripheral issue in time arrangement information. In addition, the execution of the proposed vigorous estimator with correlation to the first MSE blunder capacity utilizing reproduction information, taking into account root mean squared lapse (RMSE) are likewise examined in this paper. It was found that the robustified backpropagation learning calculation utilizing FA-LMedS beat the first backpropagation learning calculation.

KEYWORDS: ANNs, Time Series Outliers, Backpropagation, Firefly Algorithm, Least Median Squares

1 INTRODUCTION

The backpropagation calculation is in view of the feed forward multilayer neural system for an arrangement of inputs with indicated known characterizations. The calculation permits multilayer feed forward neural systems to take in info yield mappings from preparing specimens [1]. Once every section of the specimen set is introduced to the system, its yield reaction will be analyzed by the system as for the example information design. The yield reaction is then contrasted with the known and sought yield and the blunder quality is computed, where the association weights are balanced. The backpropagation calculation is in light of Widrow-Hoff delta learning tenet in which the weight conformity is done through mean square mistake (MSE) of the yield reaction to the specimen info [2]. The arrangement of these specimen examples are over and again displayed to the system until the blunder worth is minimized.

Despite the fact that ANNs have effectively caught the premium and worry of numerous specialists in numerous fields because of its widespread capacity as capacity approximator, the surely understood backpropagation learning calculation which is in view of the mean's minimization square mistake (MSE) expense capacity, is not vigorous in the vicinity of exceptions that may bring about lapse in information preparing procedure [3]. MSE is a blunder measure between the real and coveted yield that is utilized as a part of the well known backpropagation learning calculation of multilayered feed forward neural systems (MFNNs) preparing. [3] concur

* **Corresponding Author:** Saadi bin Ahmad Kamaruddin, Computational and Theoretical Sciences Department, Kulliyyah of Science, International Islamic University Malaysia, Jalan Istana, Bandar Indera Mahkota, 25200 Kuantan, Pahang Darul Makmur, MALAYSIA. saadiahmadkamaruddin@iium.edu.my

that this prevalent calculation is not totally strong in the vicinity of anomalies. Indeed, even a solitary anomaly can demolish the whole neural system fit [4].

Essentially, acquiring great information is the most entangled piece of guaging [5]. In this manner, accomplishing finish and smooth genuine information are very nearly zero likelihood. Anomalies are information seriously going astray from the example set of the larger part information. It has been accounted for that the event of anomalies reaches from 1 percent to more than 10 percent in common routine information [6][7]. Taking into account past studies [8-10] the presence of these anomalies represents a serious risk to the standard or traditional slightest squares investigation.

In time arrangement examination, the investigators need to depend on information to recognize which point in time are anomalies to assess the suitable remedial moves to be made so that the abnormal occasions can be evaluated precisely. Hypothesis and practice are larger part worried with straight systems, such ARMA and ARIMA models [11]. Then again, numerous arrangement display design which can't be clarified by a liner system which trigger the need of non-direct models, for instance bilinear models [12] and non-straight ARMA models (NARMA) [13].

2 PROBLEM FORMULATIONS

Where determining is worried, there rise different models in copious endeavors around there. In a present study, Padhan [14] checks that the SARIMA model is performs the best determining in bond creations in India. Notwithstanding, numerous different past studies have demonstrated something else; the Neural System is said to have beated established anticipating procedures and other factual technique [15-17]. To epitomize this, Kaastra & Boyd [15] have executed BPNN and ARIMA to anticipate what the future volumes would be, and built up the NN estimating as the measuring stick to the ARIMA model. Meanwhile, Franses and Griensven [16] find that ANNs have a tendency to beat straight models in the figure of trade rates once a day. Next, month to month development materials estimates have been created in a Malaysian setting by [18][19], utilizing both SARIMA and ANN procedures, where inevitably ANN was observed to be unrivaled. Despite the fact that ANNs is a standout amongst the most encouraging application in the region of anticipating, however the systems are dependably not powerful in taking care of information with anomalies which as a rule happen, in actuality information.

In the field of vigorous insights [9][20], numerous routines to manage the issue of exceptions have been proposed. They are intended to act appropriately when the genuine basic model goes astray from the presumptions, for example, ordinary lapse conveyance. Robust routines recognize and evacuate remote information before the model is manufactured, however a greater amount of them, including vigorous estimators, ought to be effective and solid regardless of the fact that exceptions show up. At the same time, they ought to perform well for the perceptions that are near the accepted model.

The easiest thought to make the customary neural system learning calculation more powerful to exceptions is to supplant the quadratic mistake with another symmetric and constant misfortune capacity, bringing about the nonlinear impact capacity. Such nonlinearity ought to decrease the impact of substantial mistakes. Vigorous misfortune capacities can be in view of the strong estimators with demonstrated capacity to endure diverse measures of peripheral information. Supplanting the MSE execution capacity with another strong capacity results in powerful learning strategy with the decreased effect of exceptions.

In this study, the issue of strong preparing of backpropagation neural systems (BPNN) time arrangement models in light of using the measurable powerful estimators is tended to. ANNs is picked in this exploration in light of the fact that they have been utilized with accomplishment as a part of numerous zones of logical and specialized orders including software engineering, building, medication, automated, material science and psychological sciences. The most well-known zone in which feed forward neural systems have discovered broad application is capacity close estimation because of its capacity as a widespread capacity approximator [3]. The greater part of past endeavors enhanced just on feedforward adapting so as to neurocomput for the most part the M-estimators [21].

One of the first vigorous learning calculations, the LMLS (Slightest Mean Log Squares) system, was presented by [8]. He proposed the logistic mistake capacity, got from the lapses' presumption produced with the Cauchy circulation. This commitment was considered as referential by different creators who attempted to develop more productive capacities. The thought of supposed M-estimators [20] was connected by Chen and Jain [22] in utilizing the Hampel's hyperbolic digression as another blunder rule. For this execution capacity extra scale estimator β, characterizing the extent of residuals suspected to be anomalies, was likewise presented. Hector et al. [4] observed that a hearty calculation for nonlinear autoregressive (NAR) models utilizing the summed up most extreme probability (GM) sort estimators beat the minimum squares technique in taking care of anomalies. In a study by [23], the toughening plan was connected to diminish the estimation of β with the advancement of preparing. There were additionally approaches with execution capacities in view of the tau-estimators [24] and the LTS (Slightest Trimmed Squares) estimator, while starting information examination with the MCD (Least Covariance Determinant) estimator was proposed [25]. [3] have exhibited the Reenacted Toughening for Minimum Median of Squares (SA-LMedS) calculation, applying the recreated tempering procedure to minimize execution measured by the median of squared residuals. A few endeavors to make the learning techniques for spiral premise capacity organizes more vigorous, after the methodologies for the sigmoid systems, have been additionally made [26]. The latest vigorous learning systems to be said are hearty co-preparing in view of the authoritative connection examination proposed by Sun and Jin [27], and powerful versatile learning utilizing direct network disparity methods [28].

