

## Article

# Accelerated Exploration for Long-Term Urban Water Infrastructure Planning through Machine Learning

Junyu Zhang <sup>1,2</sup>, Dafang Fu <sup>1,2,\*</sup>, Christian Urich <sup>1,3</sup> and Rajendra Prasad Singh <sup>1,2</sup>

<sup>1</sup> Joint Research Centre for Water Sensitive Cities, Southeast University-Monash University Joint Graduate School (Suzhou), Southeast University, Suzhou 215123, China; junyu.zhang@monash.edu (J.Z.); christian.urich@monash.edu (C.U.); rajupsc@seu.edu.cn (R.P.S.)

<sup>2</sup> Department of Civil Engineering, Southeast University, #2Sipailou, Nanjing 210096, China

<sup>3</sup> Department of Civil Engineering, Monash University, Clayton, VIC 3800, Australia

\* Correspondence: fdf@seu.edu.cn; Tel.: +86-185-5182-4285

Received: 15 October 2018; Accepted: 1 December 2018; Published: 5 December 2018



**Abstract:** In this study, the neural network method (Multi-Layer Perceptron, MLP) was integrated with an explorative model, to study the feasibility of using machine learning to reduce the exploration time but providing the same support in long-term water system adaptation planning. The specific network structure and training pattern were determined through a comprehensive statistical trial-and-error (considering the distribution of errors). The network was applied to the case study in Scotchman's Creek, Melbourne. The network was trained with the first 10% of the exploration data, validated with the following 5% and tested on the rest. The overall root-mean-square-error between the entire observed data and the predicted data is 10.5722, slightly higher than the validation result (9.7961), suggesting that the proposed trial-and-error method is reliable. The designed MLP showed good performance dealing with spatial randomness from decentralized strategies. The adoption of MLP-supported planning may overestimate the performance of candidate urban water systems. By adopting the safety coefficient, a multiplier or exponent calculated by observed data and predicted data in the validation process, the overestimation problem can be controlled in an acceptable range and have few impacts on final decision making.

**Keywords:** urban planning; water infrastructure; adaptation planning; artificial neural network; multi-layer perception

## 1. Introduction

Long-term strategic planning on urban infrastructures is often obsessed with future uncertainties such as the state of the world (e.g., economic situation, climate) or state of the city (e.g., population growth). These uncertainties are not statistical in nature which makes them hard to predict. One of the most convincing examples is the “Shrinking City” event in Dresden since 1990, where 7 predictions have been made during 15 years to predict the population growth and guide the city planning but none of them turned out to be right [1,2].

To deal with this issue, computational tools have been developed to look into more future scenarios and offer more reliable plans, such as Adaptation tipping points [3], Robust decision making [4], Info-gap [5]. The adaptation tipping points offered shifting between different strategies and plans but no guarantee of success adaptation due to lack of system performance evaluation. The robust decision-making and info-gap both aim to explore as much future as possible and evaluate the robustness of candidate plans by trade-off on the target.

As an improvement exploring planning tools have been developed to model the performance of different infrastructure plans under different scenarios, such as Adaptive policy making [6],

Adaptation pathways [7] and Dynamic adaptive policy making [8]. The adaptation pathways are able to simulate the dynamic of different infrastructure and the adaptation among them under relatively small range of future scenarios. Meanwhile, the adaptive policy making looks into wide range of future scenarios without lack of infrastructure adaptation. As the improvement of them the dynamic adaptive policy making tries to consider both but could only work out plans for independent strategies.

The limitation of the current tools is they are not able to evaluate the adaptation of a real-world combined system (centralized + decentralized) as such simulation is excessively time-consuming. More precisely, one of the major challenges on reducing the time consumption in such exploration planning tools is the robustness problem. The more detailed designs to be modelled (especially spatial distributed decentralized systems) and the more scenarios to be considered, the more time it will take, the more robust the plan can be.

Unfortunately, there are only few methods or tools that could reduce the exploration time while maintain the exploration range. This problem is being addressed in this paper by integrating the neural network method (multi-layer perceptron) with an explorative model that simulates possible urban infrastructure adaptation, to study the feasibility of using machine learning to reduce the computational time in such exploration.

In recent years, Artificial Neural Networks (ANNs), as a data-drive, self-adaptive and non-linear forecasting tool was applied in various fields such as natural resource management [9–11], pattern recognition [12,13], medical diagnosis [14] and decision making [15,16]. As a matter of factor, the methods and its derivative tool are often used in short-term decision makings or predictions (event scale) rather than long-term planning (strategy scale). To cope with the exploration model, the machine learning algorithm was designed and trained to predict urban water infrastructure performance for individual events while the decision on planning was made based on microscopic strategy performance distribution.

In this paper, the above accelerated explorative long-term planning method was proposed and tested. The following works have been conducted: (1) a comprehensive statistical trial-and-error analysis method is proposed and tested to avoid local optimization of network structure. (2) a neural network was integrated in the explorative adaptation planning to significantly reduce the simulation time, performance was tested and analyzed; (3) a correction method was proposed and tested to minimize the overestimation problem of the designed exploration framework.

## 2. Methods

### 2.1. Site Description and the Exploration

The case was carried out in Scotchman's Creek catchment, locates at the southeast of Melbourne CBD. The catchment is mostly located within Monash City council but a part of the catchment (6%) is situated within Whitehorse City council. It has an area of approximately 10.36 km<sup>2</sup> and a population of approximately 25,000 residents.

The council started to introduce rainwater tanks to households since 2005 to deal with the unpredictable rainfall events (e.g., reduce peak flow during highly intensive rainfall event, store rain water during drought season). Although the council tried to set up a progressive goal of rainwater tank uptake rate in the area, there were several obstacles in making such a plan: (1) The spatial distribution of rainwater tanks will largely influence the flood resistance in the catchment resulting from them. Thus, the promoting of higher rainwater tank uptake rate cannot be easily determined compared to upsizing pipe systems; (2) The population growth in the area could infect the construction of houses and buildings which increases the impervious surfaces in the catchment as well as the opportunity for uptake rainwater tanks; (3) The flood-resistance robustness of the combined drainage system (under different rainwater tank uptake ratio and pipe system capacity) was unclear.

Thus, a long-term (2015–2035) evolution of the urban development, climate change and water infrastructure adaptation were simulated by DAnCE4Water (Dynamic Adaptation for enabling City

Evolution for Water) [17,18] to set up a robust plan of progressive goals for both rainwater take uptake ratio and drainage pipe system upsizing. With the initial city scenario established based on the real-world catchment in 2015, DANCE4Water ran in a 5-year interval to simulate the transformation of the city and assess the urban water system performance with different drainage infrastructure updates under all possible development scenarios.

The development scenario consists of two parameters: the population growth rate (PGR) and the climate change factor (CCF). The 5-year population growth rate is ranged in [0.03,0.06] which calculated based on the maximum annual growth rate (0.012 per year) in the area according to the 1990–2015 census data from the Australian Bureau of Statistics. DANCE4Water would replace old buildings and construct new ones according to the increased population through its urban development module (UDM) [17,18]. The 5-year climate change factor is a coefficient used to magnify the 5-year designed storm. Initialized to 1.00, CCF is assumed to change every 5 years within three rates: 0.95X, 1.00X or 1.05X.

Three drainage update options were tested in this paper: (1) business as usual, (2) uptake rainwater harvesting tanks and (3) upsize drainage pipes. “Business as usual (BAU)” maintained the existing infrastructures from the previous step. The more BAU was taken, the less contribution would be done in reducing flooded junctions. “Uptake rainwater harvesting tank (RWHT)” increased the current probability of households installing rainwater harvesting tanks by 5%. The more RWHT was taken, the more decentralized systems would be built to reduce the runoff and peak flow. “Upsize drainage system (PIPE)” upgrades the drainage network, which was divided into 4 groups according to their diameters. Each upgrade enlarged one group of pipes, from the large one to the small one. The more PIPE was taken, the higher capacity of the drainage network would be.

