Modeling Opponent Strategy in Multi-Issue Bilateral Automated Negotiation Using Machine Learning

Fatemeh Mohammadi-Ashnani
MSc Graduate in Information Technology Engineering, Cyberspace Research Lab, Department of Computer Engineering, Faculty of Engineering, College of Farabi, University of Tehran, Iran
mohammadi.fateme@ut.ac.ir

Zahra Movahedi, Kazim Fouladi-Ghaleh*
Assistant Professor, Cyberspace Research Lab, Department of Computer Engineering, Faculty of Engineering, College of Farabi, University of Tehran, Iran
{zmovahedi, kfouladi}@ut.ac.ir

Received: 2020/10/23 Revised: 2021/03/06 Accepted: 2021/03/30

Abstract— **With the emergence of the World Wide Web, Electronic Commerce (E-commerce) has been growing rapidly in the past two decades. Intelligent agents play the main role in making the negotiation between different entities automatically. Automated negotiation allows resolving opponent agents' mutual concerns to reach an agreement without the risk of losing individual profits. However, due to the unknown information about the opponent's strategies, automated negotiation is difficult. The main challenge is how to reveal the optimal information about the opponent's strategy during the negotiation process to propose the best counter-offer. In this paper, we design a buyer agent which can automatically negotiate with the opponent using artificial intelligence techniques and machine learning methods. The proposed buyer agent is designed to learn the opponent's strategies during the negotiation process using four methods: "Bayesian Learning", "Kernel Density Estimation", "Multilayer Perceptron Neural Network", and "Nonlinear Regression". Experimental results show that the use of machine learning methods increases the negotiation efficiency, which is measured and evaluated by parameters such as the rate agreement (RA), average buyer utility (ABU), average seller utility (ASU), average rounds (AR). Rate agreement and average buyer utility have increased from 58% to 74% and 90% to 94%, respectively, and average rounds have decreased from 10% to 0.04%.**

Keywords— Multiagent System; Automatic Negotiation, Machine Learning; Opponent Strategy Learning; Opponent's Modeling; E-Commerce; Bayesian Learning; Kernel Density Estimation; Artificial Neural Network.

1. INTRODUCTION

Electronic Commerce (also known as E-commerce) consists of the purchasing and selling products or services through electronic systems like the web [1, 2]. Buyers and sellers are the two main entities that play an essential role in such a system. They attempt to negotiate with each other on single issue (e.g., cost) or multiple issues (e.g., cost, time, etc.) about each product with highest profitability. The negotiation can be performed between only one buyer and seller (one-to-one negotiation) or between one buyer and multiple sellers (one-to-many negotiation).

Buyers and sellers' entities could be either a human-type or an agent-type. Agents are autonomous and intelligent entities that aim to achieve specific goals upon the surrounding environment using observations that are perceived and controlled through their sensors and actuators. Intelligent agents often interact with each other in a group for solving complex goals that are insolvable using an individual agent. This interaction creates Multi-Agent Systems (MAS), which is a promising paradigm that enables the negotiation process in large-scale, dynamic, open, and heterogeneous distributed computing systems (as web) [3, 4].

Based on the entities' types, the negotiation process is categorized into three modes: human-human negotiation, human-agent negotiation, and agent-agent negotiation [5]. Due to the rapid growth of the digital world and E-commerce, agent-agent negotiation is a necessary which enables the negotiation process to be autonomous (i.e., automated negotiation) by replacing human decision-making.

Automated negotiation aims to automatically resolve agents' mutual concerns and reach an agreement to improve the current state in an optimal manner [6, 7]. It should be noted that agents in a MAS are assumed to be rational; i.e., they always seek to maximize (or optimize) their profit. Furthermore, opponent agents in a MAS may have contradictory goals. For instance, buyer agents' goal is to purchase more with the lowest price, and seller agents' goal is to sell more with the highest price. In such a context, the automated negotiation process should satisfy the MAS's two parties considering their objectives.

To design an effective automated negotiation system, two main components should be considered [8]: the negotiation protocol and the negotiation strategy; the former consists of specified rules between agents in the negotiation process such as negotiation states, state transition rules, actions, etc. The negotiation protocol is common and should be respected by all agents participating in the negotiation process. The negotiation strategy is the actions performed by agents for reaching an agreement according to the negotiation protocol. The strategy is private and should not be shared between agents.

One of the known fields in designing automated negotiation systems among rational decision-maker agents is Game theory [9, 28]. According to this field, discovering the other opponents' strategies during negotiation process plays an important role in successful developing automated negotiation system. However, agents do not tend to share their strategies with the opponents in a MAS due to the risks of losing their benefits. To solve this problem, it is required to learn the opponent's strategies through opponent modeling. Designing the best model for an opponent's behavior is an effective factor in improving the quality of the negotiation outcomes. It can further increase automated negotiation benefits; i.e., achieving a win-win agreement for the two involved parties (e.g., utilities) while minimizing negotiation cost (e.g., average rounds and agreement rate). In recent years, a number of approaches have been proposed for learning the opponents' behaviors through negotiation process [8, 9]. These approaches include machine learning and swarm intelligence techniques.

Moreover, the management of negotiation processes is another concern that should be taken into consideration. In one-to-many negotiation processes, a buyer agent is negotiating concurrently in parallel to learn and model a number of seller agents' behaviors (as opponents) to select the most profitable negotiation. The central management is not effective in such situations. On the one hand, each seller agent has its own strategy that should be designed, adopted, and adjusted if necessary, by the central management system, during the decision process. On the other hand, the intensity of message exchanges is very high in such a system. Handling the strategies of all the sellers and their adaptations complicates the functionality of central management systems and might cause a single failure point. It is then recommended to consider a distributed management system for one-to-many negotiations.

However, to our best knowledge, none of the referred approaches considers distributed management system for concurrent multi-issue negotiations between one-to-many agents in a MAS environment.

In this paper, we propose a distributed approach that overcomes the presented shortcomings in an automated negotiation system where contradictory multiple issues are negotiated concurrently between a buyer agent and multiple seller agents. The aim of the proposed approach is to reach a win-win agreement where both buyer and seller agents' utilities increase. For learning the opponents' strategies, we apply three machine learning techniques: Bayesian Learning (BL), Multilayer Perceptron Neural Network (MLP), and Kernel Density (KD). The contributions of this paper are summarized as below:

- We provide the modeling of buyer and sellers' agents and their utility functions considering a defined interval for reserved values as agent's strategy.
- We propose adjusting buyer' strategy by learning the strategy of concurrent sellers' agents in each round of negotiation with absolutely no knowledge about the opponents' agents. Bayesian Learning, Artificial Neural Network (MLP), and Kernel Density are applied for the proposed learning design.
- We propose an automated negotiation algorithm that uses proposed learning approaches for predicting the sellers' strategies based on the previous exchanged counter-offers in each round during the negotiation process.
- We design a distributed management system to handle the negotiation algorithm in an effective manner. The system considers a number of threads relies to the sellers' agents that allow the parallel executions of multiple concurrent sellers' agents with no need for a central management system.