In a paper by [21], another hearty learning calculation in view of the iterated Slightest Median of Squares (LMedS) estimator was displayed. The novel methodology is a great deal more powerful and fundamentally speedier than the SA-LMedS technique [3]. It accomplishes additionally better imperviousness to wrong preparing information. To make the preparation transform more powerful, change was made on the execution capacity as well as uproot iteratively information suspected to be anomalies. Also, an estimated system to minimize the LMedS slip basis was proposed. Meanwhile, Shinzawa, Jiang, Iwahashi and Ozaki [29] proposed molecule swarm improvement on minimum median squares (PSO-LMedS) as a hearty bend fitting technique for optical spectra. They found that, contrasted with standard bend fitting utilizing slightest squares (LS) estimator, the proposed strategy can effectively decrease undesirable impacts of sign to-commotion (SN) proportion and can yield more precise fitting results.

In the in the mean time, Xin-She Yang in 2007 from Cambridge College built up another metaheuristic calculation, in particular firefly (FA) calculation [30-35]. The firefly calculation was found to perform better contrasted with molecule swarm streamlining in taking care of abnormal state of clamor [36]. In this study, we acquaint another methodology with robustify the backpropagation learning calculation of nonlinear neural system time arrangement models utilizing FA-LMedS estimator. This paper intends to think about the execution of LS, M-estimators, ILMedS, PSO-LMedS and FA-LMedS in backpropagation calculation of both BPNN-NAR and BPNN-NARMA models. Whatever remains of this paper is composed as takes after. The related literary works are given in segment 2, and the foundation of information utilized as a part of this study is portrayed in the accompanying area, segment 3. Under segment 4, the technique review is additionally given, with the strategy used to examine the information is clarified. Besides, the exploratory settings and results are displayed in segment 5. At last, segment 6 finishes up the paper, in addition to a proposal for future attempt is additionally given.

3 DATA BACKGROUND

In this research, there were three different simulation data were used.

Background Noise Data points were selected at random and then substituted with probability δ with a background noise uniformly distributed in the specific range.

Case 1- In order to test our algorithm on the 1-D approximation task, the function by [8] was considered in this research, as also employed by previous works such as [3][21-23][37].

$$y = |x|^{-\frac{2}{3}} \tag{1}$$

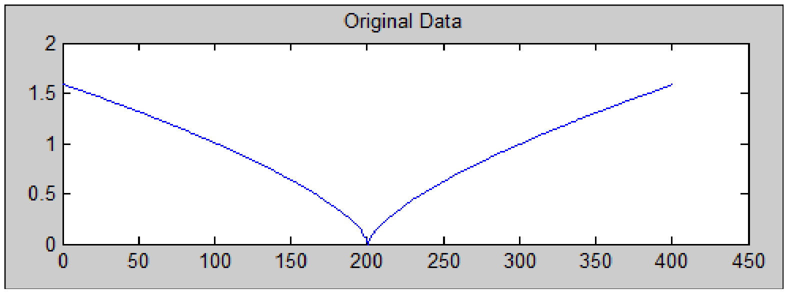

Fig. 1. Plot of data points from the function in Case 1 with no outliers, δ=0

The data used for this experiment consist of N=400 points that were generated by sampling the independent variable in the range of [-2, 2] with interval 0.01.

Case 2- Another 1-D function to be approximated was as considered in many articles [21-23] defined as:

$$y = \frac{\sin(x)}{x} \tag{2}$$

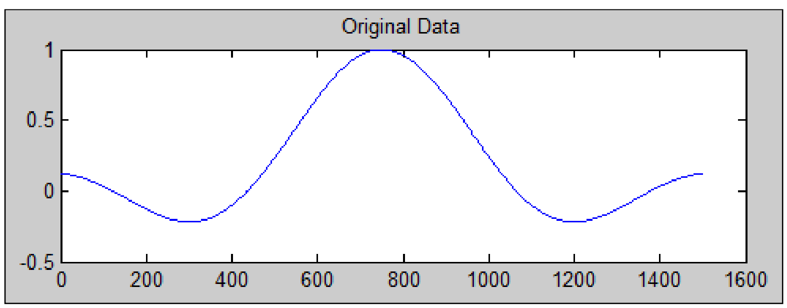

Fig. 2. Plot of data points from the function in Case 2 with no outliers, δ=0

The data used for this experiment consist of N=1500 points that were generated by sampling the independent variable in the range of [-7.5, 7.5] with interval 0.01.

Case 3- The second approximation was as suggested by [3] and [21] which can be defined as:

$$y = x_1 e^{-x_1^2 - x_2^2} \tag{3}$$

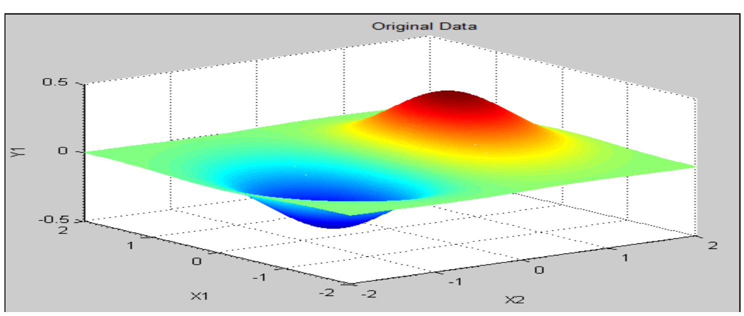

Fig. 3. Plot of data points from the function in Case 3 with no outliers, δ=0

The data points were created by sampled function on the regular 16 x 16 grid. Here, the data set were generated by sampling the independent variables, x1, x2 $\in$ [-2, 2] with interval 0.01.

4 METHODOLOGY

In the examination flowchart in Figure 4, the exploration procedure can be plainly seen. Here, the current hearty estimators on backpropagation neural system were actualized. To answer the fundamental goal of the study, the conceivable half breed powerful estimators in nonlinear autoregressive (NAR) and nonlinear autoregressive moving normal (NARMA) of neural system time arrangement were done utilizing MATLAB R2012a. At this stride, MATLAB scripts or codings were composed parallel to the numerical detailing done before. After that, the execution of the proposed robustified neural system models were analyzed utilizing reenactment information; 1-D and 2-D utilizing the standard execution measure, root mean square lapse (RMSE). At that point the powerful BPNN-NAR and BPNN-NARMA technique were tried on benchmark information. The near results were attracted those strides.

