Abstract - Transshipment problem is an example of a transportation problem that deals with the shipment of a homogenous product from various supply origins through different transshipment points to various demand destinations with the objective of minimizing the total transshipment cost. Transshipment problem has been solved using the Modified Distribution Method (MODI) where the transshipment problem is converted to an equivalent transportation problem by classifying the transshipment points as being parts of the supply origins and the demand destinations. The method cannot be used to predict the total cost of shipping a homogenous product from various supply origins through different demand destination to a particular demand destination. In this paper we have deployed the Modified Neural Network model to find the optimal solution of a transshipment problem. The model accepts both normalized and un-normalized data sets as input. The model was simulated using C++ programming language. The result has shown that the Modified Neural Network Model is more efficient and effective to predict the total cost of shipping a homogenous product from various supply origins to a particular demand destination through different transshipment points.

Keyword: Supply points, Demand points, Transshipment points, Neural Network and Transshipment Problem.

I INTRODUCTION
Transportation Problem is a subclass of a linear programming problem that deals with the shipment of a homogenous product from various supply origins, to different demand destinations with the objective of minimizing the total transportation cost in the problem. It can be classified as direct or indirect Transportation Problem [1].

The direct transportation problem deals with the shipment of homogenous product from various supply origins to different demand destinations. The mathematical model for transportation problem is stated as [2]:

\[\text{Minimizing } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \]  

Subject to the supply constraints:
\[\begin{align*}
x_{11} + x_{12} + x_{13} + \ldots + x_{1m} = a_1 \\
x_{21} + x_{22} + x_{23} + \ldots + x_{2m} = a_2 \\
\vdots \\
x_{m1} + x_{m2} + x_{m3} + \ldots + x_{mn} = a_m
\end{align*}\]

The demand constraints:
\[\begin{align*}
x_{1i1} + x_{1i2} + x_{1i3} + \ldots + x_{1in} = b_1 \\
x_{2i1} + x_{2i2} + x_{2i3} + \ldots + x_{2in} = b_2 \\
\vdots \\
x_{mi1} + x_{mi2} + x_{mi3} + \ldots + x_{min} = b_m
\end{align*}\]

And the demand constraints:
\[\begin{align*}
x_{1 j1} + x_{1 j2} + x_{1 j3} + \ldots + x_{1 jm} = b_1 \\
x_{2 j1} + x_{2 j2} + x_{2 j3} + \ldots + x_{2 jm} = b_2 \\
\vdots \\
x_{mj1} + x_{mj2} + x_{mj3} + \ldots + x_{mjm} = b_m
\end{align*}\]

Where \(a_i\) is the cost of transportation of a unit from the \(i^{\text{th}}\) source to the \(j^{\text{th}}\) destination, and the quantity \(x_{ij}\), to be positive integer or zero, represents the optimal quantity of the product that can be transported from the \(i^{\text{th}}\) origin to the \(j^{\text{th}}\) destination.

But instead of direct shipments to destinations, the commodity can be transported to a particular destination through one or more intermediate or trans-shipment points. Each of these points in turn supply to other points. Thus, when the shipments pass from destination to destination and from source to source, we have a trans-shipment problem. Since transshipment problem is a particular case of transportation problem hence to solve transshipment problem, we first convert transshipment problem into equivalent transportation problem and then solve it to obtain optimal solution using MODI method of transportation problem.

In a transportation problem, shipments are allowed only between source-sink pairs. In many applications, this assumption is too strong. The mathematical model for the converted transshipment problem is stated as:

\[\text{Minimizing } Z = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} c_{ij}x_{ij} \]  

Subject to:
\[\begin{align*}
x_{i1} + x_{i2} + x_{i3} + \ldots + x_{im} &= a_i \\
x_{i1} + x_{i2} + x_{i3} + \ldots + x_{i(n+m)} &= t_i \\
\vdots \\
x_{ij} + x_{i2} + x_{i3} + \ldots + x_{ij} &= b_j \\
x_{ij} + x_{i2} + x_{i3} + \ldots + x_{i(m+n)} &= t_j
\end{align*}\]

\[\sum_{i=1}^{m} a_i = \sum_{j=1}^{m+n} b_j \]
These constraints are similar to the constraints of a transportation problem with \( m + n \) sources and \( m = n \) destination with difference that here there are no \( x_{ij} \) and \( x_{ij} \) term and that \( b_i = 0 \) for \( j = 1, 2, 3, \ldots, m \) and \( a_i = 0 \) for \( i = m + 1, m + 2, \ldots, m + n \).

