

# Artificial Neural Network Approach for Business Decision Making applied to a Corporate Relocation Problem

**Malik Jamal Haddad**

School of Mechanical and Design Engineering,  
University of Portsmouth, Portsmouth, UK.

**David Adrian Sanders**

School of Mechanical and Design Engineering,  
University of Portsmouth, Portsmouth, UK.

## ABSTRACT

**This paper presents a new Artificial Neural Network approach to making a business decision. A corporate relocation problem is considered as an example for a business decision and the new approach is applied to select a city for corporate relocation and to rank a set of potential alternatives. Selecting the location of corporate real estate can be key to optimizing an organization's success. This is the first time Artificial Neural Networks have been used for this sort of business application. The Neural Network behaved satisfactorily and provided 91.76% accuracy when tested against randomly generated test sets. Five potential cities were considered: New York City, Washington D.C. Atlanta, Los Angeles and Portland. Decision makers identified six criteria: Financial Considerations Employee Availability, Support Services, Cultural Opportunities, Leisure Activities, and Climate. A suitable city is recommended that provides an appropriate solution and the outcome of the new approach is also used to rank potential cities based on their suitability.**

**Keywords:** Artificial Neural Networks; Corporate Relocation; Multiple Criteria; Decision Making.

## INTRODUCTION

This paper presents a new Artificial Neural Network (ANN) approach to assist with business decisions. The new approach is applied to a real world corporate relocation problem considered in [1]. Results from applying the new approach are validated against results presented in [1] and advantages of the new approach are presented.

Reducing operating costs and enhancing commercial competitiveness have often been reasons for corporate relocation. Christersson and Rothe [4] suggested five reasons for corporate relocation: Economic; Social; Environmental; Organizational identity, culture and image. Hassanain et al. [5] claimed that the main reason for relocation was often to reduce cost. Corporate relocation decisions are important long-term strategic decisions in the life time of any organization. Relocation decision often directly affect organization's existence and effectiveness [2-4].

Many researchers have used the term “*Black Box*” to represent the corporate relocation decision process [6]. Haddad et al. [1] presented a framework that demystified part of that “*Black Box*” and helped decision makers to understand the corporate relocation decision process, selecting an appropriate location that achieved their anticipated goals using Multiple Criteria Decision Making (MCDM) methods. A real world example was presented for ranking five cities in the United States of America based on their suitability for corporate relocation. The new approach described in this paper used MCDM concepts to calculate the overall score for the alternatives used in training and testing the Neural Network. This is the first time ANN have been used in this way and for this sort of application.

The next Section briefly describes corporate relocation decisions. Section 3 describes LSTM Neural Networks and Section 4 presents the new approach to making business decisions and applies it to a corporate relocation problem. Section 5 validates the results of the new approach, Section 6 discusses the results and Section 7 presents some concluding remarks and future work.

### **CORPORATE RELOCATION DECISIONS**

Corporate relocation is often considered as the physical move of an organization from old sites to new sites and often included all the processes and services required to successfully complete the relocation. A corporate relocation decision is often seen as a major and long term strategic decision in the life span of an organization. Relocation decisions often affected the organizations’ existence and competitiveness [2-4].

Several factors should be taken into consideration when making relocation decisions: cost of relocation, employees’ satisfaction and disruption to work progress during the relocation process. Researchers have suggested that the main reason for corporate relocation was to improve corporate profitability [7 & 8]. Others claimed that cultural, geographical, and legal factors could be the most important reason for relocation [4, 9 & 10]. Relocation has a number of effects on an organization and its employees. Glatte [3] proposed four corporate relocation concepts:

1. Site selection.
2. Site effect.
3. Site development.
4. Site design.

This paper is concerned with site selection. Christersson et al. [10] considered relocation decisions as an added value function for Real Estate Management. Rothe and Heywood [11] claimed that the corporate relocation process was not part of an organizations’ “day-to day” activities, and stressed that it was the function of a Corporate Real Estate (CRE) department since it often included physical relocation. All the services and processes required to successfully complete the relocation and all the post move, settling in and adjustments required after the relocation are conducted by the CRE department.

Corporate relocation decisions can be prone to distortion as decisions can often be made in fuzzy, risky and uncertain conditions, where many assumptions might be considered, and outcomes could have major effects. Using more complex and computationally inexpensive scientific algorithms, for example AI and ANN, could help achieve better decisions.

Corporate relocation decisions involve many factors and have major effects on organization survival. ANN could produce reliable decisions when large number of inputs are considered.

### LSTM NEURAL NETWORKS

Despite their biologically inspired name, ANNs, are mathematical algorithms often used to model non-linear and complex functions. The basic unit of an ANN is called a perceptron and one is shown in Figure 1. They are responsible for conducting mathematical operations and modeling non-linear functions. ANNs are often constructed by combining several perceptrons together to create a layer of perceptrons, adding multiple layers to create a Network as shown in Figure 2.

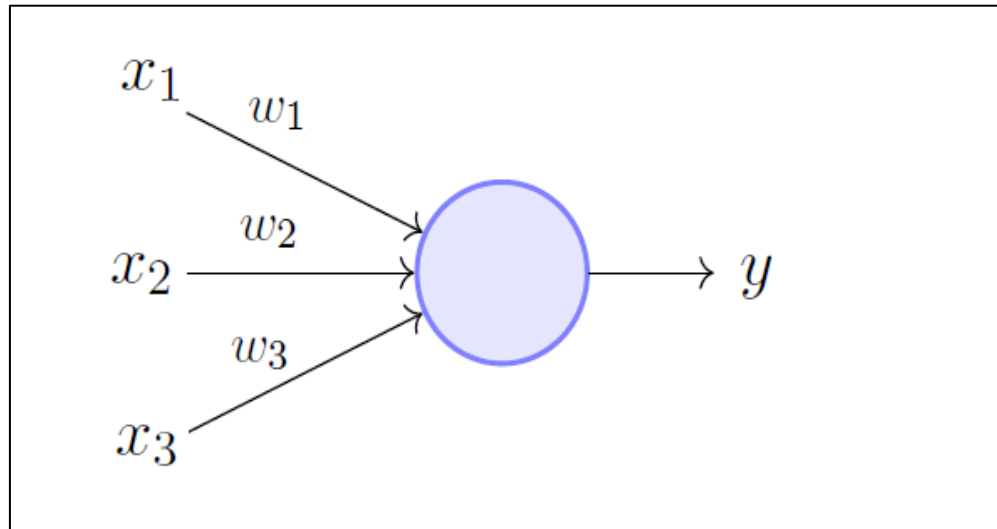


Figure 1. Single perceptron.