4.1 Robust Backpropagation Algorithm

The most important part of the study is the mathematical formulation improvement part of backpropagation neural network algorithm using statistical robust estimators. To make robust the traditional backpropagation algorithm based on the M-estimators concept for reducing outlier effect, the squared residuals ε_i^2 in the network error by another function of the residuals

$$E = \frac{1}{N}\sum_{i}^{N} \varepsilon_i^2 , \tag{4}$$

and this yields,

$$E = \frac{1}{N}\sum_{i}^{N} \rho(\varepsilon_i) , \tag{5}$$

where N is the total number of samples available for network training. We are deriving the updating of the network weights based on the gradient decent learning algorithm. To prevent the loss of generality, a feedforward neural network with one hidden layer will be implemented in this study. The weights from the hidden neurons to output neurons, $W_{j,i}$ are expressed as

$$\Delta W_{j,i} = -\alpha \frac{\partial E}{\partial W_{j,i}} = -\frac{\alpha}{N}\sum_{i}^{N} \frac{\partial \rho(\varepsilon_i)}{\partial W_{j,i}}$$

$$= -\frac{\alpha}{N}\sum_{i}^{N} \varphi(r_i).\frac{\partial f_j}{\partial net_j}.O_i , \tag{6}$$

where α is a user-supplied learning constant, O_i is the output of the ith hidden neuron, $O_j=f_j(net_j)$ is the output of the jth output neuron, $net_j=\sum_i W_{ji}O_i$ is the induced local field produced at the input of the activation function associated with the output neuron (j), and f_j is the activation function of the neurons in the output layer. In this work, a linear activation function (purelin) will be used in the output layer's neurons. The weights from the input to hidden neurons $W_{j,i}$ are updated as

$$\Delta W_{ji} = -\alpha \frac{\partial E}{\partial W_{ji}} = -\frac{\alpha}{N}\sum_{i}^{N} \frac{\partial \rho(\varepsilon_i)}{\partial W_{j,i}}$$

$$= -\frac{\alpha}{N}\sum_{i}^{N}\sum_{j} \varphi(r_i).\frac{\partial f_j}{\partial net_j}.W_{j,i}.\frac{\partial f_i}{\partial net_i}.I_i, \tag{7}$$

where I_i is the input to the i^{th} input neuron, $net_j = \sum_i W_{j,i} O_i$ is induced local field produced at the input of the activation function associated with the hidden neuron (i), and f_j is the activation function of the neurons in the hidden layer. We have the intention to use the tan-sigmoid function as the activation function for the hidden layer's neurons because of its flexibility.

The least-median-of-squares (LMedS) method estimates the parameters by solving the nonlinear minimization problem:

$$\min med_i \varepsilon_i^2 \tag{8}$$

That is, the estimator must create the littlest quality for the median of squared residuals processed for the whole information set. It creates the impression that this strategy is extremely strong to false matches furthermore to anomalies attributable to terrible limitation [6]. Not care for the M-estimators, on the other hand, the LMedS issue can't be moderated to a weighted minimum squares issue. It is maybe not feasible to scribble down a direct equation for the subordinate of LMedS estimator. Thus, deterministic calculations will most likely be unable to capacity to minimize that estimator. The Monte-Carlo strategy [6][38] has been polished to take care of this issue in some non-neural applications. Stochastic calculations are additionally recognized as the improvement calculations which utilize irregular hunt to accomplish an answer. Stochastic calculations are in this way generally moderate, however probability it will locate the worldwide least. One entirely well known streamlining calculation connected to minimize a LMedS-based system slip is reenacted toughening (SA) calculation. SA is a wonderful calculation in light of the fact that it is moderately broad and it has the propensity not to get stuck in either the neighborhood least or most extreme [3]. Then again, [21] finds that iterated LMedS (ILMedS) has a tendency to beat the SA-LMedS.

4.2 Firefly Algorithm (FA)

The FA was developed by Xin-She Yang at Cambridge University in 2007 based on the flashing pattern of tropical fireflies [30][32], and cuckoo search algorithm which was inspired by the brood parasitism of some cuckoo species [31]. In the simplest case for maximum optimization problems, the brightest, I of a firefly for a particular location, x could be chosen as $I(x) \propto f(x)$. However, the attractiveness β is relative and it should be judged by other fireflies, hence it will differ with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed by the media, thus the attractiveness is varied with the varying degree of absorption. In the simplest form, the light intensity $I(r)$ varies according to the inverse square law

$$I(r) = \frac{I_s}{r^2} \tag{9}$$

where I_s is the intensity at the source. For a stated medium with a fixed light absorption coefficient γ, the light intensity I varies with the distance r. That is

$$I = I_0 e^{-\gamma r} \tag{10}$$

where I_0 is the initial light intensity. In order to avoid singularity at r = 0 in the expression I_s/r^2, the combined effect of both the inverse square law and absorption can be approximated as the following Gaussian form

$$I(r) = I_0 e^{-\gamma r^2} \tag{11}$$

Since the attractiveness is proportional to the light intensity seen by other fireflies, the attractiveness β of a firefly can be define as

$$\beta_{\exp} = \beta_0 e^{-\gamma r^2} \tag{12}$$

where β_o is the attractiveness at r = 0. Since it is often faster to calculate $1/(1+r^2)$ than an exponential function, the above function can be approximated as

$$\beta_{inv} = \frac{\beta_0}{1+\gamma r^2} \tag{13}$$

These attractiveness expressions define a characteristic distance $\Gamma=1/\gamma$ over which the attractiveness is changing significantly from β_o to $\beta_o e\text{-}1$ for the βexp function and $\beta_o/2$ for the βinv function. In the real time implementation, the attractiveness function $\beta(r)$ can be monotonically decreasing functions such as the following generalized form

$$\beta(r) = \beta_0 e^{-\gamma r^m}; (m \geq 1). \tag{14}$$

For a fixed γ, the characteristic length becomes

$$\Gamma = \gamma^{-1/m} \to 1; (m \to \infty). \tag{15}$$

On the other hand, for a specific length scale Γ in an optimization problem, the parameter γ can be used as a typical initial value. That is

$$\gamma = \frac{1}{\Gamma^m}. \tag{16}$$

The distance between any two fireflies i and j at x_i and x_j, respectively is the Cartesian distance,