The exploration randomly selected a PGR, a CCF and a drainage infrastructure update within the available range and applied to the base city scenario. The UDM would then generate a future scenario of the city while the performance of the combined system (the number of flooded junctions in the catchment area along the drainage network) would be evaluated by SWMM. The result city scenario was saved as the base city scenario for the next 5-year decision (see Figure 1).

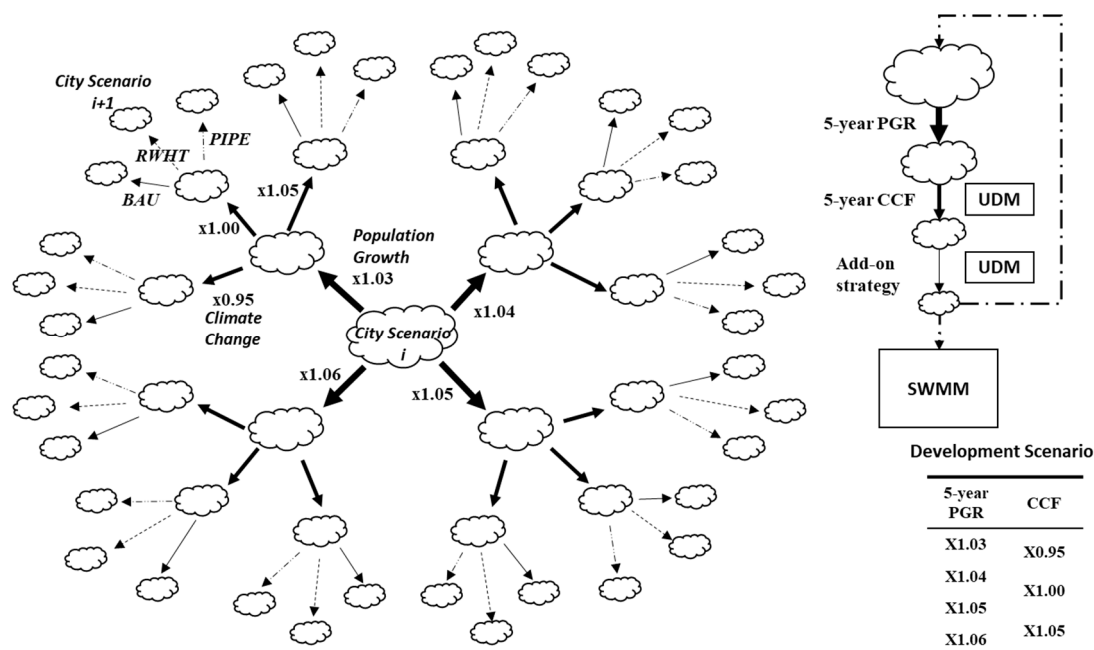


Figure 1. Designed exploration of the Scotchman's Creek catchment area.

The result scenarios were classified by the drainage infrastructure status (e.g., how many steps of BAU, RWHT and PIPE were adopted respectively). The corresponding distribution of system

performance (flooded junctions) for each status was calculated. As only one strategy was taken in each decision step, the status contains the year information as well. If the number of flooded junctions of a status was below the target (110 in 2020, 100 in 2025, 90 in 2030 and 80 in 2035, which is 100%, 91%, 82%, 73% of the flooded junctions in 2015) in over 95% of the cases, the status would be considered “robust.” The “robust” statuses were connected in a time line to form a drainage infrastructure implementation pathway as the long-term plan in this case study.

To compare the proposed acceleration exploration method, the plan was first explored through the above traditional exploration. The 20-year planning took 2.93 million simulations including 1.73 million explorations with uniformed input values and 1.2 million with random input values for the last two decision steps. The uniformed input values were listed in Figure 1, with 36 scenarios in 2020 (4 PGRs \* 3 CCFs \* 3 add-on strategies),  $36^2$  in 2025,  $36^3$  in 2030 and  $36^4$  in 2035. The random explorations selected result scenarios in 2025 and 2030, PGR and CCF within range of [1.03,1.06] and [0.95,1.05]. The whole exploration took 1 year and 4 months with 32 instances in the DAnCE4Water cloud server while the result was saved in a SQLite database containing the input values and output values for every simulation.

## 2.2. The Accelerated Exploration and ANN Design

The proposed accelerated exploration started with a normal exploration and paused when a certain amount of simulation had been finished. These simulations would be used as the training set to train an ANN while the exploration continued. The exploration then stopped when another certain amount simulation had been finished. These extra simulations would be used for validation. The ANN would be trained with different structures and settings and tested on the validation simulations. The errors of the validation would be used to choose the best structure and setting, and the ANN would do the rest of exploration by predicting with the scheduled PGR, CCF and add-on strategies (as the normal exploration) but skipping the UDM and SWMM process.

The results in the reference exploration (the scenarios as well as the evaluated system performance) were classified into three sets: the training set (size: 0.1%, 1% or 10%), the validation set (size: 10%) and the test set (size: the remaining data).

The training set was used to train the network (e.g., weights) while the validation set was for adjusting the structure of the network (e.g., number of nodes) [4]. The test set was used to assess the performance of a trained and validated network. In most literature [14,19–25], as the network structure are usually pre-defined or tested by trial-and-error, the validation sets are usually disused or replaced by the test sets. Under such substitution, the performance of the network is only meaningful for certain sets (the ‘test sets’), which have been optimized during the training, rather than for the untrained data which we expect more precise predictions.

### 2.2.1. Type of ANN

There are several groups of networks such as Feedforward Networks (e.g., Multi-layer Perceptron [26], the Probabilistic Neural Network [27], the Dynamic Neural Network [28]), Recurrent Networks (e.g., Elman Network [29], Autoregressive Networks [30]), Polynomial Networks (e.g., Ridge Polynomial Networks [31], Function Link Network [32]), Modular Networks, Support Vector Machine and so forth. [33].

Among these extensive types of ANNs and their derivations, The multi-layer perceptron (MLP), a feedforward multilayer network with non-linear node functions, is the most commonly encountered one [33,34]. Practically, MLP shows successful generalization capability, effectiveness and efficiency in forecasting time series [10,11,19,23], as well as great compatibility coping with different optimization methods or existing models [19,35]. Although MLP is usually the better choice or at least the same performance with respect to other proposal networks [33], there remain certain delimitations that have a remarkable impact on the training accuracy and efficiency. Such aspects include the structure of the network, the activation function of nodes, the existence of bias units, the quality and quantity of

training and validation datasets, the choice of training algorithm and parameters and so forth. In this paper, the MLP network will be adopted while the design process of these aspects will be investigated and adapted to the case study. The network will be established using PyBrain [36], a modular Machine Learning Library for Python.

### 2.2.2. The Structure of MLP Network

The MLP usually consists of nodes(units) arranged in three types of layer: the input layer, the hidden layer(s) and the output layer. As Figure 2 shows, each node (unit) has its own output value  $y$  and is connected by real-valued weights  $w$  to all (and only) the nodes of the subsequent layer. For the  $i$ th node in the  $l$ th layer  $n_{il}$ , let  $S_{il}$  be the set of nodes that connect to  $n_{il}$ ,  $f(x)$  be the activation function of  $n_{il}$ , the output value is calculated using Formula (1):

$$y_{n_i^l} = f\left(\sum_{n_j^m \in S_{il}} w_{ji}^{ml} y_{n_j^m}\right) \quad (1)$$

where  $y_{n_i^l}$  is the output value the  $i$ th node in the  $l$ th layer;  $w_{ji}^{ml}$  is the weight of the connection between this node and the  $j$ th node in the  $m$ th layer;  $y_{n_j^m}$  is the output value of the  $j$ th node in the  $m$ th layer;  $f(x)$  be the activation function of this node.

