TIME SERIES PREDICTION MODELS IN BILATERAL NEGOTIATIONS

A Thesis

by

Gevher Yesevi

Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Artificial Intelligence

Özyeğin University
January 2024

TIME SERIES PREDICTION MODELS IN BILATERAL NEGOTIATIONS

Approved by:

Asst. Prof. Reyhan Aydoğan, Advisor
Dept. of Computer Science
Özyeğin University

Asst. Prof. Emre Sefer
Dept. of Computer Science
Özyeğin University

Asst. Prof. Faik Boray Tek
Dept. Artificial Intelligence and Data Engineering
Istanbul Technical University

Date Approved: November 10, 2023

To my family, who inspire and strengthen me every day.

ABSTRACT

This thesis explores the dynamics of agent-based negotiations, with a focus on understanding opponent's behavior and predicting their offering patterns to make strategic decisions. Guessing the utility of the opponent's upcoming offers valuable insights for the agent's subsequent moves. The research aims to predict the opponent's future offers by employing diverse learning algorithms in various situations to measure their effectiveness in comprehending negotiation behavior. The prediction study comprises two parts; one investigating the impact of these models in one-to-one negotiations, specifically tailored for the agent's own experiences, while the other examines the performance of predictive models in a tournament setting for all agents. A learning process with three distinct targets have been established to assess the prediction models: (i) estimating the agent's utility of the opponent's next offer by considering only its offer history, (ii) estimating the agent's utility considering opponent-related variables, and (iii) estimating the opponent's utility using opponent-related variables. According to the experimented results, the best learning approach is incorporated into an agent design to observe the impacts of having predictions of future utility values on the agent's negotiation success. The thesis evaluates these models in diverse negotiation scenarios and highlights promising outcomes for the proposed methods. It also introduces a novel negotiation strategy called 'Negoformer', which incorporates predictions into the offering strategy and investigates their impact on the outcome of negotiations. The experiments showcased the success of Negoformer compared to other agents in various negotiation success metrics, such as individual utility value and social welfare score.

ÖZETÇE

Bu tez, etmen tabanlı müzakerelerin dinamiklerini araştırırken rakip etmenlerin davranışlarını anlama ve stratejik kararlar almak için teklif örüntülerini tahmin etmeye odaklanır. Rakibin gelecekteki tekliflerinin faydasını tahmin etmek, etmenin sonraki hamleleri için değerli içgörüler sunmaktadır. Araştırma, etmen davranışlarını anlama yetkinliğini ölçmek için çeşitli öğrenme algoritmalarını kullanarak rakibin gelecekteki tekliflerini tahmin etmeyi amaçlamaktadır. Tahminleme becerilerine odaklanan çalışma iki bölümden oluşmaktadır: ilki etmenin kendi deneyimleri baz alınarak tasarlanmış tekli müzakerelerde bu modellerin etkisini incelemekteyken diğeri turnuva boyunca tüm etmenler için tahmin modellerinin performansını incelemektedir. Tahmin modellerini değerlendirmek için üç farklı hedefe yönelik bir öğrenme süreci oluşturulmuştur: (i) teklif geçmişinden yararlanarak rakibin bir sonraki teklifinin etmene faydasını tahmin etmek, (ii) rakiple ilgili değişkenleri dikkate alarak etmenin faydasını tahmin etmek, ve (iii) rakiple ilgili değişkenleri kullanarak rakibin faydasını tahmin etmek. Deneylerde gözlemlenen en iyi öğrenme modeli, gelecekteki fayda değerlerini tahmin etmenin etmenlerin müzakere başarısı üzerindeki etkilerini gözlemlemek için etmen tasarımına dahil edilmiştir. Çalışmalar çeşitli müzakere senaryolarında kullanılan bu modellerin değerlendirme sonuçlarını sunarak önerilen metodların umut verici olduğunu ortaya koymaktadır. Ayrıca, müzakere stratejisine tahminleri dahil eden 'Negoformer' isimli yeni bir teklif stratejisi sunulmaktadır ve bu stratejinin başarısı otonom müzakere senaryolarında test edilmiştir. Deney sonuçları, Negoformer stratejisinin karşılaştırılan diğer etmenlere göre bireysel fayda ve toplumsal fayda gibi müzakere metriklerinde daha başarılı olduğunu ortaya koymaktadır.

ACKNOWLEDGEMENTS

During those years that I've been working with the time series, I tried my best to find the meaning that connects those little moments and make a series out of them. Marking these moments is important I have learnt. Otherwise there wouldn't be any series worth trying to understand. The most important people who gave meaning to that series of precious moments in my life were the greatest motivators that helped me during this study. My beautiful family, my crazy friends; I owe them my sincere thanks and appreciation for their patience and unconditional love. I want to thank my advisor Reyhan Aydoğan for her mentorship and the great working environment that she provided for us members of the Interactive Intelligence Lab, where I met remarkably brilliant yet humble engineers. I also want to thank each and every one of my colleagues; especially Anıl Doğru and Berk Buzcu for their contribution and support they showed throughout this study. It was such a joy to learn with and from them. Last but not least, I owe a special thanks to the biggest support in my life, Onur Keskin. I've been lucky enough to have his support in this study and in my life, and I am grateful for that.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Agents in automated negotiations make decisions based on various factors, such as time constraints, the competitiveness of the negotiation domain, and the level of collaboration displayed by their opponents [5, 6]. Advanced strategies are designed to detect opponent behaviors and respond accordingly. Therefore, understanding the underlying motives behind these behaviors becomes a critical research question in agent-based negotiation systems. By identifying the driving forces behind other agents' actions, negotiating agents can strategically optimize their negotiation outcomes and aim to maximize their utility or enhance social welfare.

In automated negotiations, strategic decisions such as accepting the opponent's counter-offer and making offers are heavily influenced by the employed strategies, which are often based on target utility calculations [7, 8, 9]. Therefore, predicting the opponents' target utilities for future rounds presents a promising approach to understand their behaviors and act upon accordingly. Existing literature has explored several methods to detect opponent moves and target utilities to adapt negotiation strategies accordingly [10, 8]. For example, Williams *et al.* propose a Gaussian process to guess the opponent's concession rate [11]. Apart from this, there are studies that focus on modeling their own utilities by employing different techniques, such as the works that use reinforcement learning to determine the target utility based on exchanged offers and remaining negotiation time [12, 13]. These studies implicitly consider the opponent's behavior. Additionally, Chen *et al.* suggest employing transfer learning in negotiations to leverage previous negotiation experiences [14].

With the increasing demand for effective decision-making in complex domains like logistics and transportation [5, 15, 13], there is a growing interest in interpreting the behaviors of other agents in these contexts. An important aspect of this interpretation involves learning meaningful patterns and trends in an agent's behavior, and time series analysis offers a promising approach for building such predictors. In automated negotiation, Li *et al.* have employed time series analysis to recognize the opponent's strategy during negotiations [16]. Inspired by their work and recognizing the significance of estimating target utility in negotiations, this thesis shows the impacts of using time series predictors to predict the utility of the opponent's upcoming offers. In addition to that, the thesis proposes a novel agent design 'Negoformer' which is empowered with the future utility value predictions to make strategic decisions.

The primary objectives of this thesis can be summarized as follows: (i) introducing deep learning-based models, namely Long short-term memory (LSTM) [17], Transformer models [2], Informer models [3], and Autoformer models [4] to predict the utility of the opponent's future offers, (ii) investigating the impact of the opponent's strategy and the size of the negotiation domain on the performance of the developed predictors, and (iii) incorporating the best performing trained models into the agent's offering strategy so that the agent can use the insights it gains from the predictions of next steps to make decisions and make a wise decision. By predicting the utility of the opponent's future offer(s), this work aims to equip agent developers with resilient and robust negotiation strategies. Such advancements could significantly improve decision-making in automated negotiation scenarios and lead to more successful and adaptive agent behaviors.

The subsequent chapters of this thesis are as follows: Chapter 2 examines the existing literature, while Chapter 3 delves into the theoretical groundwork for the subsequent analyses. Chapters 4 and 5 represent the empirical core of the research, where a systematic evaluation of predictive models in autonomous negotiations is conducted. Chapter 6 elucidates how predictive models can be practically utilized during a negotiation and proposes a novel agent design, Negoformer. Finally, Chapter 7 concludes the thesis with future research directions.

CHAPTER II

RELATED WORK

Developing a negotiating agent is a challenging task that requires consideration of several aspects of negotiations. It must incorporate strategies for observing its environment and responding effectively to ensure a successful negotiation. When an agent holds the necessary skills and reaches a high level of sophistication, its goals go beyond simply maximizing its own profit and includes the satisfaction of the other agents involved in the negotiation as well. Ideally, the agent should also prioritize social welfare depending on the context. Considering these, the essential characteristics that a successful agent should include the following:

- An effective offering strategy considering multiple factors
- Acceptance strategy
- A reliable opponent model to deal with the uncertainty about the opponent
- Utilizing the observation of past interactions
- Getting insights about future interactions

The offering strategy defines what offer to make at any moment during negotiation and utilizes it to guide its actions. Within the literature, there exist various well-established and widely adopted offering tactics. One such approach considers the temporal aspect of negotiation, which are called as time-based strategies. Time-based strategies estimate the target utility at each timestamp while taking into account the remaining time. Faratin *et al.* have introduced two time-based offering tactics, Conceder and Boulware, which employ a function to determine the concession amount based on the slope of the utility function [8]. An agent employing Conceder tactic begins with its highest favored offer and gradually concedes over time. The Boulware tactic, on the other hand, requires the agent to concede only when the negotiation deadline approaches, which is associated with a larger utility slope. Similarly, Vahidov *et al.* developed four time-dependent concession strategies (competitive, compromising, competitive-then-compromising, and compromising-then-competitive) based on Bezier curves [18]. They evaluated the performance of these strategies in different levels of negotiation task complexities.

The downside of using the time-dependent strategies is that they are more vulnerable to exploitation by other strategies. Therefore, the agents may prefer behavior-dependent tactics as well. According to Faratin *et al.*, the prior attitudes of the opponents can serve as a basis for determining one's conduct in a negotiation. Agents employing a behavior-based strategy, mimic their opponent's behavior. For instance, the Tit-for-Tat tactic evaluates the last two offers and assesses the changes between them [19].

Depending on the negotiation rounds, a measurement of just two steps might not encompass the entirety of an opponent's negotiation strategy. The opponent may deploy a dynamic strategy that is difficult to detect using behavior-based techniques. To address this, several enhanced methods have emerged, accounting for the number of negotiation steps, as demonstrated in the work of Tunalı *et al.* [20]. Rather than focusing solely on the last two rounds, their research centers on a window of offers and measures the changes in utility. A potential trade-off in this approach is using a fixed window size, where the number of negotiation steps might fluctuate dynamically. Consequently, it may not consistently capture the dynamic nature of negotiations. Building upon this method, Keskin *et al.* propose a hybrid approach that combines time-based and behavior-based tactics, resulting in significant improvements when compared to individual strategies [9].

Another crucial aspect of negotiation involves establishing an appropriate acceptance strategy. While offering strategies hold significant importance, their power is fully realized when complemented by a well acceptance strategy. Once the target agent utility is determined, the agent can compare it with the utility of the offer it receives and make informed decisions regarding whether to accept or reject it.

Baarslag *et al.* conducted an evaluation of acceptance conditions in negotiations within their study [21]. Their work demonstrates that acceptance can be determined through various methods. For instance, AC_{next} involves comparing the agent's utility derived from the target offer with that of the received offer and accepts if the latter yields a higher utility value. $AC_{threshold}$ accepts an offer if its utility is higher than the threshold utility value. On the other hand, AC_{time} takes the remaining negotiation time into consideration and makes a decision accordingly. Depending on the observed variables, an agent can be equipped with a customized acceptance strategy.

The agent lacks any information about its opponent at the beginning of negotiations. However, with the progression of the negotiation, it can discern the opponent's behavior through offer exchanges and attempt to model the opponent's preferences. Consequently, the employed opponent model equips the agent with insights into the potential utility values that the opponent might derive from any offer, based on their historical offer exchange data.

There are various studies about opponent modeling in the literature. One of the most commonly employed opponent models is the Frequency model, which constructs opponent preferences based on the frequency of offer values and changes on negotiation issues [22]. Similarly, in line with the Frequency model, Tunalı *et al.* devise a model that considers item frequency, incorporating additional considerations related to time windows [23]. Hindriks *et al.* introduce a Bayesian Learning-based opponent model and its associated advantages [24]. Another noteworthy example is Keskin *et al.*'s Conflict-based Opponent Model (CBOM), which establishes an initial belief that agents tend to make concessions over time and updates this belief when confronted with a conflicting offering pattern from the opponent [25].

When the agent gains insights into the opponent's preferences, it becomes feasible to analyze the potential offer space and evaluate key metrics related to fairness and offer quality. Utilizing this offer space, the agent can assess the proximity of the current offer to the Pareto Efficient Frontier, the Nash Product, and the Kalai-Smorodinsky Points [26]. The Nash point represents the optimal offer in terms of gains for both agents, where the product of both utilities is maximized, and it resides on the Pareto Efficient Frontier. The Kalai point signifies the point at which it yields the fairest solution to both agents. The objective is to stay close to the Pareto Efficient Frontier since it encompasses the best possible offers for the agent under any circumstances.

Hindriks *et al.* also emphasize the significance of offer moves within the offer utility space [27]. They define six distinct moves based on the chosen points within the offer space and how their locations change over time. These moves are labeled as *fortunate*, *selfish*, *concession*, *unfortunate*, *nice*, and *silent*. Evaluating the current offer's distance from the Nash and Kalai points and observing the moves during offer exchanges can provide insights into an agent's efficiency and its commitment to social welfare. Thus, the establishment of a good opponent model and accurate estimation of the Pareto Frontier hold importance.

While the previously mentioned strategies revolve around the analysis of the opponent's historical actions, the ability to predict the opponent's future behaviors can empower a negotiating agent to act strategically. To this end, Williams *et al.* utilize Gaussian processes to predict the utility of the opponent's next offer [11], assuming that the opponent concedes over time. This allows the agent to estimate its opponent's future concessions and plan accordingly. They set the concession rate of the agent dynamically according to the Gaussian processes, and compared their agent's performance with other state-of-the-art agents.