The rest of the paper is organized as follows: in section 2, we survey the previous works in the context of learning opponents' strategy. Section 3 describes the proposed approach. In section 4, we present experimental results. Section 5 provides a summary and discussion, and in section 6, we give the conclusion and the future work.

2. Related work

Automated negotiation is the process that occurs between a group of agents with incomplete information about each other. The agents are attempting to reach an agreement on some issues gaining the most profitable outcome for themselves. To ensure a successful negotiation, agents need to discover the opponents' strategies for proposing acceptable offers in a limited time. In this regard, the agents' behaviors (exchanging offers) should be learned and modeled. A number of approaches are proposed previously that could be divided in two main approaches: swarm intelligence techniques and machine learning methods. In the following, we review the previous works that applied these techniques for learning opponents' strategies during automated negotiation processes. The comparison is performed according to the following properties: number of issues (one-issue, multi-issue), number of negotiation participants (one-to-one, one-to-many, many-to-many), the learning category (online, offline), and the learning method (game theory, swarm intelligence and machine learning).

Table 1 represents the related works on automated negotiation.

2-1 Machine Learning Techniques in Automated Negotiation

Machine learning is a promising approach that enables automated modeling through learning from data [8, 9]. The learning techniques could be online or offline. The offline learning requires a training phase based on available historical data for the learning process, enabling them to be more accurate for proposing new offers. The online learning category comprises techniques that can learn and improve the estimation values in each round of the on-going negotiation process without requiring any historical data (the training phase is absent).

The provided related works in our context use four main machine learning approaches: Bayesian learning [11, 12, 13, 14, 15, 16], Non-linear Regression [17, 18], Kernel Density Estimation [19, 20, 24] and Artificial Neural Network [21, 22, 23].

Bayesian learning applies probability approaches to estimate and predict opponents' strategy parameters based on the received and accepted offers in each round. The key factor in this approach is to define relevant hypotheses and the right inputs for the learning step. Moreover, the computational complexity could be expensive when the number of hypotheses increases. The nonlinear regression approach considers a decision function with unknown parameters such as the opponent's deadline or opponent's tactic (e.g., time or behavior-dependent tactics [25]). The aim is to derive the function used by the opponent based on the proposed offer values to apply

TABLE 1. OPPONENT'S BEHAVIOR LEARNING IN AUTOMATED NEGOTIATION

Reference	Learning category	Learning method	Issue	Number of opponents
Zeng, D. et al. [11, 12]	Online	Bayesian learning	multi-issue	one-to-one
Zhang, et al.[13, 14]	Online	Bayesian learning	multi-issue	one-to-one
Eshragh et al. [15]	Online	Bayesian learning	multi-issue	one-to-one
Imran et al. [16]	Online	Bayesian learning	multi-issue	one-to-one
Yu et al. [17]	Online	Bayesian learning & Non-linear regression	multi-issue	one-to-one
Chongming Hou [18]	Online	Non-linear regression	one-issue	one-to-one
Mezni et al. [19]	Online	Kernel density estimation & PSO	multi-issue	one-to-many
Coehoorn et al. [20]	Offline	Kernel density estimation	multi-issue	one-to-one
Moosmayer et al. [21]	Offline	Artificial neural network	one-issue	one-to-many
Chun et al. [22]	Offline	Artificial neural network	one-issue	one-to-many
Bagga et al. [23]	Offline	Artificial neural network & Deep reinforcement learning	one-issue	one-to-many
Kolomvatsos et al. [27]	Online	Kernel density estimation & PSO	multi-issue	one-to-many
Kolomvatsos et al. [29]	Online	ABC	multi-issue	one-to-many

for the future opponent's offer values. The initial guess is one of the key factors for the success of this method.

The approach proposed by Zeng and Sycana's [11, 12] is one of the earliest researches conducted in the field of automated negotiation using Bayesian learning. Before the negotiation process starts, a set of hypotheses about the reserved value of the opponents are made based on the obtained information from the previous negotiations. During the negotiation, the hypothesis and information scope are updated using the Bayesian learning method.

In [13, 14], Zhang et al. suggested an approach for bilateral multi-issue negotiation so that both participants gain the maximum benefits. In this approach, Bayesian learning has been applied for predicting the opponent's preferences by analyzing the previous offers. The authors also proposed a counter-offer proposition algorithm to adopt an effective trade-off between high-weighed and low-weighted issues based on the opponent's predicted preference.

Eshragh et al. [15] provided a fuzzy-based opponents' preferences modeling in multi-issue, multi-participant negotiation systems. The seller's agent evaluates and estimates the parameters of the opponents' preferences model using the recursive Bayesian filtering technique based on the fuzzy scores and some arguments about the offered proposal that were received from opponents' agents.

Authors in [16] considered a bilateral negotiation in the electricity market context using Bayesian learning. They proposed a utility-based strategy model by combining the trading reward and the perception of the remaining negotiation time for both buyer and seller agents. Furthermore, an adaptive agent-tracking strategy is provided for seller agents to estimate and update the reserved value of buyer agents in each round using Bayesian learning.

Yu et al. [17] combined the Nonlinear Regression method with Bayesian learning for predicting the reservation value as well as the opponent's time limit (deadline) in a one-to-one dynamic negotiation process. First, the unknown parameters of the time-dependent behavior are estimated using nonlinear regression. Second, the hypotheses of Bayesian learning are improved based on the calculated estimated values.

Authors in [18] used three decision tactic functions [25] as offer modeling (time-dependent tactic, resource-dependent tactic, behavior-dependent tactic) in one issue negotiation process. They applied nonlinear regression to identify a tactic type for estimating the reservation value and the deadline. To improve the initial guess, they provided some heuristics for the mentioned tactics.

We proposed a Bayesian Learning-based technique for opponent modeling in automated multiagent negotiation [30]. In this work, Bayesian Learning method increases the efficiency of negotiation, which is measured in terms of average buyer utility (ABU) and average seller utility (ASU). ABU and ASU have increased from 90% to 94% and 27% to 31%, respectively.