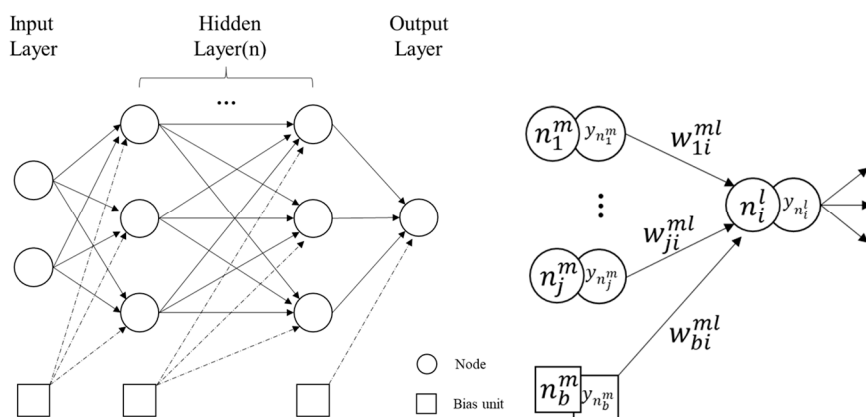


Figure 2. Structure and value propagation of MLP.

The input layer receives the input data while the output of output layer refers to the predicted results. Thus, both only requires only 1 layer to fulfill the task. The number of nodes in these layers are determined according to the number of input variables and target variables [37]. In some cases, the input and output variables are linearly normalized to (0,1) or (−1,1), to avoid computational problems or to meet algorithm requirement [24,38,39]. In this study, such methods were not applied because: (1) with the exploration continues, the input variables will always exceed the range of the existing records while the output variable also has the chance. (2) the weights may undo the scaling.

The number of hidden layers and its nodes has a significant impact on MLP training [37,40]. Simple networks maybe less accurate in learning the problem while complex networks may take excessively long training time. one hidden layer is usually sufficient in most cases [14,19–25,33,41–43] while sometimes multiple hidden layers shows better learning on certain problems [35].

The number of nodes in hidden layer is usually determined through trial-and-error method [19,23,43]. The range of attempts is usually within 1 to 20 [14,19–25], or 3 times the number of input variables [43]. The best number of nodes was the one having the smallest mean-square error (MSE) and root-mean-square error (RMSE) and the highest correlation coefficient ( $r$ ) for the validation data set. [11]

In this paper, the designed MLP consists 1 input layer, 1 hidden layer and 1 output layer. There will be 5 nodes in the input layer representing climate change factor, population, the number of decision take for BAU, RWHT and PIPE within the 20 years and 1 node in the output layer referring to the



flooded junctions. No variables will be normalized. The number of nodes in the hidden layer will be determined within 1 to 20 through trial-and-error method.

### 2.2.3. The Activation Functions

The role of activation function (AF) in MLP is to non-linearize the linear combination of weights and node values passing through from the previous layer. Practically, there are three types of AFs: (1) the analytic AFs, which are classic functions such as Gaussian, Sigmoid and Tanh; (2) the fuzzy AFs, which has faster convergence in training; and (3) the adaptive AFs, which improves the nonlinear response of the network [40]. Although the fuzzy AFs perform better on specific problems [44], there is little evidence on the advantage of such AFs in practice. On the other hand, the adaptive AFs also suffer from a more complex and error-prone training algorithm [40]. Thus, only classic analytic AFs are considered in this study.

For nodes in the hidden layer, most commonly used AFs are the logistic sigmoid function [34,38,41], the tanh function [35,43,45]. These two functions are similar in shape while different in output ranges (sigmoid: [0,1], tanh: [-1,1]). For the output layer, most researchers adopt linear function [11,35,41,45].

In this paper, the log-sigmoid function has been used for the hidden layer nodes while linear function has been applied in the output layer to test their performance on handling random noise.

### 2.2.4. Bias Unit

The bias unit is an extra set of nodes added to all layers but the output layer, which helps to get a better and quicker learning of the network. The output value of a bias unit is fixed value while the weights of connection from the bias unit to the subsequent nodes are still adjustable. The addition of bias unit introduces a threshold value that may influence the activation of the subsequent nodes [24,37], or, from another perspective, helps to move the AF in the subsequent nodes along the x-axis for better learning results. Thus, in most cases, bias units always contribute positively to the network.

### 2.2.5. Learning Algorithm and Parameter Setting

The traditional and most commonly used training method for MLP is the two-step error-backpropagation method [14,19,24]. Firstly, the input vector is fed into the input layer, propagating forward through hidden layer(s) to the output layer. Then, the error is calculated in the gradient descent and propagated backward from the output layer through the hidden layer(s) to the input layer, which modifies the weights for every connection between nodes. The training repeats until the network's overall error are less than a predefined learning rate, or until the number of maximum epochs is reached. Learning rate is a damping factor applied to weights correction during training [40], indicating the amount that the weights are updated. Epoch is a measure of the number of times all of the training vectors are used once to update the weights. Obviously, when dealing with huge datasets, it is super time consuming if all the weights are recomputed for each training vector. Thus, there is also a batch-learning term for the backpropagating method, which feeds multiple training samples in one forward/backward pass. The number of samples in one pass is called batch size while such one forward/backward process is count as one iteration.

As the original backpropagation method is likely to be slow [41], improved strategies such as Second-order On-Line training methods have been developed. Although these second-order training algorithms are likely to converge significantly faster than first-ordered backpropagation [37], they require more complex data preprocessing as well as more storage and computational costs. Luckily, there are also several improved first-order backpropagation methods. The most commonly used is the Backpropagation with Momentum [22,24], which significantly speed up the training process. The momentum is an inertial factor applied to the weights during the back propagate process, which aims to maintain the direction of weight changing [40]. The addition of momentum accelerates convergence where the learning quality is good while precisely reduces the number of oscillations where bad [37].

The settings of training parameters are more likely to be empirical and case-dependent. In most cases, the start/fixed learning rate will be in the range of [0.01,0.3] [21,22,25,34] while the end learning rate within [0.00013,0.001] [19,21]. The number of epochs usually depends on the training data size and the computational capacity, ranging from 200 to 15,000 [19,21,22,24,34,35,42]. Momentum is typically set to 0.9 [22], although the optimal value might be task-specific [21,24,34].

The designed network structure and learning parameters are shown in Table 1. All combinations of structure and learning parameters were tested with the first 0.1% of data and validated with the following 0.05% data. After the best structure was determined, the network was again tested with different size of training set size to find the best application pattern. The validation set size is half of the training set. The best performing structure and application pattern were applied to the case study to study the feasibility of ANN in supporting long-term planning.

**Table 1.** Designed Neural Network Parameters.

| Type | Structure |       |      | Activation Function | Bias Units | Learning Settings                        |                            |
|------|-----------|-------|------|---------------------|------------|--|----------------------------|
|      | Name      | Layer | Node |                     |            |  |                            |
| MLP  | input     | 1     | 5    | -                   | True       | training size <sup>1</sup><br>batch size | 0.1%, 1%, 10%<br>1         |
|      | hidden    | 1     | 1–20 | sigmoid             | True       | learning rate<br>learning rate decay     | 0.01, 0.1, 0.3<br>1.0      |
|      | output    | 1     | 1    | linear              | False      | momentum<br>epoch                        | 0.1–0.9<br>500, 1000, 5000 |

<sup>1</sup> Training size is the percentage of total data used as the training set, tested after the ANN structure being determined.