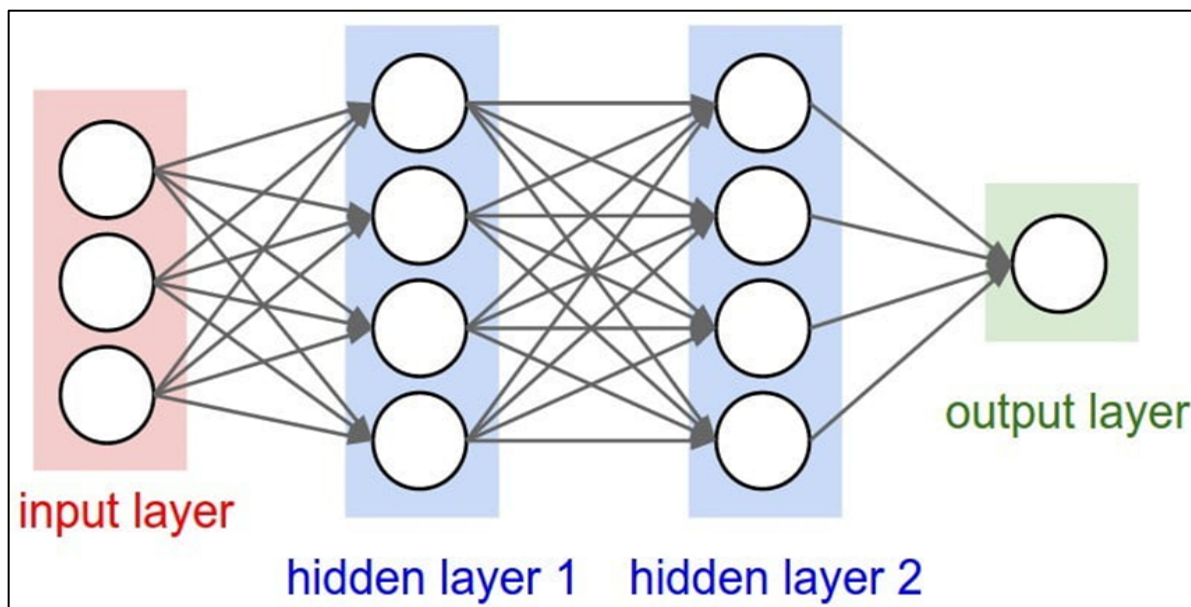


Figure 2. ANN model

A perceptron can be trained on a set of samples using a “learning algorithm”. To train a Network, the perceptron weights are adjusted in proportion to the difference between the correct output, and the perceptron output, for each sample. The aim is to minimize the difference between the

correct output and the Network output. This is achieved by slightly adjusting the weights by small values “learning rate” and running all the samples again. Running all the samples in a set and adjusting the weights to minimize the difference is “one epoch”.

An ANN often consists of an input layer and an output layer and hidden layers in between. Different types of ANN are used for different types of applications. This paper considered a LSTM Neural Network. LSTM Neural Networks are often regarded as a type of Recurrent Neural Network. They were first introduced by Hochreiter and Schmidhuber [12] in 1997. Since then, many researchers have worked on simplifying their structure, and enhancing their efficiency and accuracy [13 & 14]. Due to the growth of statistical modeling in science, researchers are using more complex methods and algorithms, such as ANNs, to deal with problems concerned with patterns and prediction [15]. ANNs can show excellent predictive power compared to traditional approaches. ANNs are receiving increased attention as powerful, flexible modeling techniques for predicting patterns in data [16]. LSTM have been successfully used in many fields of science including handwriting recognition, text completion, vehicle trajectory prediction and pattern recognition [16 & 17]

LSTM networks evaluate mappings from input sequence to output sequence by calculating the network units’ activations [18]. Deep LSTM Networks are often created by stacking several LSTM layers. Input sets go through multiple non-linear layers with each layer identifying specific features of a set. After data were processed through the layers, the network could recognize the appropriate identifiers for classifying the data to appropriate classes [19]. Due to their classification and predictive capabilities, ANNs could produce suitable and reliable outcomes for corporate location problems.

### **THE NEW APPROACH APPLIED TO CORPORATE RELOCATION**

To demonstrate the new business decision making approach, it was applied to a corporate relocation problem considered in [1]. The corporate relocation problem identified six decision criteria: Financial Considerations, Employee Availability, Support Services, Cultural Opportunities, Leisure Activities, and Climate; Seasonal, and Year Round. Five potential cities were identified and are shown in Figure 3. The five cities were: New York City, Washington D.C., Atlanta, Georgia, Los Angeles, California, and Portland, Oregon.



**Figure 3. Map of the United States of America showing five potential cities to relocate considered in this paper.**

Decision maker identified the most important criteria and provided weights showing their relevant importance. Also performance measures for all the alternatives (cities) with respect to all decision criteria were identified. The performance measures signified how suitable the alternatives were with respect to each criterion. A High performance measure indicated good suitability. If two alternatives showed the same performance with respect to a specific criterion, then their performance measures with respect to that criterion were set equal. The performance measures of the five cities and the criteria weights are shown as a decision matrix in Table 1.

**Table 1. Decision Matrix for the Corporate Relocation problem considered in [1].**

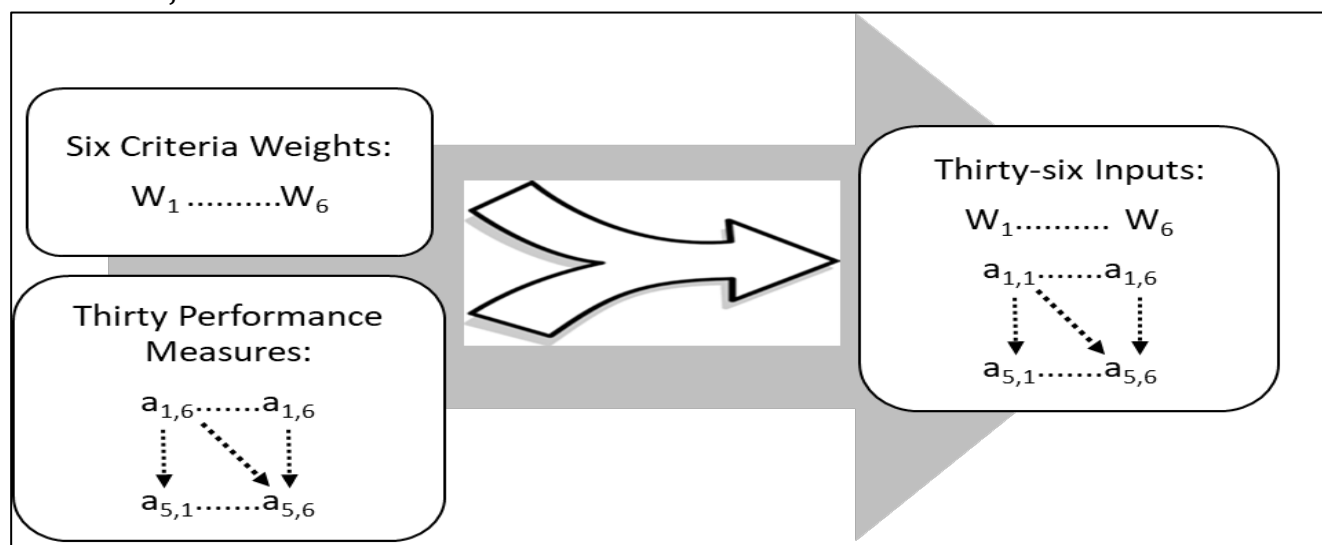
Alternative	A1 N.Y.C.	A2 Washington n D.C.	A3 Atlanta	A4 L.A.	A5 Portland
Criteria					
C1: Financial Considerations = 0.428	0.313	0.119	0.176	0.346	0.046
C2: Employee Availability = 0.207	0.064	0.098	0.168	0.493	0.177
C3: Support Services = 0.207	0.416	0.062	0.116	0.284	0.122
C4: Cultural Opportunities = 0.041	0.300	0.215	0.105	0.307	0.073
C5: Leisure Activities = 0.063	0.060	0.107	0.160	0.315	0.359
C6: Climate = 0.053	0.082	0.082	0.173	0.442	0.220

