*Available online www.jsaer.com*

***Journal of Scientific and Engineering Research*, 2020, 7(9):177-184**

**Research Article**

**ISSN: 2394-2630**
**CODEN(USA): JSERBR**

# Algorithm for Predicting Opponent's Future Moves in an Electronic Negotiation using Artificial Neural Network

**Bennett, E.O.[1], Oniyinde, A.O.[2], Sako, D. J. S.[3]**

Department of Computer Science, Rivers State University, Port Harcourt, Nigeria
E-mails:[1]bennett.okoni@ust.edu.ng, [2]a.oinyinde@gmail.com, [3]sunday.sako@ust.edu.ng

**Abstract** Several models applied to electronic negotiations, particularly in predicting an opponent's future move(s) are seen to be restrictive, less accurate and lacking homogeneity. These models are also faced with the problem of limited or uncertain knowledge and conflicting preferences. This research is aimed at developing an algorithm for predicting a negotiator's counterpart future moves in an e-negotiation using Artificial Neural Network. Design methodology details the negotiation process between land agent and an intending land buyer. Land agent tries to maximize its own utility while improving the buyer's satisfaction level to arrive at a successful negotiation through collective satisfaction. Buyer accepts or rejects proposed offer by land agent based on his potential land value criteria. The neural network utilizes its learning ability to study the negotiation patterns and to predict the variables that affect the outcome of price negotiation. These variables include: land location, land type, size, density and access road. The predictive model was implemented using MATLAB. Results showed that negotiations greater than or equal to 85% of land values were accepted by agents and negotiations less than 80% of land values were rejected by agents. The applied neural network model resulted to lower Mean Square Error training regression of 0.92 and regression model produced 1.26 which indicated a linear relationship for the different rejected and accepted negotiations made by the agents. In comparing the non-linear neural network model to the regression model, a more effective data fit was produced.

**Keywords** Negotiation, Artificial Neural Network, Prediction, Electronic Negotiation System

## 1. Introduction

Negotiations are of vital significance in both informal and official dealings, including customer negotiations, service level negotiations with suppliers or joint contract negotiations, negotiation of selling and rental terms, provision of services and several other legal agreements. Such negotiations will most often have implications on ties between businesses, their competitiveness, and their credibility on the long run. A successful negotiation impacts significantly on commercial success by creating tighter business bonds, presenting durable and excellent solutions. Whenever a party fails to negotiate a good deal, the consequences may have a lasting adverse impact on the business or organization, which could give business rivals an opening to gain competitive advantage.

Many computer science techniques, in particular Artificial Intelligence, have also been used in designing and developing applications that support at least one negotiator [1]. The advent of the internet together with emerging computing and networking technologies, have created new possibilities and platforms for application design and delivery that can aid the various forms of negotiations and negotiating parties. There have been some research efforts in this area to provide solutions for assisting human negotiators. An overview of e-negotiations and Negotiation Support Systems (NSS) and Negotiation Software Agents (NSA) is presented in [2]. Electronic Negotiation Systems (ENS) provides a communication channel for several parties, via the Internet, to resolve a specific problem or to accomplish a common objective. In addition to the communication channel, the main feature of these systems is that they enable embedded information systems to improve information collection, processing, and transmission capabilities, and support decision-making and problem solving. Negotiation itself is broad as it is practiced in virtually every field and aspect of life especially in business. ENS attempts to integrate the various aspects of e-negotiations process and different e-negotiation systems. In negotiation, the ability to predict an opponent's move or next offer is key. Predicting the agent's behaviour and using those prediction results to maximize agents own benefits is one of the crucial issues in the negotiation process [3]. A negotiating agent must produce offers based on its scope, since it has limited opponent information and insufficient processing ability. Several methods and models have been applied to predict the negotiating actions (moves) of the opponent, but are known to be restrictive, less accurate and lack homogeneity as shown in [4,5]. Thus, the adoption of Artificial Neural Network (ANN); a model that can generalize and forecast unseen data by inferring unseen relationships on unseen data after learning from the initial inputs and their relationships, ANN does not place limitations on input parameters as compared to many other prediction models.