Coehoorn and Jennings [20] proposed a method for learning the opponent's preferences in a multi-issue time-dependent negotiation process. They applied KDE to estimate the opponent's weight for an issue by finding a relation between the difference of the two last offers and the weight of an issue derived from previous negotiations.

Authors in [19] considered the negotiation process in the context of service selection and composition in cloud environments. The Kernel Density estimation approach is used to resolve unknown Quality of Adaption (QoA) parameters of candidate services to enable predicting the service behavior. The input data of a KDE is the negotiated penalty values that were assigned to candidate services.

Moosmayer et al. [21] used an artificial neural network to study the trend of annual prices in the negotiation process of participants. They predicted whether the reservation value, target value, and initial offer affect the results of the negotiations between sellers and buyers. Neural network analysis is used for the flexibility of predicting the effective factors on price. In comparison with regression analysis, the neural network has a lower standard error and is showed that the target factors have a significant role in B2B price negotiations.

Authors in [22] provided a neural network-based model to predict the supplier's offer during the complex and variant negotiation process. They considered both non-offer (e.g., inventory level and scheduled production level) and offer dependent information (e.g., past and current offer, deadline) as inputs of the neural network.

In [26], an architecture for learning the opponent's behaviors in a multi-issue bilateral negotiation process has been proposed. The proposed architecture comprises a behavior prediction logic component that applies an artificial neural network for behavior prediction.

In [23], a deep reinforcement learning approach is used to determine the action to take (policy) by a buyer agent in a particular state of the concurrent bilateral negotiation environment. The authors applied artificial neural networks to train the strategy used as the initial policy inputs for the RL approach.

2-2 Swarm Intelligence Learning-Based Methods in Automated Negotiation

Swarm intelligence (SI) is a concept that considers the collective behavior of decentralized, self-organized systems such as MAS. These approaches include Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO) and simulate the foraging operation of birds, honey bees, ants, respectively. In the domain of automated negotiation, SI is applied in many fields such as cloud computing, service composition, and e-commerce.

In our context, agents are decentralized entities who could perform collective behaviors to increase their utilities for each round of the negotiation process.

Authors in [27] adopted Particle Swarm Optimization (PSO) algorithm to adjust the weights of multiple issues in e-commerce applications to find the optimal agreement in a one-to-many negotiation process. An extension of the previous work is provided in [29] by the same authors but using an Artificial Bee Colony (ABC) algorithm.

Authors in [19] proposed a negotiation-based model for service selection and composition in the cloud environment. They used a combination of KDE technique and PSO approach to estimate the service behavior for revealing the incomplete knowledge through negotiation between service providers' agents in order to optimize the service composition process.

3. Proposed Methodology

In this section, we describe our proposed approach for the automated negotiation problem. Agents negotiate with each other to reach an agreement that satisfies their strategies and results in the highest profit at the end of the negotiation process. Three machine learning techniques (BL, KDE, ANN) are used to learn and model the opponents' strategies. First, we represent the proposed automated negotiation system and the related parameters. Second, the proposed algorithm is described. Finally, we model the opponents' strategies using the three techniques.

3-1.The Proposed System

The system comprises one buyer and multiple sellers agents (one-to-many) negotiating concurrently over multiple issues (e.g., price, quality, deadline, etc.) to reach an agreement. A reserved value corresponds to each agent (as a strategy) within the interval [low_i; $high_i$], which should be satisfied during the negotiation process. This value is hidden from other agents in the system. There is a deadline value related to each agent, and the negotiation continues until the deadline is reached. The buyer's deadline and the seller's deadline are denoted as T_b and T_s, respectively. Moreover, the objective of agents is contradictory: the buyer tends to buy a high-quality product at a low price, while the seller tends to sell its product at a high cost.

Agents attempt to model and learn the opponents' strategies using machine learning techniques. The seller agent initializes the negotiation process by proposing an offer. For the next rounds, the offer will be given alternatively (i.e., either the buyer agent or the seller agent proposes). If the buyer accepts the offer proposed by the seller, the negotiation process terminates; otherwise, the buyer proposes a new offer after the learning step. Agents gain utility when the negotiation results in an agreement. There is zero utility if the deadline is passed or the negotiation is terminated by one of the agent entities. Fig.1 illustrates the proposed system. It is worth noting that the learning techniques are assumed to be run on each agent to provide a distributed negotiation management system.

Negotiated issues are directly proportional (P) or inverse proportional (IP) to the utility. In direct proportion, the utility increases as the value of the issue increases, and in inverse proportion, the utility decreases as the value of the issue increases. For example, the price is directly related to the seller's utility, which means that as the price goes up, the seller's utility also increases.

3-2.The proposed algorithm

In this section, we describe the proposed algorithm (see Algorithm 1). Inputs are reserved values low_i, $high_i$, and the current time t and output is the best proposed offer.

In the first step, the seller' offer is calculated as (1) (line 3):

$$OSeller_i = low_i + (1 - \phi(t)) * (high_i - low_i) \quad (1)$$

Where $OSeller_i$ denotes the seller's offer for issue i in each round of negotiation. low_i ($high_i$) is the lowest (highest) reserved value for each issue, and $\Phi(t)$ corresponds to the time-dependent strategy function, which is formulated using (2):

$$\Phi(t) = (1 - k)(1 + \left(\frac{min(t, T)}{T}\right)^{\frac{1}{\psi}}) \quad (2)$$

Where T denotes the deadline, and the k and ψ parameters correspond to a random value in [0; 1]. This function returns a value within the interval [0, 1]: $0 \leq \Phi \leq 1$ ($\Phi(0) = k$ and $\Phi(T) = 1$). The seller's offer is then sent to the buyer. In the second step, the utility of buyer agents is evaluated according to the opponent's offer value using (3) and (4), as below (lines 4 - 8):

