OPEN ACCESS
International Journal of Innovations in Science & Technology
Check for updates
IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130
Recognized Journal

Double Auction Used Artificial Neural Network in Cloud Computing

Original Article

Muhammad Adeel Abbasa[1], Zeshan Iqbal[1]
[1]Department of Computer Science University of Engineering and Technology Taxila, Pakistan
Corresponding Email: rajaadeelabbas1122@gmail.com, zeshan.iqbal@uettaxila.edu.pk

Citation | I. Z. Abbasa. A. M, "Double Auction used Artificial Neural Network in Cloud Computing," Int. J. Innov. Sci. Technol., Special Issue, pp. 65-76, 2022.
Received | June 5, 2022; **Revised** | June 15, 2022; **Accepted** | June 28, 2022; **Published** | June 30, 2022.
DOI| https://doi.org/10.33411/IJIST/2022040506

Double auction (DA) algorithm is widely used for trading systems in cloud computing. Distinct buyers request different attributes for virtual machines. On the other hand, different sellers offer several types of virtual machines according to their correspondence bids. In DA, getting multiple equilibrium prices from distinct cloud providers is a difficult task, and one of the major problems is bidding prices for virtual machines, so we cannot make decisions with inconsistent data. To solve this problem, we need to find the best machine learning algorithm that anticipates the bid cost for virtual machines. Analyzing the performance of DA algorithm with machine learning algorithms is to predict the bidding price for both buyers and sellers. Therefore, we have implemented several machine learning algorithms and observed their performance on the bases of accuracy, such as linear regression (83%), decision tree regressor (77%), random forest (82%), gradient boosting (81%), and support vector regressor (90%). In the end, we observed that the Artificial Neural Network (ANN) provided an astonishing result. ANN has provided 97% accuracy in predicting bidding prices in DA compared to all other learning algorithms. It reduced the wastage of resources (VMs attributes) and soared both users' profits (buyers & sellers). Different types of models were analyzed on the bases of individual parameters such as accuracy. In the end, we found that ANN is effective and valuable for bidding prices for both users.

Keywords: Cloud Computing, Double Auction Algorithm, Supervised Machine Learning, Regression Problem, Artificial Neural Network

Acknowledgment. We acknowledge that our work is not published by someone else. We would like to thank all our colleagues who supported us, and we are also thankful to all the authors who contributed to this research.

Conflict of Interest: The author(s) declare that the publication of this article has no conflict of interest.

Project details. This is a research work under the supervision of Dr. Zeshan Iqbal.

Author's Contribution. All of my co-authors contributed to this research in the form of data collection, write-up, and improvement.

Introduction

The cloud market allows its user to access the network on request and exchange configurable resources over the cloud network by using the internet [1]. In the cloud market, different sellers offer a variety of services to their users and charge according to their demand for a specific period [2]. CDARA act as an agent between customers and providers; furthermore, it fulfills their demands as per their request [3]. CustomerCustomers request virtual machines and their attributes are usually set by cloud providers. The attributes include several factors like CPU, RAM, HDD, bandwidth, and time. CDARA algorithm assigns the resources according to their demand and as well as their bids price that is suggested by the providers [4]. The major issue in DA algorithm is getting a good quality of service (QoS) at a reasonable price and finding the optimal bidding price according to its demand. , A variety of learning methods have been used to improve the performance of DA algorithm, but no one worked on optimal bidding price. The main problem occurs when multiple bid prices come in the auction algorithm, and we need to achieve equilibrium prices during the algorithm's execution[5]. The resource allocation process depends upon the bidding price. If the customer bid price is less than the seller's bid price, no one gets resources in terms of virtual machines. All customer's and provider's bids are arranged in descending order because a higher bid density has a chance to get resources first. All resource allocation process is done on bid density [4]. Higher bid density provides high profit for providers, and it is good for their company, revenue always runs businesses, and in return, they provide us well services (VMs). DA also provides the freedom to get resources from multiple service providers at a lower price. If one provider is selling the services at a higher price, on the other side, another provider is selling at a low price. DA automatically allocates a resource from the lower service provider to the higher bid density user. The bidding process decides which customer gets a resource. The main task is DA is resource allocation, and resource allocation depends on the bidding process, so our main focus is on the bidding process. If we work on the bidding process, we can improve the performance of DA, and it will generate better results as compared to previous algorithm. We will add learning process for bidding policy because nowadays competition is high and we need to achieve Qos. Predicting the price for VMs for customer is major problem so we will use learning algorithm to predict the price.