II REVIEW OF RELATED WORKS
Several researchers have solved the Transportation Problem with various methods to obtain an optimal solution. The simplex method developed by Dantzig provides an algorithm which consists in moving from one vertex of the region of feasible solution to another in such a way that the value of the objective function at the succeeding vertex is less in a minimization problem. [3] stated that the computational procedure in the simplex method is based on the fundamental property that the optimal solution to a linear programming problem, if it exists, occurs only at one corner point of the feasible region.

[4] formulated an OR model for finding an optimal solution for a transportation problem. All the optimal solution algorithms for solving Transshipment Problem need an initial basic feasible solution.

[5] uses the following methods in finding the initial feasible solution:

- North West Corner Method
- Least cost method
- Row minima Method
- Column minima method
- Vogel Approximation method

The algorithm deployed the stepping stone and the modified distribution methods (MODI) in testing for optimality of a transshipment problem.

[6] proposed a heuristic algorithm for solving transportation problems with mixed constraints and extend the algorithm to find a more-for-less (MFL) solution, if one exists. The more-for-less (MFL) paradox in a transportation problem occurs when it is possible to ship more total goods for less (or equal) total cost, while shipping the same amount or more from each origin and to each destination and keeping all the shipping costs non-negative.

[7] proposed a method, called separation method-based on the zero point to find an optimal solution problem where transportation cost, supply and demand are intervals. They developed the separation method without using the midpoint and width of the interval in the objective function of the fully interval transportation problem.


This paper deploys a modified neural network model to find the optimal solution of a transshipment problem.

III MATERIALS AND METHOD
Neural Networks are models designed to imitate the human brain through the use of mathematical models. A neural network consists of a set of artificial neurons (nodes) grouped in a number of layers [9]. Figure 1 illustrates a single layer neural network with three input elements and three neurons. In the network each element of the input vector \( p \) is connected to each neuron input through the weights matrix \( w \).

![Figure 1: A-layer neural network (source: Jamal, Ibrahim and Salam, 2009)](image1)

The neural network used the learning rule to modify the weights and the biases of the network and the training process for selecting parameters for a given problem. The training procedure used in multilayer perceptron neural network is one of the supervised learning algorithms called the back propagation algorithm. The back propagation algorithm uses the computed output error to adjust the weights so as to minimize the error in its predictions on the training dataset.

Back propagation neural network requires that all training input data must be normalized between 0 and 1 for training. It cannot be used to train un-normalized input data.

IV CONCEPTUAL FRAMEWORK OF THE PROPOSED SYSTEM
Conceptual design of a system is concerned with making a prototype of the proposed system. It provides a description of the proposed system in terms of a set of integrated ideas and concepts about what the system should do, behave and look like. The conceptual framework of the proposed system is illustrated in Figure 2.

![Figure 2: Conceptual framework of the proposed system](image2)

The proposed system accepts both normalized and un-normalized input data. It normalizes the data using the normalized function \( \lambda \). There are two basic normalization techniques namely max-min normalization and decimal scaling techniques. In this work we use the decimal scaling technique to normalize the data by moving the decimal unit of values of the attributes as shown in equation (9).

\[ v' = \frac{v}{10^j} \]

(9)

where \( j \) is the smallest integer such that \( \max |v'| < 1 \).
The Multi-Layer Perceptron Neural Network model for the transportation problem in two propagations namely, the forward and backward propagations. The algorithm uses the activation function to predict the calculated output. The activation function used in a multi-layer perceptron neural network is a sigmoid function which is expressed in equation (11).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

Where $\delta_k$ is the error calculated (actual) output in the previous layer and $w_{jk}$ is the weight of the connection between the ith unit in the next layer and the jth unit.

$$E_r = \frac{1}{2} \sum_i (y_i - \sigma(x_i))^2$$

Where $E_r$ is the mean square error calculated by equations (18) and (19). The overall performance (net error) of the Multi-Layer Perceptron Neural Network is measured by the mean square error (MSE) expressed in equation (13).

$$\delta_{ij} = \sum_k \delta_k w_{jk} \sigma'(x_j)$$

Where $\delta_{ij}$ is the error calculated by equations (1 and 2).