Cao *et al.* introduce a 'segmented linear regression strategy' (SLR) model, which is designed to predict the opponent's upcoming offer patterns and adapt the model in a dynamic manner [28]. They used behavior-oriented and technical-oriented negotiation theories to form the agent's negotiation strategy and tested the strategy in computer-to-computer and human-to-computer negotiation settings with single issue.

Another research direction explores building a prediction model for the opponent's strategy. Hou *et al.* uses nonlinear regression to learn the tactic of the opponent [29]. Similarly, Li *et al.* propose using a time series prediction model to classify an opponent's strategy from a predefined set of strategies [16]. They utilize agents from the negotiation platform called Genius [30] and employ the LSTM model for recognizing the opponent's strategy. Inspired by these studies, the first phase of this thesis also utilizes LSTM model, along with other deep-learning based models. However, rather than predicting the predefined strategies, this thesis focuses on predicting the utility values for the future rounds in an automated negotiation setting with multiple issues.

Leveraging state-of-the-art predictive models can be highly effective in predicting the utility values obtained from future offer exchanges. Given their success in various research fields, this thesis investigates the application of several prediction models to time series prediction tasks. The prediction models explored in this thesis include LSTM [17], Transformers [2], Informer models [3], and Autoformer models [4].

CHAPTER III

BACKGROUND

This chapter offers background information on the primary components addressed in this thesis. It begins by exploring the dynamics of automated negotiation, covering the negotiation protocol, some essential negotiation strategies, and the interpretation of offering behaviors. Furthermore, it delves into explaining the predictive models employed across the studies in the thesis for learning time series data.

3.1 Automated Negotiation

In automated negotiation, agents engage in negotiations over a finite set of issues, denoted as $\mathcal{I} = \{1, 2, \ldots, n\}$. Each issue $i \in \mathcal{I}$ has a range $\mathcal{D}_i$ of possible values, and an outcome, $O \in \Omega$, represents a complete assignment of issues with Ω being the Cartesian Product of the ranges of instantiations for each issue $\Omega = \mathcal{D}_1 \times \mathcal{D}_2 \times \ldots \times \mathcal{D}_n$. A utility function maps negotiation outcomes to real numbers within the range [0, 1], indicating the desirability of each outcome. In other words, the utility function serves as a mathematical representation of the agent's preferences and typically additive utility functions are used in agent-based negotiations. Equation 1 shows the formal representation of an additive utility function, where w_i represents the weight or importance of the negotiation issue I_i (i.e., issue weight), O_i represents the value for issue i in offer O, and V_i is the valuation function for issue i, providing the desirability of the issue value. It is assumed that $\sum_{i \in n} w_i = 1$ and the domain of V_i is (0, 1) for any i. An issue value is preferred when its valuation V_i is higher. Negotiating agents utilize their utility functions to determine what offers to make and decide when to accept their opponent's offers.

$$\mathcal{U}(O) = \sum_{i=1}^{n} w_i \times V_i(O_i) \tag{1}$$

Every negotiation must adhere to a protocol. *Stacked Alternating Offers Protocol* (SAOP) is a protocol proposed by Aydoğan *et al.* [31]. Under this protocol, specific actions and conditions dictate under which conditions the decisions can be taken and when the negotiation stops. The interactions cease either when agents reach a consensus or when they reach a predetermined deadline. The negotiation starts with an initial offer made by one of the agents. At each round, the agent that received the offer can choose to (i) accept the received offer, (ii) propose a counteroffer, or (iii) end the negotiation without reaching an agreement. This interaction follows a turn-taking approach and persists until an agreement is achieved or the deadline is met. It is important to note that, initially, agents lack knowledge about their opponent's preferences and negotiation strategies. However, they may attempt to gradually infer these preferences and strategies over time by analyzing offer exchanges.

A sophisticated agent should have a well-defined target utility for its offering strategy to decide on how and when to act. These target utility estimations consider various aspects like time, the behavior of the opponent, or a combination of them. Since these variables are subject to change at each negotiation round, the estimated target utilities are usually defined as a function of these variables.

Negotiation strategies serve as the cornerstone of an agent design process. While an agent's design can be enriched with elements such as opponent models and predictions for upcoming rounds, the effective utilization of these enhancements depends on having a well-structured negotiation strategy. Therefore, it is crucial to employ tactics that effectively control offering behavior and acceptance criteria.

In the process of selecting negotiation strategies, the agent developer must thoroughly assess the negotiation environment, define its constraints, and subsequently opt for the most suitable negotiation tactic. For instance, certain factors like the size of the negotiation domains significantly impacts the offer space, thereby impacts the complexity of the offer generation process. Likewise, the overall negotiation time plays an important role since it potentially impacts the concession decisions, particularly when there is a risk of failing to achieve a consensus by the end of the negotiation. It is also critical to investigate the full offer space and how it affects the behavior of competing agents. However, in cases where time is restricted, the agent may not be able to thoroughly investigate all possible choices. Consequently, agent strategy development must encompass consideration of all these factors with the goal of producing robust performance under varying conditions. While incorporating insights into the agent's decision-making process holds value as proposed in the studies of this thesis, it is equally important to ensure that the agent processes and adapts to these insights properly.

The hybrid negotiation strategy proposed by Keskin *et al.* [9] combines the time based concession strategy [18] with an improved behavior-based tactic [9]. Equation 2 shows the target utility function employed by this hybrid strategy.

$$\mathcal{TU}_{Hybrid} = (t^2) \times \mathcal{TU}_{Time} + (1 - t^2) \times \mathcal{TU}_{Behavior} \tag{2}$$

The time-based tactic TU_{Time} used in the target utility function of hybrid strategy is introduced by Vahidov *et al.* [18], as shown in Equation 3. This time-based tactic includes three main parameters in its target utility function; P_0, P_1, and P_2. P_0 represents the initial target utility value, P_2 shows the final target utility value, and P_1 shows the slope between these utility values. If P_1 is equal to the average of P_0 and P_2, this means there is a linear slope. If this value is greater than the average value, the agent tends to act more Boulware (i.e., not concedes for a while but tends to concede more when the deadline approaches), otherwise it will have more tendency to concede.

$$\mathcal{TU}_{Time} = (1 - t)^2 \times P_0 + [2 \times (1 - t) \times t \times P_1] + t^2 \times P_2 \tag{3}$$

The behavior-based tactic used in hybrid strategy is an improved mimicking tactic inspired by Tit-For-Tat [9][19]. The Tit-For-Tat tactic uses a fixed empathy score throughout the negotiation to adjust the level of mimicking. The difference of Behavior-based tactic used in the hybrid strategy is that it uses a time-based empathy score instead of a fixed constant. It also considers a window of opponent's offers while calculating ΔU, rather than checking for the difference of last two offers as in the original Tit-For-Tat equation.

Equation 4 shows the calculation of the target utility of Behavior-based tactic. In the formulation, $U(O_j^{t-1})$ shows the agent's utility for the previous offer. μ symbolizes the time-dependent mimicking parameter to scale the change of utility, as shown in Equation 6. This value is multiplied with ΔU, which represents the utility change over the last n consecutive offers, as shown in Equation 5. W_i used in the formulation indicates the weights of each pair of utility differences.

$$\mathcal{TU}_{Behavior} = U(O_j^{t-1}) - \mu \times \Delta U \tag{4}$$

$$\Delta U = \sum_{i=1}^{n}[W_i \times (U(O_h^{t-i}) - U(O_h^{t-i-1}))] \tag{5}$$

$$\mu = P_3 + t \times P_3 \tag{6}$$

The aforementioned strategies can be summarized as follows:

- ***Time-based tactics***: These strategies calculate a target utility by considering the remaining time, and generate an offer with that target utility. The agents use a function of normalized negotiation time to calculate the utility of their coming offer. These agents tend to concede on their target utility as time goes by. The main difference between the Conceder and Boulware agents is the concession curve they show [8]. While the Conceder agent concedes quickly, the Boulware agent tends to concede only when the deadline is approaching.

- ***Behaviour-based tactics***: The opponents with this strategy are expected to be opponent-aware and act accordingly. The agents that adopt a behavior-based strategy mimic the behavior of the opponent agent. For this reason, the agent calculates the difference of utilities between the opponent's two consecutive offers and applies a proportional concession amount.

- ***Hybrid strategy***: The agents employing this strategy decide on their approach by considering both remaining time and the behaviors of the agent they negotiate with. The target utility is determined as a linear combination of the target utilities calculated by time-based and behavior-based strategies. The principal intuition is when time is not significant (e.g., at the beginning of the negotiation), the agent pays more attention to its opponent's behavior during deciding on its next offer's target utility. As the deadline approaches, the agent cares more about the remaining time and tends to find an agreement urgently. This study uses two versions of this agent with and without opponent modeling.

In order to examine the changes in the behavior of negotiating agents throughout the negotiation, Hindriks *et al.* establish six negotiation moves [1]. A move is defined by the difference in utility values for both sides regarding the subsequent offers. These moves are labeled as *fortunate*, *nice*, *concession*, *selfish*, *unfortunate*, and *silent*. Table 1 demonstrates the calculation of move types where ΔU_A and ΔU_{Op} represent the utility difference for the agent and its opponent.

Table 1: Move specification of a negotiating agent [1]

	Self Difference	**Opponent Difference**
Silent	$\Delta U_A = 0$	$\Delta U_{Op} = 0$
Nice	$\Delta U_A = 0$	$\Delta U_{Op} > 0$
Concession	$\Delta U_A < 0$	$\Delta U_{Op} > 0$
Unfortunate	$\Delta U_A < 0$	$\Delta U_{Op} < 0$
Fortunate	$\Delta U_A > 0$	$\Delta U_{Op} > 0$
Selfish	$\Delta U_A > 0$	$\Delta U_{Op} < 0$

3.2 Predictive Models for Learning Time Series Data

The rapid improvement in technology has opened up possibilities for exploring different approaches for learning tasks. While the focus on machine learning research is predominantly on advancing the processing techniques for text and image cases, learning from time series data still keeps its significance in many of real life problems.

Time series data refers to a sequence of data points collected and recorded over time [32]. This type of data is prevalent in various domains, including finance, economics, weather forecasting, sensor data analysis, and many others. Analyzing and predicting patterns within time series data can provide valuable insights for decision-making and future planning.

Like any learning task, learning from time series data is a challenging task due to many dependencies on the data characteristics. Before implementing any learning method to the data, it is important to observe the unique characteristics of it and select the proper algorithms wisely. The main challenges of time series data are as follows:

- **Temporal dependencies:** Time series data shows temporal dependencies since every data point is affected by its prior observations. Making accurate predictions or understanding important trends depends on accurately capturing these dependencies. However, it is challenging to effectively model and exploit these dependencies.

- **Non-stationarity:** Time series data frequently displays non-stationarity, which means that its statistical characteristics change over time. The distribution of the data can vary according to trends, seasonality, and other underlying patterns, which makes it difficult to model and predict future behavior with good accuracy.

- **Noise and outliers:** Time series data is subject to noise and outliers, which can obscure patterns and have an impact on learning models' effectiveness. Maintaining the accuracy and robustness of the models depends on the ability to detect and handle noisy or abnormal data inputs.

- **Irregular sampling:** Time series data may have timestamps that are inconsistent or unevenly spaced, which makes it difficult to handle missing data or precisely align observations. Specialized methods are needed to handle the gaps and provide reliable analysis and modeling when dealing with irregular sampling.

Considering all these unique characteristics of time series data, selecting the most proper model architecture for learning task is challenging. It requires a detailed data exploration to identify the most suitable algorithm for the target data. A sequence-based data might contain a trend, seasonal variations, cyclical fluctuations or irregular variations over time. It is also important to observe if there are other accessible factors that might have an impact on the data behavior.

There are a number of learning options, including the statistical models like ARIMA, hybrid methods, and neural network models like recurrent neural networks (RNNs). This section goes through some of these models that can be used for time series prediction tasks.

3.2.1 Autoregressive Integrated Moving Average (ARIMA)

Time series analysis models are statistical models that are designed for analyzing a collected time series data over time and forecast the future values according to their trends, seasonality and other temporal dependencies. There are several types of time series analysis modeling options that are widely used in forecasting tasks mostly in univariate series.

ARIMA models decompose the sequential data into three parts and builds a model by creating a linear combination of these components. The name of the model comes from the components 'autoregression' (AR), 'integrated' (I), and 'moving average' (MA). It is a well-performing and explainable algorithm, however it is not able to capture the dependencies between different variables.

3.2.2 Recurrent Neural Network (RNN) Models

The Recurrent Neural Network (RNN) is a sequence-based deep learning model, where the outputs from previous steps are fed into the subsequent steps. This mechanism allows the RNN to carry knowledge forward, functioning like a memory, which makes it suitable for tasks involving sequence and order, such as language modeling, speech recognition, and image recognition. Similar to other neural network models, RNN employs backpropagation to calculate and minimize errors [33]. However, during the backpropagation process, the gradients can either become very small or very large due to the accumulated information from previous steps, a phenomenon known as the vanishing/exploding gradient problem. To mitigate this issue, several variations of RNN models have been developed, with one of the most commonly used ones being Long Short-Term Memory (LSTM) models.

Long Short-Term Memory (LSTM) Models: LSTM is a specialized architecture within the framework of Recurrent Neural Networks (RNN) [17]. This architecture utilizes memory gates to control the inputs that need to be used in a recurrent way. These memory gates enhance the model's ability to make predictions for historical datasets characterized by sequential patterns.

The gating mechanism in LSTM models enables the preservation of inputs from specific time steps and the transfer of information retained at those time steps to subsequent stages. This feature, combined with its activation functions, establishes nonlinear relationships between inputs and generates predictions accordingly. LSTM models tend to excel when working with substantial datasets.