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \tag{17}$$

where x_i, k is the k^{th} component of the spatial coordinate x_i of i^{th} firefly. In 2-D case, the distance between any two fireflies i and j can be written as

$$r_{i,j} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2}. \tag{18}$$

The movement of a firefly i is attracted to another brighter firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{i,j}^2} * (x_j - x_i) + \alpha \in_i, \tag{19}$$

where the second term is due to the attraction and third term is randomization with α being the randomization parameter, and $\in_i$ is a vector of random numbers being drawn from a Gaussian distribution or uniform distribution. For example, the simplest form is $\in_i$ can be replaced by rand - 1/2 where rand is a random number generator uniformly distributed in [0, 1]. For most implementation, usually $\beta_o=1$ and $\alpha \in [0,1]$ [30]. It is important pointing out that xi is a random walk biased towards the brighter fireflies. If $\beta_o=0$, it becomes a simple random walk. Furthermore, the randomization term can easily be extended to other distributions such as Levy flights [30]. The parameters γ now characterizes the contrast of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA algorithm behaves. In theory, $\gamma \in [0,\infty)$, but in actual practice, $\gamma \in 0(1)$ is determined by the characteristic length Γ of the system to be optimized. Thus, for most applications, it typically varies from 0.1 to 10 [30][35].

The descriptive measures for model selection was root mean square error (RMSE) which can be written as

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N} \varepsilon(t)^2}{N}} \quad (20)$$

Comparable the same number of past exploration endeavors illustrated, it is conceivable to prepare the FNNs with median neuron information capacity with inclination based calculations [21]. In these systems, basic summation in the neuron info is supplanted by the median information capacity, which likewise causes non-differentiability of the system mistake capacity. The RBP learning calculation was produced to take out the impact of moderate union for the low slope greatness brought about by the level areas of sigmoid actuation capacities. This is the motivation behind why the calculation is more suitable to the LMedS lapse capacity, just essentially on the grounds that legitimate estimation of the slope's indication is more probable than fitting estimation of its precise worth [21]. The progressions of the estimated RBP calculation for the LMedS slip criteria can be composed as takes after:

Step 1: Utilization backpropagation to figure subsidiaries of the MSE execution capacity, εmed
Step 2: Upgrade the system weights
Step 3: Adjust components of information with foundation commotion
Step 4: Calculate the network LmedS performance ε_{med}
Step 5: If the LMedS execution is minimized to the expected objective, or the quantity of ages surpasses the most extreme number of ages, quit preparing. Generally go to the first step

The basic NAR-ANN formulation can be represented as below;

$$H(x) = purelin\left[\sum_{j=1}^{m} w_{jk}\left[\tanh\left(\sum_{i=1}^{l} w_{ij}\left[x(t-1), x(t-2), \dots, x(t-n_y)\right] + \varepsilon(t)\right)\right]\right] \quad (21)$$

The finalized NARMA-ANN formulation can be represented as below;

$$H(x) = purelin\left[\sum_{j=1}^{m} w_{jk}\left[\tanh\left(\sum_{i=1}^{l} w_{ij}\left[\begin{array}{l}[x(t-1), x(t-2), \dots, x(t-n_y), \\ \varepsilon(t-1), \varepsilon(t-2), \dots, \varepsilon(t-n_\varepsilon)] + \varepsilon(t)]\end{array}\right]\right)\right]\right] \quad (22)$$

where
$H(x)$ is the estimated model,
$x(t-1), x(t-2), \dots, x(t-n_y)$ are lagged input terms,
$\varepsilon(t-1), \varepsilon(t-2), \dots \varepsilon(t-n_\varepsilon)$ are lagged residual terms, and the lagged residual terms are obtained recursively after the initial model (based on the input and output terms) is found.
Hence, $\varepsilon(t)$ are the white noise residuals.
l is the input neurons with index i
m is the hidden neurons with index j
n is the output neurons with index k

5 RESULTS AND DISCUSSIONS

In light of the Tables 1, 2 and 3, the customary calculation delivered the best results in view of the littlest RMSE values for the perfect information without anomalies. This result is parallel with the case that MSE blunder capacity is ideal for the information without anomalies by [3][21][39].

On the other hand, the circumstance is changed for the information containing falsely produced anomalies where the MSE-based technique totally loses its effectiveness. This can be demonstrated by the strategy's breakdowns as indicated in Tables 1, 2 and 3. The strong calculations of FA-LMedS perform essentially better contrasted with MSE-based expense capacity with the most minimal RMSE values in every one of the three cases, as indicated in Tables 1, 2 and 3. This is perhaps because of the reason that firefly calculation perform better for larger amount of clamor [36] and when it fused into backpropagation neural system preparing calculation, it focalize at quicker rate with least feedforward neural system outline [40]. As said by [32], the fireflies calculation is the uncommon instance of quickened molecule swarm streamlining calculation. FA-LMedS keep up to endure with little mistakes for the information comprising anomalies more than 60 percent, as demonstrated in Tables 1, 2 and 3.

Taking into account the experimental consequences of this exploration, there are three general conclusions can be drawn concerning the tried calculations:

i. The learning calculation taking into account the MSE capacity works best when the information are spotless or not defiled by exceptions.
ii. The FA-LMedS calculation appears to beat the MSE expense capacity for the information with gross slips.
iii. The hearty NARMA-ANN strategy performed the best by means of FA-LMedS contrasted with the powerful NAR-ANN technique.

6 CONCLUSIONS AND RECOMMENDATIONS

In this paper we displayed novel adjusted backpropagation neural system learning calculation taking into account the half breed firefly calculation with slightest middle of squares, otherwise called FA-LMedS. Our calculation is not just vigorous to the vicinity of different measures of anomalies additionally quicker and more precise contrasted with the first backpropagation calculation which is in view of MSE expense capacity. The execution prevalence of our system in correlation over different calculations, in the vicinity of distinctive level of exceptions, was exactly illustrated.

For the perfect information or the information which don't experience the ill effects of exceptions issue, the traditional backpropagation in light of the MSE calculation performed extremely well. In addition, for information comprising gross mistakes and exceptions up to 70 percent, it can be presumed that the model manufactured by the system prepared with the FA-LMedS learning calculation is more exact than for the conventional strategy. This implies that the new proposed hearty stochastic calculation in view of the LMedS estimator has shown even enhanced heartiness over the first mistake capacity, whereby it figured out how to endure up to 70 percent exceptions while keeping up an exact model on both reproduction information.