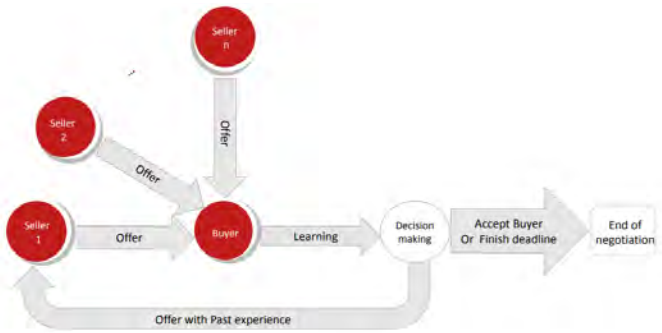


Fig.1. The proposed system

Algorithm 1 Automated negotiation algorithm

```
    Input: $lowBuyer$ Minimum Buyer Offer, $highBuyer$ Maximum Buyer Offer, $lowSeller$ Minimum Seller Offer, $highSeller$ Maximum Seller Offer, $t$ current time
    Output: BestOffer
 1: $AcB = -1$                ▷ The offer is not accepted by Buyer
 2: While (deadline $\leq t$) or ($AcB == -1$)
 3:     Calculate $OSeller$ using equation 1
 4:     For each issue $i$
 5:         If $OBuyer \in P$ then
 6:             Calculate UBuyer using equation 3
 7:         Else
 8:             Calculate the UBuyer using equation 4
 9:     Calculate final utility of buyer using equation 5
10:     If $UBuyer_i > UBuyer_{i-1}$ then
11:         $AcB == 1$
12:         Send the final utility for all agents
13:         BestOffer = OSeller
14:         Break
15:     Else
16:         $OBuyer = learning(OBuyer)$
17:         Check the duplication of OBuyer
18:     For each issue $i$
19:         If $OSeller \in P$ then
20:             Calculate USeller using equation 6
21:         Else
22:             Calculate USeller using equation 7
23:     Calculate final utility of seller using equation 8
24:     If $USeller_i > USeller_{i-1}$ then
25:         $AcS == 1$
```

$$U(OBuyer_i) = \frac{OSeller_i - low_i}{high_i - low_i} \text{ if issue } i \text{ is P} \tag{3}$$

$$U(OBuyer_i) = \frac{high_i - OSeller_i}{high_i - low_i} \text{ if issue } i \text{ is IP} \tag{4}$$

where *OBuyer$_i$* is the offer that corresponds to the *i*th issue of the buyer. If the offer is directly proportional (P) to the utility, (3) is applied; otherwise, the offer is considered indirectly proportional (IP) to the utility, calculated using (4). The final utility value for the buyer is then estimated as (5) (line 9):

$$FinalUtility(Buyer) = \sum_{i=1}^{n} W_i.U(OBuyer_i) \tag{5}$$

where *n* refers to the number of issues, and *W$_i$* is the weight corresponding to each issue *i*. In this stage, the buyer compares the utility of the current offer to the previous one. If the current utility is more, the buyer will accept the negotiation. The received utility value is then broadcasted to the other agents. Otherwise, if the previous utility is more, a new offer (*OSeller$_i$*) will be introduced using the machine learning methods described in the next section (lines 10-17). During the negotiation, all offers are stored to avoid duplicated offers. The seller calculates the utility of each issue *i* after receiving the buyer offer (*OBuyer$_i$*) as (6) and (7) (lines 18-22):

$$U(OSeller_i) = \frac{OBuyer_i - low_i}{high_i - low_i} \text{ if issue } i \text{ is P} \tag{6}$$

$$U(OSeller_i) = \frac{high_i - OBuyer_i}{high_i - low_i} \text{ if issue } i \text{ is IP} \tag{7}$$

The final utility value for the seller is then estimated as (8) (line 23):

$$FinalUtility(Seller) = \sum_{i=1}^{n} W_i.U(OSeller_i) \tag{8}$$

If the current utility is more, the seller declares his agreement. Otherwise, he proposes another offer using equation 1 (lines 24 and 25). It continues until the agent's deadline is reached or negotiation is accepted by the buyer.

3-3.Bayesian Learning

BL is used to estimate the next opponent's offer in each round. To estimate more effectively, given an issue, we consider an interval of offers value for that issue. We assume that each agent proposes the maximum or minimum value of the defined interval. The interval is divided into equal cells, and a random value *X$_i$* is considered within each cell. These values are considered to be the reserved value of the opponent. Next, it is decided which value will be sent to the opponent using the Bayesian learning method.

Based on the random values (*X$_i$*) and the history of the opponent's offers so far, the adjusted offer seller is calculated by (9) using a linear regression [17]:

$$OSeller_i(t) = p_o + (p_i^x - p_o)(\frac{t}{t_i^x})^b \tag{9}$$

where p_o is the initial buyer offer, p_i^x is the random value in each cell, and *b* is the regression coefficient and is calculated by (10) [17] as:

$$\begin{aligned} b &= \frac{\sum_{i=1}^{t_b} t_i^* P_i^*}{\sum_{i=1}^{t_b} t_i^{*2}} \\ P_i^* &= In\frac{P_0 - P_i}{P_0 - P_i^x} \\ t^* &= In\frac{t}{t_i^x} \end{aligned} \tag{10}$$

Then, the correlation coefficient between the adjusted seller offer and the history of the seller offers is calculated using (11) [17]:

$$\gamma = \frac{\sum_{i=1}^{t_b}(P_i - \bar{P})(\hat{P}_i + \bar{\hat{P}})}{\sqrt{\sum_{i=1}^{t_b}(P_i - \bar{P})^2 \sum_{i=1}^{n}(\hat{P}_i + \bar{\hat{P}})^2}} \tag{11}$$

where refers to the similarity of the adjusted parameters and the history of the opponent offers, which $\bar{P_i}$ and $\bar{\hat{P}}$ denote the adjusted offer and the average of the adjusted offer, respectively.

Based on the above equations, given a hypothesis, we calculate the probability of the offering seller using the Bayesian probability as (12):

$$P(H_i \mid OSeller) = \frac{P(OSeller \mid H_i)\, P(H_i)}{\sum_{k=1}^{N_{all}} P(OSeller \mid H_k) P(H_k)} \quad (12)$$

In this equation, the number of hypotheses is equal to the number of cells. Therefore, we have P(H_i) = 1/(N_{all}). $P(H_i|OSeller)$ indicates the probability of *OSeller* under hypothesis H_i. In the proposed model, *OSeller* is the history of the opponent's offers, and it determines which of the *X* values is the actual opponent's reserved value. The posterior probability P(H_i|*OSeller*) is obtained according to the new values of *OSeller* in the next round. The agent updates the P(H_i)/ using $P(H_i|OSeller)$. The $P(OSeller|H_i)$ is equal to γ.