Financial Considerations was identified as the most important criterion and it was given the highest weight, its weight was set to 0.428, followed by Employee Availability and Support Services which were equally important, their weights were set to 0.207. The fourth most important criterion was Leisure activities, its weight was set to 0.063, followed by Climate, its weight was set to 0.053.

The least important criterion was Cultural Opportunities. It was given the lowest weights. Its weight was set to 0.041.

Los Angeles scored highest with respect to Financial Considerations followed by New York City, Atlanta and Washington D.C. respectively. Portland scored lowest with respect to Financial Considerations. Los Angeles scored highest with respect to Employee Availability followed by Portland, Atlanta and Washington D.C. respectively. New York City scored lowest with respect to Employee Availability. New York City scored highest with respect to Support Services followed by Los Angeles, Portland and Atlanta respectively. Washington D.C. scored lowest with respect to Support Services. Los Angeles scored highest with respect to Cultural Opportunities followed by New York City, Washington D.C. and Atlanta respectively. Portland scored lowest with respect to Cultural Opportunities. Portland scored highest with respect to Leisure Activities followed by Los Angeles, Atlanta and Washington D.C. respectively. New York City scored lowest with respect to Leisure Activities. Los Angeles scored highest with respect to Climate followed by Portland and Atlanta respectively. Washington D.C. and New York City had similar climate. Both cities were given the same value and scored lowest with respect to Climate.

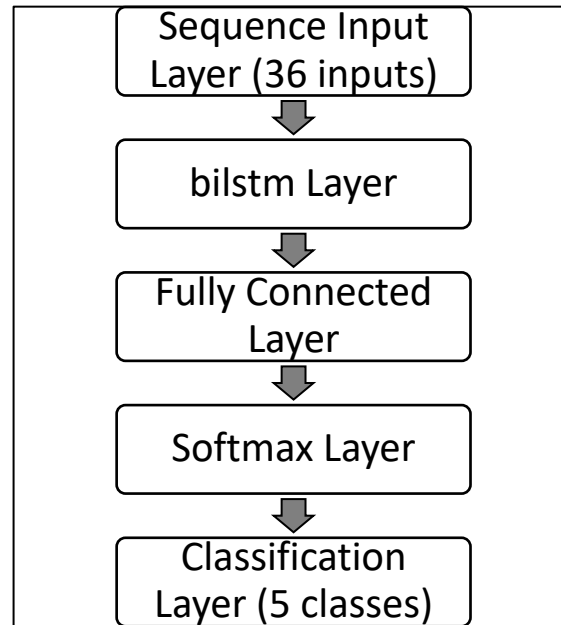
The first LSTM Neural Network used in this paper considered thirty-six inputs and five outputs. Inputs to the Network were the six criteria weights and the thirty performance measures in Table 1 comparing all alternatives with respect to all criteria as shown in Figure 4 where  $W_j$  represented the weight of criterion  $j$  and  $a_{i,j}$  represented the performance measure of alternative  $i$  with respect to criterion  $j$ .



**Figure 4. 36 inputs to the LSTM Network.**

The structure of the LSTM Neural Network used is shown in Figure 5. It consisted of five layers:

- Layer 1. Sequence Input Layer with thirty-six inputs.
- Layer 2. Bilstm Layer(100,'OutputMode','sequence').
- Layer 3. Fully Connected Layer with five nodes.
- Layer 4. Softmax Layer.
- Layer 5. Classification Layer.



**Figure 5. LSTM Neural Network used in this paper.**

MS Excel was used to create randomly generated criteria weights and performance measures. Twenty thousand sets were generated, each set containing six criteria weights and thirty performance measures. Randomly generated sets and the overall scores were used as input sets to train a Long Short Term Memory (LSTM) Neural Network. The overall score of the alternatives was calculated using Weighted Sum Model (WSM):

$$P_i = \sum_{j=1}^n W_j \cdot a_{ij} \quad (1)$$

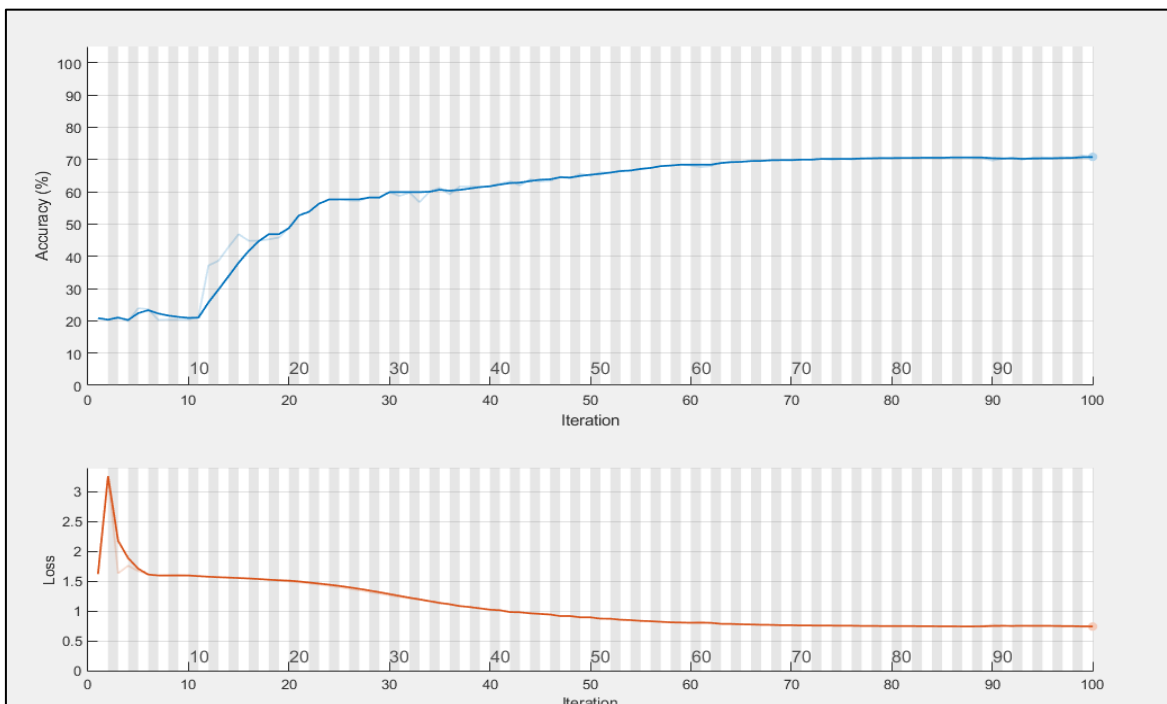
In the above equation  $P_i$  represents the overall score of alternative  $i$ ,  $W_j$  represents the weight of criterion  $j$ , and  $a_{ij}$  represents the performance measure of alternative  $i$  with respect to criterion  $j$ . The best solution to a problem was often the alternative that maximized the value of  $P_i$ . The overall scores were used to identify the best alternative and to rank the set of alternatives.