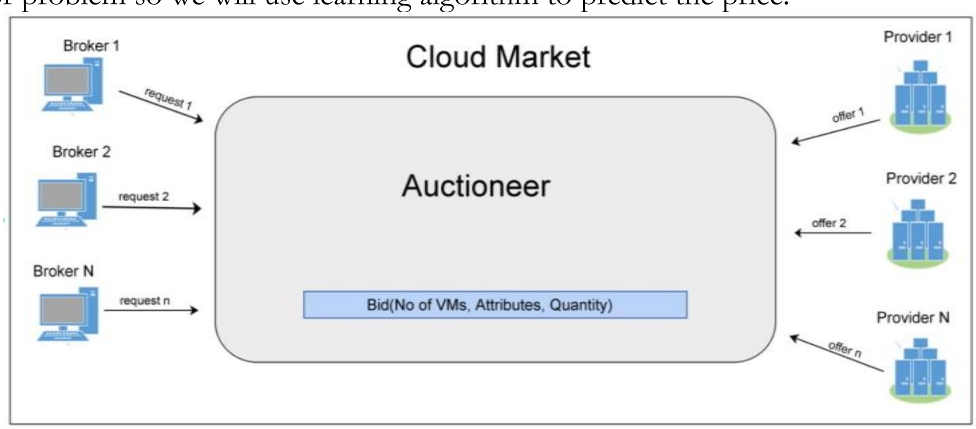

Figure 1: Cloud Environment

In Figure 1, the cloud computing structure is discussed as in other researchses [6], and it also explains how DA algorithm works as an agent between users and providers. Auctioneer takes input from customers and providers and compares their offers and demands according to their bid cost. User requests for virtual machine (VM) that consists of different attributes such as Hard Disk (HDD), Random Access Memory (RAM), Bandwidth (BW), Central

Processing Unit (CPU), and Time. Provider offers a number of VMs and demanded prices. In figure 1 broker acts as a user requesting a virtual machine and its bid price. In the end, it assigns virtual machines to the user at their request.

On the other hand, sellers get the price and provide the services offered to the auction algorithm. Here auctioneer represents as DA algorithm. We need to analyze different machine learning algorithms for predicting the price of VMs. If we use the predicted price of VM in DA algorithm, we can get better results than the simple DA algorithm. The machine learning technique is very useful for one-sided algorithms [7][8]. In the next section, we will discuss different resource allocation algorithms and machine learning algorithms for predicting the bidding price.

Related Work

In this section, we have studied resource allocation and its issues in a cloud environment and also comprehensively analyzed the learning mechanism (LM) in this literature. Trading of virtual machines CDARA is the problem by referring to the cloud market and profits for both sides of customer and sellers [3]. The proposed model has CDARA that raises the income of buyers and sellers. Resource Allocation Problem (RAP) was used as an optimization problem in an auction where N customers and sellers offer their bids according to requirements. However, the main issue is whether we can increase providers' profit and social welfare without increasing the number of total resources [9]. The DA algorithms had a paramount trading system to obtain competitive equilibrium prices and how they worked under different conditions during the bidding process. Linear regression was used for resource allocation, and logistic regression had used to confirm if the resource was allocated or not all things were done with the help of machine learning algorithms but also to examine the multi-dimensional problem in cloud resources distribution system [10]. Smith implemented the DA algorithm and achieved average price (AP) by perfuming neoclassical in his experiment and also predicting the equilibrium price with partial data [11]. The bidding policy system uses reinforcement learning (RL) in DA, and this system assists in air conditioning and heating ventilation system [12]. The bidding policy did not require a special framework based on showing attributes and conversion of the indoor temperature of a building and trading payment system [13]. In 2020 DA was used as resource allocation in the blockchain and offloading method used for computing in mobiles. Low memory, limited storage, and computation capacity are low in mobile devices. Due to this reason, we can perform huge tasks like cryptocurrency mining due to the low power of resources. To avoid this problem, G-TRAP had introduced in edge computing and cloud computing. Offloading task performed by DA algorithm where it takes from cellular phone sends to servers were computation process and give return to auction algorithm[14]. Offloading task was requested by mobile devices and services are provided by datacenters, all these services run in the application where blockchain is used, but the problem is they only focus on resource distribution and do not involve a learning mechanism [15]. The online gaming platform has a consensus problem where all the users want to get specific resources with partial data [16]. For bidding policy, iterative learning was used in DA algorithm, where they got an equilibrium price at the end [17]. Several learning techniques have been proposed DA algorithm with equilibrium and varying policies to depict the asymmetric data for sellers and buyers. Different genetics [18] and RL [9] were used. Our need was to find a general learning method that finds the planning scheme in cloud computing by using DA algorithm and increased knowledge. The best way to fulfill the conditions in the learning framework is Experience Weight Attraction (EWA) which combines the attributes of belief learning and RL. EWA was used in DA algorithm that was the combination of belief learning and RL, then altered EWA in an appropriate learning system by extracting discrete learning behavior [19]. This proposed model was economically good and guided both-sided participants; this effort is closed to our work. Several ML algorithms have been used in past years, but some worked on limited rows in the data set.