### 2.3. Trial and Error

The performance of learning results was assessed by the root-mean-square error (RMSE), which is a commonly used index in machine learning [14,20,21,34]. The lower RMSE it is, the better prediction the module makes [19].

RMSE is defined as the absolute value of the estimated error between the predicted result and the observed result, calculated by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (O_i - P_i)^2}{n}} \quad (2)$$

where  $O_i$  is the observed result;  $P_i$  is the predicted result.

As the unit of RMSE is case-dependent, the correlation coefficient ( $r$ ) [14,20,21,34] was adopted to compare the training performance with other studies.

$$r = \frac{\sum_{i=1}^n (P_i - \bar{P})(O_i - \bar{O})}{\sqrt{\sum_{i=1}^n (P_i - \bar{P})^2 \sum_{i=1}^n (O_i - \bar{O})^2}} \quad (3)$$

where  $O_i$  is the observed result;  $P_i$  is the predicted result;  $\bar{O}$  is the mean value of the observed result;  $\bar{P}$  is the mean value of the predicted result.

Practically, as the decision in long-term infrastructure implementation planning is not scenario-based but strategy-based, the distribution of predict results for each strategy combination should be more convincing than RMSE. Thus, the prediction distribution of outputs was also adopted in this study as the other performance indicator

## 3. Results and Discussion

### 3.1. ANN Structure and Training Parameters

As mentioned in the previous section, all combinations of structure (number of hidden nodes) and learning parameters (learning rate, momentum and number of epochs) were tested with the first

0.1% of all data (training size = 0.1) and validated with the following 10% of data. For each parameter, the distributions of RMSE for each candidate value under all possible combinations are shown in Figure 3.

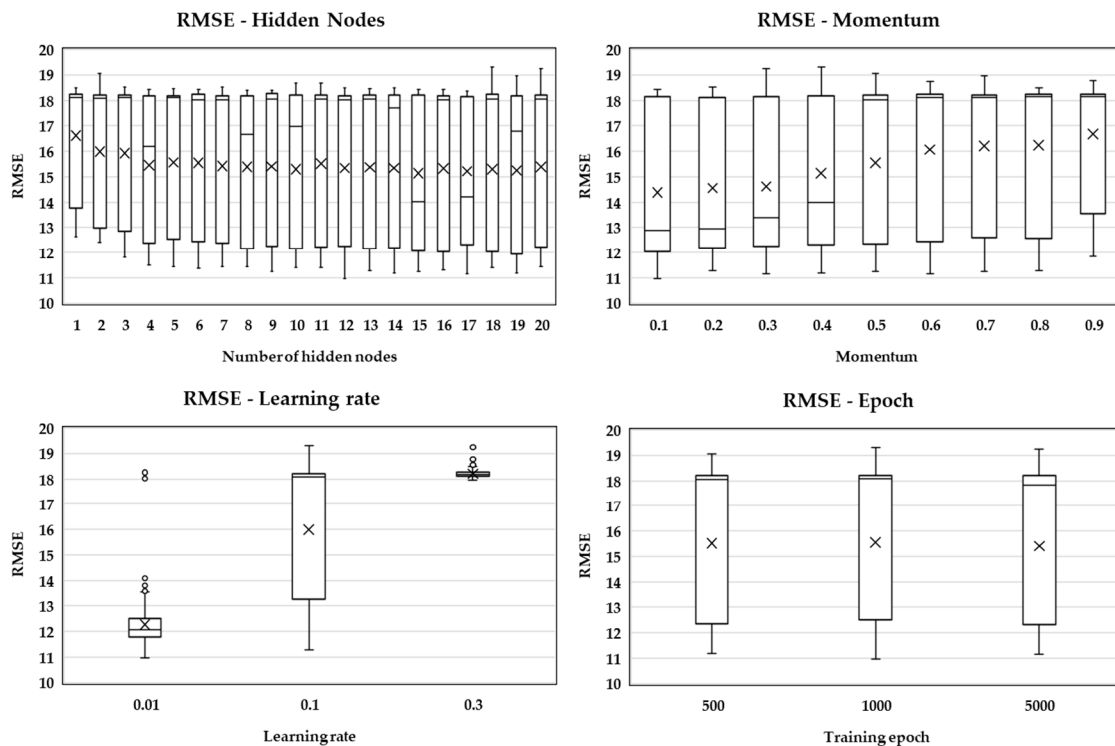


Figure 3. RMSE Distributions under different manipulated variables.

By adopting ANN(MLP) in urban water infrastructure performance prediction, the RMSE of such method ranges from 10.97–19.33 nodes with the observed flooded junctions ranging from 20 to 146. For the number of hidden nodes, setting 1 node caused the highest average RMSE (16.62) which may due to the strongest linearity of the network. With the number of hidden nodes rises to 4 nodes, the average RMSE drops gradually to 15.46 where the non-linearity starts to develop effect. From 4 nodes to 20 nodes, the average RMSE keeps stable within (15.13,15.56). Although there is no significant difference in the average RMSE with the number of hidden nodes changing, the distributions of RMSE still have dramatic and irregular variations. These distributions are characterized by the minimum, maximum, Q1, Q3 and mid-values, which indicates 100%, 75%, 50%, 25%, 0% chance of getting a higher RMSE than the given value, respectively. Thus, the lower these values are, the better performance of the network we will get.

As shown in Table 2, the MLP network with 15 nodes was always in the top 5 well-performed structure and has significant advantages in low mid-value compared to others. The 17 nodes network is slightly better than the 15 nodes one on minimum, Q3 and maximum as well as slightly poor on Q1 and mid-value. Thus, the network of 15 and 17 hidden nodes are selected as the candidate structure for the following studies.

Following the same process, the rest parameters are then determined: momentum = 0.1, learning rate = 0.01, epoch = 5000.

The candidate network was again tested with different size of training set size to find the best application pattern (see Table 3). The result indicates that network with 15 nodes performs better than the 17 nodes one under the select learning parameter, which is within 3 times the number of input variables [38]. Training with the first 10% data will have a significant improvement in reducing the RMSE while maintaining an acceptable time-saving capacity (reduce 80% of the time).



The best performing structure and application pattern (Table 3) were then applied to the case study. The overall RMSE for the whole observed data and the predicted data is 10.5722 and the detailed performance of MLP prediction is shown in Figure 4. The overall RMSE is slightly higher than the validation result (9.7961).