There are three essential gate mechanisms utilized in the LSTM model to control

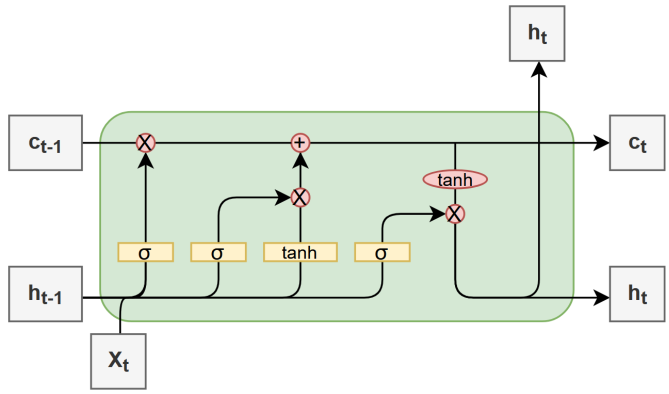


Figure 1: LSTM layer

the data flow as illustrated in Figure 1: the input gate, the memory gate (or forget gate), and the output gate. The input gate determines which new information is retained in the long-term memory. The forget gate determines whether to retain or discard information from the long-term memory and controls the quantity to be retained. The outputs of the input gate and the forget gate combine to reshape the long-term memory, which is then passed to the output gate. The output gate receives new input values, the information stored in the short-term memory and recently generated long-term memory and then returns the updated short-term memory.

3.2.3 Transformer Models

Transformers are one of the best performing models in natural language processing [2]. This model created a groundbreaking impact in the machine learning research thanks to the novel architecture called *self-attention mechanisms*. This mechanism enables the model to process sequences in parallel; therefore, making them highly efficient for a wide range of sequence-based tasks such as language translation, text generation and image recognition. The power of transformer models to capture contextual information and learn from large datasets has contributed to their widespread use and improved performance in a many applications. Figure 2 shows the blocks and components utilized in the model architecture.

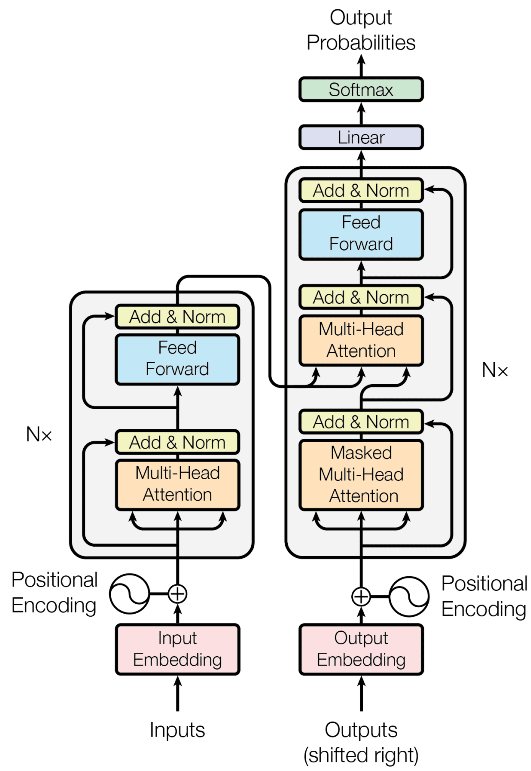


Figure 2: Transformer model blocks [2]

Although the model has been shown to be quite efficient in many fields, it does have significant limitations. Three main challenges of regular transformer models are listed as follows:

1. The quadratic computation of self-attention

2. The memory bottleneck in stacking layers

3. The speed plunge in predicting long outputs

The following methods modify the Transformer model to find solutions to these problems and make it usable in time series predictions.

Informer Models: The Informer was developed for time series forecasting tasks on long sequences of data [3]. The algorithm was designed to efficiently manage temporal dependencies in time series data and is based on the Transformer architecture, which was initially proposed for natural language processing. In order to properly handle time series data, the Informer algorithm makes a number of changes to the Transformer design. The approach specifically includes a new encoder that takes advantage of the temporal relationships in the data and a new decoder that makes predictions for the future using both the input data and predicted values from past time steps. The model architecture is shown in Figure 3.

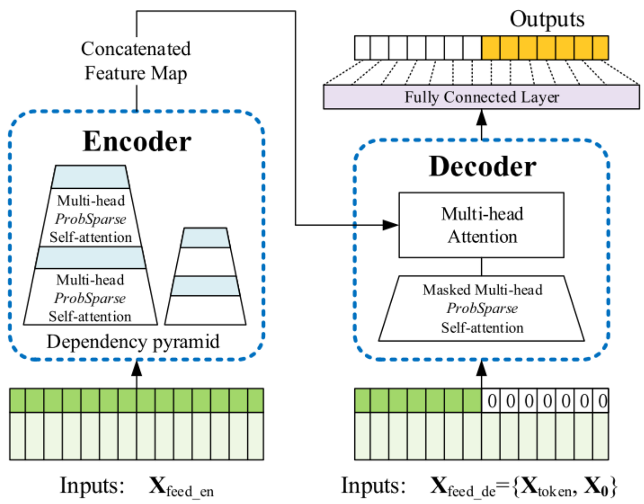


Figure 3: Informer model architecture [3]

The Informer algorithm's encoder is composed of several self-attention layers, similar to the original Transformer model. However, the Informer encoder uses a sliding window of fixed length rather than a sequence of input tokens to capture the temporal dependencies within the input sequence. The sliding window method reduces the computational cost of the model while improving the encoder's ability to capture long-range relationships in the data. The decoder in the Informer algorithm uses an autoregressive approach to generate future predictions. Particularly, the decoder uses the projected values from earlier time steps' ground truth values as input at each time step. This enables the forecasting process to consider both the predicted and actual values, which helps mitigating error propagation over time.

The Informer approach contains these adjustments as well as a number of additional performance-enhancing methods, such as layer normalization, residual connections, and a hybrid loss function that combines mean squared error with mean absolute error. Overall, with a focus on capturing the temporal connections within the data and minimizing error propagation over time, the Informer technique is a fast and effective approach for time series forecasting on long sequences.

Autoformer Models: Autoformer models successfully modify the self-attention mechanism of the Transformers to 'Auto-Correlation' mechanism to decompose the components of the data and process them separately [4]. This way, calculating relationship between the current and past values becomes easier and more achievable. This mechanism employs the Fast Fourier Transform to calculate the auto-correlation. With that, it can find the period-based dependencies.

Figure 4 shows the complete Autoformer structure. The encoder and decoder blocks are utilized similar to the vanilla Transformer model and includes the Auto-Correlation mechanism instead of self-attention mechanism. The model shows good performance on many domains compared to its alternatives.

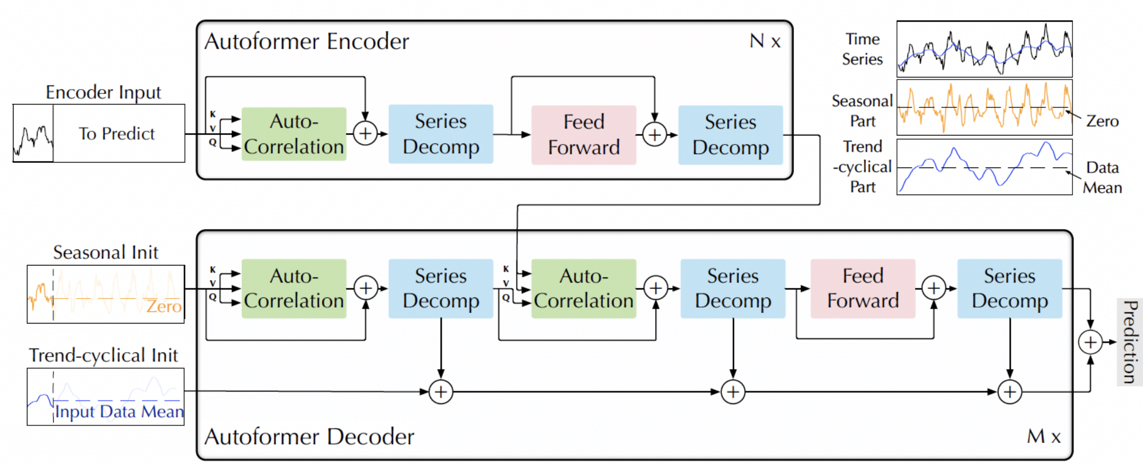


Figure 4: Autoformer model architecture [4]

CHAPTER IV

LEARNING OPPONENT'S BEHAVIOR BASED ON AGENT'S OWN EXPERIENCES

Understanding agent behavior can encompass various aspects since a sophisticated agent comprises multiple components that collectively determine its negotiating approach in diverse conditions, thereby shaping its overall behavior. The strategies utilized by agents to determine the best offers, assess valuable offers, and decide on the most favorable acceptance moments form the key elements of this behavior. The utility values that agents can obtain for a given offer play a significant role in shaping the strategies that ultimately define their behaviors. Therefore, the focus of this study is on learning the agent behavior through the prediction of agents' utility values based on the agent's own experiences.

While the primary objective is to learn agent behaviors by their utility values and assess the performance of selected machine learning models, it is crucial to remind that this objective can be fulfilled in different sophistication levels. To make wise decisions, agents might use an opponent model and/or a strategy that helps the agent to navigate throughout a negotiation session. Identifying the significance of these indicators in the prediction models is a key goal of this study, as it holds importance in any machine learning project. Therefore, a learning framework that incorporates different indicators in the prediction model is employed for each experimented machine learning model [34]. The following sections introduce this learning framework and present the conducted study with different settings applied to the machine learning models, shedding light on the crucial indicators that contribute most significantly to learning agent behaviors.

4.1 Learning Framework

The primary objective of this study is to develop a prediction module capable of generating utility value forecasts for the next steps in bilateral negotiations. By doing so, the decision-making process of agents can be enhanced, enabling them to avoid making offers with lower utility than their opponent's next offer, thereby preventing the loss of potential gains. In scenarios where agents negotiate with the same opponent multiple times, this predictive capability allows them to anticipate their counterpart's behavior and make strategic decisions accordingly, facilitating the achievement of a consensus more efficiently. These predictions play a crucial role in capturing opponent behavior trends and identifying their strategies, even with the collected data during a negotiation session. Once they are obtained, the decision making design of the agent can be enriched with these insights.

Before constructing a prediction model to capture these trends and strategies, it is essential to prepare the learning framework that will be utilized with the models. Since the chosen variables directly influence the model performances, it is crucial to identify the significant indicators and input them into the model wisely.

Consequently, we proposed a learning framework [34]. The proposed learning framework consists of three key objectives:

- ***Objective 1:*** Predicting the utility of its opponent's coming offer for itself, $U_A(O_{t+1})$, based on remaining time and offer exchanges so far.

- ***Objective 2:*** Improving the performance of the prediction gained in *Objective 1* by considering additional features such as estimated opponent utility, Nash Distance, and opponent's moves.

- ***Objective 3:*** Predicting its opponent's estimated utility of its opponent's next offer, $\widehat{U}_{Op}(O_{t+1})$, based on remaining time, offer exchanges so far and additional features used in *Objective 2*.

Accordingly, Figure 5 depicts the inputs used in the utility value prediction to achieve the aforementioned objectives. In any analysis, a negotiating agent can monitor all offers made during the negotiation and calculate its utilities by using its utility function ($< U_A(O_0), ..., U_A(O_t) >$) at time t. To achieve the first objective, a recommended approach is to utilize a time series predictor that considers both the remaining time (t_{remain}) and the agent's utility for a group (referred to as 'window' throughout the study) of past consecutive offer exchanges ($< U_A(O_{t-k}), ..., U_A(O_t) >$). The size of the window k can be easily customized in various learning settings, allowing for flexible adjustments according to specific preferences and requirements. Similar to the window size, the length of the predictions n can be customized as well.

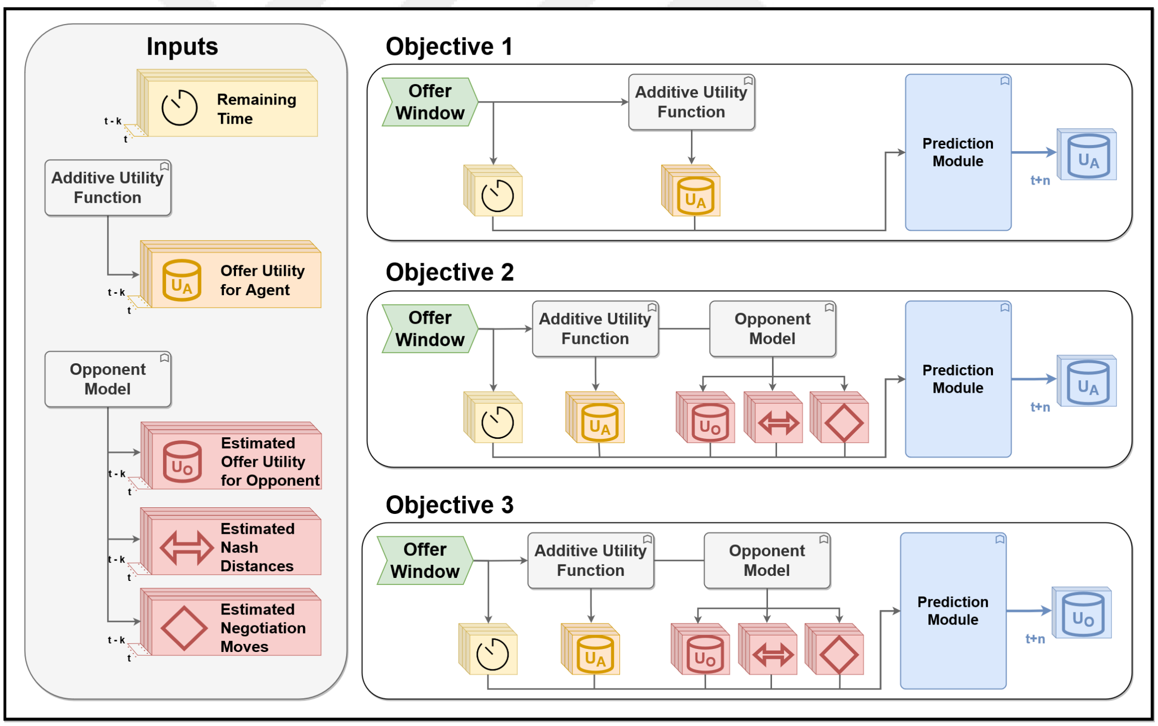


Figure 5: Inputs and outputs for utility value prediction of coming offers

For *Objective 2*, the incorporation of opponent's estimated utilities in the offer exchanges, negotiation move analysis, and Nash distances is suggested. While the agent lacks direct access to the opponent's preferences, employing opponent modeling allows it to learn the opponent's preferences over time in terms of utilities. As a result, the agent can utilize additional estimated information, including the opponent's utility for a given offer, negotiation move types, and the Nash distances of the offers. It is essential to note that the accuracy of the chosen opponent model may impact the performance of the predictions for the utility value of upcoming offers. Existing opponent modeling studies from the literature can be utilized for this purpose [10].