The proposed hearty calculations for preparing neural systems can likewise be utilized as a part of uses other than capacity rough guess and framework distinguishing proof, for example, design acknowledgment, framework ID, machine learning, counterfeit consciousness, monetary danger administration, automated, and in addition quality control and streamlining.

In future endeavors, a few options might be contemplated in this examination, which are:

i. To build the quantity of cycle number and swarm size molecule FA-LMedS further watch the execution.
ii. To find quicker and more capable different options for minimize the LMedS slip model capacity, e.g. by controlling the upgraded adaptations of the RBP technique.

iii. To apply other quicker preparing calculation into the system and further analyze the outcomes, for example, traincgf [3] and trainlm [39].
iv. To apply other existing factual strong stochastic estimators and soon half and half them with quick deterministic calculations.
v. To train the system utilizing diverse enactment capacities as said by [1].
vi. To further enhance the backpropagation learning calculation utilizing enhanced rendition of slightest trimmed squares (LTS) such those as proposed by [39][41][42].
vii. To test the new FA-LMedS calculation on close estimation undertaking of a 2-D winding as recommended by [21], and in addition Information set 4 to Information set 14 as proposed by [39].
viii. To analyze the normal time execution (in seconds) for all the tried calculations in reproduction studies as indicated by [21].
ix. To analyze the new vigorous neural system technique with the direct improved neural system routines utilizing molecule swarm enhancement [43][44].
x. To further cross breed neural system with enhanced form firefly calculation such the one proposed by [45], called Half and half Transformative Firefly Calculation (HEFA).
xi. To further investigate diverse mix of system parameters, for example, number concealed hubs, number of information slacks, number of yield slacks.
xii. To think about the execution of the proposed calculation with the current ones, for example, M-estimators, iterative slightest middle squares (ILMedS) and molecule swarm on minimum middle squares (PSO-LMedS).

ACKNOWLEDGEMENTS

Special thanks dedicate to Universiti Teknologi MARA and Malaysian Ministry of Education (MOE) for supporting this research under the Research Grant No. 600-RMI/FRGS 5/3 (137/2014) and No. 600-RMI/DANA 5/3/CIFI (65/2013).

REFERENCES

[1] Sibi, P., Jones, S. A. and Siddarth, P. 2013. Analysis of Different Activation Functions Using Back Propagation Neural Networks. *Journal of Theoretical and Applied Information Technology*, 473, pp. 1264-1268.
[2] Engelbrecht, A. P. 2007. Computational Intelligence: An introduction. Wiley.
[3] El-Melegy, M. T., Essai, M. H. and Ali, A. A. 2009. Robust training of artificial feedforward neural networks. *Found Comput Intell*, 1, pp. 217–242.
[4] Hector, Claudio, M. and Rodrigo, S. 2002. Robust Estimator for the Learning Process in Neural Network Applied in Time Series. *Artificial Neural Networks-ICANN 2002*, pp. 1080-1086.
[5] Gitman, L. J. and McDaniel, C. 2008. The future of business, The essentials, South-Western Pub.
[6] Zhang, Z. 1997. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 151, pp. 59-76.
[7] Pearson, R. K. 2002. Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 101, pp. 55-63.
[8] Liano, K. 1996. Robust error measure for supervised neural network learning with outliers. *IEEE Trans. Neural Networks,* 71, pp. 246-250.
[9] Huber, P. J. 1981. Robust Statistics. John Wiley and Sons, New York.
[10] Rousseeuw, P. J. and Leroy, A. M. 1987. Robust regression and outlier detection, Wiley, New York.
[11] Box, G. E., Jenkins, G. M. and Reinsel, G. C. 1994. Time series analysis, forecasting and control, Ed. Prentice Hall.
[12] Gabr, M. M. 1998. Robust estimation of bilinear time series models. *Comm. Statist, Theory and Meth*, 271, pp. 41-53.
[13] Connor, J. T. and Martin, R. D. 1994. Recurrent neural networks and robust time series prediction. *IEEE Transactions of Neural Networks*, 52, pp.240-253.
[14] Padhan, P. C. 2012. Use of univariate time series models for forecasting cement productions in India. *International Research Journal of Finance and Economic*s, 831, pp. 167-179.
[15] Kaastra and Boyd, M. S. 1995. Forecasting futures trading volume using neural networks. *The Journal of Futures Markets*, 18, pp. 953-970.
[16] Franses, P. H. and Griensven, K. V. 1998. Forecasting exchange rates using neural networks for technical trading rules. *Studies in Nonlinear Dynamics and Econometrics*, 42, pp.109–114.