A regression tree is used for the optimization of bidding policy and linear programming to increase the performance of CDARA [20]. The problem in CDARA is the bidding policy; we need to increase throughput instead of focusing on the algorithm's execution time. The regression tree creates problems when we increase the data size and change the output when new data is added [20]. And EWA did not perform well in a small size data set [15] because it is reinforcement learning. It always works on feedback, and it improves itself by taking responses from the environment, so that is why it did not perform well. Deep Deterministic Policy Gradient (DDPG) is used for bidding policy in the whole sales market for predicting the energy bid price, but we are dealing with cloud services. They used large data for the learning process because it worked on feedback [21]. Q learning is applied in CDARA by using a bidding policy [22], but the problem is that a small dataset could not help in Q learning. This learning always requires feedback with limited rows Q learning system, which cannot provide the best results. Bidding police are discussed for cloud instances, but the problem is that it is solely on assumptions. Real-time entries are not discussed. Therefore, cloud instances require a real-time environment and can handle large amounts of data instead of assumption entries [23]. CDARA needed the best learning algorithm that could handle massive datasets and maximize the performance in resource allocation, in addition to finding optimal bidding costs. Therefore, we are looking for the best ML algorithm which forecasts the bidding cost in the manner of VMs attributes. In addition, we need to use different traditional and advanced machine learning algorithms to predict the final bidding cost for each customer and provider.

Materials and Method

Combinatorial Double Auction (CDA)

CDA consists of the allocation model, which allocates the virtual machines and their proposed offers in the form of bids that require groups of different resource attributes [4]. The bids are calculated given the below format.

Bid (buyer) = bidPerCPU * CPU + bidPerMemory * Memory + bidPerBandwidth * Bandwidth + bidPerStorage * Storage (1)

Bid (seller) = bidPerCPU *CPU + bidPerMemory * Memory+bidPerBandwidth * Bandwidth +bidPerStorage * Storage (2)

Winner Selection

CDA selects the resources based on equations (4) and equation (6). If the bid density offered by customer equation (4) is greater or equal to provider bid equation (6), then the resource assigned to the customer is reasonable at the final trade price (FTP) equation (11). The total number of customers and their demands [3].

$$M_i = \sum_{k=1}^{l} (a_k^i * q_k^i) \quad (3)$$

Bid density of buyer.

$$bd_i = \frac{b_i}{\sqrt{M_i}} \quad (4)$$

Total number of providers and their demands

$$M_j = \sum_{k=1}^{l} (a_k^j * q_k^j) \quad (5)$$

Bid density of seller.

$$bd_j = \frac{b_j}{\sqrt{M_j}} \tag{6}$$

Pricing Model

Assigning of resources is determined by the paid the price by the customer by utilizing the pricing model [3]. In the cloud market, this model is best for customers and providers to encourage participation. Sum of demanded attributes for a virtual machine by the customer [3]. Table 1 shows the attributes description, and each attribute defines a distinct role in each equation.

$$tq_i = \sum_{k=1}^{l} q_k^i$$

The average bid cost for the customer is computed by the total requested resources.

$$ap_i = \frac{b_i}{tq_i} \tag{7}$$

Sum of total resources offered by the seller[3].

$$tq_j = \sum_{k=1}^{l} q_k^j$$

The average bid cost for the provider is computed by the total requested resources.

$$ap_j = \frac{b_i}{tq_j} \tag{8}$$

Concluding TP of customer i and provider j is calculated based on the average price of customer and provider.