In summary, the second predictor is fed by the following inputs:

- Agent's utilities for the offer window, $< U_A(O_{t-k}), ..., U_A(O_t) >$,
- Estimated opponent utilities for the offer window, $< \widehat{U}_{Op}(O_{t-k}), ..., \widehat{U}_{Op}(O_t) >$,
- Nash distances for the offer window, $< \Delta N(O_{t-k}), \Delta N(O_t) >$,
- Estimated moves for the offer window, $< m(O_{t-k}, O_{t-k+1}), ..., m(O_{t-1}, O_t) >$,
- Remaining negotiation time, t_{remain}.

For achieving the third objective, the same inputs employed in *Objective 2* are utilized, but the model's output is defined as the estimated opponent utility for the next offer, $\widehat{U}_{Op}(O_{t+1})$. All offer-related inputs can be preserved as a time series throughout the negotiation process, enabling the application of supervised learning models to comprehend the sequential behaviors of negotiating agents and make predictions for the utility values accordingly. Given the significance of the selected inputs during negotiation sessions, it becomes essential to choose an appropriate multivariate time series prediction method.

The learning framework is intentionally designed to offer flexibility, making it adaptable to different negotiation settings with varying machine learning models, opponent models, and window sizes as per specific requirements. Consequently, this study employs this learning framework and tests the performance of different predictive models on these objectives.

4.2 Prediction Approach

In this phase of the thesis, a prediction module is developed to guess the utility values of the next coming offer for the agent engaged in repeated negotiations with selected counterparts. This predictive module offers several advantages, including gaining valuable insights about opponent behavior, fostering trust and cooperation, adaptability in response to changing opponent behaviors, enabling long-term planning, and improving negotiation efficiency. As the agent interacts repeatedly with the same opponents, the predictive model's ability to foresee their behaviors becomes increasingly valuable. The baseline agent can dynamically adjust its negotiation strategies by utilizing the predicted utility values, which leads to more effective communication and collaboration with the selected agents over time.

To accurately predict the next-step utility values in repeated negotiations, choosing a suitable learning method capable of capturing the dynamics of negotiation sessions is significant. In this study, two models, LSTM and Transformer, are meticulously selected for their distinct strengths that align well with the task at hand. LSTM and Transformer models are suitable for predicting next-step utility in time series data due to their capabilities in handling sequential dependencies and complex patterns. LSTM's recurrent architecture enables it to capture long-term dependencies, which makes it effective in understanding how past interactions influence current utility values in repeated negotiations. On the other hand, Transformers' self-attention mechanism enables focusing on relevant interactions and processing the negotiation history in parallel. This characteristic makes the model adaptable to complex and dynamic negotiation environments. Utilizing these two algorithms can provide valuable insights into their respective strengths and weaknesses, along with their suitability to the given negotiation dataset's characteristics.

The mentioned models receive sequential information using a sliding window approach. This approach involves selecting a window size based on a specific number of time steps. During each iteration, the window slides by one step, and the models utilize the data within the window. The inputs for each iteration are not only dependent on the current step but also include the previous inputs captured by the window, as illustrated in Figure 6.

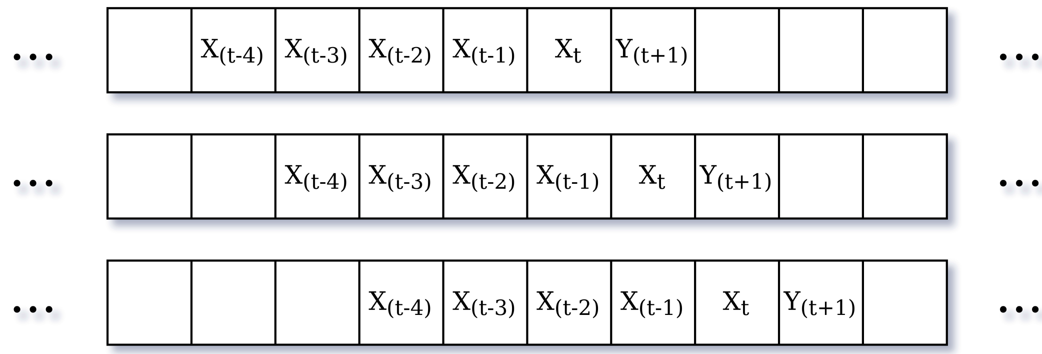


Figure 6: Sliding window approach of model input and output

4.2.1 Long Short-Term Memory (LSTM) Model Architecture

Recurrent Neural Network (RNN) deep learning architectures are well-suited for processing sequential inputs, which makes it highly relevant for handling the time-dependent nature of repeated negotiations. As a widely used neural network, RNN's inherent ability to model temporal dependencies within its architecture allows it to capture the evolving patterns and behaviors in the negotiation history. To address the vanishing gradient problem commonly encountered in deep neural networks, a variation called LSTM (Long Short-Term Memory) [17] is introduced. The vanishing gradient problem occurs when the calculated gradient approaches zero during back-propagation, resulting in the loss of valuable information during model training. LSTM overcomes this challenge by incorporating memory gates, which effectively control the input flow and enable the preservation of information from long sequences throughout the training process. This way, LSTM can process the negotiation history in a unidirectional manner, ensuring crucial information is retained and utilized to make accurate predictions in the context of repeated negotiations.

Figure 7 shows the proposed LSTM model on the left-hand side, while the right-hand side demonstrates the general LSTM architecture.

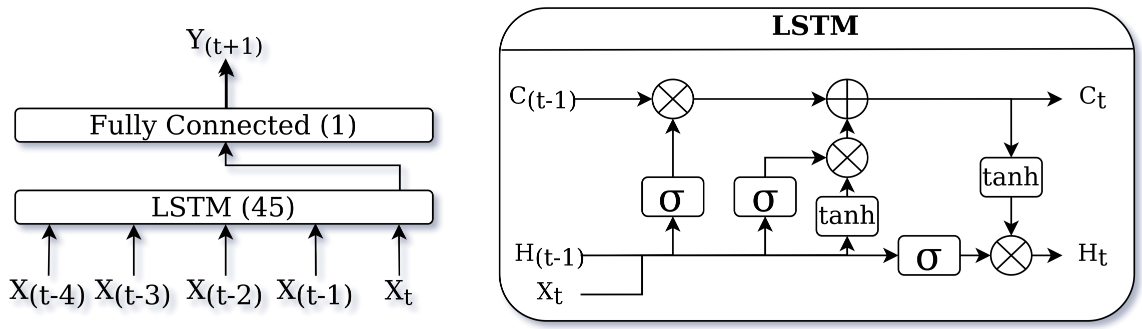


Figure 7: Proposed LSTM model and the LSTM architecture

In our study, we utilized an LSTM architecture that consists of one LSTM layer with 45 units and a fully connected layer for making predictions, as illustrated in Figure 7. The window size is selected as five, therefore the model takes a sequence of the last five inputs: $X_{(t-4)}$, $X_{(t-3)}$, $X_{(t-2)}$, $X_{(t-1)}$, $X_{(t)}$, and predicts the next value $Y_{(t+1)}$.

The selected parameters for this LSTM architecture are tuned to efficiently capture the sequential patterns and dependencies present in recurring negotiations while mitigating unnecessary complexity. The details of these parameters are as follows:

- Batch size: 32
- Loss function: 'Mean Squared Error'
- Optimizer: 'RMSProp'
- Learning rate: 0.001
- Number of epochs: 50

It should be noted that the term 'window size' can be referred to as 'time steps' in several resources, as these terms are used interchangeably in the context of RNNs.

4.2.2 Transformer Model Architecture

Transformer models became notable for their self-attention mechanism, which allows them to establish dependencies between inputs and outputs without relying on recurrences in neural networks (i.e., allowing data to be processed regardless of a direction). The attention mechanism links the positions of input and output sequences and traverses among these sequences to observe where to pay attention. This approach enables the model to prioritize relevant information effectively. Compared to recurrent layers that require $O(n)$ sequential operations, the Transformer architecture employs a constant number of operations, leading to lower complexity and efficiency, especially in tasks involving very long sequences.

With the introduction of Transformer models, the attention of the deep learning researchers has been shifted remarkably on self-attention mechanisms and the improvements that can be obtained with them. Due to its performance on sequential datasets, the application of these mechanisms on time series predictions are also started to gain interest. Recent studies show that Transformer-based architectures can be used to predict time series with few alterations [35]. Wu *et al.* employed an encoder-decoder approach with a window size of five for training the time series and predicting the next step [36]. Inspired by these findings and after experimenting with various Transformer-based architectures, the Transformer model in this study uses an encoder-decoder model with shifted inputs for predicting utility values in negotiation. This architecture, as illustrated in Figure 8, is specifically designed to predict time series data.

The left-hand side of Figure 8 shows the model's encoder. The encoder in this model takes the first three inputs of the sequence ($X_{(t-4)}$, $X_{(t-3)}$, $X_{(t-2)}$) and passes them through two Transformer blocks. The study that introduce attention layers by Vaswani *et al.* employs input embedding for the 'token2vector' operation. However, in this research, a fully connected layer is utilized rather than input embedding to

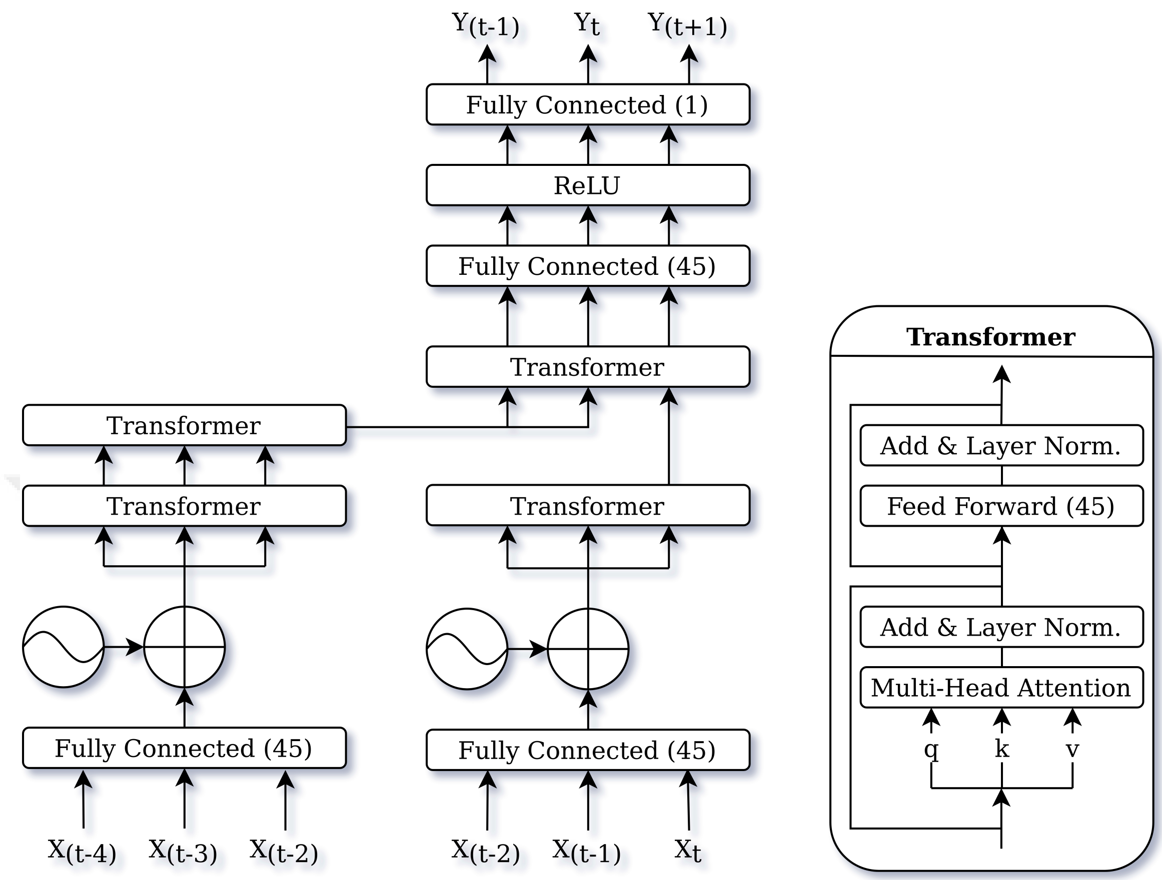


Figure 8: Proposed Transformer-based encoder-decoder model

better suit the requirements of time series prediction tasks. Employing fully connected layers in the encoders aligns with the approach used by Wu *et al.* [36]. Other than this alteration, the other components of the original Transformer structure remain unchanged.

The model's decoder is demonstrated at the middle section of Figure 8. As shown in the figure, it takes the last 3 inputs of the sequence of determined window ($X_{(t-2)}$, $X_{(t-1)}$, X_t), and predicts the shifted outputs ($Y_{(t-1)}$, Y_t, $Y_{(t+1)}$). The decoder part of the model also uses a fully connected layer instead of input embedding, similar to the encoder. The decoder contains two Transformer blocks which are the same blocks used in the encoder, with the only difference of the inputs. There, the outputs obtained from the encoder are fed into the second Transformer block of the decoder as inputs.

After including and processing these additional inputs in the second Transformer block, the decoder utilizes two fully connected layers and return the final outputs.