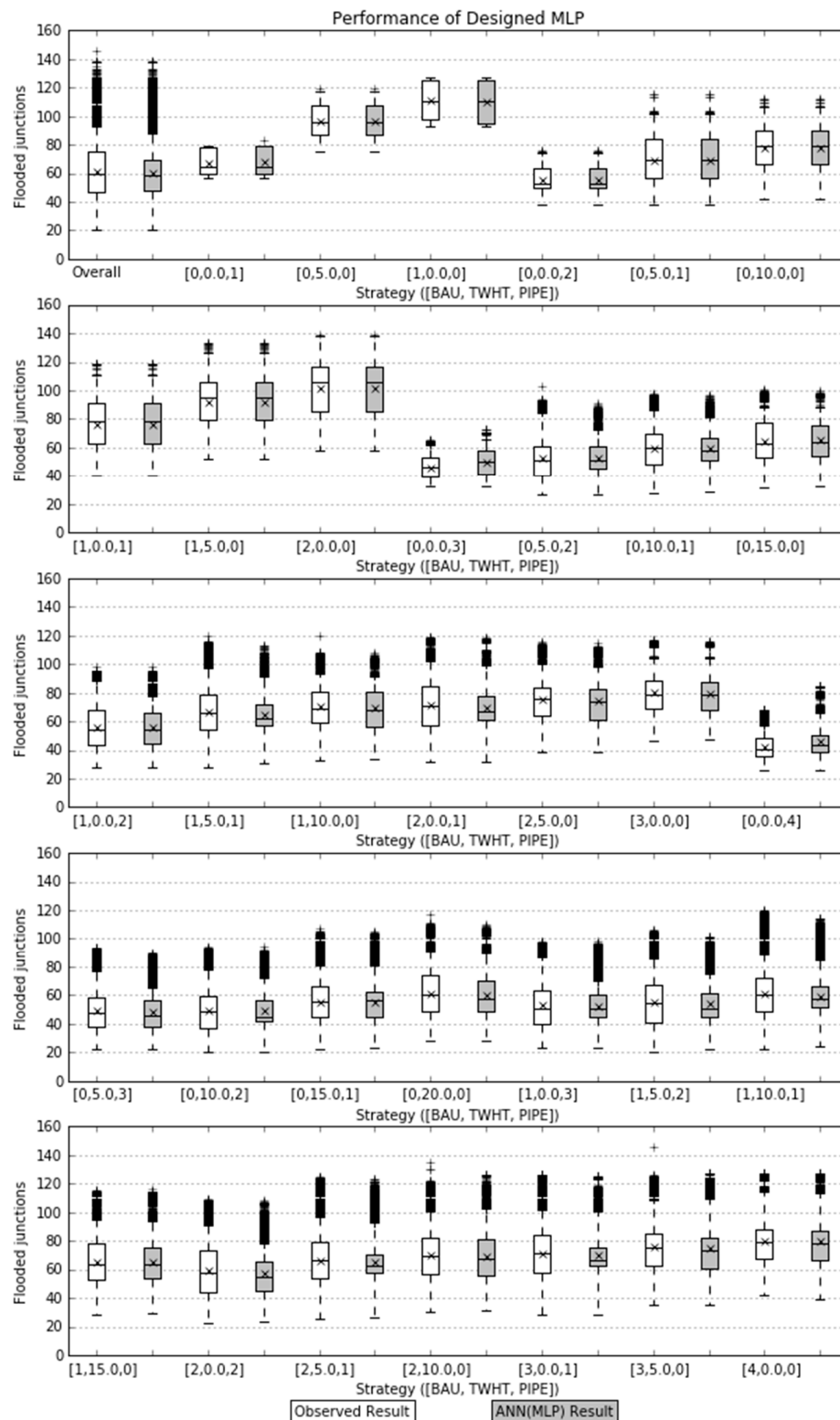


Figure 4. ANN performance for different strategy combinations (supported by Matplotlib [46]).

**Table 2.** Comparison of performance distribution for different number of hidden nodes.

|     | 1st | RMSE  | 2nd   | RMSE  | 3rd   | RMSE  | 4th   | RMSE  | 5th   | RMSE  |
|-----|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Min | 12  | 10.97 | 14/17 | 11.17 | 19    | 11.18 | 9     | 11.25 | 15    | 11.26 |
| Q1  | 19  | 11.95 | 18    | 12.04 | 16    | 12.05 | 15    | 12.08 | 13    | 12.16 |
| Mid | 15  | 14.02 | 17    | 14.20 | 8     | 16.67 | 19    | 16.79 | 10    | 16.98 |
| Q3  | 17  | 18.15 | 5     | 18.17 | 10/12 | 18.19 | 13/14 | 18.20 | 15    | 18.21 |
| Max | 17  | 18.37 | 8     | 18.39 | 9     | 18.41 | 6     | 18.42 | 15/16 | 18.43 |

**Table 3.** ANN performance under different training set sizes.

|                | Training Size | Hidden Nodes | Learning Rate | Momentum | Epoch | RMSE    |
|----------------|---------------|--------------|---------------|----------|-------|---------|
| Validation set | 0.001         | 15           | 0.01          | 0.1      | 5000  | 11.5051 |
|                | 0.01          |              |               |          |       | 11.8653 |
|                | 0.1           |              |               |          |       | 9.7961  |
|                | 0.001         | 17           |               |          |       | 12.2593 |
|                | 0.01          |              |               |          |       | 12.5760 |
|                | 0.1           |              |               |          |       | 11.9862 |
| Test set       | 0.1           | 15           | 0.01          | 0.1      | 5000  | 10.5722 |

The correlation coefficient ( $r$ ) of the test set was 0.821, which was preferable compared to  $r_s$  in the other close applications of ANN (flood discharge: 0.683–0.851 [47], open-channel junction velocity field: 0.035–0.884 [48], drought effects on surface water quality: 0.819–0.922 [49], BOD in river: 0.505–0.821 [19]).

Taking account of the tremendous amount of data in this case study, the above result suggested the proposed statistical trial-and-error method for determining network parameters is feasible and reliable on selecting the best structures.

### 3.2. Performance on Supporting Long-Term Planning

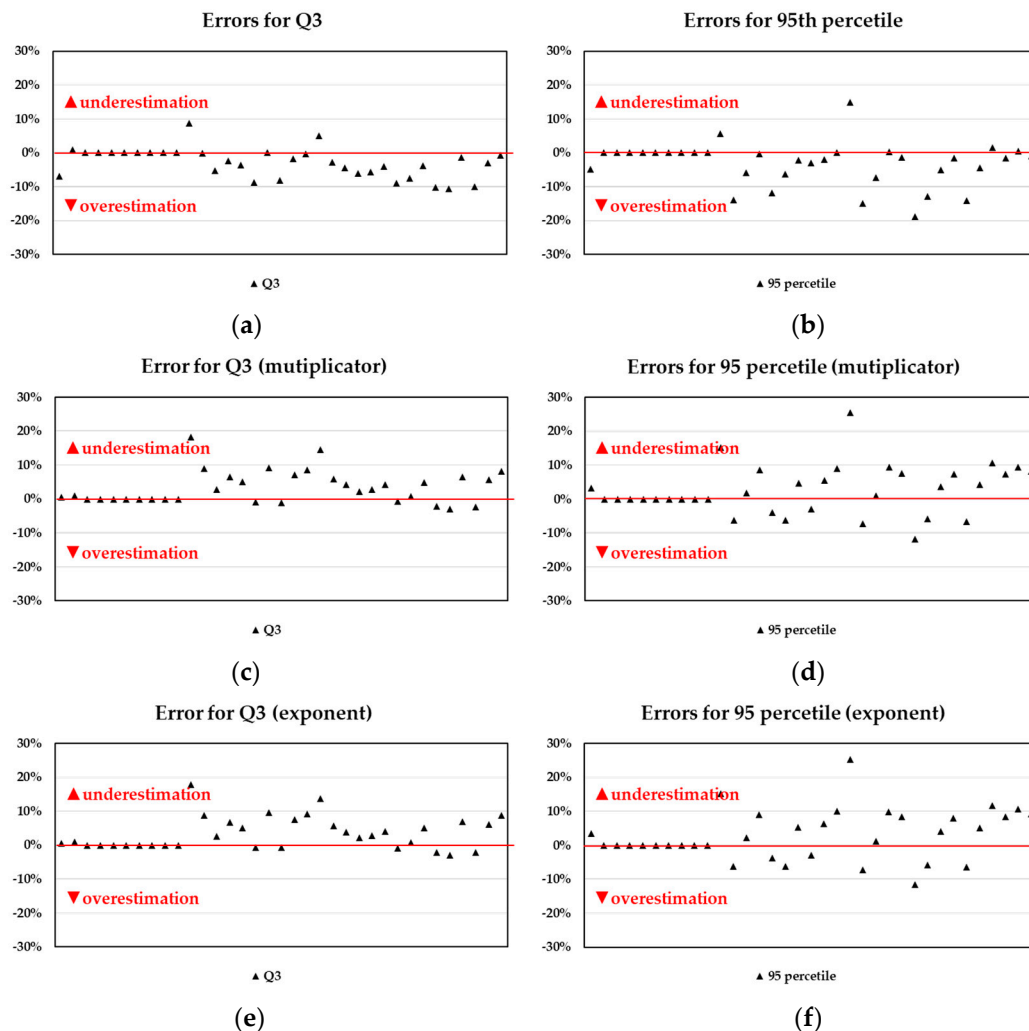
To analyze the performance variations of different implementation strategy combinations for the urban water system in the case study, boxplots are again used while the upper end of the whiskers is set to 95th percentile (Figure 4). In other word, the probability of a certain system performing better than this upper end is 95%. Thus, the accuracy on the 95th percentile and Q3 is practically more important than that of mid-value, Q1 and minimum.