MATLAB was used to set the training options of the LSTM Network, the layers of the LSTM Network and the structure of the layers. A LSTM Neural Network was created with thirty-six inputs, 100 hidden units in the bilstm Layer, and five output classes. An adaptive momentum estimation algorithm was used in this architecture with an initial learning rate of 0.01 and maximum number of 100 epochs.

A (20,000x36) matrix was imported to MATLAB and used as training and testing sets. The matrix was split in a ratio 3:1 for training and testing sets respectively. Figure 6 shows Network training progress, the Network accuracy shown as a blue curve in the upper part of Figure 6 and the Network loss shown as a red curve in the lower part of Figure 6. The Network accuracy represented a measure of the performance of the Network and assessed how accurately the Network behaved over each epoch. It compared the Network outcome against the true outcome. The Network loss represented the total number of errors made for each sample during the training process. Loss value indicated how poorly the Network behaved after each epoch. To optimize the Network

performance, the Network accuracy should be maximized and the Network loss should be minimized.

The initial learning rate was set to 0.01 and the maximum number of epochs was set to 100. As Network training progressed, the Network training accuracy increased. The Network loss decreased. By the end of 100 epochs the Network training accuracy was above 70%.



**Figure 6. Screen shot of Network training progress considering 36 inputs and 5 output classes, Network training accuracy increasing up to 70% (above) and Network loss decreasing (below).**

Figure 7 shows the training outcome showing that the Network required 10 minutes and 28 seconds to complete 100 epochs using an initial learning rate 0.01, as the training progressed Network accuracy increased and Network loss decreased. By the end of the 100 epochs the Network training accuracy reached 70.79%.

```
>> ChoiceNet=trainNetwork(XTrain,YTrain, layers,options);
Training on single CPU.
```

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:08	20.81%	1.6175	0.0100
50	50	00:05:14	64.90%	0.8941	0.0100
100	100	00:10:28	70.79%	0.7404	0.0100

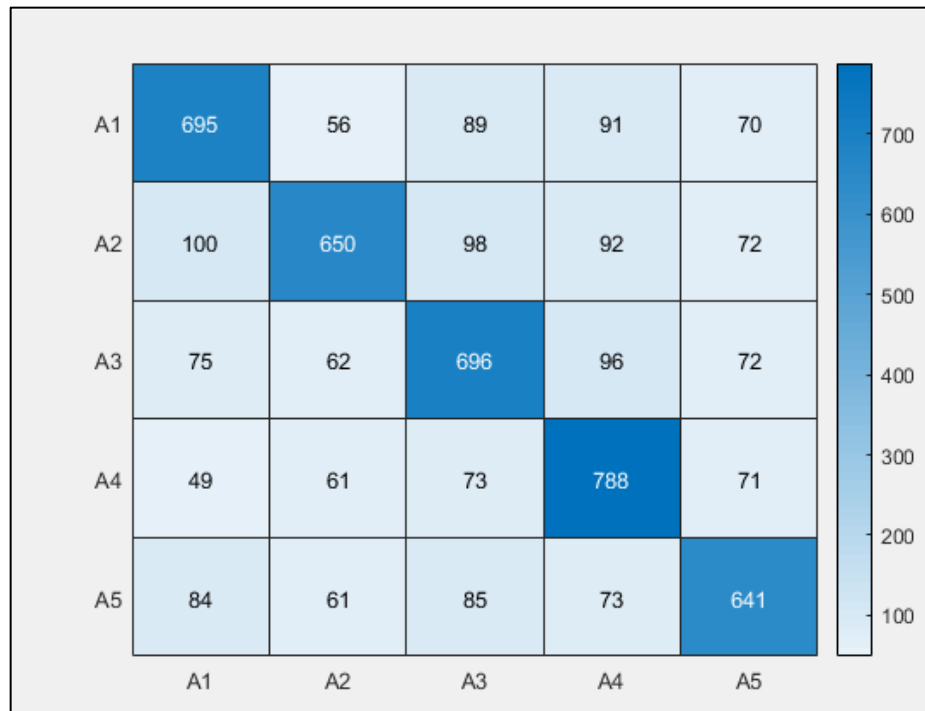
**Figure 7. Screen shot of Network training outcome considering 36 inputs and 5 output classes.**

The Network accuracy reached 69.4% when tested with the testing set.

Figure 8 shows the confusion chart produced from testing the Network with the testing set. A confusion chart shows the predictions produced from the Network (Vertical Axis) and the number