In order to have a consistent output schema, the window size used in the adjusted Transformer model is kept as five, same as LSTM model. The remaining details of the model parameters are listed as following:

- Batch size: 32
- Loss function: 'Mean Squared Error'
- Optimizer: 'RMSProp'
- Learning rate: 0.001
- Number of epochs: 50

4.3 Experimental Setup

The selected model architectures require historical negotiation sessions to model the opponent agent's behaviors and effectively capture them during negotiations. To accomplish this, multiple negotiations were conducted using the GeniusWeb negotiation framework to collect the negotiation sessions and use for training the models. GeniusWeb is the Python-based web version of Genius [30] negotiation platform, which allows agent developers to design agents in Python and supports various machine learning libraries for time series predictions. However, direct use of the agents that are available in Genius is not possible on this platform. Therefore, a negotiation tournament is prepared in GeniusWeb by integrating a baseline agent, and the sessions have been collected for this baseline agent.

The baseline agent is developed to use in the negotiation tournaments and the learning process. The developed baseline agent employs a hybrid offering strategy [9] with a frequency-based opponent model [23]. This baseline agent then used in the negotiations against five available agents, namely Boulware, Conceder, behavior-based, hybrid, and another hybrid strategy with opponent modeling.

The negotiation tournaments are designed by including various domains and preference profiles for bilateral negotiations to assess the performance of proposed approaches. Nine negotiation domains that are available in the GeniusWeb negotiation platform are selected based on their characteristics and representativeness for improving the model performances. Table 2 shows the information of selected domains in which the negotiation sessions are held for training and testing purposes. The negotiation dynamics can be different according to the negotiation domains. Therefore, two large, two medium, and two small-sized domains have been selected for training sessions, whereas one domain for each size group has been selected for testing purposes. The baseline agent interacts with all opponents in each negotiation domain and both preference profile options that the domain presents. The negotiations get repeated ten times for each negotiation configuration with a deadline of three minutes for each session. After completing the negotiation sessions with five opponents, two preference profiles, and six domains for ten times, 600 different negotiation sessions were obtained to use in the training process. Addition to that, 60 more negotiation sessions are collected after negotiating with five opponents, two profiles, and three different domains twice for testing purposes.

Table 2: Domain details for both training and test data

	DomainID	**Category**	**Issue & Value List**	**Offer Space**	**Opposition**
Train	1	Small	[2, 3, 2, 2, 2, 3, 3]	432	0.189
	2	Small	[8, 2, 2, 8, 2]	512	0.096
	3	Medium	[5, 11, 8, 9]	3,960	0.260
	4	Medium	[17, 4, 5, 3, 2]	2,040	0.056
	5	Large	[26, 20, 8, 4]	16,640	0.281
	6	Large	[26, 22, 10, 8]	45,760	0.255
Test	7	Small	[2, 3, 2, 2, 11, 2]	528	0.080
	8	Medium	[4, 3, 2, 2, 5, 5, 3]	3,600	0.191
	9	Large	[26, 8, 26, 2]	10,816	0.280

Once the tournament is complete and the data is collected, they are separated based on the size of domains since it is important to observe how the negotiation performances, hence the prediction performance, are impacted by the size of negotiation domains. Therefore, collected negotiation sessions have been grouped by domain sizes before using in model training. When the training data becomes ready, training sessions for each objective have been conducted with prepared data sets.

4.4 Results

After training neural networks (LSTM and Transformer) on 600 negotiation sessions with the provided configurations, predictions were generated for the opponent's next-step utility values using the test data. The accuracy of these predictions was assessed using two evaluation metrics: root-mean-square error (RMSE) and mean absolute percentage error (MAPE). The predictions were made for a total of 60 negotiation sessions, and the final results were obtained by averaging the calculated RMSE and MAPE metrics. Additionally, an analysis was conducted to understand the impact of the domain size on each objective. Notably, *Objective 1* and *Objective 2* share the same target variable, which is the utility value of the agent for the opponent's next offer. *Objective 3* uses the same opponent-related input variables as *Objective 2* during training, but its target is to predict the opponent's own utility. As a result, the evaluations were conducted, considering these differences in three main objectives.

The performance of the predictions for *Objective 1* and *Objective 2* was assessed first. The MAPE scores for each model on different domain sizes are presented in Figure 9. The blue bars represent the prediction performance on all test scenarios. The results indicate that *Objective 1* achieved a prediction with 17% and 18% MAPE score using Transformer and LSTM, respectively, with the Transformer slightly outperforming LSTM. Furthermore, it is evident that incorporating opponent-related features, such as the estimated opponent's utility of previous offers, their Nash distances, and estimated opponent moves, leads to a decrease in prediction errors. Thus, *Objective 2* is successfully achieved.

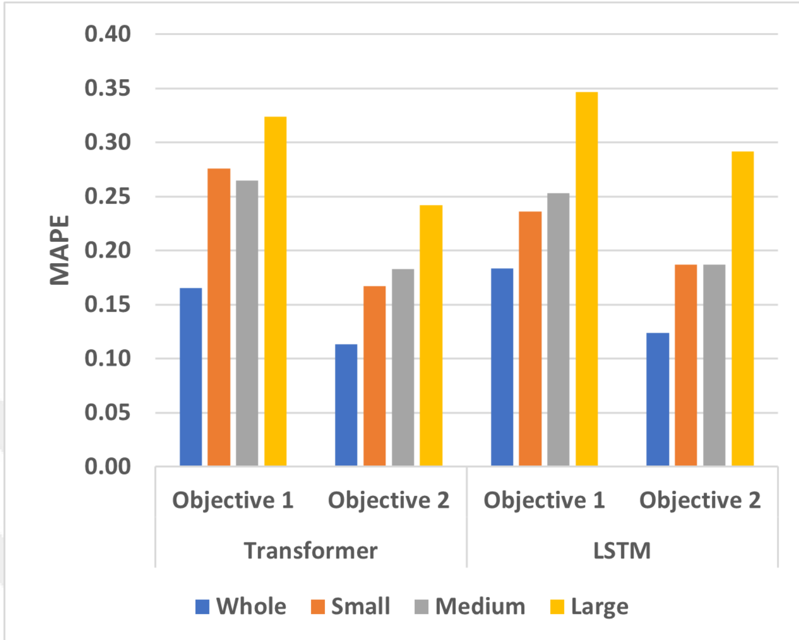


Figure 9: Model comparison among different domain sizes for *Objectives 1 & 2*

To explore the impact of domain size on the performance of Transformer and LSTM models, separate predictors were trained for different scenarios, categorized as small, medium, and large domains, represented by orange, grey, and yellow bars, respectively. Notably, predicting agents' utilities on large domains yielded the highest error rates regardless of the model used. This is likely due to the complexity of possible offer distributions in large negotiation domains, where a regular negotiation session covers only a small portion of the offer space, potentially limiting the model's generalization ability. Conversely, models trained with the entire dataset exhibited superior performance as they were exposed to more diverse examples during training. Figure 10 provides a detailed analysis, summarizing both MAPE and RMSE scores for each setting of *Objective 1* and *Objective 2*. Evidently, the RMSE and MAPE scores of *Objective 2* are lower than those of *Objective 1*, which indicates improvements in both models when enriched with additional opponent-related variables, as expected.

Domain	Model	MAPE		RMSE	
		Obj 1	*Obj 2*	*Obj 1*	*Obj 2*
Whole	Transformer	0.17	0.11	0.10	0.08
	LSTM	0.18	0.12	0.12	0.08
Small	Transformer	0.28	0.17	0.17	0.11
	LSTM	0.24	0.19	0.16	0.13
Medium	Transformer	0.26	0.18	0.13	0.11
	LSTM	0.25	0.19	0.13	0.11
Large	Transformer	0.32	0.24	0.16	0.13
	LSTM	0.35	0.29	0.22	0.18

Figure 10: Model comparison among different domain sizes for *Objectives 1 & 2*

Evaluating *Objectives 1* and *2* is crucial for understanding the opponent's behavior from the agent's perspective of utility change and predicting the opponent's utilities for their upcoming offers. As most opponents generate offers based on their own utility, the ability to predict their utilities becomes vital. Hence, *Objective 3* aims to learn the opponent's utility, even though the agent is unaware of the opponent's utility function. To assess the predictor's performance, the predictions are compared with the estimated utilities obtained through the adopted opponent model first. Subsequently, these predictions are compared with the actual utilities calculated using the opponent's utility function in the system. Note that the agent does not have access to these actual utilities during the training process, and the chosen opponent modeling approach may influence the performance of this comparison as the models are trained using estimations. This comparison offers valuable insights into the efficacy of the opponent modeling technique employed by the agent.

Figure 11 presents a comparison of predictions with estimated and actual opponent utilities. As anticipated, both models exhibit higher error rates when predicting actual opponent utilities. Nevertheless, the predictions for estimated values indicate the beneficial influence of utilizing opponent-related inputs in predicting opponent utility.

This observation suggests that enhancing the accuracy of the utilized opponent model could potentially lead to lower error rates in predicting actual opponent utilities, leading further exploration as a potential research path. Overall, the performance of LSTM models appears to be marginally superior to Transformers in all domain categories, except for small domains.

Domain	Model	MAPE		RMSE	
		EST	*REAL*	*EST*	*REAL*
Whole	Transformer	0.12	0.22	0.14	0.19
	LSTM	0.11	0.19	0.12	0.17
Small	Transformer	0.08	0.20	0.08	0.15
	LSTM	0.11	0.27	0.11	0.17
Medium	Transformer	0.16	0.30	0.18	0.22
	LSTM	0.14	0.29	0.15	0.20
Large	Transformer	0.12	0.39	0.12	0.19
	LSTM	0.09	0.27	0.09	0.17

Figure 11: Model comparison among different domain sizes for *Objective 3* according to estimated & real values

The analysis also includes an examination of how the general models perform against different opponents, taking into account their distinct characteristics. Figure 12 displays the error distribution of the models across various opponent strategies for *Objectives 1* and *2*. Initially, one might assume that predicting the utilities of opponent's next offers would be relatively straightforward when they adopt a time-based concession strategy, as these offers follow a function of normalized negotiation. However, it's essential to note that *Objectives 1* and *2* aim to learn the agent's utility, not the opponent's utility. Consequently, due to the varying nature of the domains and opposition, there may not be a consistent pattern in terms of the agent's utility, as highlighted in Table 2. In other words, the agent's received utilities may exhibit some fluctuations.

On the contrary, behavior-based strategies are expected to demonstrate a more regular pattern since they consider the opponent agent's utilities. As a result, the predictors' performance against such agents is lower compared to behavior-based agents. The performance of the predictors is better for *Objective 2* due to the consideration of opponent-related features. Although the models perform well against behavior-based opponents, their performance significantly drops when facing hybrid agents, whose stochastic behavior introduces challenges for the predictors.

Opponent	Model	MAPE		RMSE	
		Obj 1	*Obj 2*	*Obj 1*	*Obj 2*
Whole	Transformer	0.22	0.15	0.14	0.10
	LSTM	0.21	0.15	0.13	0.10
Behavior-based	Transformer	0.21	0.17	0.14	0.12
	LSTM	0.19	0.16	0.13	0.11
Boulware	Transformer	0.29	0.18	0.15	0.10
	LSTM	0.29	0.19	0.15	0.10
Conceder	Transformer	0.29	0.21	0.15	0.12
	LSTM	0.28	0.21	0.15	0.12
HybridW	Transformer	0.27	0.25	0.18	0.17
	LSTM	0.25	0.23	0.17	0.16
HybridWO	Transformer	0.31	0.20	0.19	0.13
	LSTM	0.30	0.20	0.19	0.13

Figure 12: Model comparison among different opponents for *Objectives 1 & 2*

Figure 13 presents the model comparison against each opponent strategy for *Objective 3*, showcasing the metrics for estimated and actual opponent utilities. Generally, next-step utility predictions against Hybrid strategies exhibit lower performance compared to time-based and behavior-based strategies, which is expected due to their higher complexity in involving both time-based and behavior-based approaches. However, predictions against Hybrid strategies show better results when comparing *Objective 3* predictions with the real opponent utilities in terms of MAPE. This could be attributed to the estimation of opponent's offers by the employed opponent model.

Opponent	Model	MAPE		RMSE	
		EST	*REAL*	*EST*	*REAL*
Whole	Transformer	0.11	0.23	0.12	0.18
	LSTM	0.11	0.21	0.12	0.18
Behavior-based	Transformer	0.09	0.15	0.10	0.15
	LSTM	0.09	0.16	0.10	0.16
Boulware	Transformer	0.11	0.25	0.12	0.17
	LSTM	0.11	0.22	0.12	0.16
Conceder	Transformer	0.10	0.28	0.10	0.16
	LSTM	0.10	0.25	0.10	0.16
HybridW	Transformer	0.12	0.23	0.14	0.21
	LSTM	0.14	0.21	0.15	0.20
HybridWO	Transformer	0.12	0.19	0.13	0.17
	LSTM	0.12	0.19	0.14	0.18

Figure 13: Model comparison among different opponents for *Objective 3* according to estimated & real values

Overall, it is observed that there is no significant difference between the LSTM and Transformer results. The RMSE and MAPE values were similar for each objective, as shown in Figure 14. The similarity in model performances can also be interpreted by the prediction examples illustrated in Figure 15. Although the models could be refined by adding more complexity and potentially achieving higher success, this study utilized basic architectures to focus on the prediction generalization capability across different domain sizes and employed opponent strategies.