$$atq_j^i = \left(\frac{ap_i + ap_j}{2}\right) \tag{9}$$

Customer i required to pay a specified price to provider j, and paid the price is calculated based on allocated items.

$$ap_j^i = atq_j^i * ap_j^i \tag{10}$$

The final cost is assigned to, the customer by the provider as the sum of the paid the price, and all this process is done after the negotiation between the customer and provider [3].

$$Tp_j^i = \sum_{k=1}^{l} py_k^i \tag{11}$$

Table 1: Attribute's Description

Attributes	**Description**
a_k^i	Attribute of VMs requested by buyers.
q_k^i	Number of VMs requested by buyers.
b_i	Bid offer by buyers
a_k^j	Attributes of VMs offered by sellers.
q_k^j	Number of VMs offered by sellers.
b_j	Bid offer by sellers.

Utility Function (UF):

UF is calculated by offered bid price minus FTP. If the buyer's bid price is lower than offered bid price by the sellers, then the utility becomes zero [24].

$$UF = Bid - FTP \tag{12}$$

Methodology

Figure 2 depicts the pictorial representation of our methodology and shows the six steps involved in it. Firstly CDARA had executed recurrently over the CloudSim platform

[25] due to the need for large data for the learning process and the creation of the dataset. Then note down all demands and their respective bids that the customers submitted. However, we also recorded the offers that were offered by the sellers and their respective bids. Demands and offers contain CPU, RAM, BW, and time. With the help of these, we have noticed the final cost for both customers and providers. In the next phase, we analyzed the data gathered by CDA algorithm's output after successfully acquiring the resources. We have created the dataset on the bases of CDA output. A feature selection is made on the dataset due to the need for improvement in the performance of the auction algorithm. These features have different domains and ranges defined in table 2. Several ML algorithms (Support Vector Machine [26], Random Forest Regression [27], Decision Tree [28], and Gradient Boost Regression [29]) have been implemented with the distinct size of the dataset. Few of them have provided the best results, but dataset size is limited; however, Artificial Neural Network [30] has provided the best result among all learning algorithms with a large dataset. Subsequently executing the learning algorithm, we observe the performance of DA on the basis of loss function and accuracy. ANN produced the best result among all learning algorithms and gave 97% on a large dataset size.

Figure 2: Overview of Methodology

DataSet Description

Our dataset has dependent variables (final trade price) and independent variables (CPU, RAM, HDD, BW, Time, User Bid, and Provider Bid) total data set contains eight variables and 40,000 entries. Each feature has its domain and range, CPU domain is 200 and range is 1000, RAM domain is 256 and range 2048, HDD domain is 12000 and range is 48,000, BW domain is 120 and range is 960, and time domain is 10 and range is 60. All offers and demands exist within these limits.

Table 2: Domain & Ranges

Features	Minimum	Maximum	Mean	Standard Deviation
CPU (MHz)	200	1000	600	236.64
RAM (MB)	256	2048	960	686.32
BW (Byte/sec)	120	960	468	286.24
Time (T/24*hr)	10	60	35	17.07
HDD (MB)	12000	48000	3000	13416.57
Bid Price Customer($)	0.159	1.006	0.556	1.88
Bid Price Provider($)	0.058	0.401	0.203	0.078

Final Trade Price($)	0.108	0.703	0.379	0.116

Result and Discussion

ANN has been implemented with different types of dense layers (DL) by using Keras API [31]. First of all, we have loaded 40,000 rows of the dataset that was developed based on DA algorithm output, and each row contains a dependent and independent variable. The final trade price (FTP) depends upon the CPU, RAM, HDD, BW, and time. FTP defined the UF, and this function tells the profit to providers. Dataset is divided into two sections; training data is labeled in one section, and the second one is labeled by the testing dataset. For the learning process, we have used a training dataset, and for the validation process, we have used test data. The training data set has 70%, and the remaining 30% has testing data of the dataset. Testing and training data were loaded into data frames and mix them for further processing. Different features were normalized by calculating their standard deviation and mean [32]. Preprocessing is done by the normalization technique, and this technique helps us to develop our model. We construct three models using different hidden layers to forecast the FTP and compute the loss function. Our target is to forecast the real cost of the customer and provider's bid cost allocated by the DA. We construct the sequential model by using different inputs (CPU, RAM, HDD, and BW) with the help of DL. In the initial phase, a few layers were used to predict the FP was not accurate, and after that, we added more layers with the help of the regulator, and finally, we reached the optimal solution. Nodes are significant while predicting the FTP, we categories hidden layers into three sections (small, medium and large). Initially we used 13 hidden layers and epoch size was 40 to 80 but error rate is so high so we increased the numbers of hidden layers. We have used 42 hidden layers to reduce the error rate. This result was good as compare to initial stage but not optimal so we need to find optimal solution so we again increase the number of hidden layers at the end, found optimal solution by using 87 hidden layers. Figure 3 depicts the final stage of training by using Mean Square Error as cost function; it tells us about the average loss value of entire dataset.