## 2. Related Works

With the advent and advancements of computer technologies, solutions to negotiation represents a new weapon to approach the negotiation problem. Various models have been applied to automate the negotiation process and its selected activities. These models vary from decision-making models of negotiation to learning methods for supporting the negotiation, based on a variety of mechanisms including: Game Theory, Possibilistic Decision Theory, Possibilistic Case-based Reasoning, Probabilistic Decision Theory, Constraint Based reasoning, Heuristic search, Bayesian Learning, Reinforcement Learning and evolutionary computation.

Early techniques were established on Game Theory. However, Game Theory makes a number of assumptions including knowledge of circumstances [6]. In order words, the rules of meeting must be understood, interests must be clearly stated and interests of opponents must also be known. Game theory has two crucial drawbacks that makes it not often considered. First, is the assumption that a negotiating agent or party has unlimited processing and reasoning power and second, is its assumption that all negotiating agents share similar information or knowledge. These limitations were eliminated by the introduction of decision-functions.

The Bayesian learning model enables updating the knowledge or beliefs of one agent about other agents [7]. Before negotiation starts an agent acquires knowledge. This knowledge can be acquired from past experiences and indirect knowledge; and are usually about negotiating agents or parties and the environment. The above knowledge is encoded in the form of subjective probability distributions. In Bayesian Learning, it is necessary to specify the strategy spaces and type spaces in order to be able to learn. Ideally, these spaces are defined generically enough to allow learning of a rich variety of opponent profiles. This model introduces various reasonable assumptions about the structure of opponent profiles as well as about an opponent's negotiation strategy to ensure learning an opponent model is feasible. These assumptions include: structural and rationality assumptions. However, this model presents the problem of uncertainty; because there is no general way to represent and process the uncertainty within the background knowledge and the prior probability function; it is also computationally intensive and expensive especially when applied to complex models.

The strategic reasoning of a negotiating agent is usually computationally intractable. In such situations it can be supported in the search for the best strategy by some heuristic approaches. [8] Suggest approximating the rational choice of negotiation strategies with the use of decision functions. This is established on the concept of heuristic approaches and techniques, which can be used for measuring successful offers or counter-offers during negotiations. This approach defines three negotiation tactics (functions), they include: time-dependent tactics, resource-dependent tactics, and behaviour dependent tactics.

Generally, Mathematical models yield worse results when compared to non-linear models of regression as proven by past experiments. As efficient and accurate as models of non-linear regression are, they have a drawback of being restrictive, because they involve known functions of an opponents' behaviour in negotiation, and mathematical models have empirically proven to be less accurate when compared to Artificial Neural Networks (ANN). Thus, the adoption of ANN which has been proven to be universal approximators, provides sufficient hidden layer neurons while assuming that the activation function is bounded and non-constant [9].

## 3. System Design

The structural and behavioral details of the system including the following key components: knowledge base, Artificial Neural Network, negotiation mechanism, and negotiation outcome is shown in Figure 3.1.