3-4.Artificial Neural Network

In this work, a multilayer perceptron neural network with three inputs, a hidden layer, and an output layer is designed to predict the opponent's reserved value. The function of the developed neural network is defined as (13):

$$Output = f\ (Input_1, Input_2, Input_3) \qquad (13)$$

where $Input_1$ is the lowest offer, $Input_2$ is the highest offer, and $Input_3$ is the value of *x*. The hidden layer has 3 Sigmoid neurons and the output layer uses a linear neuron. The output of the network determines whether the calculated value is the opponent's reservation value or not. The weights of the artificial neural network are obtained at the training phase using the backpropagation method. After the learning process, there is a need for the adaptation strategy. Because several values might be selected as the reservation value. We assume that the opponent proposes offers that are close to its reserved value. For that, we calculate the distance between the selected *x*s and the opponent offers. The *x* with the lowest distance is selected as the next offer.

3-5.Kernel Density Estimation

The opponent's reservation value is estimated using (14) [29]:

$$CDF(x) = \frac{1}{N}.\sum_{i=1}^{N}\frac{1}{2}.(1 + erf(\frac{x - P_{si}}{\sqrt{2}})) \quad (14)$$

where *x* is the random number of each cell, and P_{si} is the history of the opponent's offer value. *erf* is considered to be 0.14.

Once calculating the kernel density of *x*'s, the largest one will be delivered as an offer to the opponent.

4. Experimental results

In this section, we represent the experimental results that are given from our proposed approaches for modelling opponents' strategies. We compare our work with the approach proposed in [29] where ABC model is represented for estimating the optimal offer based on the collective behaviors (offers) of seller agents.

4-1.Dataset

In order to evaluate the effectiveness of our proposed approaches, we have used the real S&P 500 stock data where are recorded stock prices for all companies currently found on the S&P 500 index during five years (https://www.kaggle.com/camnugent/sandp500). There are six columns in the database, including date, high, low, close, volume, and name. High, low, and close represent the highest price reached in the day, the lowest price reached in the day and the optimal price for both agents, respectively. Volume attribute is the number of shares traded, and name represents stock's ticker name. We use the high, low, close, and proposed offer (price) attributes as four issues in our evaluation. The close attribute is the desired value that both buyer and sellers' agents tend to reach it. The proposed offer is calculated using our proposed approach during the negotiation process.

4-2Initial Setting

In the empirical evaluation section, one hundred negotiations with fifty threads were conducted for four issues (high, low, close, and proposed current offer). The first proposed offer from the seller is a random number in interval [low, high]. The seller and the buyer's deadline are considered a random number between [6, 20] (see section 4.4).

4-3.Model Evaluation Criteria

Four criteria have been used for the model evaluation according to [29].

1) *Agreement ratio(AG)*

This criterion specifies the number of negotiations that come to an end by reaching an agreement. *R* represents the total number of negotiations; *S* is the number of negotiations accepted by the buyer, and *B* is the number of the negotiations accepted by the seller. The agreement ratio is obtained by (15):

$$AG = \frac{B + S}{R} \qquad (15)$$

The agreement ratio determines the degree of agent satisfaction. The high level of this criterion represents that many agreements have come to an end, and both agents have benefitted.

2) *Average Buyer utility (ABU)*

This criterion represents the buyer utility during the negotiation and obtained from (16):

$$ABU = \frac{1}{B + S}\sum_{i=1}^{B+S} Max_i(U_b) \qquad (16)$$

The buyer utility $Max_i(U_b)$ is calculated so that the highest utility is regarded among the current negotiations with several sellers. That is, the buyer ultimately accepts the negotiation, which has the highest utility. It is assumed that when the buyer reached an agreement in a negotiation, he does not inform the seller and waits for the other negotiations' answer. Also, time does not affect the final result (the term time refers to the response time, not the negotiation rounds).

3) *Average Seller utility (ASU)*

This criterion specifies the utility of the seller. This criterion is an average of the seller utility. The utility is calculated only when the agents have reached an agreement and estimated from (17):

$$ASU = \frac{1}{B+S}\sum_{i=1}^{B+S} U_s \quad (17)$$

which U_s is the utility of the seller in the negotiation process.

4) Average rounds (AR)

This criterion is the number of rounds required for the negotiations to be reached an agreement. $T = \min(T_s, T_b)$ indicates the needed time and the resources related to the negotiation results. The high average of rounds indicates the need for more time and resources. This criterion is defined by the (18) and (19):

$$AR = \frac{1}{B+S}\sum_{i=1}^{B+S} \frac{Z.t^{*} + (1-Z)t^{*}}{T} \quad (18)$$

$$Z = \begin{cases} 1 & \text{if the buyer accept the seller's offer at time } t^{*} \\ o & \text{if the seller accept the buye's offer at time } t^{*} \end{cases} \quad (19)$$

4-4. The Relationship Between Evaluation Criteria and Deadline

As it was said, a deadline is assigned to each agent (a deadline is the number of rounds each agent participates in the negotiation). To determine the minimum deadline, an experiment has been designed where 100 negotiations were conducted with a deadline between 3 and 11. The experimentations are repeated 200 times. It is worth mention that we confirm the relationship between utility and time using both the ABC method (see [29]) and our proposed BL method. According to the negotiation protocol, the negotiation comes to an end when the buyer accepts the offer or when the deadline passes. Suppose the negotiation does not reach an agreement. In that case, the utility of the two sides is zero; however, by considering whether the seller accepts or not, the agreement ratio (*AG*) and average rounds (*AR*) change. If the seller does not accept the offer, all four mentioned criteria would be zero. If the seller accepts the offer, depending on the number of acceptance and the negotiation time, the agreement ratio and the average rounds are determined as a value within the interval [0, 1]. According to the experimental results, if the deadline is considered equal to 3, the agreement ratio is zero. Therefore, the minimum value of the deadline is equal to 4. In Fig.2, the deadline is considered a number between 3 and 11. Due to the use of mean in equations 15, 17, and 18, the value of these three criteria is decreased. In Fig.2a, the horizontal axis represents the deadline, and the vertical axis represents the agreement ratio value. As it was seen, when the deadline is three, the agreement ratio is approximately equal to zero. As the deadline increases, the agreement ratio also increases. According to the definition of the model, the buyer ultimately accepts the negotiation with the highest utility. So, the low deadline does not affect the buyer utility (see Fig.2b). Fig.2c also shows that the deadline decreasing has a direct effect on the seller utility. Finally, Fig. 2d illustrates that when the deadline is low, less round number is required to reach an agreement. This case is a good point, but it should be taken into account that both buyer utility and agreement ratio are also decreased, and it is therefore possible that the agents quit the negotiation. Oobviously, as the deadline increases, a higher agreement ratio could be obtained. According to the experimental results, to have a good agreement ratio and the buyer utility, the minimum amount of the deadline is considered equal to 6. To carry out the experiments, it is required to limit the deadline interval. we specify the deadline interval as a value between [6, 20].