[17] Pei, L., Chen, S. H., Yang, H. H., Hung, C. T. and Tsai, M. R. 2008. Application of artificial neural network and SARIMA in portland cement supply chain to forecast demand. *Natural Computation*, 3, pp. 97-101.
[18] Kamaruddin, S. B. A., Ghani, N. A. M. and Ramli, N. M. 2013. Determining the Best Forecasting Method to Estimate Unitary Charges Price Indexes of PFI Data in Central Region Peninsular Malaysia. *AIP Conf. Proc.*, 11522, pp. 1232-1239.
[19] Kamaruddin, S. B. A., Ghani, N. A. M. and Ramli, N. M. 2014. Best Forecasting Models for Private Financial Initiative Unitary Charges Data of East Coast and Southern Regions in Peninsular Malaysia. *International Journal of Economics and Statistics*, 2, pp. 119-127.
[20] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. and Stahel, W. A. 1986. Robust Statistics, The approach based on influence functions. Wiley, New York.
[21] Rusiecki, A. 2012. Robust Learning Algorithm Based on Iterative Least Median of Squares. *Neural Process Lett*, 36, pp. 145-160.
[22] Chen, D. S. and Jain, R. C. 1994. A robust back propagation learning algorithm for function approximation. *IEEE Trans. Neural Networks*, 53, pp. 467–479.
[23] Chuang, Su, S. and Hsiao, C. 2000. The annealing robust backpropagation ARBP learning algorithm. *IEEE Trans Neural Networks*, 115, pp. 1067–1076.
[24] Pernia-Espinoza, V., Ordieres-Mere, J. B., Martinez-de-Pison, F. J. and Gonzalez-Marcos 2005. A TAO - robust backpropagation learning algorithm. *Neural Network*, 182, pp. 191–204.
[25] Rusiecki, A. 2008. Robust MCD-based backpropagation learning algorithm. *Artificial Intelligence and Soft Computing*, pp. 154-163.
[26] Chuang, C., Jeng, J. T. and Lin, P. T. 2004. Annealing robust radial basis function networks for function approximation with outliers. *Neurocomputing*, 56, pp. 123–139.
[27] Sun, S. and Jin, F. 2011. Robust co-training. *International Journal of Pattern Recognition Artificial Intelligence*, 257, pp. 1113–1126.
[28] Jing, X. 2012. Robust adaptive learning of feedforward neural networks via LMI optimizations. *Neural Networks*, 31, pp. 33-45.
[29] Shinzawa, H., Jiang, J. H., Iwahashi, M. and Ozaki, Y. 2007. Robust curve fitting method for optical spectra by least median squares LMedS estimator with particle swarm optimization PSO. *Analytical sciences*, 23, pp. 781-785.
[30] Yang, X. S. 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press, United Kingdom.
[31] Yang, X. S. and Deb, S. 2009. Cuckoo Search via Levy Flights. *Proceedings of IEEE World Congress on Nature & Biologically Inspired Computing*, pp. 210-214.
[32] Yang, X. S. 2009. Firefly Algorithms for Multimodal Optimization. Proceeding of the 5th Symposium on Stochastic Algorithms, Foundations and Applications, Eds. O. Watanabe and T. Zeugmann, Lecture Notes in Computer Science, 5792, pp. 169-178.
[33] Yang, X. S. and He, X. 2013. Firefly Algorithm: Recent Advances and Applications. *International Journal Swarm Intelligence*, 11, pp. 36-50.
[34] Yang, X. S. 2010a. Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley and Sons, USA.
[35] Yang, X. S. 2010b. A New Metaheuristic Bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization, Eds. J. R. Gonzales et al., Springer, SCI, 284, pp. 65-74.
[36] Pal, S. K., Rai, C. S. and Singh, A. P. 2012. Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non-Linear Optimization Problems. *International Journal of Intelligent Systems and Applications*, 10, pp. 50-57.
[37] Rusiecki, A. L. 2005. Fault tolerant feedforward neural network with median neuron input function. Electronics Letters, 4110, pp. 603-605.
[38] Aarts, E., Korst J. and Michiels, W. 2005. Simulated annealing. *Search methodologies*, pp. 187-210.
[39] Beliakov, G., Kelarev, A. and Yearwood, J. 2012. Derivative-free optimization and neural networks for robust regression. *Optimization*, 6112, pp. 1467-1490.
[40] Nandy, S., Sarkar, P. P. and Das, A. 2012. Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training. *International Journal of Computer Applications*, 4222, pp. 8-16.
[41] Zioutas, G., Avramidis, A. and Pitsoulis, L. 2007. Penalized Trimmed Squares and A Modi-Fication of Support Vectors for Un-Masking Outliers in Linear Regression. *REVSTAT–Statistical Journal*, 51, pp. 115-136.
[42] Alfons, A., Croux, C. and Gelper, S. 2013. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 71, pp. 226-248.
[43] Rakitianskaia, A. S. and Engelbrecht, A. P. 2012. Training feedforward neural networks with dynamic particle swarm optimization. *Swarm Intelligence*, 63, pp. 233-270.
[44] Brajevic, I. and Tuba, M. 2013. Training Feed-Forward Neural Networks Using Firefly Algorithm. *Recent Advances in Knowledge Engineering and Systems Science*, pp. 156-161.

[45] Abdullah, A., Deris, S., Mohamad, M. S. and Hashim, S. Z. M. 2012. A new hybrid firefly algorithm for complex and nonlinear problem. *Distributed Computing and Artificial Intelligence*, pp. 673-680.