Cost function

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{13}$$

$$\hat{y}_i = \text{predicated Value } F(x)$$

$$y_i = \text{oberved value}$$

$$n = \text{numbers of sample in } y$$

Mean square error [33] helped us reach an optimal point where we have five main features such as CPU, RAM, HDD, BW, and time; with the help of matplotlib [34] library to draw the best-fitted line of our model by using ANN model. Figure 4 shows the best-fitted line. On x-axis, the independent and y-axis dependent variable is defined.

Samimi et al. [3] performed a simple experiment with 20 customers and 7 providers in an auction algorithm. Each customer requested his demands according to its requirement and submitted their respective bids for the virtual machine as well. On the other side, providers demanded their offers in the auction without using any technique like an integer program or any learning process for bid values (Virtual Machine). At the end of the auction, only 9 customers got the resources, as shown in figure 4 below.

S. A. Tafsiri et al. [4] implemented integer programming to increase the success ratio for winner users. They performed with 20 users and 7 providers in the auction algorithm, and they got better results as compared to a simple auction algorithm without using integer programming. With integer programming, 14 got the resources that were much better than previous results. But it can still improve, so we decided to introduce the learning mechanism to increase the success ratio in a double auction.

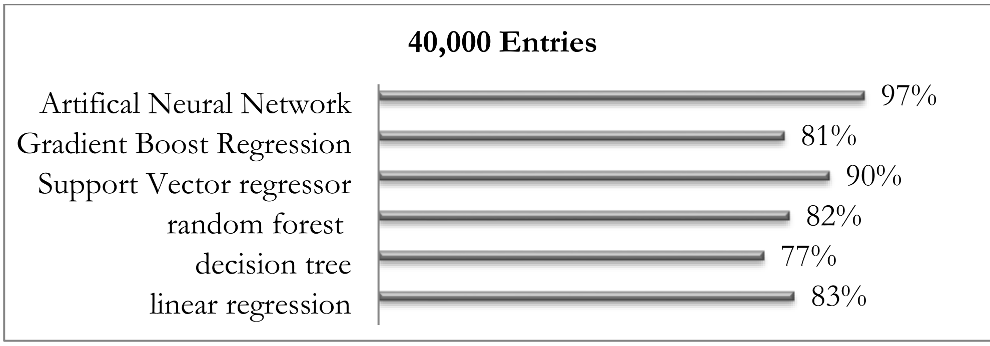

Figure 3: Accuracy over 40,000 entries

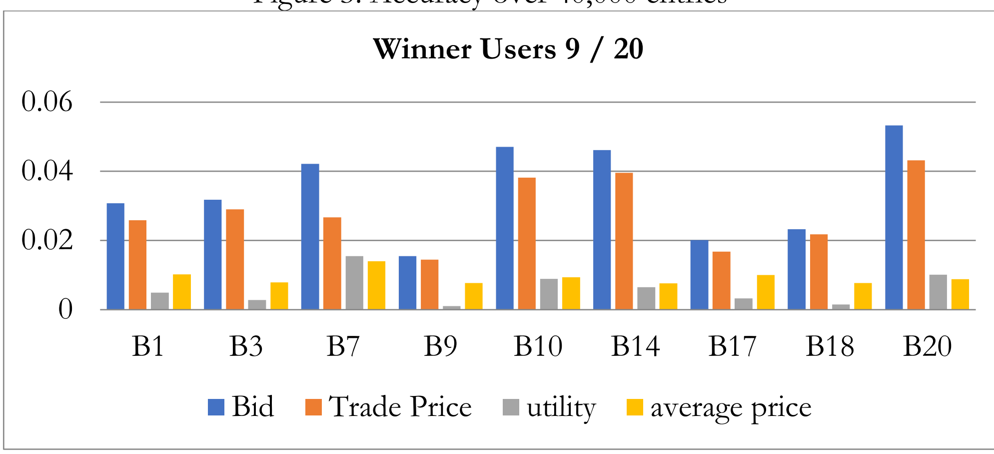

Figure 4: Simple Double Auction Algorithm

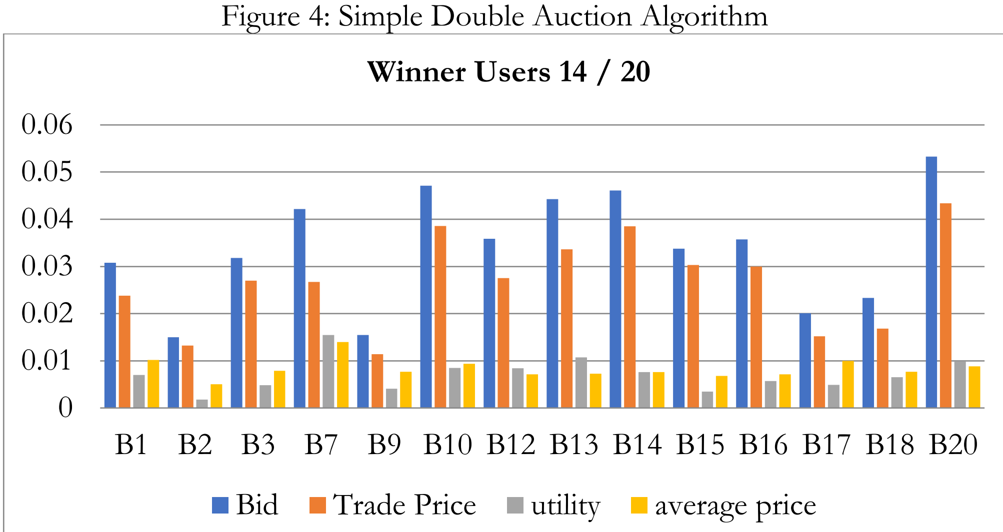

Figure 5: Integer Programming

Double Auction with Learning Mechanism:

DA algorithm is implemented by using predicted bid values with different learning algorithms, and without LM at the end, we observed that LM performed best. In addition, ANN provided the foremost results among all other learning algorithms. We experimented with twenty users who requested resources (VMs) and seven providers who offered their services (VMs), and we observed that the probability loss of instances is reduced by using ANN predicted values.

In figure 6, we have used ANN predicted values in the double auction, and we got tremendous results. We introduced 20 users and 7 providers in the above experiment, and both users and providers used ANN values. After using the ANN values for bid, 18 users got the resources that are much better than the simple double auction and integer programming.

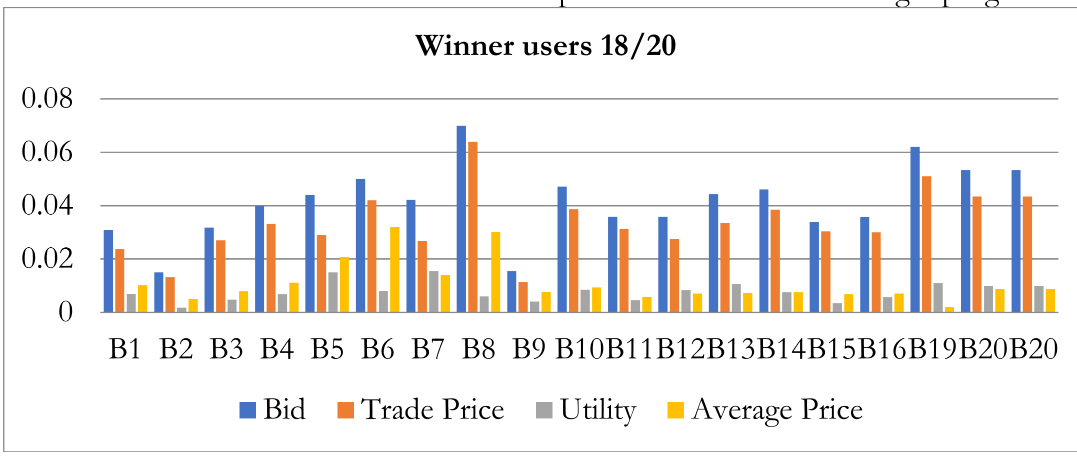

Figure 6: DA used ANN Values

In figure 6, there are three main attributes such as TP, UF, and the average cost of every user got the resources in DA and sky-craping bid price by B8 at 0.707, on the flip side, the minimum bid price by B2 at 0.0155, furthermore towering TP obtained by B19 at 0.0518 and rock bottom TP was achieved by B9 at 0.0115.