The back propagation error in the hidden layer is calculated by the formula in equation (7).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

The back propagation error in the output neuron is calculated by using the formula in equation (7).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

Where $\delta_k$ is the error calculated (actual) output in the previous layer and $w_{jk}$ is the weight of the connection between the ith unit in the next layer and the jth unit.

$$E_r = \frac{1}{2} \sum_i (y_i - \sigma(x_i))^2$$

Where $E_r$ is the mean square error calculated by equations (18) and (19). The overall performance (net error) of the Multi-Layer Perceptron Neural Network is measured by the mean square error (MSE) expressed in equation (13).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

Where $\delta_k$ is the error calculated (actual) output in the previous layer and $w_{jk}$ is the weight of the connection between the ith unit in the next layer and the jth unit.

$$E_r = \frac{1}{2} \sum_i (y_i - \sigma(x_i))^2$$

Where $E_r$ is the mean square error calculated by equations (18) and (19). The overall performance (net error) of the Multi-Layer Perceptron Neural Network is measured by the mean square error (MSE) expressed in equation (13).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

Where $\delta_k$ is the error calculated (actual) output in the previous layer and $w_{jk}$ is the weight of the connection between the ith unit in the next layer and the jth unit.

$$E_r = \frac{1}{2} \sum_i (y_i - \sigma(x_i))^2$$

Where $E_r$ is the mean square error calculated by equations (18) and (19). The overall performance (net error) of the Multi-Layer Perceptron Neural Network is measured by the mean square error (MSE) expressed in equation (13).

$$\delta_k = (1 - \sigma(x_k)) \sum_j \delta_j w_{jk}$$

Where $\delta_k$ is the error calculated (actual) output in the previous layer and $w_{jk}$ is the weight of the connection between the ith unit in the next layer and the jth unit.
Table 3: Initial solution using VAM

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>200</td>
</tr>
<tr>
<td>Y</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>A</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Demand</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>150</td>
<td>250</td>
<td>500</td>
</tr>
</tbody>
</table>

Using the Modified Distribution method, the optimal solution of the transportation problem is shown in Table 4:

Table 4: Optimal solution using MODI

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>supply</th>
<th>u_i</th>
<th>v_j</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>200</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>Demand</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>150</td>
<td>250</td>
<td>500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since opportunity cost corresponding to each unoccupied cell is positive, therefore the total transportation cost is:

\[
\text{total cost} = 100 \times 7 + 100 \times 8 + 50 \times 4 + 250 \times 0 = 2450
\]

Applying the Modified Neural Network using C++ simulator, the total transportation cost is calculated as follows:

Enter the number of input neurones
2
Enter the number of neurones in the hidden layer
2
Enter the number of neurones in the output layer
1

INPUT PARAMETERS
200
300

OUTPUT PARAMETERS
250

The weights between the input and the hidden layers are:
input[1].weight[1]=0.00125126
input[1].weight[2]=0.563585
input[2].weight[1]=0.193304
input[2].weight[2]=0.80874
The weights between the hidden and the output layers are:
hidden[1].weight[1]=0.585009
hidden[2].weight[1]=0.479873
The input to hidden layer [1] is: 1
The input to hidden layer [2] is: 1
The output quantity [1] is: 0.743622
the error is 249.256
The diff 0.256378
The error 249.256
The output_hidden error 47.5203
the sum 50.6035
The Input_hidden error 0
The Input_hidden error 0
the sum 50.6035
The Input_hidden error 0
The Input_hidden error 0
The weights between the hidden and the output layers are:
The weights between the input and hidden layers are:
input[1].weight[1]=0.00125126
input[1].weight[2]=0.563585
input[2].weight[1]=0.193304
input[2].weight[2]=0.80874
The input to hidden layer [1] is: 1
The input to hidden layer [2] is: 1
The output quantity [1] is: 1
the error is 249
The unit cost between the input and the hidden layers are:
input[1].cost[1]=7
input[1].cost[2]=8
The unit cost between the hidden and the output layers are:
hidden[1].cost[1]=1
the sum of neural 18.71893
the sum of neural 2240.106
The total Transportation cost ==248.825
Press any key to continue . . .
V DISCUSSION OF RESULT

The results have shown that the optimal solution of the transshipment problem is obtained using MODI method and the Modified Neural Network Model. The MODI method can be used by the producer to predict the total cost of transportation of a commodity from various supply origins to different demand destinations through various transshipment points while the Modified Neural Network model can be used by the producers or consumers to predict the total cost transportation from various supply origin to a particular demand destination through different transshipment points. Secondly, the results have shown that the optimal solution of a transshipment problem with large constraints and variables can be solved using the Modified Neural Network Model.

VI CONCLUSION

Transshipment problem is a multilayer perceptron problem with three layers namely input layer as the sources, hidden layers as transshipment points and output layers as the demand destination points. Transshipment problem with very large constraints and variables can best be solved using the Modified Neural Network Model. The model can be used to predict the total cost of shipping a homogenous product from various supply origins through different transshipment points to a particular demand destination.

VII REFERENCES