For strategies containing only rainwater tanks ([0,5.0,0], [0,10.0,0], [0,15.0,0] and [0,20.0,0]), the first two combinations are all included in the training set and share the same distribution with the observed results. For the latter two strategies, the 95th percentile errors are  $-0.24\%$  and  $-1.26\%$  respectively while the Q3 errors being  $-2.28\%$  and  $-5.68\%$ . This suggests the designed MLP network is effective and has relatively good performance in predicting strategies with spatial randomness. The performance of purely decentralized systems may have stronger and more linear relation with the rainfall events and urban permeability (related to buildings/population), which makes the prediction of these purely decentralized strategies better than mix strategies.

For the same reason, the purely business as usual strategies also have good predictions: for [3,0.0,0], Q3 =  $-0.22\%$  and 95th =  $0.18\%$ ; for [4,0.0,0], Q3 =  $-0.77\%$  and 95th =  $-0.88\%$ . As no additional systems were implemented in these scenarios, the designed network performs well in generalizing the relation between water system performance and rainfall events and urban permeability.

For the overall performance, the MLP result has similar minimum, Q1 and mid-value compared to the observed result (min: 20, 20; Q1: 48.1, 47.0; mid: 58.3, 60.0). Whereas the predicted values have a narrower range (20.0–88.44) than the observed ones (20–93) despite the outliers. Such phenomena indicate that the prediction in the high-value events (poorly performed water system in practice) tend to aggregate to the Q3. This suggests that, from an overview perspective, the adoption of ANN supported planning may raise the chance of overestimating the performance of urban water systems.

To make this proposed method applicable and reliable in practice, the error distributions of the result are investigated to solve the overestimating problem. As shown in Figure 5, all errors of Q3 lie between  $(-10.56\%, 8.76\%)$  and 95th percentile between  $(-18.91\%, 14.95\%)$ . The majority of these errors are negative, indicating universal overestimations of the urban water system.



**Figure 5.** Error distribution of MLP predicted result and corrected result ((a,b) observed errors for 95th percentile and Q3; (c,d) corrected errors for 95th percentile and Q3 by multiplication; (e,f) corrected errors for 95th percentile and Q3 by exponent).

As Table 4 shows, the adoption of safety coefficient could effectively raise the error from negative to positive (from overestimation to under estimation) while slightly enlarge the standard deviation of the errors.

**Table 4.** Mean  $\pm$  SD error of adopting the safety coefficient.

|                 | Observed Error       | Multiplicator       | Exponent            |
|-----------------|----------------------|---------------------|---------------------|
| Q3              | $-2.29\% \pm 4.28\%$ | $3.38\% \pm 4.73\%$ | $3.43\% \pm 4.72\%$ |
| 95th percentile | $-3.13\% \pm 6.34\%$ | $2.63\% \pm 7.15\%$ | $2.96\% \pm 7.32\%$ |

As these errors are related to the network structure and its final status, a safety coefficient, which comes from the validation process, is adopted to adjust the final output of the network. By investigating the observed data and the predicted data in the validation set, a multiplicator or exponent can be

calculated out and applied for the test set. As the 95th percentile is the dominant factor of this case study, the safety coefficient also comes from the 95th percentile of the validation (multiplier:1.0910, exponent:1.0272).

The result of correction is shown in Figure 5. There is no obvious difference between correction with multiplier and exponent. The corrected errors of Q3 lied in  $-3.05\%$  to  $18.24\%$  (multiplier) and  $-2.96\%$  to  $17.87\%$  (exponent) while that of the 95th percentile in  $-11.69\%$  to  $25.41\%$  (multiplier) and  $-11.60\%$  to  $25.36\%$  (exponent).

As shown in Table 5, the accelerated exploration identified all robust drainage infrastructure status in the reference exploration while overestimated three. The corrected accelerated exploration identified most robust drainage infrastructure status in the reference exploration while underestimated one. The underestimated one has no influence on the plan generation as there is no connectable route in the previous decision year. Thus, the correction is essential and effective to raise the robustness of the proposed accelerated exploration.

**Table 5.** Robust progressive goal for Scotchman's Creek.

|      | Reference Exploration | Accelerated Exploration | Corrected Accelerated Exploration |
|------|-----------------------|-------------------------|-----------------------------------|
| 2020 | [0,0,1] <sup>1</sup>  | [0,0,1]                 | [0,0,1]                           |
| 2025 | [0,0,2]               | [0,0,2]                 | [0,0,2]                           |
| 2030 | [0,0,3]               | [0,0,3]                 | [0,0,3]                           |
|      | [0,5,2]               | [0,5,2]                 | [0,5,2]                           |
|      | [0,10,1]              | [0,10,1]                | [0,10,1]                          |
|      | [0,15,0]              | [0,15,0]                | -                                 |
|      | [1,0,2]               | [1,0,2]                 | [1,0,2]                           |
| 2035 | -                     | [1,5,1]                 | -                                 |
|      | [0,0,4]               | [0,0,4]                 | [0,0,4]                           |
|      | [0,5,3]               | [0,5,3]                 | [0,5,3]                           |
|      | [0,10,2]              | [0,10,2]                | [0,10,2]                          |
|      | -                     | [1,5,2]                 | -                                 |
|      | -                     | [2,0,2]                 | -                                 |

<sup>1</sup> [BAU,RHWT(%),PIPE].

Notably, for 95th percentile, the majority of errors are controlled within  $\pm 10\%$ . The two outliers represent the two pure strategies of upgrading pipes, [0,0,0,3] and [0,0,0,4]. Although there are great errors on these two strategies (underestimation of water system), the origin system performance of them is good enough that the errors have no influence on identifying them as good strategies (not influencing decision). This error also indicates that different from purely decentralized strategies, such as purely centralized strategies which have only relations with rainfall events, do not have a preferable prediction at all.

Such a result indicates that when using the MLP to predict a black box problem, such as the urban water system in the case study, there should be at least two related input factors for each variable (the candidate infrastructure, e.g., pipe, rwht) to ensure reliable prediction.

#### 4. Conclusions

In this study, an accelerated exploration planning method was proposed by integrating the neural network method (multi-layer perceptron) with an explorative model (DAnCE4Water), to significantly reduce the simulation time of generating a robust long-term water system adaptation plan. The proposed method was applied to a case study in Scotchman's Creek, Melbourne, Australia. Results showed the proposed method can cut down 80% of the simulation time while offering the same plan.