4-5. Performance Results

The main challenge of this study is learning the opponent's behavior and the answer to the question whether learning the opponent's behavior affects the agents' utility? We estimate the opponent's strategy using three methods of machine learning.

The first method applies the combination of the regression analysis and Bayesian learning, in which we have two parts: 1) learning and 2) strategy adaptation. Points are the opponent reservation value and are selected randomly. The correlation coefficient is calculated using the regression analysis (learning part). The probability of the occurrence of each reservation point is obtained using Bayesian methods (strategy adaptation part).

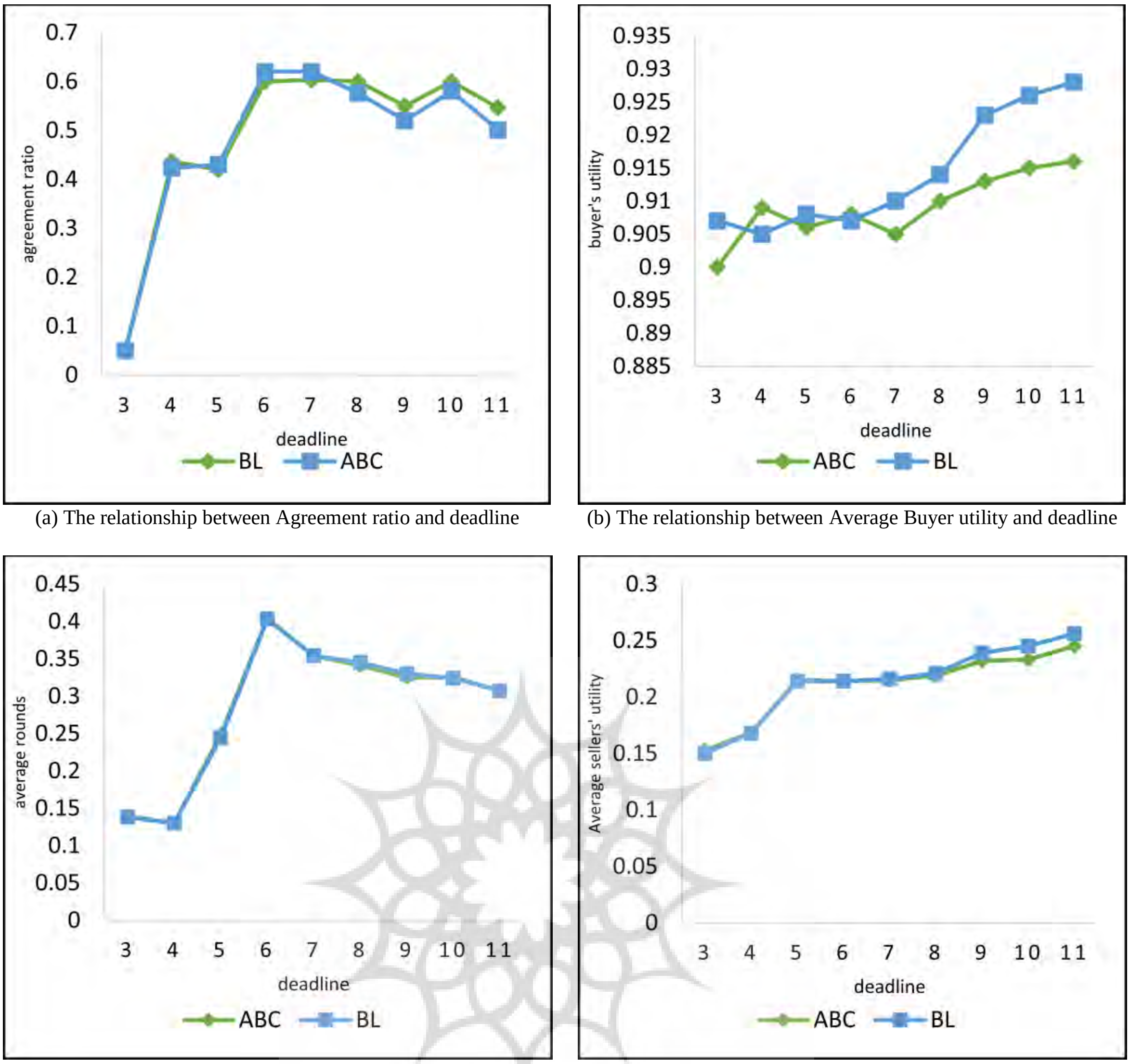
The second method is the artificial neural network. At first, the neural network is trained by the data obtained from the Bayesian method. At each phase, high price, low price, the opponent offer, and the offer's acceptance or non-acceptance were stored. These data were used as the training data and applied for the test data of the S&B stock database.

Then as in the previous method, points are considered as the opponent reservation value. The neural network predicts which one can be an actual value. Since several points might be selected according to the described negotiation protocol, the opponent remains close to the desired close price. Consequently, Euclidean distance is used, and the point with the shortest distance to the opponent's offer is selected.

The final method is applying the Kernel Density Estimation method. This method obtains the Kernel Density Estimation After choosing the default points, and the point which has the lowest value is chosen. As mentioned initially, the Kernel Density Estimation method is one of the offline methods; however, it was applied online in this study. As it estimates, using the opponent offers history.

The proposed methodologies are based on the one-buyer multi-seller multi-issue automated negotiation. The buyer and sellers' agents have no information about each other. Using machine learning methods, we learn and reveal the opponent (seller) strategy, and according to that, we propose the offer to the buyer's agent. This allows decreasing the number of the negotiation round while increasing the buyer utility significantly.

Fig.3 represents the experimentation results. In Fig.3a, the agreement ratio criteria are calculated and illustrated for the three mentioned machine learning methods. The artificial neural network has the highest and ABC [29] methodology has the lowest agreement ratio. Fig. 3b and 3c show the buyer utility and the seller utility criteria. The highest utility is also achieved by the artificial neural network method. And finally, Fig. 3d represents the average rounds. The artificial neural network requires fewer rounds for agreement.