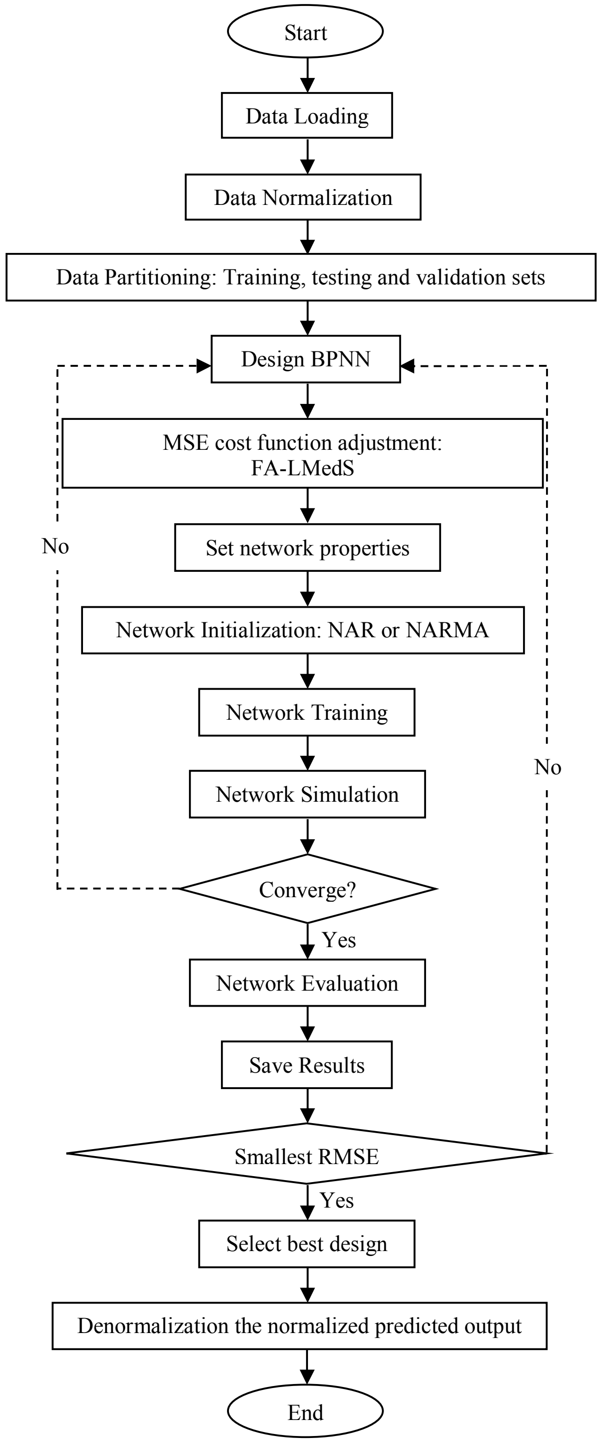

Fig. 4. Flowchart of proposed robust BPNN-NAR and BPNN-NARMA

Table 1. The RMSE scores for test function in Case 1

Max Lags of inputs Ny)	Max Lags of errors Ne)	Hidden	Swarm Size	Iteration	Percentage of Outliers δ)	Algorithm			
						MSE		FA-LMedS	
						NAR	NARMA	NAR	NARMA
2	2	5	5	5	0	0.0165	0.0144	0.0006	0.0006
2	2	10	5	5	0	0.0317	0.0232	0.0009	0.0009
2	2	20	5	5	0	0.0559	0.0373	0.0010	0.0009
2	2	5	5	5	10	2537.2571	2243.7403	0.0073	0.0055
2	2	10	5	5	10	7806.4086	6968.4181	0.0040	0.0048
2	2	20	5	5	10	3197.1237	2696.4526	0.0024	0.0013
2	2	5	5	5	20	2691.9232	2483.4234	0.0005	0.0069
2	2	10	5	5	20	2681.6996	1561.1672	0.0041	0.0073
2	2	20	5	5	20	3837.5206	3669.8813	0.0092	0.0026
2	2	5	5	5	30	3436.0919	4352.2928	0.0003	0.0050
2	2	10	5	5	30	3483.9997	3468.8221	0.0047	0.0056
2	2	20	5	5	30	8194.4084	9399.368	0.0068	0.0071
2	2	5	5	5	40	4842.7778	3768.5752	0.0060	0.0024
2	2	10	5	5	40	12529.7165	12785.6565	0.0042	0.0018
2	2	20	5	5	40	2616.636	2040.8494	0.0023	0.0020
2	2	5	5	5	50	4613.7007	4797.0081	0.0028	0.0070
2	2	10	5	5	50	6696.0732	7111.9116	0.0071	0.0073
2	2	20	5	5	50	5870.884	7450.8175	0.0007	0.0028
2	2	5	5	5	60	6304.1263	5309.1228	0.0037	0.0034
2	2	10	5	5	60	15210.3341	19075.8857	0.0005	0.0039
2	2	20	5	5	60	2440.4538	3412.5697	0.0266	0.092
2	2	5	5	5	70	5608.1938	5209.1128	0.0561	0.0530
2	2	10	5	5	70	5159.2982	5390.3442	0.0445	0.0937
2	2	20	5	5	70	3338.7296	4171.1588	0.0182	0.0149
3	3	5	5	5	0	0.2266	0.9105	0.0331	0.0381
3	3	10	5	5	0	0.1888	0.5777	0.0704	0.0363
3	3	20	5	5	0	0.8679	0.3433	0.0756	0.0610
3	3	5	5	5	10	2422.4903	2354.6119	0.0631	0.0243
3	3	10	5	5	10	2961.5131	2019.4265	0.0733	0.0626
3	3	20	5	5	10	1703.9431	1457.9506	0.0702	0.0752
3	3	5	5	5	20	2782.5779	2530.3214	0.0629	0.0421
3	3	10	5	5	20	4023.1339	2749.3138	0.0740	0.0395
3	3	20	5	5	20	2388.1344	3923.2193	0.0390	0.0408
3	3	5	5	5	30	2340.4295	2597.6883	0.0995	0.2822
3	3	10	5	5	30	2565.6173	2779.8586	0.0981	0.0690
3	3	20	5	5	30	8138.588	10017.3766	0.0137	0.0487
3	3	5	5	5	40	3929.4474	4579.1315	0.0687	0.0609
3	3	10	5	5	40	3069.3175	2993.2578	0.0176	0.0983
3	3	20	5	5	40	3252.8795	3788.1158	0.0870	0.0507
3	3	5	5	5	50	4103.3899	3686.9169	0.0843	0.0731
3	3	10	5	5	50	3213.4135	3489.5679	0.0251	0.0554
3	3	20	5	5	50	2344.8786	2746.5734	0.0076	0.0121
3	3	5	5	5	60	4979.5091	4377.9622	0.0736	0.0320
3	3	10	5	5	60	4662.0489	47763.665	0.0452	0.0295
3	3	20	5	5	60	7100.6318	8801.7551	0.0989	0.0926
3	3	5	5	5	70	1786.7835	1608.6392	0.0020	0.0019
3	3	10	5	5	70	2300.8852	2439.9802	0.0312	0.0362
3	3	20	5	5	70	3513.0521	4485.2789	0.0628	0.0402