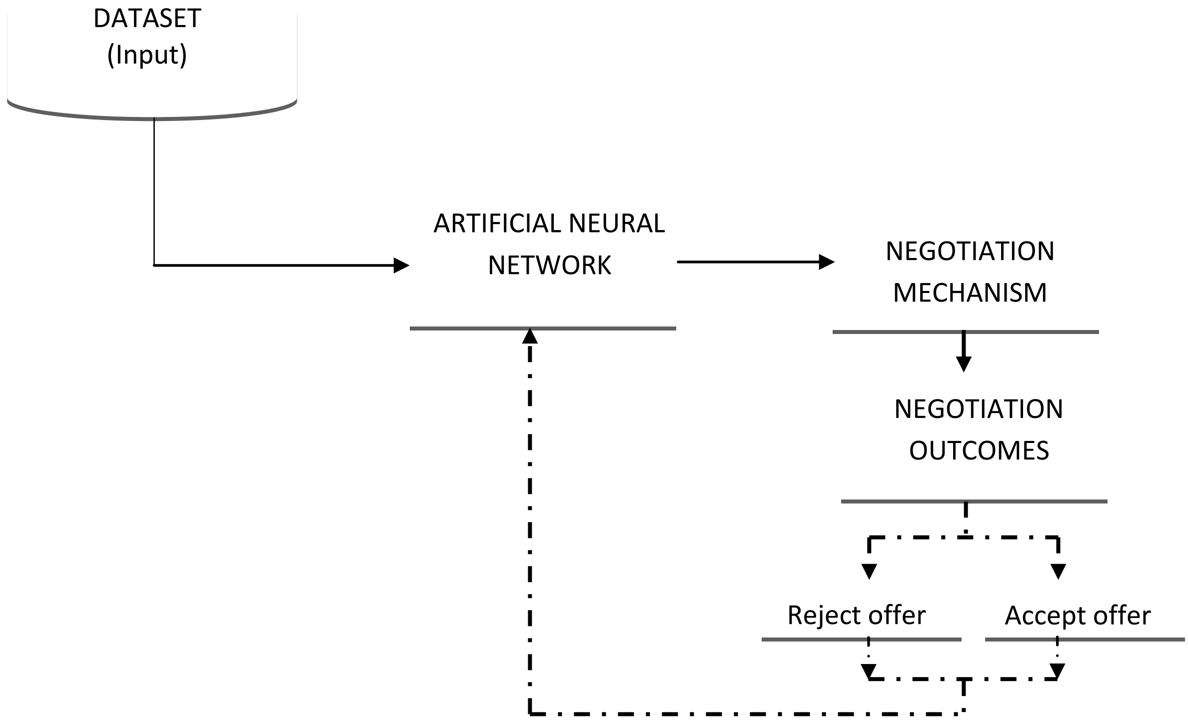


*Figure 3.1: Architecture of the System*

### 3.1. Input Parameters

Negotiation occurs between a real estate agent and a buyer. Land agent holds a key role in this model, as he sets the initial value. Through mutual satisfaction, the land agent tries to increase its utility while improving the opponent's satisfaction level to reach an agreement. The criteria for land procurement serve as inputs to the system, as shown in Table 3.1.

**Table 3.1:** Input Parameters

| S\|N | Input Parameters | Variable | Specification (Value) |
|---|---|---|---|
| 1 | Size (in plots) | P | 1, 2, 3,...n |
| 2 | Location | L | semi-rural area, rural area, semi-urban area, urban area |
| 3 | Land Type | T | Swamp, dry land, upland, water front |
| 4 | Density | D | Low, Medium, High |
| 5 | Access Road | A | Yes or No |

The above parameters serve as input to the system and are used to derive the utility function which generates (predicts) the value of the specified land criteria.

### 3.2. Artificial Neural Network

Our predictive network is trained using past negotiation data with parameters from Table 3.1 to adequately capturethe dynamics of negotiations. The classical error back-propagation algorithm being the most popular learning technique for neural networks was adopted in our multilayer feed-forward neural network.The number of input neurons to the ANN model is equal to the number of independent variables, while the number of output neurons is equal to the number of functions being approximated by the model. Here, the input set value X denotes the land property criteria (size, location, land type, density and access road). Each node in the input layer has a signal Xi as network input, multiplied by a weight (W) value between the input layer and the hidden layer and produces Y which is the output (Land value).

An agent's utility can be calculated by summing up each preference (behaviour) weight value, expressed below:

$Z_t = \sum_{i-1}^{n} w_i e_i(X_i)$ (3.1)
Where,
$Z_t$ is the land specification
$X_i$ is the value of preference i in land specification scenario $Z_t$
$e_i$ is the evaluation function for preference $X_i$.
And output layer is given as:
$y_i = f(z)$ (3.2)
Where,
$f$ is an activation functionof Z.
The model needs to learn from both the input weights (priorities) $w_i$ and the evaluation function $e_i(X_i)$ as seen in equation (3.1) in order to adequately generate (predict) an opponent's preference (utility function). Evaluation functions assume a space between (0 and 1) such that the sum of weights equal 1 (normalization). In this network, the satisfaction level of an agent is given a value between 0 and 1. 0 signifies a rejected offer, while 1 signifies an accepted offer.
The normalization of weights is given by:
$w_i = \frac{r_i}{n(n+1)}$ (3.3)
Where,
$r_i$ is the rank of weight $Wi$ in a preference
n is the number of issues.
In training the model, the network is scanned to identify input neurons with weights of minimum errors using gradient descent. Iteratively, parameters (weights) are adjusted until one with the lowest error is reached. As network parameters are adjusted, the error decreases. Training stops when a laid down condition is met, this condition describes the training rate. A low learning rate is usually more accurate as a high rate can exceed the lowest point since the slope of the hill changes constantly. The training rate value of this model is fixed at 70. Graphically, from the first iteration, a downhill (descending) step is taken as specified by the gradient descent; same is done for subsequent iterations. This process continues until it reaches the floor of the graph or a point where a downhill movement or step can no longer be made (local minima or convergence). At this step, the value of parameters is the best for producing the desired output or result. Thus, completing the training.

### 3.3. Negotiation Mechanism

The interaction between negotiating parties is regulated by a negotiation protocol that defines the rules of how and when proposals can be exchanged. This negotiation mechanism adopts the alternating-offers protocol where negotiating parties exchange offers in turns. That is, the negotiators take turns selecting the next negotiation action until the negotiation is finished. Any agent can start the negotiation by making an offer. An agent can choose three actions as a reply, including accepting the offer; rejecting the offer with generating a new one to the opponent; and walking away (ending the negotiation). If negotiating agents have proposed an offer in turn which is not accepted, then a complete negotiation round is finished and a new round will begin. If it is not an offer, the negotiation has finished. An agent is allowed to make one offer in a round.
Negotiations do not run infinitely, hence the use of a deadline. Deadlines can be round-based or time-based. The round-based deadline is adopted by this negotiation mechanism. Here, negotiation halts once (Rounds > Deadline). A negotiation is said to have reached deadline if the number of negotiation rounds supersedes the maximum number of rounds (deadline) indicated in the negotiation protocol. A negotiation is terminated and declared inconclusive if negotiation deadline is reached. The deadline of this negotiation system is an integer value set as ten (10).
An agreement is reached if and only if an offer proposed by one agent is accepted by the other agent. Once an agreement is reached negotiation ends and a digital contract is drafted.
The notations applied in our ANN model is shown in Table 3.2.

**Table 3.2:** Basic Notations used

| S\|N | Notation | Meaning |
|---|---|---|
| 1 | Agt∪{a, b} | a finite set of agent names |
| 2 | a | Real Estate agent |
| 3 | b | Buyer |
| 4 | Offer | a set of offers over the negotiationdomain |
| 5 | Action⊆ Bid ∪{accept, reject, end} | a set of possible actionsthat can be taken during the negotiation where end denotes that the agent walks away. |
| 6 | Round | a round of negotiation |
| 7 | Round ⊆ $N^+$ | is the set of round numbers. Rounds are numbered from one (1) onwards, if i is the current round, then the next round is numbered (incremented as) i + 1 |
| 8 | Action(a, r) | this denotes an action agenta ∈ Agt took in a particular round |
| 9 | D : Round × $N^+$ | is a predicate that denotes whether or not the negotiation deadline has been reached and is given by (Round > Deadline) |

The algorithm for the negotiation process is given below:

**Algorithm: Negotiation**

```
1.  Start
2.  //Initialize land specification
3.  Require: (noOfPlot, location, landType, density, acessRoad)
4.  Initialize buyer land specification x (landValue)
5.  if (noOfPlot.equals("i") && location.equals("Li") && landType.equals("Ti") && density.equals("Di")
    && accessRoad.equals("Ai"))
6.          price =  x * i;
7.          int location = (int) (Li * price);
8.          int landType = (int) (Ti * price);
9.          int density = (int) (Di * price);
10.         int AccessRoad = (int) (Ai * price);
11.         calculate_price(" location + landType + density + AccessRoad);
12.         //Predict Land Value
13.         int predicted_value = calculated_price;
14. End if
15. action[accept, reject, end]
16. Agt[a, b]
17. Deadline = 10;
18. while round <= Deadline Do
19.         a.propose(offer);
20.         if (Agt)=/= null
21.               b.receive(action, offer);
22.               b.response(action);
23.               a.receive(action, offer);
24.               a.response(action);
25.         round += 1;
26.         if (b.response(action))== accept OR (a.respone(action)) == accept
27.                Price = offer;
28.                Print("Negotiation successful")
29.                Print(contract);
30.                Go to 46;
31.         else if (b.response(action))== reject
32.                b.propose(offer);
33.         else if (a.response(action)) == reject
34.                a.propose(offer);
35.         else if (b.response(action))== end OR (a.respone(action)) == end
36.                print("Negotiation ended by opponent./n");
37.                Go to 46;
38.         End if
39.         End if
```

```
          End if
          End if
          End if
    End Do
    If round > Deadline
          Print("Negotiation is inconclusive");
    End
```

### 3.4. Experimental Specification

In order to demonstrate the feasibility and effectiveness of an ANN-based predictive model we have used past data collected by a Real Estate Company in Port Harcourt, Nigeria. The collected dataset captures past negotiation data within the scope of this research. The negotiation case under study is a scenario where a real estate agent and a buyer enter into negotiation over a landed property. In the dataset, five issues were considered: size, location, land type, density and access road. The initial price of one (1) plot of land was in the range ₦ PP1to ₦ PP4 but the price varied based on property specifications.

### 3.5. Predicted Output

The outcome of the prediction is based on the land criteria (specification) selected by the buyer as shown in Table 3.3. These criteria have parameters containing values that serve as input to the matrices that forecast the price of the specified land property.

**Table 3.3:** Predicted Land Price Specification Output

| Input Parameters | Specification | Notation | Value |
|---|---|---|---|
| Size | Number of Plots | $P_i$ | $P\{1,2,\ldots n\}$ |
| Location | semi-rural area | $L_1$ | $0.35 * P_i$ |
| | rural area | $L_2$ | $0.25 * P_i$ |
| | semi-urban area | $L_3$ | $0.60 * P_i$ |
| | urban area | $L_4$ | $0.75 * P_i$ |
| Land Type | swamp | $T_1$ | $0.20 * P_i$ |
| | dry land | $T_2$ | $0.55 * P_i$ |
| | upland | $T_3$ | $0.35 * P_i$ |
| | water front | $T_4$ | $0.30 * P_i$ |
| Density | Low | $D_1$ | $0.30 * P_i$ |
| | Medium | $D_2$ | $0.55 * P_i$ |
| | High | $D_3$ | $0.65 * P_i$ |
| Access Road | Yes | $A_1$ | $0.80 * P_i$ |
| | No | $A_2$ | $0.20 * P_i$ |
| Predicted Price | If $A_1$ | Y | $[S_i- (L_i+ D_i)] + (T_i+ A_1)$ |
| | If $A_2$ | | $[S_i- (L_i+ D_i+ A_1)] + T_i$ |

## 4. Results and Discussion

It was observed that offers were accepted at a bargain greater or equal to 85% of the potential land price, while lower than 85% of the potential price were rejected. The trained neural network consists of the four negotiation cases, with each having two price offers (rejected and accepted offers and their rates when compared to the potential price of the property) as shown in Table 4.1, and 10 hidden nodes.

**Table 4.1:** Negotiations Based on Land Specification

| S\|N | Land Specifications (Criteria) | | | | | Potential Price | Offers (Buyer) (₦) | Offer Rate (%) | Label |
|---|---|---|---|---|---|---|---|---|---|
| | Size (No of Plots) | Location | Land Type | Density | Access Road | | | | |
| 1 | 1 | semi-rural | water front | medium | no | ₦ PP1 | ₦ R1 | 74% | Rejected |
| | | | | | | | ₦ A1 | 90% | Accepted |
| 2 | 1 | rural | dry land | low | no | ₦ PP2 | ₦ R2 | 60% | Rejected |
| | | | | | | | ₦ A2 | 93% | Accepted |
| 3 | 1 | Semi-urban | swamp | low | yes | ₦ PP3 | ₦ R3 | 73% | Rejected |
| | | | | | | | ₦ A3 | 85% | Accepted |
| 4 | 1 | urban | swamp | high | yes | ₦ PP4 | ₦ R4 | 75% | Rejected |
| | | | | | | | ₦ A4 | 88% | Accepted |

To further understand the negotiation offer acceptance and rejection rate shown in Table 4.1, a line graph is depicted in Figure 4.1 showing the highest and lowest rates of both accepted and rejected offers.