Conclusion

Various supervised learning algorithms had different results in DA by using bidding strategy, but in this paper, ANN brings off tremendous results for forecasting the bid price for customers and providers as reported by their requests and demands. This algorithm has worked best in DA by providing the bidding predicted value and improving DA's performance by using LM. In addition, ANN provided supreme results in *comparison* to traditional ML algorithms. Classical machine learning algorithms function well on less number of entries in the dataset, and initially, it contained 500 entries, but later on, we increased the size of the dataset than we observed with 16,000 entries. Unfortunately, some algorithms work well, but some are not. Data is increasing daily, so we need to develop a large dataset to observe the behavior of learning algorithms with 40,000 entries because the large size of the dataset performed well with ANN model.

On the other hand, decision tree, linear regression, random forest, support vector, and gradient boosting regression worked well but with limited dataset size. ANN can deal with a large-size dataset. Even if we increase the size of our dataset, the performance of DA will not affect, and it deals easily with all kinds of issue that comes due to data. In the future, we can add more attributes to our dataset, like the operating system and model of CPU, and we can also increase the domain and ranges of CPU, RAM, HDD, BW, and time. Online gaming can utilize this concept where customers demand different attributes; furthermore, all types of auctions can use this concept to find optimal bidding policies such as web store, Live Auctioneers, and eBid.

References

[1] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015, doi: 10.1016/J.IS.2014.07.006.

[2] A. Youssef, A. Almishal, and A. E. Youssef, "Cloud Service Providers: A Comparative Study," *www.ijcait.com Int. J. Comput. Appl. Inf. Technol.*, vol. 5, no. April, pp. 2278–7720, 2014.

[3] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing," *Inf. Sci. (Ny).*, vol. 357, pp. 201–216, Aug. 2016, doi: 10.1016/J.INS.2014.02.008.

[4] S. A. Tafsiri and S. Yousefi, "Combinatorial double auction-based resource allocation mechanism in cloud computing market," *J. Syst. Softw.*, vol. 137, pp. 322–334, Mar. 2018, doi: 10.1016/J.JSS.2017.11.044.

[5] K. M. I. Baig. M. S, Imran. A, Yasin. A. U, Butt. A. H, "Natural Language to SQL Queries: A Review," *Int. J. Innov. Sci. Technol.*, vol. 4, no. 1, pp. 147–162, 2022.

[6] S. W. and I. M. Sehir e N, Shehzad M.A, Aslam M.S, "Optimize Elasticity in Cloud Computing using Container Based Virtualization," *Int. J. Innov. Sci. Technol.*, vol. 2, no. 1, pp. 1–16, 2020.

[7] P. Hummel and R. Preston McAfee, "Machine learning in an auction environment," *J. Mach. Learn. Res.*, vol. 17, pp. 1–37, 2016.

[8] Q. M. A. and M. . Kiran. I, Siddique. Z, Butt. A. R, Mudassir. A. I, "Towards Skin Cancer Classification Using Machine Learning and Deep Learning Algorithms: A Comparison," *I nternational J. Innov. Sci. Technol.*, vol. 3, no. special issue, pp. 110–118, 2021.

[9] A. Darmann, U. Pferschy, and J. Schauer, "Resource allocation with time intervals," *Theor. Comput. Sci.*, vol. 411, no. 49, pp. 4217–4234, 2010, doi: 10.1016/j.tcs.2010.08.028.

[10] J. Zhang, N. Xie, X. Zhang, K. Yue, W. Li, and D. Kumar, "Machine learning based resource allocation of cloud computing in auction," *Comput. Mater. Contin.*, vol. 56, no. 1, pp. 123–135, 2018, doi: 10.3970/cmc.2018.03728.

[11] V. L. Smith, "An Experimental Study of Competitive Market Behavior," *https://doi.org/10.1086/258609*, vol. 70, no. 2, pp. 111–137, Oct. 2015, doi: 10.1086/258609.

[12] M. Van Otterlo and M. Wiering, "Reinforcement learning and markov decision processes," *Adapt. Learn. Optim.*, vol. 12, pp. 3–42, 2012, doi: 10.1007/978-3-642-27645-3_1/COVER/.