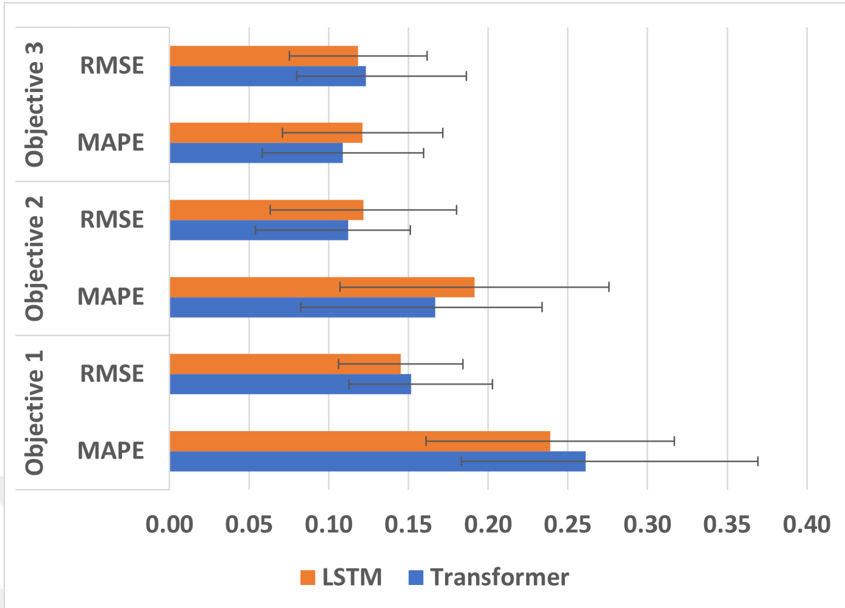
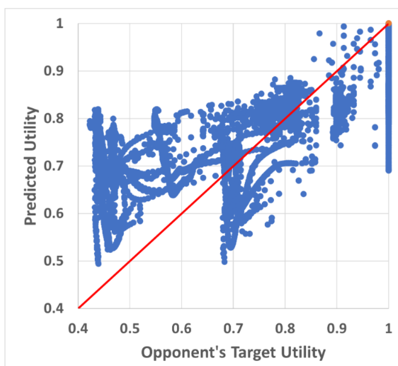
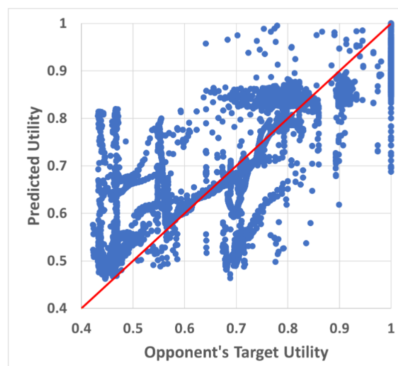


Figure 14: Averaged RMSE and MAPE results for different objectives



(a) LSTM - RMSE: 0.135



(b) Transformer - RMSE: 0.104

Figure 15: Comparing predicted utilities in *Objective 3* for a large domain

CHAPTER V

LEARNING OPPONENT'S BEHAVIORS FROM ALL AGENT EXPERIENCES

Performance of the negotiating agents are usually tested or assessed in a tournament setting where each agent negotiates with each other in various negotiation settings. Consequently, after establishing the learning structure with the agent's own experiences in the initial study, two additional research questions emerged to further explore the effectiveness of the models:

1. Can a negotiating agent benefit from modeling opponent's behavior based on its own experiences?

2. Which machine learning algorithms are best suited for predicting opponent's next move in a negotiation considering the perspectives of multiple agents?

The second study in this thesis addresses these questions by departing from the approach that focuses on training the models based on one agent's negotiation experiences. Unlike the previous study, we gather historical data collected by different agents and use it to train a model. For this purpose, we run several tournaments where each agent negotiate with each other in bilateral fashion on different negotiation scenarios. This approach aimed to create more robust and generalized models that could effectively capture the intricate dynamics and diverse strategies exhibited in automated negotiations.

By exploring these research questions, the thesis endeavors to provide a comprehensive understanding of the predictive models' applicability and adaptability in autonomous negotiations, especially in scenarios involving different opponents with varying strategies and preferences. Therefore, this study limits the objectives in the learning framework with *Objective 2* and *Objective 3*, considering the performance shown in the previous study. The results of this study shed light on the most suitable machine learning algorithms for modeling agent behaviors and predicting opponent's next utility values in a tournament setting, ultimately contributing to more effective decision-making and negotiation strategies in real-world applications.

5.1 Learning Framework

A comprehensive evaluation of diverse learning approaches has been conducted to determine the most effective method for predicting agent utilities and gaining insights into agent behaviors in autonomous negotiations. In light of the research questions raised following the initial study, the learning framework in this study has been adapted to examine how enhanced negotiation data influences the prediction outcomes.

The first study involves defining a baseline agent with a hybrid strategy and frequency-based windowed opponent model and employing machine learning models to predict upcoming utility values based on historical offer exchanges between agents. This study broadens its scope to encompass a dynamic negotiation tournament setting that incorporates the viewpoints of various agents employing different strategies and opponent models in model training. Unlike the first study, the predictive models are designed for general adaptability rather than being trained for a specific baseline agent's negotiation experiences.

Figure 16 simply illustrates the distinction between the two studies in terms of the focused agent interactions during modeling. The left-hand side of the figure portrays the agent interaction structure of the prior study, which included a baseline agent. In that study, prediction inputs were derived from the agent's negotiation experiences with other agents. On the other hand, the right-hand side of the figure illustrates the focused interaction structure of this study, where all agents' experiences are gathered and utilized in model training.

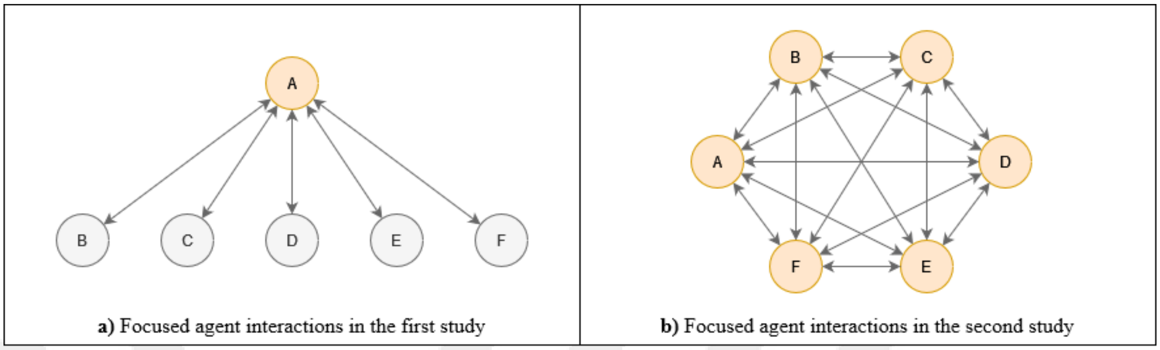


Figure 16: Environmental differences of the studies in terms of agent interactions

In addition to this fundamental difference that shapes the input negotiation histories, the second study refines the objective-based learning framework employed in the previous study, as illustrated in Figure 17. The outcomes of the initial study revealed that the predictions for *Objective 2* exhibited a more successful learning process when compared with the predictions for *Objective 1*. It is important to recall that both objectives aim to predict the utility of the opponent's future offers for the agent itself, $U_A(O_{t+n})$, with *Objective 2* incorporating the opponent model related features in training, while *Objective 1* does not.

Consequently, this study exclusively evaluates the model performances for *Objective 2* and *Objective 3*, which utilize the same input variables but aim to predict different variables. These objectives are once again summarized as follows:

- ***Objective 2:*** Predicting the utility of its opponent's coming offer for itself, $U_A(O_{t+n})$, based on remaining time, offer exchanges so far, estimated opponent utility, Nash Distance, and opponent's moves.

- ***Objective 3:*** Predicting its opponent's estimated utility of its opponent, $\widehat{U}_{Op}(O_{t+n})$, based on remaining time, offer exchanges so far and additional features used in *Objective 2*.

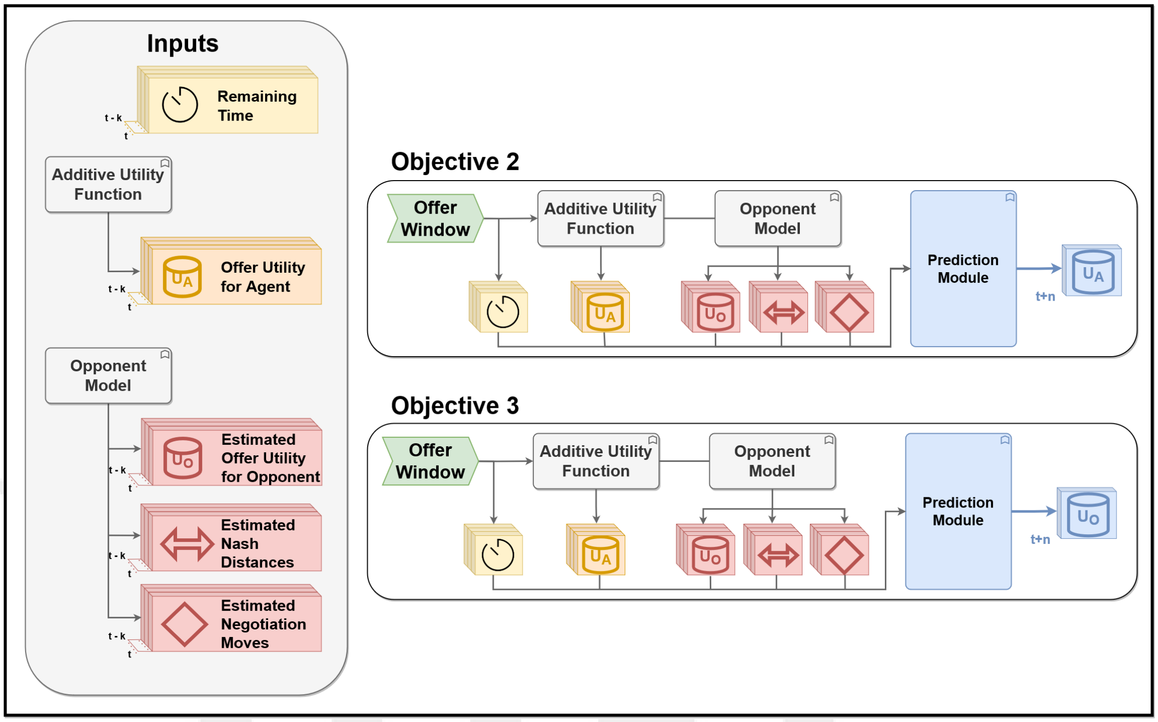


Figure 17: Inputs and outputs for utility value prediction of coming offers

5.2 Prediction Approach

To further enhance the investigation of agent behaviors and utility value predictions in the dynamic realm of automated negotiations, the study strategically incorporates two advanced transformer-based models and one RNN-based model: 'Informer', 'Autoformer', and 'LSTM'.

These cutting-edge models have been specifically adapted and fine-tuned to excel in time series predictions, showcasing their remarkable capabilities in handling sequential data and effectively capturing long-term dependencies. The inclusion of the Transformer-based models serves as a pivotal step to tackle the research questions posed in this study with greater precision and depth. Leveraging the strengths of Informer and Autoformer, the study seeks to delve deeper into the complexities of agent interactions in a tournament setting, where the predictors involve the negotiation experiences of all agents competing in the tournaments.

By exploring the potentials of these state-of-the-art models, the study aims to shed light on the most effective machine learning algorithms for modeling agent behaviors and accurately predicting utility values, thus empowering negotiators with invaluable insights to optimize decision-making strategies and achieve favorable outcomes in tournament setting involving agents with different strategies.

Apart from the changes mentioned for this study, another essential adjustment is made regarding the window sizes utilized by the models. In the previous study, a window size of five was employed, which can be insufficient for capturing the intricate negotiation dynamics observed in a tournament setting. To address this potential issue, the window size is substantially increased in this study to 96, allowing the models to perceive negotiation patterns from a broader perspective and reducing the risk of overfitting.

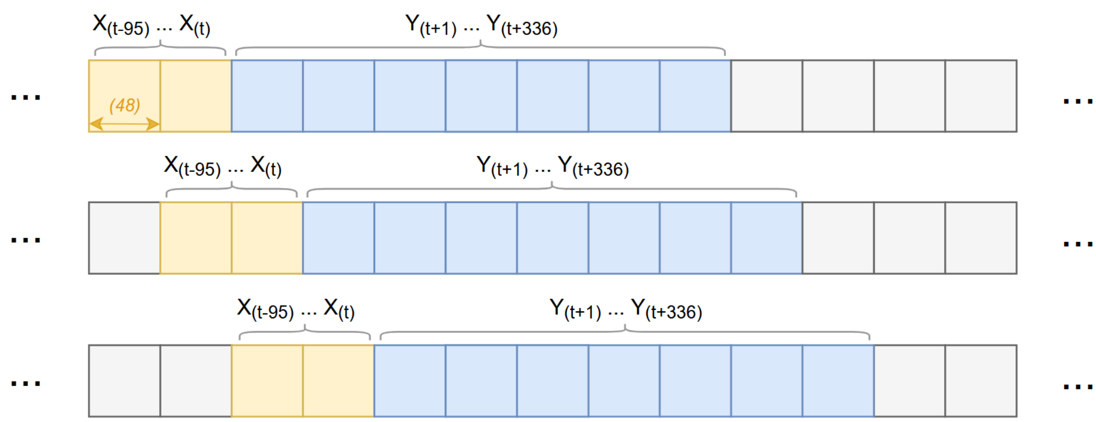


Figure 18: Updated sliding window approach of model input and output

As demonstrated in Figure 18, each iteration now considers the last 96 data points as inputs. Furthermore, instead of predicting the single utility value of the next offer, the models in this study are designed to forecast the next 336 values. Moreover, the windows are not fully sliding at each iteration; rather, they slide by half of the window size, meaning that in the subsequent iteration, the window shifts by 48 data points and incorporates 96 consecutive points as inputs. Note that the window and prediction sizes preferred in this study are also shown to be effective choices in the experiments held in the original Autoformer study [4].

This adaptive approach allows the models to maintain a more comprehensive view of the negotiation history and facilitate more accurate predictions by having a deeper understanding of agent behaviors in the autonomous negotiations context.

5.2.1 LSTM Models

While the Transformer-based models hold high potential for success in this study, the LSTM model remains a strong candidate for the task at hand. Its established effectiveness in handling sequential data and its interpretability make it a valuable addition to the modeling. By including LSTM alongside the Transformer models, we can gain a more comprehensive understanding of the strengths and weaknesses of these diverse approaches, enhancing the depth of our analysis and insights into time series prediction.