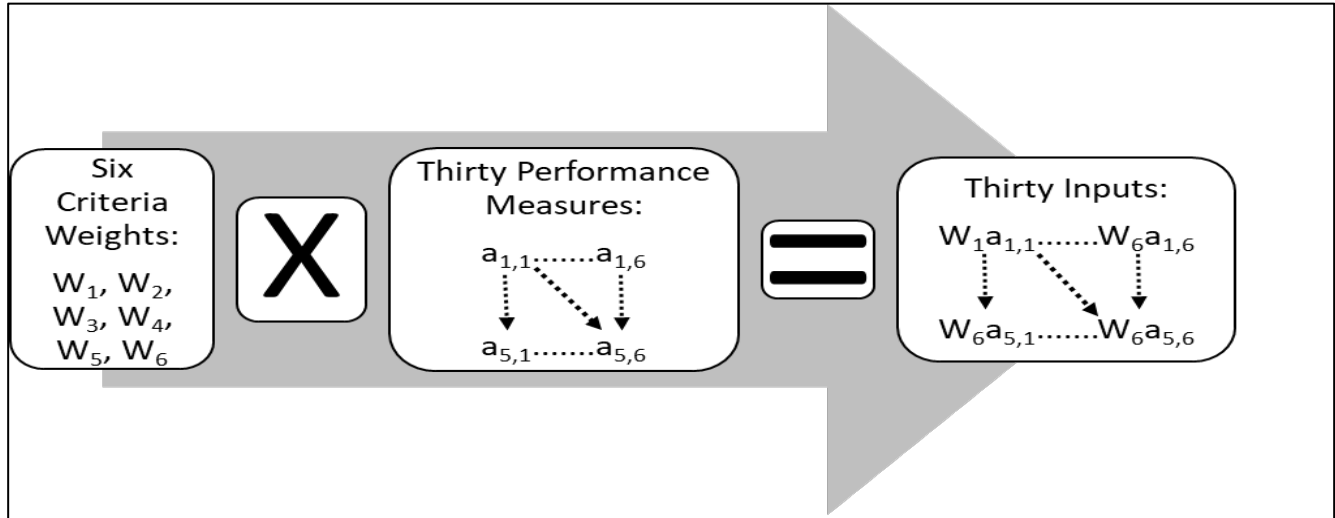
of correct labels from the testing set (Horizontal axis). Network correct predictions are shown diagonally in the darker blue boxes.  $A_1$  was correctly predicted by the Network 695 times but was incorrectly predicted as  $A_2$  56 times, 89 times as  $A_3$ , 91 times as  $A_4$  and 70 times as  $A_5$ .  $A_2$  was correctly predicted by the Network 650 times but was incorrectly predicted 100 times as  $A_1$ , 89 times as  $A_3$ , 91 times as  $A_4$  and 72 times as  $A_5$ .  $A_3$  was correctly predicted by the Network 696 times but was incorrectly predicted 75 times as  $A_1$ , 62 times as  $A_2$ , 96 times as  $A_4$  and 72 times as  $A_5$ .  $A_4$  was correctly predicted by the Network 788 times but was incorrectly predicted 49 times as  $A_1$ , 61 times as  $A_2$ , 73 times as  $A_3$  and 71 times as  $A_5$ .  $A_5$  was correctly predicted by the Network 641 times but was incorrectly predicted 84 times as  $A_1$ , 61 times as  $A_2$ , 85 times as  $A_3$  and 73 times as  $A_4$ .



**Figure 8. Screen shot of the confusion chart used to assess Network accuracy.**

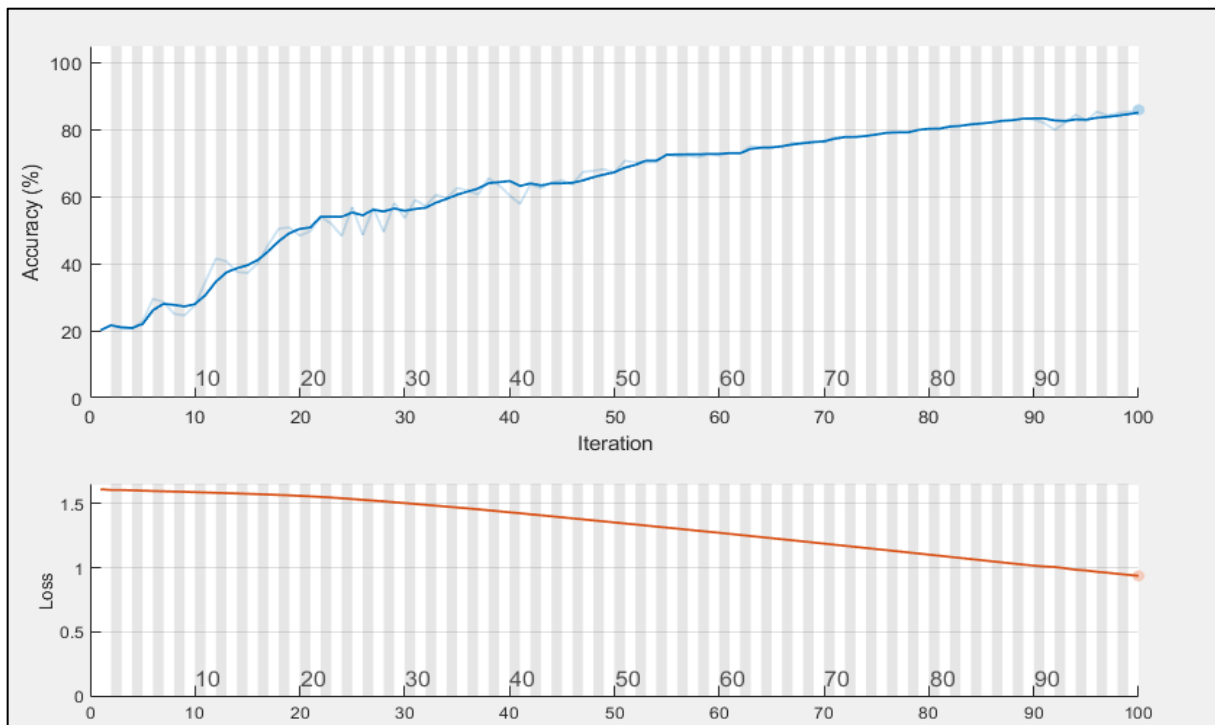
The Network produced acceptable accuracy. Different values for initial learning rate and maximum number of epochs were considered. A best compromise between learning time and overall accuracy of the Network was made.

Inputs to the LSTM Network were modified to improve Network accuracy. Performance measures were multiplied by the criteria weights to reduce the number of inputs to the Network. The performance measures of each alternative with respect to each criterion were scaled by that criterion weight to create thirty inputs to the Network as shown in Figure 9 where  $W_j$  represented the weight of criterion  $j$  and  $a_{i,j}$  represented the performance measure of alternative  $i$  with respect to criterion  $j$ .



**Figure 9. Scaling performance measures by criteria weights to reduce number of inputs.**

The same Network structure shown in Figure 5 was used. An adaptive momentum estimation algorithm was used in this architecture with an initial learning rate of 0.001 and maximum number of 100 epochs. A (20,000x30) matrix was imported to MATLAB and used as training and testing sets. The matrix was split in a ratio 3:1 for training and testing sets respectively. Figure 10 shows Network training progress with an initial learning rate 0.001 and 100 epochs considering thirty inputs and five output classes. As Network training progressed, the Network training accuracy increased, shown as a blue curve in the upper section of Figure 10 and the Network loss decreased, shown as a red curve in the lower section of Figure 10. By the end of 100 epochs the Network training accuracy was around 86%.



**Figure 10. Screen shot of Network training progress considering 30 inputs and 5 output classes, Network training accuracy increasing up to 70% (above) and Network loss decreasing (below).**

Figure 11 shows the training outcome showing that the Network required 13 minutes and 48 seconds to complete 100 epochs using an initial learning rate 0.001, as the training progressed Network accuracy was increasing and Network loss was decreasing. By the end of the 100 epochs the Network training accuracy reached 85.96%.

```
>> ChoiceNet2=trainNetwork(ITrain,OTrain,layers2,options);
Training on single CPU.
```

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:06	20.15%	1.6127	0.0010
50	50	00:08:05	67.12%	1.3541	0.0010
100	100	00:13:48	85.96%	0.9373	0.0010

Figure 11. Screen shot of Network training outcome considering 30 inputs and 5 output classes.