Instead of modifying the network parameters, the network structure and settings in this paper were determined through a comprehensive statistical trial-and-error analysis (evaluating for all possible

parameter combination). With 10% of the training data, the validation error (10% data) was 9.7961, the overall prediction RMSE was 10.5722 (80% data) and the correlation coefficient ( $r$ ) was 0.821. This suggests that the ANN could have stable and reliable with good designed network and low proportion of training data. It also emphasizes the necessity of network design which did take time in the trial-and-error analysis but having promising return in total time saving and accuracy.

The ANN showed diverse capacity on predicting the performance of different type of flood-resisting strategies. The estimation of purely decentralized strategies (scenarios with RWHT only) and purely BAU strategies is far more accurate than that of mixed strategies. Meanwhile, the purely centralized strategies (scenarios with PIPE only) had the worst prediction. Considering the input variables related to the strategy, it is obvious that the performance estimation would be more accurate if more flood-related input variables are related to it (two for RWHT and BAU while one for PIPE). Thus, more flood-related input variables should be considered (for each strategy) in future studies.

The proposed exploration method raised the chance of overestimating the performance of urban drainage systems ( $-3.13\% \pm 6.34\%$  flooded junctions than observed). By adopting the safety coefficient, a multiplier or exponent calculated by observed data and predicted data in the validation process, the overestimation problem was controlled in an acceptable range and have very limited impacts on final decision making ( $2.63\% \pm 7.15\%$  flooded junctions than observed). Such correction is effective in practice as the real-world goal for planning is either above or below a certain target. Instead of reducing the error which is a tough task, the correction shifts the error along one direction (to more underestimate side) to ensure the reliability of the given plan. As the error came from the method, the safety coefficient calculated by the validation data could be reasonable to some extent.

Although the proposed accelerated exploration method was proved to be efficiency in time saving (saved 80% of exploration) and effective (offered similar decisions after correction), there are still several aspects requires further studies. (1) The training set used in this study followed a “real-world exploration” time sequence, which means there were much more simulations in the later decision steps than in the earlier steps. Such setting may have influence on the network performance. Further studies have to be conducted on the composition of the training set to ensure efficient and effective training; (2) Further investigation in the cause of the universal overestimation have to be conducted to optimize the algorithm or training pattern. (3) More case studies should be carried out to further validate and improve the proposed accelerated exploration method.

**Author Contributions:** Conceptualization, J.Z., D.F., C.U. and R.P.S.; Methodology, J.Z. and C.U.; Software, J.Z.; Validation, J.Z.; Formal Analysis, J.Z.; Data Curation, J.Z. and C.U.; Writing-Original Draft Preparation, J.Z.; Writing-Review & Editing, R.P.S.; Visualization, J.Z.; Supervision, D.F.; Project Administration, D.F.; Funding Acquisition, D.F.

**Funding:** The research was co-funded by National Key R&D Program of China (Grant No. 2018YFC0809900) and the Priority Academic Program Development of the Jiangsu Higher Education Institution.

**Acknowledgments:** Thanks to Professor Dafang Fu and Christian Urlich for the creative leadership and for motivation to finish the research. Thanks to Chenli Wu, who continuously supports us to finish the work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Moss, T. ‘Cold spots’ of Urban Infrastructure: ‘Shrinking’ Processes in Eastern Germany and the Modern Infrastructural Ideal. *Int. J. Urban Reg. Res.* **2008**, *32*, 436–451. [[CrossRef](#)]
2. Wiechmann, T.; Pallagst, K.M. Urban shrinkage in Germany and the USA: A Comparison of Transformation Patterns and Local Strategies. *Int. J. Urban Reg. Res.* **2012**, *36*, 261–280. [[CrossRef](#)] [[PubMed](#)]
3. Kwadijk, J.C.J.; Haasnoot, M.; Mulder, J.P.M.; Hoogvliet, M.M.C.; Jeuken, A.B.M.; van der Krogt, R.A.A.; van Oostrom, N.G.C.; Schelfhout, H.A.; van Velzen, E.H.; van Waveren, H.; et al. Using adaptation tipping points to prepare for climate change and sea level rise: A case study in the Netherlands. *Wiley Interdiscip. Rev. Clim. Chang.* **2010**, *1*, 729–740. [[CrossRef](#)]

4. Lempert, R.J.; Groves, D.G.; Popper, S.W.; Bankes, S.C. A General, Analytic Method for Generating Robust Strategies and Narrative Scenarios. *Manag. Sci.* **2006**, *52*, 514–528. [[CrossRef](#)]
5. Ben-Haim, Y. *Info-Gap Decision Theory: Decisions under Severe Uncertainty*; Academic Press: Cambridge, MA, USA, 2006.
6. Walker, W.E.; Rahman, S.A.; Cave, J. Adaptive policies, policy analysis, and policy-making. *Eur. J. Oper. Res.* **2001**, *128*, 282–289. [[CrossRef](#)]
7. Haasnoot, M.; Middelkoop, H.; Offermans, A.; Beek, E.V.; Deursen, W.P.A.V. Exploring pathways for sustainable water management in river deltas in a changing environment. *Clim. Chang.* **2012**, *115*, 795–819. [[CrossRef](#)]
8. Haasnoot, M.; Kwakkel, J.H.; Walker, W.E.; Ter Maat, J. Dynamic adaptive policy pathways: A method for crafting robust decisions for a deeply uncertain world. *Glob. Environ. Chang.* **2013**, *23*, 485–498. [[CrossRef](#)]
9. Mustafa, M.R.; Rezaei, R.B.; Saiedi, S.; Isa, M.H. River suspended sediment prediction using various multilayer perceptron neural network training algorithms—A case study in Malaysia. *Water Resour. Manag.* **2012**, *26*, 1879–1897. [[CrossRef](#)]
10. Singh, S.; Reddy, C.S.; Pasha, S.V.; Dutta, K.; Saranya, K.R.L.; Satish, K.V. Modeling the spatial dynamics of deforestation and fragmentation using Multi-Layer Perceptron neural network and landscape fragmentation tool. *Ecol. Eng.* **2017**, *99*, 543–551. [[CrossRef](#)]
11. Ruben, G.B.; Zhang, K.; Bao, H.; Ma, X. Application and Sensitivity Analysis of Artificial Neural Network for Prediction of Chemical Oxygen Demand. *Water Resour. Manag.* **2017**, *32*, 273–283. [[CrossRef](#)]
12. Ripley, B.D. *Pattern recognition and neural networks*; Cambridge University Press: Cambridge, UK, 2009; pp. 233–234.
13. Kumar, R.; Singh, B.; Shahani, D.T. Recognition of single-stage and multiple power quality events using Hilbert–Huang transform and probabilistic neural network. *Electr. Power Compon. Syst.* **2015**, *43*, 607–619. [[CrossRef](#)]
14. Sun, W.Z.; Jiang, M.Y.; Ren, L.; Dang, J.; You, T.; Yin, F.F. Respiratory signal prediction based on adaptive boosting and multi-layer perceptron neural network. *Phys. Med. Biol.* **2017**, *62*, 6822–6835. [[CrossRef](#)] [[PubMed](#)]
15. Ivey, R.; Bullock, D.; Grossberg, S. A neuromorphic model of spatial lookahead planning. *Neural Netw.* **2011**, *24*, 257–266. [[CrossRef](#)] [[PubMed](#)]
16. Erdem, U.M.; Hasselmo, M. A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *Eur. J. Neurosci.* **2012**, *35*, 916–931. [[CrossRef](#)] [[PubMed](#)]
17. Urich, C.; Rauch, W. Exploring critical pathways for urban water management to identify robust strategies under deep uncertainties. *Water Res.* **2014**, *66*, 374–389. [[CrossRef](#)]
18. Urich, C.; Sitzenfrie, R.; Kleidorfer, M.; Bach, P.M.; McCarthy, D.T.; Deletic, A.; Rauch, W. Evolution of urban drainage networks in DAnCE4Water. In Proceedings of the 9th International Conference on Urban Drainage Modelling, Belgrade, Serbia, 4–6 September 2012.
19. Raheli, B.; Aalami, M.T.; El-Shafie, A.; Ghorbani, M.A.; Deo, R.C. Uncertainty assessment of the multilayer perceptron (MLP) neural network model with implementation of the novel hybrid MLP-FFA method for prediction of biochemical oxygen demand and dissolved oxygen: A case study of Langat River. *Environ. Earth Sci.* **2017**, *76*, 503. [[CrossRef](#)]
20. Fan, X.; Wang, L.; Li, S. Predicting chaotic coal prices using a multi-layer perceptron network model. *Resour. Policy* **2016**, *50*, 86–92. [[CrossRef](#)]
21. Mirici, M.E. Land Use/Cover Change Modelling in a Mediterranean Rural Landscape Using Multi-Layer Perceptron and Markov Chain (Mlp-Mc). *Appl. Ecol. Environ. Res.* **2018**, *16*, 467–486. [[CrossRef](#)]
22. Saeidi, S.; Mohammadzadeh, M.; Salmanmahiny, A.; Mirkarimi, S.H. Performance evaluation of multiple methods for landscape aesthetic suitability mapping: A comparative study between Multi-Criteria Evaluation, Logistic Regression and Multi-Layer Perceptron neural network. *Land Use Policy* **2017**, *67*, 1–12. [[CrossRef](#)]
23. Feng, X.; Li, Q.; Zhu, Y.; Hou, J.; Jin, L.; Wang, J. Artificial neural networks forecasting of PM 2.5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* **2015**, *107*, 118–128. [[CrossRef](#)]