The parameters of the LSTM model used for this study are as follows:

- Number of layer(s): 1
- Number of unit(s): 45
- Input length (window size): 96
- Prediction length: 336
- Batch size: 32
- Loss function: 'Mean Squared Error'
- Optimizer: 'ADAM'
- Number of epochs: 8
- Initial learning rate: 0.0001
- Reduced learning rate: True

In contrast to the parameters employed in the previous study, LSTM utilizes the 'ADAM' optimizer in this study, aligning it with the choice made for other models to maintain consistency. Rather than utilizing a fixed learning rate, the reduced learning rate approach is preferred to ensure consistency with the other models as well. The prediction length and window size remain constant across all models in this study.

5.2.2 Informer Models

The introduction of Transformer models has caused a significant increase in the number of studies attempting to adapt these models for time series prediction cases. However, as researchers explored the details of this model, it appeared that certain components of the vanilla Transformer models may not perform as efficiently in time series prediction tasks compared to their success in Natural Language Processing (NLP) and other areas.

One successful study that focuses on this situation is the Informer model, which proposes key updates on the attention mechanism and the distilling operation to decrease the complexity. Employing these changes on the vanilla Transformer architecture can yield successful results in time series prediction tasks. Due to the fact that this model leverages the inherent power of Transformers by employing necessary improvements for time series predictions, the Informer model becomes a promising alternative to the adjustments proposed in the prior study, particularly in more complex tournament scenarios with extensive data. Consequently, this model is included to this study to enhance our ability to understand agent behaviors and predict utility values with greater robustness.

The model architecture with defined input and output structure is shown in Figure 19. Consistent to the proposed sliding window approach, the encoder takes 96 inputs, the decoder takes 96 inputs for the tokens and 336 for processing the outputs, and the output size is 336.

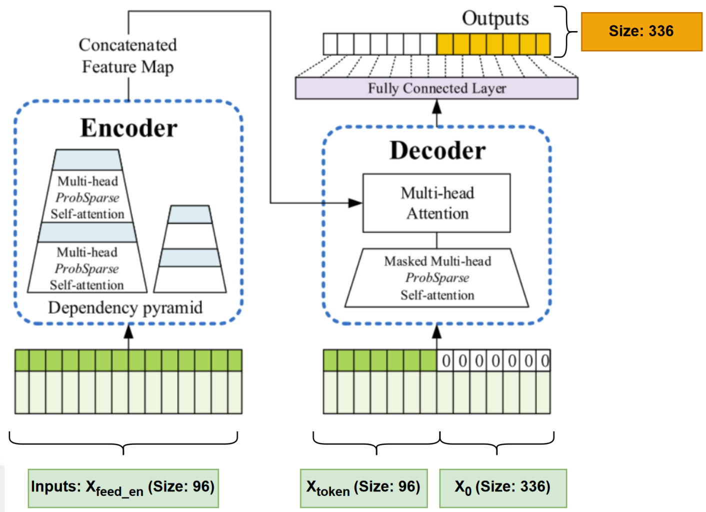


Figure 19: Informer model representation [3]

After tuning the model with different parameters, the final set of parameters are as follows:

- Number of encoder block(s): 2
- Number of decoder block(s): 1
- Input length (window size): 96
- Prediction length: 336
- Batch size: 32
- Loss function: 'Mean Squared Error'
- Optimizer: 'ADAM'
- Number of epochs: 8
- Initial learning rate: 0.0001
- Reduced learning rate: True

The other parameters of the model remains same as the default values used in the original Informer structure. The learning rate in this model starts from 0.0001, and at each epoch it is reduced to its half, as suggested by the Informer study. Also, different than the first study, the optimizer in this model is set as 'ADAM' optimizer instead of 'RMSProp' based on the study of the paper and the tuning results.

5.2.3 Autoformer Models

Another Transformer-based model architecture tailored for time series predictions is 'Autoformer,' which incorporates an auto-correlation mechanism to comprehend period-based dependencies and capture valuable information in the form of time series. The unique decomposition architecture of the Autoformer model allows it to decompose trend and seasonal elements. This feature makes it a valuable addition to this study for assessing the prediction performances.

The model demonstrates its effectiveness when compared to several alternative models derived from the Transformer architecture across diverse datasets. Considering the sequential structure of negotiation sessions involving multiple parties, employing a decomposition-based model appears to be a promising approach. Incorporating the Autoformer model into our study provides an opportunity to acquire valuable insights into the potential of this technique for comprehending agent behaviors and forecasting utility values within the context of automated negotiation tournaments.

In line with the Informer model settings, the Autoformer model complies to the input and output structure defined by the proposed sliding window approach. Based on the model tuning assessments conducted in the Autoformer study and our parameter experimentation, the list of parameters employed in the model is as follows:

- Number of encoder block(s): 2
- Number of decoder block(s): 1
- Input length (window size): 96

- Prediction length: 336
- Batch size: 32
- Loss function: 'Mean Squared Error'
- Optimizer: 'ADAM'
- Number of epochs: 8
- Initial learning rate: 0.0001
- Reduced learning rate: True

The remaining model parameters are maintained at their default values. Similar to the Informer model, the Autoformer model utilized in this study applies the reduced learning rate option to manage the model's complexity.

5.3 Experimental Setup

Following the completion of the initial study, it became evident that the models exhibited promising and consistent outcomes. This prompted the question of how these models would perform when the training process encompassed the perspectives of all agents, rather than being limited to that of a baseline agent. This inquiry led to the second phase of the research, which aimed to analyze the chosen objectives (*Objectives 2 and 3*) within a tournament setting, with an increased number of agents and a broader array of domains. Because of these changes, the experimental setup required some adjustments.

The essential settings, such as grouping the collected negotiation sessions by domain size categories, are retained. However, the number of agents are expanded from five to 14, including those agents used in the previous study. These strategies are selected thanks to their proven success in ANAC competitions [37]. This expansion notably increased the volume of collected negotiation data, as it gets collected from a tournament involving 9 domains, 14 agents, with each agent negotiating with every other agent twice to gather utilities and calculations from both perspectives. Given this substantial increase in data volume, it is reasonable to employ the proposed models, as the enhancements provided by these architectures offer significant advantages in terms of time efficiency.

The domains are generated using GeniusWeb, employing the same size categories as in the earlier study. However, instead of producing one domain for each size, three domains were generated for nine different domain sizes. Table 3 provides detailed information about these domains, categorizing them as small, medium, and large.

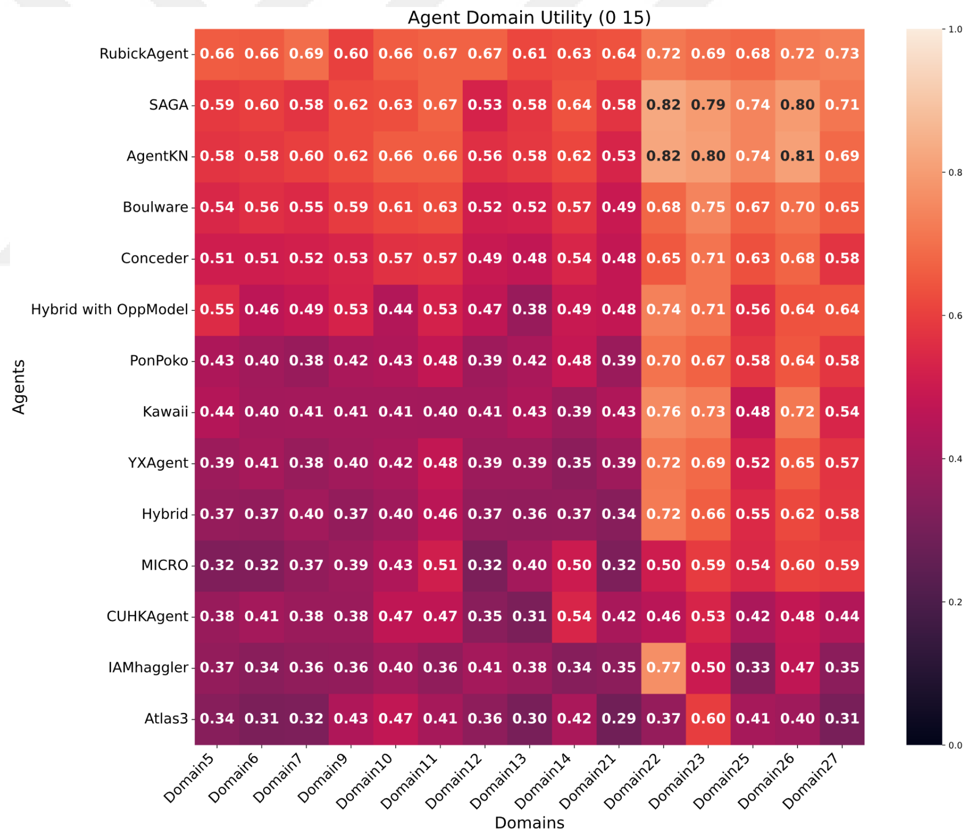


Figure 20: Heatmap of average utilities for agents & domains (Domains 0-15)

Figures 20 and 21 display the average utility values acquired by the agents in each domain during the tournament. These values serve as effective indicators while interpreting the negotiation performance of the agents and illustrate how an agent's

Table 3: Domain details of the second study for training and test data

	Domain ID	Size	Opposition	Issue & Value List
Train	1	3,125	0.3578	[5, 5, 5, 5, 5]
	2	3,125	0.2966	[5, 5, 5, 5, 5]
	3	1,296	0.3394	[6, 6, 6, 6]
	4	1,296	0.3849	[6, 6, 6, 6]
	5	1,024	0.3578	[4, 4, 4, 4, 4]
	6	1,024	0.3568	[4, 4, 4, 4, 4]
	7	729	0.3966	[3, 3, 3, 3, 3, 3]
	8	729	0.2546	[3, 3, 3, 3, 3, 3]
	9	625	0.3568	[5, 5, 5, 5]
	10	625	0.2306	[5, 5, 5, 5]
	11	256	0.3627	[4, 4, 4, 4]
	12	256	0.2617	[4, 4, 4, 4]
	13	15,625	0.3564	[5, 5, 5, 5, 5, 5]
	14	15,625	0.2696	[5, 5, 5, 5, 5, 5]
	15	7,776	0.3578	[6, 6, 6, 6, 6]
	16	7,776	0.2701	[6, 6, 6, 6, 6]
	17	4,096	0.3791	[4, 4, 4, 4, 4, 4]
	18	4,096	0.3478	[4, 4, 4, 4, 4, 4]
Test	19	3,125	0.1989	[5, 5, 5, 5, 5]
	20	1,296	0.2420	[6, 6, 6, 6]
	21	1,024	0.2881	[4, 4, 4, 4, 4]
	22	729	0.3494	[3, 3, 3, 3, 3, 3]
	23	625	0.1809	[5, 5, 5, 5]
	24	256	0.3904	[4, 4, 4, 4]
	25	15,625	0.3282	[5, 5, 5, 5, 5, 5]
	26	7,776	0.2857	[6, 6, 6, 6, 6]
	27	4,096	0.2667	[4, 4, 4, 4, 4, 4]

performance is also influenced by the strategy employed by its opponent.

The heatmaps also feature the names of the 14 agents that took part in the tournament, each of which was designed with distinct strategies. As previously mentioned, GeniusWeb does not allow using external agents to be used within the platform. Therefore, the members of the Interactive Intelligence Lab developed an alternative platform that enabled negotiations to be conducted by uploading the designed agents. Each agent listed in the table was implemented by referencing the project's GitHub

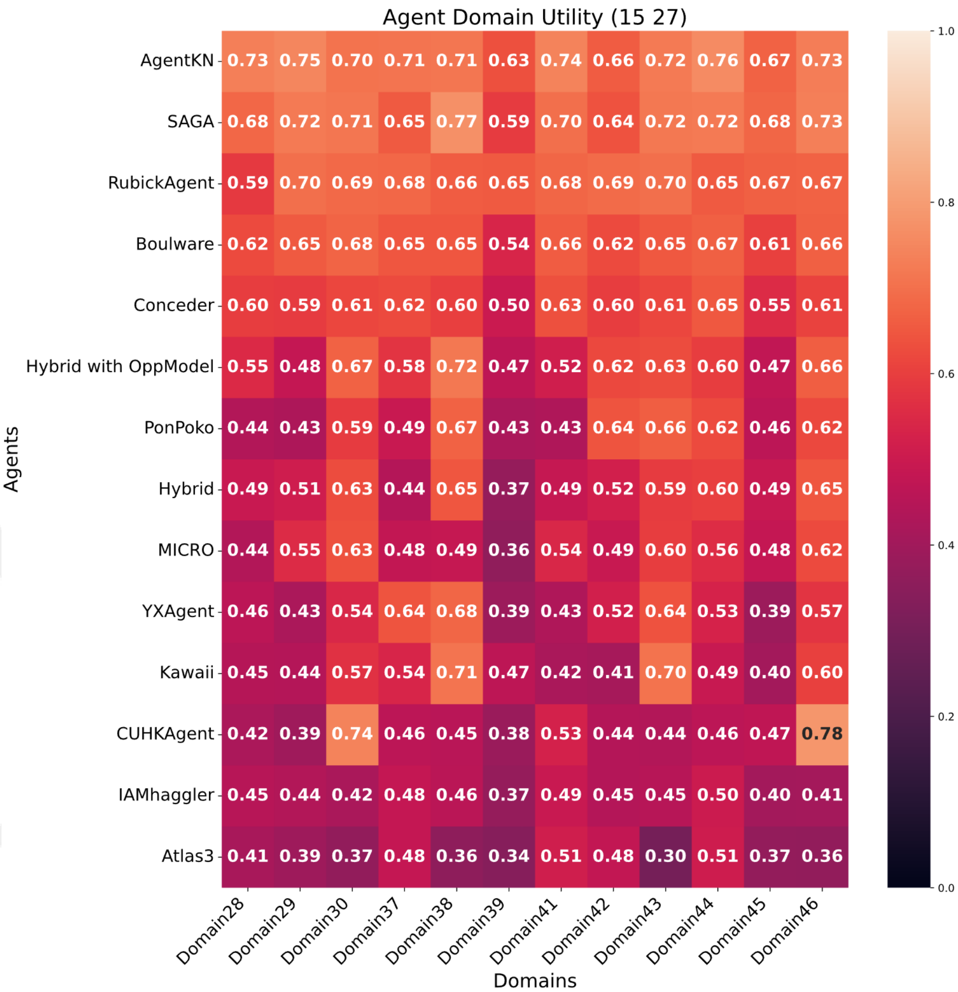


Figure 21: Heatmap of average utilities for agents & domains (Domains 16-27)

pages and their published studies. This approach facilitated the inclusion of a diverse set of agent strategies, providing a comprehensive examination of the models' performances against various negotiating styles.