The Network accuracy reached 86.58% when tested with the testing set. Figure 12 shows the confusion chart produced from testing the Network with the testing set considering thirty inputs and five output classes. The confusion chart shows the predictions produced from the Network. Network correct predictions are shown diagonally in the darker blue boxes. A<sub>1</sub> was correctly predicted by the Network 904 times but was incorrectly predicted as A<sub>2</sub> 25 times, 23 times as A<sub>3</sub>, 35 times as A<sub>4</sub> and 14 times as A<sub>5</sub>. A<sub>2</sub> was correctly predicted by the Network 880 times but was incorrectly predicted 48 times as A<sub>1</sub>, 42 times as A<sub>3</sub>, 26 times as A<sub>4</sub> and 16 times as A<sub>5</sub>. A<sub>3</sub> was correctly predicted by the Network 873 times but was incorrectly predicted 47 times as A<sub>1</sub>, 27 times as A<sub>2</sub>, 42 times as A<sub>4</sub> and 12 times as A<sub>5</sub>. A<sub>4</sub> was correctly predicted by the Network 912 times but was incorrectly predicted 46 times as A<sub>1</sub>, 21 times as A<sub>2</sub>, 39 times as A<sub>3</sub> and 24 times as A<sub>5</sub>. A<sub>5</sub> was correctly predicted by the Network 760 times but was incorrectly predicted 45 times as A<sub>1</sub>, 37 times as A<sub>2</sub>, 52 times as A<sub>3</sub> and 50 times as A<sub>4</sub>. Comparing results from Figures 8 and 12 shows that the Network produced more accurate results when the number of inputs were reduced to thirty and the initial learning rate was decreased to 0.001.

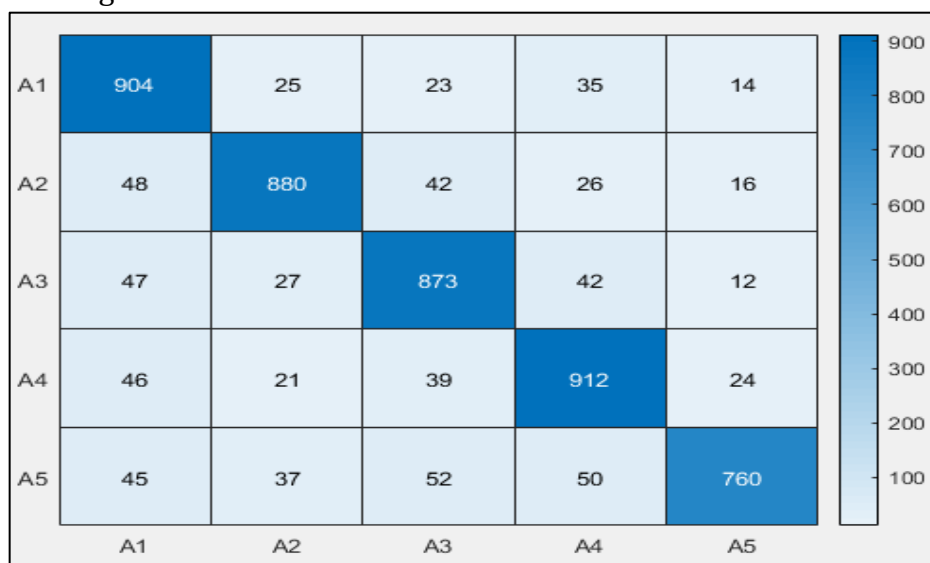
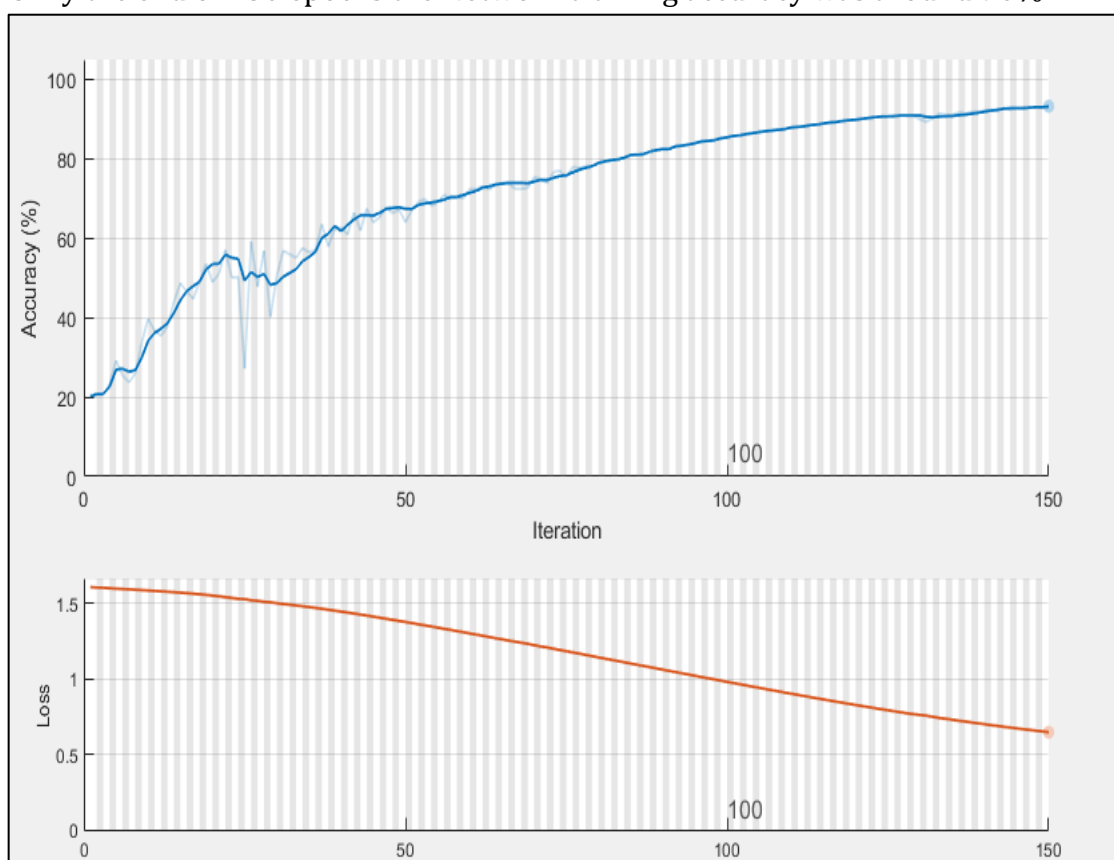


Figure 12. Screen shot of the confusion chart used to assess Network accuracy considering 30 inputs and 5 output classes.