24. Lopez, M.E.; Rene, E.R.; Boger, Z.; Veiga, M.C.; Kennes, C. Modelling the removal of volatile pollutants under transient conditions in a two-stage bioreactor using artificial neural networks. *J. Hazards Mater.* **2017**, *324*, 100–109. [[CrossRef](#)] [[PubMed](#)]
25. Abderrahim, H.; Chellali, M.R.; Hamou, A. Forecasting PM10 in Algiers: Efficacy of multilayer perceptron networks. *Environ. Sci. Pollut. Res. Int.* **2016**, *23*, 1634–1641. [[CrossRef](#)] [[PubMed](#)]
26. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning internal representation by back-propagation of errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
27. Enke, D.; Thawornwong, S. The use of data mining and neural networks for forecasting stock market returns. *Expert Syst. Appl.* **2005**, *29*, 927–940. [[CrossRef](#)]
28. Guresen, E.; Kayakutlu, G.; Daim, T.U. Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.* **2011**, *38*, 10389–10397. [[CrossRef](#)]
29. Lee, T.S.; Chen, I.F. Forecasting exchange rates using feedforward and recurrent neural networks. *J. Appl. Econ.* **1995**, *10*, 347–364.
30. Kodogiannis, V.; Lolis, A. Forecasting Financial Time Series using Neural Network and Fuzzy System-based Techniques. *Neural Comput. Appl.* **2002**, *11*, 90–102. [[CrossRef](#)]
31. Ghazali, R.; Hussain, A.J.; Al-Jumeily, D.; Merabti, M. *Dynamic Ridge Polynomial Neural Networks in Exchange Rates Time Series Forecasting*; Springer: Berlin, Germany, 2007.
32. Hussain, A.J.; Knowles, A.; Lisboa, P.J.G.; El-Deredy, W. Financial time series prediction using polynomial pipelined neural networks. *Expert Syst. Appl.* **2008**, *35*, 1186–1199. [[CrossRef](#)]
33. Ramos, E.G.; Martínez, F.V. A Review of Artificial Neural Networks: How Well Do They Perform in Forecasting Time Series? *Analítica Revista Análisis Estadístico* **2013**, *6*, 7–15.
34. Pham, T.D.; Yoshino, K.; Bui, D.T. Biomass estimation of *Sonneratia caseolaris* (L.) Engler at a coastal area of Hai Phong city (Vietnam) using ALOS-2 PALSAR imagery and GIS-based multi-layer perceptron neural networks. *GISci. Remote Sens.* **2016**, *54*, 329–353. [[CrossRef](#)]
35. Zadkarami, M.; Shahbazian, M.; Salahshoor, K. Pipeline leakage detection and isolation: An integrated approach of statistical and wavelet feature extraction with multi-layer perceptron neural network (MLPNN). *J. Loss Prev. Process Ind.* **2016**, *43*, 479–487. [[CrossRef](#)]
36. Schaul, T.; Bayer, J.; Wierstra, D.; Sun, Y.; Felder, M.; Sehnke, F. PyBrain. *J. Mach. Learn. Res.* **2010**, *11*, 743–746.
37. Ba, A.J.S. *Second-Order Methods for Neural Networks*; Springer: Berlin, Germany, 1997; pp. 201–203.
38. Piotrowski, A.P.; Napiorkowski, M.J.; Napiorkowski, J.J.; Osuch, M. Comparing various artificial neural network types for water temperature prediction in rivers. *J. Hydrol.* **2015**, *529*, 302–315. [[CrossRef](#)]
39. Zhang, G.; Eddy Patuwo, B.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [[CrossRef](#)]
40. Laudani, A.; Lozito, G.M.; Riganti Fulginei, F.; Salvini, A. On Training Efficiency and Computational Costs of a Feed Forward Neural Network: A Review. *Comput. Intell. Neurosci.* **2015**, *2015*, 818243. [[CrossRef](#)] [[PubMed](#)]
41. Bayram, S.; Ocal, M.E.; Laptali Oral, E.; Atis, C.D. Comparison of Multi Layer Perceptron (Mlp) and Radial Basis Function (Rbf) for Construction Cost Estimation: The Case of Turkey. *J. Civil Eng. Manag.* **2015**, *22*, 480–490. [[CrossRef](#)]
42. Pham, B.T.; Tien Bui, D.; Prakash, I.; Dholakia, M.B. Hybrid integration of Multilayer Perceptron Neural Networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS. *Catena* **2017**, *149*, 52–63. [[CrossRef](#)]
43. Talebi, N.; Nasrabadi, A.M.; Mohammad-Rezazadeh, I. Estimation of effective connectivity using multi-layer perceptron artificial neural network. *Cogn. Neurodyn.* **2018**, *12*, 21–42. [[CrossRef](#)]
44. Tang, J.; Deng, C.; Huang, G.B. Extreme Learning Machine for Multilayer Perceptron. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 809–821. [[CrossRef](#)]
45. Humphrey, G.B.; Maier, H.R.; Wu, W.; Mount, N.J.; Dandy, G.C.; Abrahart, R.J.; Dawson, C.W. Improved validation framework and R-package for artificial neural network models. *Environ. Model. Softw.* **2017**, *92*, 82–106. [[CrossRef](#)]
46. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [[CrossRef](#)]
47. Seckin, N. Modeling flood discharge at ungauged sites across Turkey using neuro-fuzzy and neural networks. *J. Hydroinform.* **2011**, *13*, 842–849. [[CrossRef](#)]

48. Sharifipour, M.; Bonakdari, H.; Zaji, A.H. Comparison of genetic programming and radial basis function neural network for open-channel junction velocity field prediction. *Neural Comput. Appl.* **2018**, *30*, 855–864. [[CrossRef](#)]
49. Safavi, H.R.; Malek Ahmadi, K. Prediction and assessment of drought effects on surface water quality using artificial neural networks: Case study of Zayandehrud River, Iran. *J. Environ. Health Sci. Eng.* **2015**, *13*, 68. [[CrossRef](#)] [[PubMed](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).