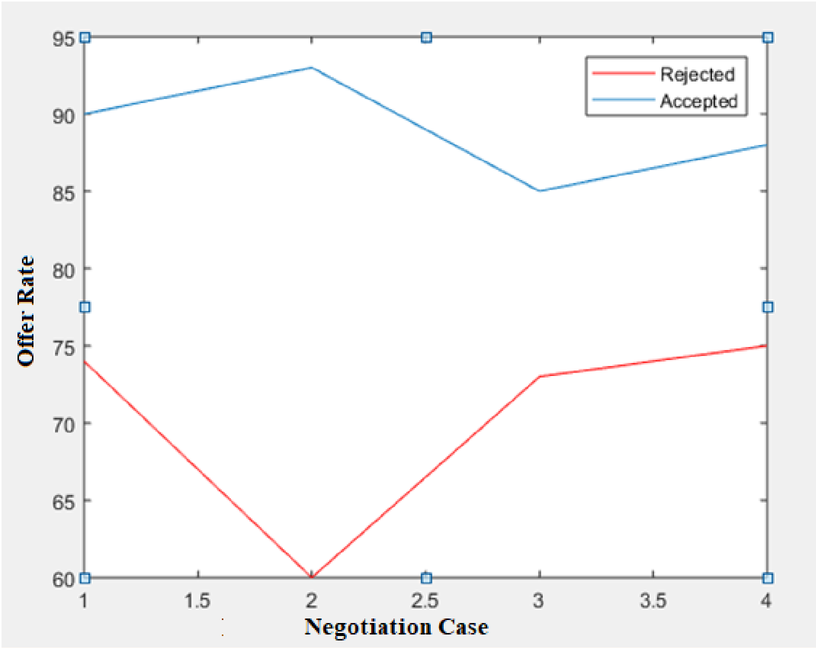


*Figure 4.1: Line Graph Representing Negotiation Outcome Based on Land Specification*

The training of the neural network was executed and the performance of the model was measured using Mean Square Error (MSE). The training error was 0.0002711 at 70 Epochs (iteration). The testing error was 0.0001032 and validation was 0.00010326 as shown in Figure 4.2. The essence of neural network was to predict the factors that influenced price negotiation outcome such as size, land location, land type, density and access road. Also, to capture the non-linear verdict involved in price negotiation.