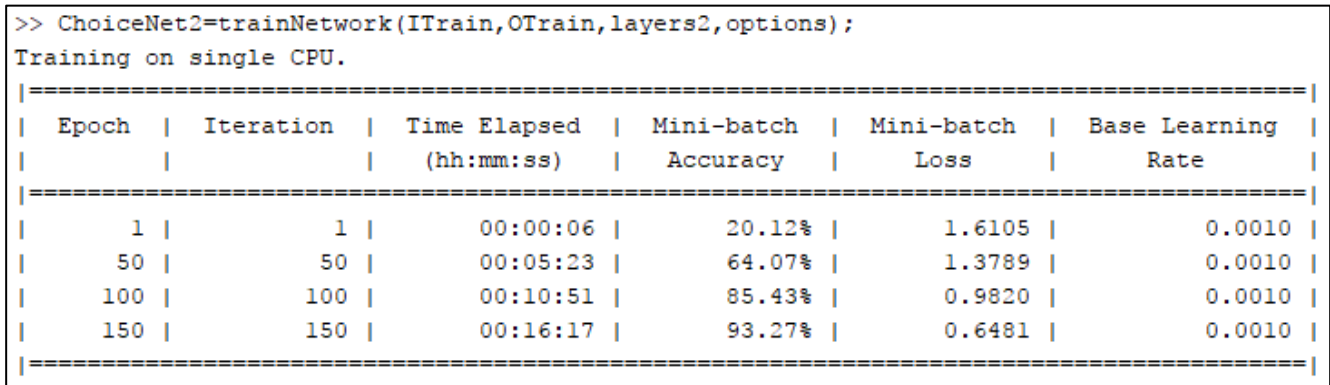
Figure 11 showed that the Network required 13 minutes and 48 seconds to complete 100 epochs with an initial learning rate of 0.001. Using smaller initial learning rate required more time to complete the 100 epochs.

The Network produced higher accuracy when the data was pre-processed so that the number of inputs were reduced to thirty as shown in Figure 11. Different values for initial learning rate and maximum number of epochs were considered. A best compromise between time and overall accuracy of the Network was conducted. The initial learning rate and maximum number of epochs were set to 0.001 and 150 respectively. Figures 13 shows Network training progress, with an initial learning rate 0.001 and 150 epochs considering thirty inputs and five output classes. As Network training progressed, the Network training accuracy increased, shown as a blue curve in the upper section of Figure 13 and the Network loss decreased, shown as a red curve in the lower section of Figure 13. By the end of 150 epochs the Network training accuracy was around 90%.



**Figure 13. Screen shot of Network training progress for 150 epochs considering 30 inputs and 5 output classes, Network training accuracy increasing up to 93.27% (above) and Network loss decreasing (below).**

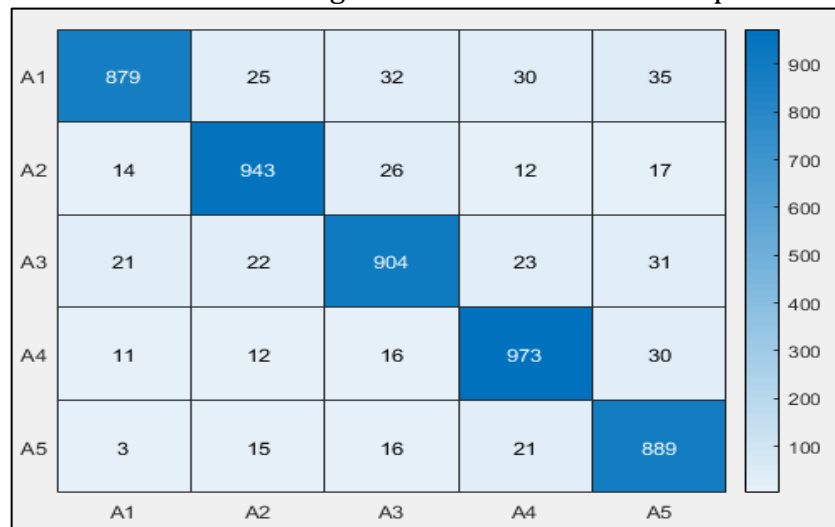
Figure 14 shows the Network training outcome showing that the Network required 16 minutes and 17 seconds to complete 150 epochs using an initial learning rate 0.001, as the training progressed Network training accuracy increased and Network loss decreased. By the end of the 150 epochs the Network training accuracy reached 93.27%.



**Figure 14. Screen shot of Network training outcome for 150 epochs considering 30 inputs and 5 output classes.**

The Network accuracy reached 91.76% when tested with the testing set.

Figure 15 shows the confusion chart produced from testing the Network with the testing set considering thirty inputs and five output classes after completing 150 epochs. The confusion chart shows the predictions produced from the Network, the Network correct predictions are shown diagonally in the darker blue boxes. A<sub>1</sub> was correctly predicted by the Network 879 times but was incorrectly predicted as A<sub>2</sub> 25 times, 32 times as A<sub>3</sub>, 30 times as A<sub>4</sub> and 35 times as A<sub>5</sub>. A<sub>2</sub> was correctly predicted by the Network 943 times but was incorrectly predicted 14 times as A<sub>1</sub>, 26 times as A<sub>3</sub>, 12 times as A<sub>4</sub> and 17 times as A<sub>5</sub>. A<sub>3</sub> was correctly predicted by the Network 904 times but was incorrectly predicted 21 times as A<sub>1</sub>, 22 times as A<sub>2</sub>, 23 times as A<sub>4</sub> and 31 times as A<sub>5</sub>. A<sub>4</sub> was correctly predicted by the Network 973 times but was incorrectly predicted 11 times as A<sub>1</sub>, 12 times as A<sub>2</sub>, 16 times as A<sub>3</sub> and 30 times as A<sub>5</sub>. A<sub>5</sub> was correctly predicted by the Network 889 times but was incorrectly predicted 3 times as A<sub>1</sub>, 15 times as A<sub>2</sub>, 16 times as A<sub>3</sub> and 21 times as A<sub>4</sub>. The confusion chart shown in Figure 18 shows a higher number of correct predictions (shown diagonally in the darker blue boxes) than the number of correct predictions shown in the confusion charts in Figures 8 and 12. Comparing results from Figures 8, 12 and 15 shows that the Network produced more accurate results when the number of inputs were reduced to thirty, Initial learning rate was decreased to 0.001 and increasing the maximum number of epochs to 150.



**Figure 15. Screen shot of the confusion chart used to assess Network accuracy considering 30 inputs and 5 output classes after completing 150 epochs.**

### **THE NEW APPROACH APPLIED TO A CORPORATE RELOCATION PROBLEM**

Haddad et al. [1] applied two well-known MCDM methods to the corporate relocation problem shown in Table 1: The Analytical Hierarchy Process (AHP) and the Preference Ranking Organization METHod for Enrichment Evaluations II (PROMETHEE II).

AHP is a MCDM method created by Thomas L. Saaty in 1971- 1975 [20]. AHP helps in solving multiple conflicting criteria problems. AHP breaks down a composite problem into simpler sub-problems then, combines the solutions of all sub-problems into one overall solution [21]. AHP applies pairwise comparisons between alternatives to assess the level by which one alternative dominates another with respect to each criterion [20]. Since its development, AHP has been used to solve multiple criteria problems in most fields of science [20 & 21].