Table 2. The RMSE scores for test function in Case 2

Max Lags of inputs Ny)	Max Lags of errors Ne)	Hidden	Swarm Size	Iteration	Percentage of Outliers δ)	Algorithm			
						MSE		FA-LMedS	
						NAR	NARMA	NAR	NARMA
2	2	5	5	5	0	0.1288	0.1202	0.0297	0.0252
2	2	10	5	5	0	0.1782	0.1824	0.0392	0.0982
2	2	20	5	5	0	0.2366	0.2595	0.0456	0.0751
2	2	5	5	5	10	1592.8770	1801.0386	0.0284	0.0803
2	2	10	5	5	10	2793.9951	2639.77614	0.0473	0.0025
2	2	20	5	5	10	1788.0502	1642.0878	0.0349	0.0092
2	2	5	5	5	20	1640.7081	1575.8881	0.0339	0.0031
2	2	10	5	5	20	1637.5895	1249.4667	0.0506	0.0231
2	2	20	5	5	20	1958.9590	1915.6934	0.0431	0.0159
2	2	5	5	5	30	1853.6698	2086.2149	0.0748	0.0664
2	2	10	5	5	30	1866.5475	1862.4774	0.0197	0.0036
2	2	20	5	5	30	2862.5877	3065.8388	0.0298	0.0016
2	2	5	5	5	40	2200.6312	1941.2818	0.0331	0.0074
2	2	10	5	5	40	3539.7339	3575.7036	0.0218	0.0069
2	2	20	5	5	40	1617.6019	1428.5830	0.1915	0.1705
2	2	5	5	5	50	2147.9526	2190.2073	0.2086	0.1941
2	2	10	5	5	50	2587.6771	2666.8167	0.1862	0.1792
2	2	20	5	5	50	2422.9907	2729.6185	0.3065	0.2302
2	2	5	5	5	60	2510.8019	2304.1533	0.1941	0.1647
2	2	10	5	5	60	3900.0428	4367.5949	0.3575	0.0967
2	2	20	5	5	60	1562.1951	1847.3141	0.1428	0.1400
2	2	5	5	5	70	2368.1625	2282.3480	0.2190	0.0242
2	2	10	5	5	70	2271.4088	2321.7114	0.2666	0.2622
2	2	20	5	5	70	1827.2190	2042.3415	0.2729	0.2027
3	3	5	5	5	0	0.2388	0.2379	0.2304	0.0171
3	3	10	5	5	0	0.1973	0.2071	0.4367	0.4006
3	3	20	5	5	0	0.1596	0.1596	0.1847	0.1636
3	3	5	5	5	10	1556.4351	1534.4744	0.2282	0.2005
3	3	10	5	5	10	1720.9047	1421.0652	0.2321	0.0266
3	3	20	5	5	10	1305.3517	1207.4562	0.2042	0.0561
3	3	5	5	5	20	1668.1068	1590.6983	0.2379	0.0445
3	3	10	5	5	20	2005.7751	1658.1054	0.2070	0.0182
3	3	20	5	5	20	1545.3589	1980.7118	0.1596	0.0331
3	3	5	5	5	30	1529.8462	1611.7345	0.1534	0.0704
3	3	10	5	5	30	1601.7544	1667.2908	0.1421	0.0756
3	3	20	5	5	30	2852.8210	3165.0239	0.1207	0.0631
3	3	5	5	5	40	1982.2833	2139.8905	0.1590	0.0733
3	3	10	5	5	40	1751.9467	1730.1034	0.1658	0.0702
3	3	20	5	5	40	18035.7409	19463.0826	0.1980	0.0629
3	3	5	5	5	50	20256.8258	19201.3461	0.1611	0.0740
3	3	10	5	5	50	17925.996	18680.385	0.1667	0.0390
3	3	20	5	5	50	1531.2996	1657.2788	0.3165	0.0995
3	3	5	5	5	60	2231.4813	2092.3580	0.2139	0.0981
3	3	10	5	5	60	2159.1776	2185.4899	0.1730	0.0137
3	3	20	5	5	60	2664.7010	2966.7752	0.1946	0.0687
3	3	5	5	5	70	1336.7062	1268.3214	0.1920	0.0176
3	3	10	5	5	70	1516.8668	1562.0435	0.1868	0.0870
3	3	20	5	5	70	1874.3137	2117.8477	0.1657	0.0843

Table 3. The RMSE scores for test function in Case 3

Max Lags of inputs Ny)	Max Lags of errors Ne)	Hidden	Swarm Size	Iteration	Percentage of Outliers δ)	Algorithm			
						MSE		FA-LMedS	
						NAR	NARMA	NAR	NARMA
2	2	5	5	5	0	0.0226	0.0252	0.0371	0.0224
2	2	10	5	5	0	0.0436	0.0497	0.0307	0.0244
2	2	20	5	5	0	0.1112	0.1186	0.0569	0.0672
2	2	5	5	5	10	4359.7566	8641.6380	0.057	0.0636
2	2	10	5	5	10	1079.2603	1365.1487	0.0525	0.0466
2	2	20	5	5	10	3988.2749	4121.7564	0.1150	0.0153
2	2	5	5	5	20	3717.8782	3244.3472	0.0756	0.0550
2	2	10	5	5	20	3073.1547	2446.1292	0.2471	0.0209
2	2	20	5	5	20	5694.1254	6723.3495	0.0424	0.0272
2	2	5	5	5	30	5730.2224	6368.4176	0.0517	0.0494
2	2	10	5	5	30	5256.0255	4664.3883	0.1019	0.0236
2	2	20	5	5	30	11503.8636	11533.4735	0.0969	0.0136
2	2	5	5	5	40	7562.8682	5504.5465	0.1372	0.0378
2	2	10	5	5	40	2471.0467	3209.0136	0.1727	0.0490
2	2	20	5	5	40	4243.4113	2721.2212	0.0336	0.0046
2	2	5	5	5	50	5179.6874	4948.2608	0.0696	0.0075
2	2	10	5	5	50	10196.5565	12361.8442	0.0269	0.0201
2	2	20	5	5	50	9697.4329	11364.5493	0.0248	0.0174
2	2	5	5	5	60	13726.6642	13780.5056	0.0156	0.0136
2	2	10	5	5	60	1727.7562	3490.8264	0.0366	0.0290
2	2	20	5	5	60	3368.2453	4460.2079	0.0435	0.0376
2	2	5	5	5	70	7393.3438	8065.0819	0.0346	0.0321
2	2	10	5	5	70	9217.7368	14128.0656	0.0939	0.0530
2	2	20	5	5	70	5306.5654	8367.3694	0.0376	0.0271
3	3	5	5	5	0	0.0937	0.1255	0.1278	0.1174
3	3	10	5	5	0	0.0749	0.0933	0.0204	0.0113
3	3	20	5	5	0	0.0381	0.0379	0.0479	0.0102
3	3	5	5	5	10	3636.3138	4549.1711	0.0711	0.0287
3	3	10	5	5	10	6106.0489	4076.2713	0.0745	0.0399
3	3	20	5	5	10	2437.9512	2242.0911	0.1014	0.0085
3	3	5	5	5	20	6265.6872	6059.5287	0.0388	0.0097
3	3	10	5	5	20	7522.1978	7795.0577	0.0373	0.0028
3	3	20	5	5	20	4214.6130	5269.1885	0.0295	0.0053
3	3	5	5	5	30	3956.8955	3710.3527	0.0513	0.0086
3	3	10	5	5	30	4083.2797	3841.2332	0.0763	0.0303
3	3	20	5	5	30	2822.7018	2875.1272	0.0566	0.0292
3	3	5	5	5	40	6906.748	9012.3312	0.0884	0.0391
3	3	10	5	5	40	4870.0912	5275.4700	0.0484	0.0050
3	3	20	5	5	40	6093.7314	6934.9149	0.2715	0.0970
3	3	5	5	5	50	9831.8046	8903.4344	0.0432	0.0059
3	3	10	5	5	50	5076.8150	4965.6695	0.0480	0.0275
3	3	20	5	5	50	7313.6695	5727.4593	0.1195	0.0193
3	3	5	5	5	60	55470.641	7144.9387	0.1076	0.0114
3	3	10	5	5	60	11215.7927	12026.7402	0.1389	0.0704
3	3	20	5	5	60	13201.5013	12719.1829	0.2692	0.0156
3	3	5	5	5	70	2128.2096	2340.6143	0.0410	0.0835
3	3	10	5	5	70	3316.8333	3011.2852	0.0623	0.0719
3	3	20	5	5	70	5455.9473	8776.2548	0.1008	0.0036