(a) The relationship between Agreement ratio and deadline

(b) The relationship between Average Buyer utility and deadline

(c) The relationship between Average Seller utility and deadline

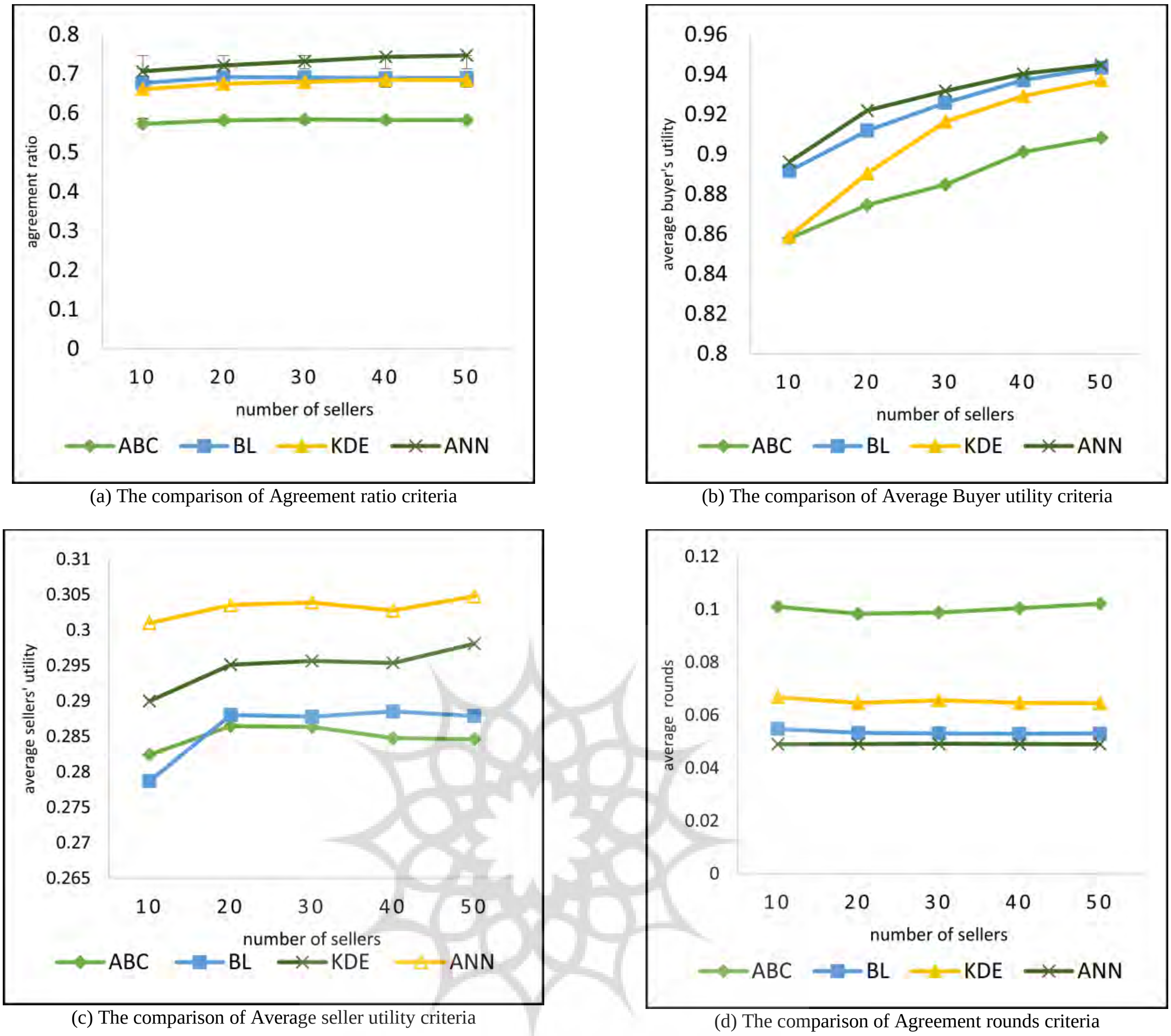
(d) The relationship between Average rounds and deadline

Fig. 2. The relationships between evaluation criteria and deadline

As summary, considering the four evaluation criteria, the three machine learning methods achieve a better result than ABC methodology. Among these methods, the best results for the four criteria are obtained by the artificial neural network method. The reason for the improvement of the results after applying learning techniques can be summarized in the sentence that learning the opponent's behavior has caused the agent to obtain more accurate information about the opponent's actions and therefore make a better decision at each stage. Therefore, its utilization rate improves.

5. Conclusion and Future Work

The focus of this study was on the machine learning methods, which in the area of auto-negotiation had considered the opponent behavior in the multiagent environment. Three methods of Bayesian learning, Kernel Density Estimation and Artificial Neural Network were examined. The approach of our study is the learning of strategy acceptance to enhance buyer utility. The buyer can negotiate with several sellers simultaneously and selects a negotiation with the highest utility. By removing the intermediates and reducing the number of massages as well as applying the online learning method, it is tried to take a step towards conducting auto-negotiation in the real world. Comparing our proposed method with the methodology of [29] showed that when the opponent's behavior is learned, the desired results are achieved (the buyer utility and the agreement ratio increase and the round average decreases). Compared to the other methods of machine learning, the Artificial Neural Network method had the best answer. But as stated in the previous section, compared to the previous method, all the three proposed methods had a better outcome. If time is essential in a negotiation, it is recommended to use Bayesian learning. If the utility is important, and the data is available, the artificial neural network is recommended. The more information we get from the opponent naturally, the better performance we will have. Therefore, we need a method that calculates the opponent's strategy, learns the preferences, learns the opponent's deadline, and learns its offering strategy. Since the opponent agents are



(a) The comparison of Agreement ratio criteria

(b) The comparison of Average Buyer utility criteria

(c) The comparison of Average seller utility criteria

(d) The comparison of Agreement rounds criteria

Fig. 3. Evaluation criteria comparisons in the proposed methodology of machine learning methods and ABC for opponent strategy learning

also rational, our behavior must be such that to obtain the information from the opponent and the opponent does not be aware of our strategy. So, we need to have some strategies to change the strategy if necessary. The other issue is investigating the prediction of future utility. After reaching an agreement, whether the continuation of the negotiation has a higher utility for us or not? The advantage of knowing this information is to increase its deadline if the continuation of the negotiation has a higher utility. If it is not the case, to leave the negotiation. This will improve the quality of the negotiation. To use the nonlinear regression model, we need to know our data well to select a model that suits it. One of the works we will do in the future is the precise analysis of the data and extracting a model for nonlinear regression.

REFERENCES

[1] Kenneth Laudon and Carol Traver. *E-Commerce: Business, Technology, Society (3rd Edition)*. Prentice-Hall, Inc., USA, 2006.

[2] Dave Chaffey. *E-business and E-commerce Management: Strategy, Implementation and Practice*. Pearson Education, 2007

[3] Sarit Kraus. Automated negotiation and decision making in multiagent environments. In *ECCAI Advanced Course on Artificial Intelligence*, pages 150–172. Springer, 2001.