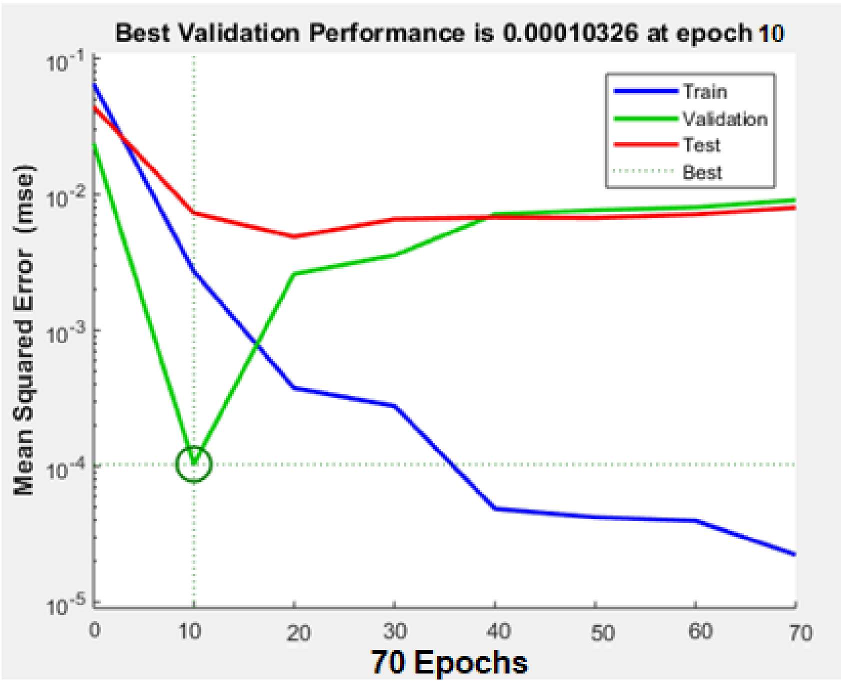


***Figure 4.2: Negotiation Trained in ANN***

When compared with Regression Model, the neural network has a better MSE as shown in Table 4.2 below. The decrease in MSE shows that the non-linear neural network model is able to produce a better fit and forecast of the data when compared to the regression model.A Regression Analysis was carried out on both models using the formulae for the equation of a straight line:

Y = mx + C.

$$m = \text{slope} = \frac{\Delta y}{\Delta x} = \frac{y2-y1}{x2-x1} \quad (4.1)$$

Where, y = mean negotiation rate, x = price of land and C = is a constant which is the y-axis intercept (first value of the negotiation rate).

**Table 4.2:** Model Comparison

| Predictive Model | Training MSE | Testing MSE | Regression |
|---|---|---|---|
| Neural Network | 0.0002711 | 0.0001032 | 0.9200000 |
| Regression Model | N/A | N/A | 1.2600000 |

## 5. Conclusion

This research presents a neural network-based algorithm for predicting the opponent's offer during the process of negotiation. The algorithm is embedded in an electronic negotiation system to guide negotiating agents in making offers during negotiation. The results of the neural network analysis show that this model can exhibit interesting negotiation strategies and, at the very least, provide a negotiator, useful information.

One potential limitation of this research relates to the generalizability of the model. Specifically, our model was applied to a particular domain, and the accuracy of its predictions and recommendations may be less adequate for other domains. Thus, in order to ensure genericity of the model, future research should venture into different negotiation domains and cases.

## References


[1]. Matwin, S., Szapiro, T., & Haigh, K. (1991). Genetic Algorithms Approach to a Negotiation Support.

[2]. Kersten, G. E., & Lo, G. (2003). Aspire: An Integrated Negotiation Support Systems and Software Agents for E-business Negotiations. International Journal of Internet and Enterprise Management, 1(3)293-315.

[3]. Mohammad Irfan Bala, Sheetal Vij, Debajyoti Mukhopadhyay (2014). Intelligent Agent for Prediction in E- Negotiation. Article in Advances in Intelligent Systems and Computing, Researchgate.

[4]. Hornik, K. (1991). Approximation Capabilities of Multilayer Feed-forward Networks. Neural Networks, 4(2), 251-257.

[5]. J. Brzostowski and R. Kowalczyk "Adaptive negotiation with on-line prediction of opponent behaviour in agent-based negotiations," Proc. IEEE/WIC/ACM Int. Conference on intelligent Agent Technology, Washington, DC: IEEE Computer Society, 2006, pp. 263-269.

[6]. Faratin, P., Sierra, C., & Jennings, N. R. (2002). Using Similarity Criteria to Make Issue Trade-Offs in Automated Negotiations. Artificial Intelligence, 142(2), 205-237.

[7]. Zeng, D., & Sycara, K. (1998). Bayesian Learning in Negotiation. International Journal of Human-Computer Studies 48, 125-141.

[8]. P. Faratin, C. Sierra, and N.R. Jennings, (1998). Negotiation decision functions for autonomous agents. Int. Journal of Robotics and Autonomous Systems, vol. 24, no. (3 - 4), pp. 159-182.

[9]. Hornik, K. (1991). Approximation Capabilities of Multilayer Feed-forward Networks. Neural Networks, 4(2), 251-257.