[13] S. Li, W. Zhang, J. Lian, and K. Kalsi, "Market-Based Coordination of Thermostatically Controlled Loads - Part I: A Mechanism Design Formulation," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1170–1178, Mar. 2016, doi: 10.1109/TPWRS.2015.2432057.

[14] R. A. U. Ullah. A, Qayyum. H, Hassan. F, Khan. M. k, "Comparison of Machine Learning

Algorithms for Sepsis Detection," *Int. J. Innov. Sci. Technol.*, vol. 4, no. 1, pp. 175–188, 2022.

[15] L. Li, Y. Li, and R. Li, "Double Auction-Based Two-Level Resource Allocation Mechanism for Computation Offloading in Mobile Blockchain Application," *Mob. Inf. Syst.*, vol. 2021, 2021, doi: 10.1155/2021/8821583.

[16] M. H. Degroot, "Reaching a consensus," *J. Am. Stat. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974, doi: 10.1080/01621459.1974.10480137.

[17] H. Dawid, "On the convergence of genetic learning in a double auction market," *J. Econ. Dyn. Control*, vol. 23, no. 9–10, pp. 1545–1567, Sep. 1999, doi: 10.1016/S0165-1889(98)00083-9.

[18] Y. Zeng, "Research on Social Talent Governance Based on Genetic Algorithm," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/6288679.

[19] Q. Yu, Y. Liu, D. Xia, and L. Martinez, "The Strategy Evolution in Double Auction Based on the Experience-Weighted Attraction Learning Model," *IEEE Access*, vol. 7, pp. 16730–16738, 2019, doi: 10.1109/ACCESS.2019.2895875.

[20] S. Verwer, Y. Zhang, and Q. C. Ye, "Auction optimization using regression trees and linear models as integer programs," *Artif. Intell.*, vol. 244, pp. 368–395, Mar. 2017, doi: 10.1016/J.ARTINT.2015.05.004.

[21] H. Xu, H. Sun, D. Nikovski, S. Kitamura, K. Mori, and H. Hashimoto, "Deep Reinforcement Learning for Joint Bidding and Pricing of Load Serving Entity," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6366–6375, Nov. 2019, doi: 10.1109/TSG.2019.2903756.

[22] X. Sui and H. F. Leung, "A Q-learning based adaptive bidding strategy in combinatorial auctions," *ACM Int. Conf. Proceeding Ser.*, pp. 186–194, 2009, doi: 10.1145/1593254.1593283.

[23] J. Song and R. Guerin, "Pricing and bidding strategies for cloud computing spot instances," *2017 IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2017*, pp. 647–653, Nov. 2017, doi: 10.1109/INFCOMW.2017.8116453.

[24] D. Minarolli and B. Freisleben, "Utility-based resource allocation for virtual machines in cloud computing," *Proc. - IEEE Symp. Comput. Commun.*, pp. 410–417, 2011, doi: 10.1109/ISCC.2011.5983872.

[25] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," pp. 1–9, 2009.

[26] W. S. Noble, "What is a support vector machine?," *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, doi: 10.1038/NBT1206-1565.

[27] M. R. Segal, "Machine Learning Benchmarks and Random Forest Regression," *Biostatistics*, no. May 2003, pp. 1–14, 2004.

[28] G. K. F. Tso and K. K. W. Yau, "Predicting electricity energy consumption: A comparison of

regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761–1768, Sep. 2007, doi: 10.1016/J.ENERGY.2006.11.010.

[29] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, no. DEC, p. 21, 2013, doi: 10.3389/FNBOT.2013.00021/BIBTEX.

[30] S.-C. Wang, "Artificial Neural Network," *Interdiscip. Comput. Java Program.*, pp. 81–100, 2003, doi: 10.1007/978-1-4615-0377-4_5.

[31] T. B Arnold, "kerasR: R Interface to the Keras Deep Learning Library," *J. Open Source Softw.*, vol. 2, no. 14, p. 296, 2017, doi: 10.21105/joss.00296.

[32] S. Gopal, K. Patro, and K. Kumar Sahu, "Normalization: A Preprocessing Stage," *IARJSET*, pp. 20–22, Mar. 2015, doi: 10.48550/arxiv.1503.06462.

[33] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Clim. Res.*, vol. 30, no. 1, pp. 79–82, Dec. 2005, doi: 10.3354/CR030079.

[34] S. Tosi, "Matplotlib for Python Developers," *Packt Publ.*, p. 308, 2009.