[4] Peter Braun, Jakub Brzostowski, Gregory Kersten, Jin Baek Kim, Ryszard Kowalczyk, Stefan Strecker, and Rustam Vahidov. *E-negotiation Systems and Software Agents: Methods, Models, and Applications*, pages 271–300. Springer London, London, 2006.

[5] Dean G Pruitt and Peter J Carnevale. *Negotiation in social conflict*. Buckingham, Open University Press, 1993.

[6] Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[7] Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of automated negotiation*. Cambridge University Press, 2014.

[8] Tim Baarslag, Mark J.C. Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5):849–898, 9 2016.

[9] Usha Kiruthika, Thamarai Selvi Somasundaram, and S. Kanaga Suba Raja. Lifecycle model of a negotiation agent: A survey of automated negotiation techniques. *Group Decision and Negotiation*, 9, 2020.

[10] S. Park and S. Yang. An automated system based on incremental learning with applicability toward multilateral negotiations. In *2006 SICE-ICASE International Joint Conference*, pages 6001–6006, 2006.

[11] Dajun Zeng and Katia Sycara. Benefits of learning in negotiation. In Proceedings of the *14th National Conference on Artificial Intelligence*, AAAI 97, pages 36–41, 1997.

[12] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48(1):125–141, 1998.

[13] Jihang Zhang, Fenghui Ren, and Minjie Zhang. Bayesian-based preference prediction in bilateral multi-issue negotiation between intelligent agents. *Knowledge-Based Systems*, 84:108–120, 2015.

[14] Jihang Zhang, Fenghui Ren, and M. Zhang. Prediction of the opponent's preference in bilateral multi-issue negotiation through bayesian learning. In *ANAC@AAMAS*, 2014.

[15] Faezeh Eshragh, Mozhdeh Shahbazi, and Behrouz Far. Real-time opponent learning in automated negotiation using recursive bayesian filtering. *Expert Systems with Applications*, 128:28–53, 2019.

[16] Kashif Imran, Jiangfeng Zhang, Anamitra Pal, Abraiz Khattak, Kafait Ullah, and Sherjeel Mahmood Baig. Bilateral negotiations for electricity market by adaptive agent-tracking strategy. *Electric Power Systems Research*, 186:106390, 2020.

[17] Chao Yu, Fenghui Ren, and Minjie Zhang. *Bilateral Negotiation Model Based on Bayesian Learning*, pages 75–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[18] Chongming Hou. Predicting agents tactics in automated negotiation. In Proceedings. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'04*, pages 127–133, 2004.

[19] Haithem Mezni and Mokhtar Sellami. A negotiation-based service selection approach using swarm intelligence and kernel density estimation. *Software: Practice and Experience*, 48(6):1285–1311, 2018.

[20] Robert M. Coehoorn and Nicholas R. Jennings. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th International Conference on Electronic Commerce*, ICEC '04, page 59–68, New York, NY, USA, 2004.

[21] Dirk C. Moosmayer, Alain Yee-Loong Chong, Martin J. Liu, and Bjoern Schuppar. A neural network approach to predicting price negotiation outcomes in business-to-business contexts. *Expert Systems with Applications*, 40(8):3028 – 3035, 2013.

[22] Chun Ching Lee and C. Ou-Yang. A neural networks approach for forecasting the supplier's bid prices in supplier selection negotiation process. *Expert Systems with Applications*, 36(2, Part 2):2961–2970, 2009.

[23] Pallavi Bagga, Nicola Paoletti, Bedour Alrayes, and Kostas Stathis. *A deep reinforcement learning approach to concurrent bilateral negotiation*, In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Pages 297-303, 2020.

[24] G. M. Farag, S. E. AbdelRahman, R. Bahgat, and A. M. A-Moneim. Towards KDE mining approach for multiagent negotiation. In *2010 The 7th International Conference on Informatics and Systems (INFOS)*, pages 1–7, 2010.

[25] Peyman Faratin, Carles Sierra, and Nick R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3):159 – 182, 1998.

[26] Mohammad Irfan Bala, Sheetal Vij, and Debajyoti Mukhopadhyay. Negotiation life cycle: An approach in e-negotiation with prediction. In Suresh Chandra Satapathy, P. S. Avadhani, Siba K. Udgata, and Sadasivuni Lakshminarayana, editors, *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*, Vol I, pages 505–512, Cham, 2014.

[27] Kostas Kolomvatsos and Stathes Hadjieftymiades. On the use of particle swarm optimization and kernel density estimator in concurrent negotiations. Information Sciences, 262:99 –116, 2014.

[28] Binmore, K., Vulkan, N. Applying game theory to automated negotiation. NETNOMICS: Economic Research and Electronic Networking 1: 1–9, 1999.

[29] Kolomvatsos, Kostas & Panagidi, Kakia & Neokosmidis, Ioannis & Varoutas, Dimitris & Hadjiefthymiades, Stathes. Automated Concurrent Negotiations: An Artificial Bee Colony Approach. Electronic Commerce Research and Applications.19:56-69, 2016.

[30] Fatemeh Mohammadi-Ashnani, Zahra Movahedi, Kazim Fouladi-Ghaleh. Using Bayesian Learning for Opponent Modeling in Multiagent Automated Negotiation, Proceedings of the *5th International Conference on Web Research, ICWR 2019*, University of Science and Culture, Tehran, Iran, 2019.

Fateme Mohammadi Ashnani received the M.S. degree in Information Technology (IT) Engineering from Faculty of Engineering, College of Farabi, University of Tehran, in 2016. She has worked on machine learning and multiagent systems. Her research interest is online machine learning.

Zahra Movahedi received the B.S. and M.S. degrees in Computer Engineering from University of Rene Descartes and University of Pierre & Marie Curie, respectively, in France. She received the Ph.D. degree in Computer Engineering from Télécom SudParis & University of Paris Saclay, France. She is currently an assistant professor of Computer Engineering at University of Tehran, Iran. Her research interests include IoT, Fog Computing, Multiagent Systems, and Blockchain.

Kazim Fouladi-Ghaleh received the M.S. and Ph.D. degrees in Electrical & Computer Engineering / Artificial Intelligence and Robotics from School of ECE, College of Engineering, University of Tehran, Iran. He is currently an assistant professor of Computer Engineering at University of Tehran, and the director of the Cyberspace Research Lab. His research interests include AI, Image Processing & Computer Vision, Deep Learning, Cybenetics and Cyberspace Studies.