PROMETHEE II is an outranking MCDM method used to produce a total ranking of alternatives. PROMETHEE II consist of a preference function describing each criterion and weights characterizing their significance. The idea behind PROMETHEE II is to apply pairwise comparisons among alternatives with respect to each criterion then broadly assess them with respect to all criteria [22].

The aim behind applying MCDM methods to this problem was to rank the set of alternatives (cities) based on their suitability using pairwise comparisons to achieve a total order of alternatives (cities). AHP and PROMETHEE II delivered the same ranking of cities: Los Angeles > New York City > Atlanta > Portland > Washington D.C.

The new approach presented in this paper was applied to the same corporate relocation problem considering six decision criteria and five potential cities to relocate [1]. Criteria weights and performance measures for all the alternative (cities) with respect to all the criteria are shown as a decision matrix in Table 1.

The trained and tested Network was used to provide an overall outcome for each city based on the criteria weights and performance measures. WSM was used to create inputs for the LSTM Neural Network. Thirty inputs were considered.

### **DISCUSSION AND RESULTS**

The new approach identified A4 (Los Angeles) as the best city to relocate and produced the following ranking of cities: Los Angeles > New York City > Atlanta > Portland > Washington D.C. The overall score of the cities were: New York = 0.211, Washington D. C. = 0.181, Atlanta = 0.189, Los Angeles = 0.236 and Portland = 0.182.

Los Angeles was identified as the best city to relocate by AHP and PROMETHEE II in [1]. The new approach also provided the same ranking of cities to relocate based on their suitability. In this way the results were validated against results published in [1].

MCDM methods do not produce a "correct" decision, rather they aid decision makers to find an appropriate solution that corresponds to their aims as well as their appreciation of the problem [22]. Since there was no perfect decision making method, Glatte [3] suggested using more than one decision making method to validate the outcome of the decision process, and to check for the

suitability of the methods used. He recommended decision makers to consider only verifiable and reliable data to reach appropriate decisions.

Since there is no correct answer in MCDM problems, and since methods were often used because of their simplicity and software availability, the new approach was operated on a MATLAB platform, considered as a straightforward and computationally inexpensive platform that does not require expert knowledge.

Since the new approach was based on ANNs, Transfer Learning (TL) could be developed to modify the new approach [23]. The new approach could be generalized to suit most choice and ranking problems for example supplier selection, human resource management, and strategic marketing decisions.

### **CONCLUSION AND FUTURE WORK**

This paper presented a new approach for making business decisions based on a LSTM Neural Network and successfully applied it to a corporate relocation problem. This is the first time ANN have been applied to this sort of problem. The new approach blended MCDM concepts with ANNs to create a simple yet effective and quick business decision making system.

The new approach was applied to a business problem considered in [1] and results were validated using results in [1]. The new approach behaved satisfactorily and was simpler, more efficient and faster. Moreover, applying the new approach did not require previous knowledge or experience in MCDM methods. The outcome of the new approach could be used to select the most suitable alternative or to rank a set of alternatives.

Currently the authors are applying LSTM Neural Networks to other decision making problems [19, 20 & 22]. Future work will consider using different types of ANNs for other real world problems [21].

### **References**

1. Haddad, M., et al., Selecting a Discrete Multiple Criteria Decision Making Method to decide on a Corporate Relocation, *Archives of Business Research*, 2019. 7(5):p. 7(5), 48-67.
2. Oladokun, T., Corporate site selection and acquisition in a Nigerian GSM communication company, *Journal of Corporate Real Estate*, 2011, 13(4), p.247-260.
3. Glatte, T., Location Strategies: Methods and their methodological limitations, *Journal of Engineering Design and Technology*, 2015, 13(3), p. 435 – 462.
4. Christersson, M. and P. Rothe, Impacts of organizational relocation: a conceptual Framework, *Journal of Corporate Real Estate*, 2012, 14(4), p.226-243.
5. Hassanain, M.A., et al., Weighted Evaluation Method for Corporate Real Estate Site Selection in Saudi Arabia, *Journal of Urban Planning and Development*, 2018, 144(1), p.05017012.
6. Arkesteijn, M., et al., Improving decision making in CRE alignment, by using a preference-based accommodation strategy design approach, *Journal of Corporate Real Estate*, 2017, 19(4), p.239-264.
7. Cumming, D., et al., Corporate Relocation in Venture Capital Finance, *Entrepreneurship Theory and Practice*, 2009, 33(5), p. 1121 – 1155.

8. Barcena-Ruiz, J. C. and M. B. Garzon, Relocation and investment in R and D by firms, Portuguese Economic Journal, 2014, 13(1), p. 25–38.
9. Brouwer, A., et al., The firm relocation decision: An empirical Investigation. Annals of Regional Science, 2004, 38(2), p. 335–347.
10. Christersson, M., et al., Social impacts of a short-distancerelocation process and new ways of working, Journal of Corporate Real Estate 2017, 19(4), p. 65-284.
11. Rothe, P. and C. Heywood, Demystifying the short-distance relocation process: 5 cases from Finland, Journal of Corporate Real Estate. 2015, 17(1), p. 160-177.
12. Hochreiter, S. and J. Schmidhuber, Long short-term memory, Neural Comput., 1997, 9(8), p. 1735-1780.
13. Gers, F. A. Learning to forget: Continual prediction with LSTM, in Proc. 9th International Conference on Artificial Neural Networks: ICANN '99, Edinburgh, UK, 1999, pp. 850–855.
14. Greff, K., et al., LSTM: A Search Space Odyssey, IEEE T Neur Net Lear., 2015, 28(10), p. 2222- 2232.
15. Olden, J.D. and Jackson, D.A., Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks, Ecological modelling, 2002, 154(1-2), p.135-150.
16. Graves, A., et al., A Novel Connectionist System for Unconstrained Handwriting Recognition, IEEE T Pattern Anal., 2008, 31(5), p. 855–868.
17. Dai, S., et al., Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction, in IEEE Access, 2019, 7(1), p. 38287-38296.
18. Sak, H., et al., Long short-term memory recurrent neural network architectures for large scale acoustic modeling, 2014.
19. Haddad, M. and D. Sanders, Deep Learning Architecture to Assist with Steering a Powered Wheelchair, IEEE Transactions on Neural System and Rehabilitation Engineering, Accepted and in Press.
20. Haddad; M., et al., Selecting a robust decision making method to evaluate employee performance, International Journal of Management and Decision Making, 2019, 18(4), p. 333 – 351.
21. Haddad, M., et al., Selecting a discrete multiple criteria decisionmaking method for Boeing to rank four global market regions, Transportation Research Part A: Policy and Practice, 2020, 134(1), p 1 -15.
22. Haddad, M. and D. Sanders.,Selecting a best compromise direction for a powered wheelchair using PROMETHEE, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2019, 27(2), p. 228-235.
23. Côté-Allard, U. et al., Deep learning for electromyographic hand gesture signal classification using transfer learning, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2019, 27(4), p. 760-771.