5.4 Results

The models were trained using negotiation tournament data for all domains, following a similar approach to the previous study. Predictions were generated for both *Objective 2* and *Objective 3* for each model. To assess their performance, two evaluation metrics were employed: the Mean Absolute Percentage Error (MAPE) and the Root of Mean Squared Error (RMSE). By evaluating the performance of Informer, Autoformer and LSTM models using these evaluation metrics, the study aims to identify the strengths and weaknesses of each architecture in predicting utility values in a tournament setting. The results were then analyzed to determine the most effective model for comprehending agent behaviors and making strategic decisions in automated negotiations where agents with diverse strategies and preferences interact with each other.

The initial outcome derived from this study encompasses the utility value predictions for both *Objective 2* and *Objective 3*, generated by all models. In Figure 22, the error scores of these models are depicted for each objective, categorized by the domain sizes. The error scores are highlighted based on the value range, with darker colors representing higher error values.

Domain	Model	MAPE		RMSE	
		Obj 2	*Obj 3*	*Obj 2*	*Obj 3*
Whole	Autoformer	1.425	0.017	0.567	0.031
	Informer	0.470	0.046	0.214	0.059
	LSTM	0.429	0.056	0.187	0.066
Small	Autoformer	1.794	0.043	0.605	0.053
	Informer	0.627	0.040	0.231	0.050
	LSTM	0.565	0.044	0.199	0.054
Medium	Autoformer	1.166	0.048	0.512	0.075
	Informer	0.329	0.053	0.172	0.062
	LSTM	0.293	0.059	0.153	0.064
Large	Autoformer	1.425	0.017	0.567	0.031
	Informer	0.436	0.032	0.197	0.041
	LSTM	0.417	0.076	0.183	0.083

Figure 22: Error scores of models on *Objective 2* and *Objective 3*

The findings of *Objective 2* indicate that, across all domain categories, the LSTM model consistently emerges as the top-performing model, whereas the Autoformer model consistently performs the poorest. Notably, the error margins between Informer and LSTM are relatively minor, with Informer closely trailing behind LSTM. However, the error rates exhibited by the Autoformer results are notably and significantly higher compared to those of the other models.

A significant observation arising from this result is the substantial difference in error scores between *Objective 2* and *Objective 3*, regardless of other breakdowns. In every domain and model, the prediction tasks associated with *Objective 3* exhibit notably lower error scores than those of *Objective 2*. It is worth noting that these two objectives use the same indicators as inputs to the model and differ only in their target variable. *Objective 3*, focused on predicting the estimated utility value of the opponent for future rounds, benefits from the opponent-related inputs utilized during training. The lower error rates observed in *Objective 3* could potentially be influenced by the opponents factoring in their own utilities when formulating an offer.

When evaluating the scores for *Objective 3*, it becomes evident that the models predicting for all domain sizes, labeled as ‘Whole’ in the score figure, exhibit remarkably low errors. The models being exposed to a wide range of diverse examples across all domain sizes might be the reason of this score for ‘Whole’ domain type. It is worth noting that the models also perform well in large domains, which can be attributed to their exposure to a more extensive offer space, rendering them more robust learning behavior.

From the results, it can be seen that the effectiveness of the model types varies across different scenarios. Particularly, when concentrating on *Objective 3*, it is evident that Autoformer consistently exhibits superior performance in most cases. This suggests that Autoformer’s decomposition feature is well-suited for this specific context.

Figure 23 presents the error scores of the models for *Objective 3*, comparing them with both estimated and actual utility values. In the context of *Objective 3* predictions, it is crucial to emphasize that the ground truth for comparing these predictions

Domain	Model	MAPE		RMSE	
		EST	*REAL*	*EST*	*REAL*
Whole	Autoformer	0.017	0.123	0.031	0.146
	Informer	0.046	0.134	0.059	0.151
	LSTM	0.056	0.139	0.066	0.159
Small	Autoformer	0.043	0.132	0.053	0.157
	Informer	0.040	0.131	0.050	0.145
	LSTM	0.044	0.136	0.054	0.151
Medium	Autoformer	0.048	0.113	0.075	0.139
	Informer	0.053	0.125	0.062	0.144
	LSTM	0.059	0.130	0.064	0.151
Large	Autoformer	0.017	0.125	0.031	0.143
	Informer	0.032	0.125	0.041	0.138
	LSTM	0.076	0.134	0.083	0.155

Figure 23: Error scores of models on *Objective 3* compared to estimated and real utility values

is the estimated utility values derived from the opponent model. As it is highlighted in the previous study results, the quality of the opponent model is of utmost importance. Following the generation of predictions, they are then compared with the actual utility values of the opponents.

It is worth noting that these actual utility values were never visible to the agent during the negotiation or model training. Consequently, the error rates of predictions in comparison to the real opponent utilities are understandably higher. However, these error scores fall within a range that can still be considered satisfactory. This is an important observation as it shows that it can contribute to the development of the agent's negotiation strategy.

CHAPTER VI

PROPOSED NEGOTIATION STRATEGY: NEGOFORMER

Through the examination of demonstrated studies, the importance of strategically employing insights derived from the predictions of future rounds becomes evident. Traditional negotiation approaches have not always incorporated these prediction methods. Yet, as the research landscape evolves, it becomes clear that utilizing such predictions can significantly improve the outcome and effectiveness of negotiation dynamics. However, it is not just the act of predicting utility values that matters; it is how these insights are transformed into effective tactics that can make a real difference. In response to this need, this thesis introduces a novel negotiation agent called 'Negoformer'.

6.1 Proposed Negotiation Strategy

In the previous studies, the data collection process featured a range of negotiation strategies, each relying on factors such as remaining time, past behaviors, or a combination of both to varying degrees. Since our exploration to evaluate the efficacy of predictions showed a promising path, we examine whether incorporating these predicted utilities into the agent's approach may result in significant advantages. Negoformer is introduced to meet this need, as it incorporates the 'predicted behaviors' into the agent's decision-making process.

This strategic approach consists of following components:

1. A novel candidate offer selection process.

2. The incorporation of a prediction module for target offer selection

3. The updated offering behavior according to the predictions.

The details of these contributions are provided in the remaining of this section.

6.1.1 Candidate Offer Selection

The opponent model utilized by the agent provides estimated opponent utility values for every potential option within the offer space, along with valuable information such as the Pareto points, Nash point, and the Kalai point, all of which are used by the agent to inform its strategy. Upon observing these estimated opponent utilities, the agent gains a clear view of the potential Pareto frontier, encompassing all the points where maximum benefit can be achieved without making sacrifices. The Pareto Frontier depicts a curve encompassing all utility values of the Pareto optimal offers.

When the agent receives an offer, it draws a circle tangential to the Pareto frontier and picks potential offers within that circle that could be used as the target offer. The radius of the circle, ϵ, is determined by the domain size. In case of negotiating in a small domain, the number of offers that fall within this circle may be relatively limited. As a result, when the domain size is small, the radius ϵ is increased to encompass a larger number of candidate offers within the circle, and conversely, it is reduced in larger domains to refine the selection of potential offers.

The size of the circle also depends on the amount of offers it encompasses. Every circle must contain at least five offer. If this criterion is not met, the radius of the circle will be enlarged until it contains at least five offers. When the circle contains a substantial amount of offers, ideally six points are determined among them as candidate offers. These candidate offers are as follows:

- ***Maximum point:*** Offer within the circle with the highest utility
- ***Pareto point:*** Pareto efficient offer within the circle
- ***Center point:*** Offer within the circle with the median utility

- ***Minimum point:*** Offer with minimum utility
- ***Closest to Nash point:*** Offer within the circle which is closest to the Nash point
- ***Closest to Kalai point:*** Offer within the circle which is closest to the Kalai point

Note that the Kalai point is the fairest point on the Pareto Frontier where minimum utilities of both parties are maximized. The Nash point is the dot product of these utility values. In most of the cases, these two values are equal. However, they might be different in negotiations within unbalanced domains. Figure 24 shows the potential offers as listed above.

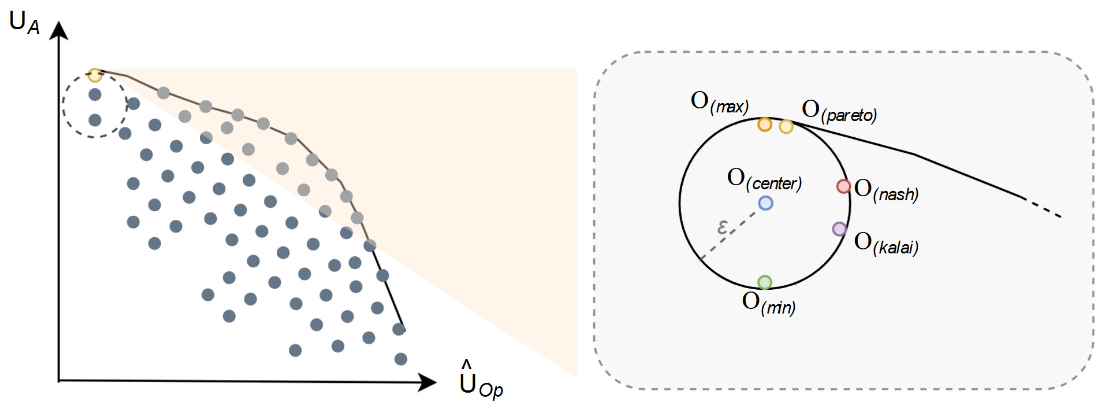


Figure 24: Illustration for determining candidate offers

6.1.2 Target Offer Selection based on Prediction Model

The use of prediction module plays role at this stage. After identifying the candidate offers, each candidate offer is individually fed into the prediction module with the history of previous offers. The chosen prediction model generates separate opponent utility value predictions for each alternative offer over the next k rounds. Note that k is the length of generated predictions. In other words, it provides insights about what the utility values of the next k rounds might look like if Negoformer were to propose that candidate offer. The agent subsequently calculates the slope of these predictions. The offer leading the most effective drop in opponent's next utility value is considered as the optimal scenario for Negoformer. Therefore, the candidate offer that results in predictions with the lowest slope becomes the next offer of Negoformer. Figure 25 provides an illustration of the target offer selection and the utilization of the prediction module.

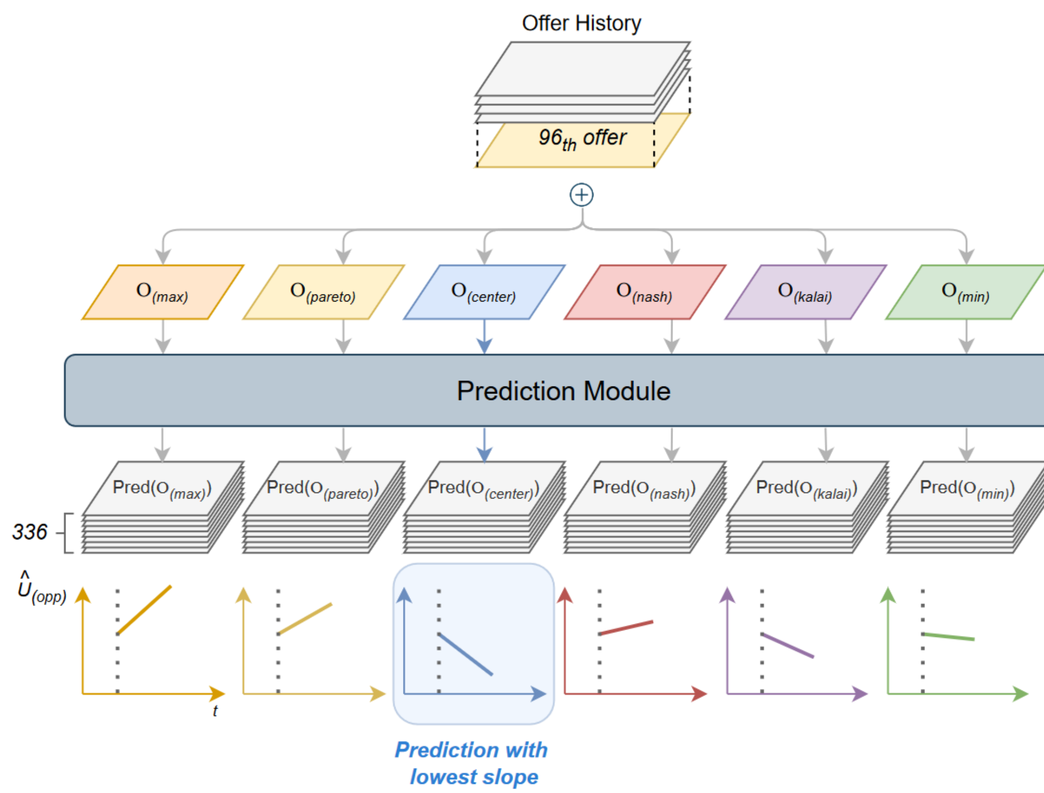


Figure 25: Selecting target offer to compare with received offer

Once the agent's next offer is determined, Negoformer employs the AC_{next} strategy as its acceptance condition. If the received offer yields a higher utility value than the selected next offer, Negoformer accepts the offer, effectively concluding the negotiation. On the other hand, if the received offer provides lower utility value than the target offer, Negoformer rejects the offer and proceeds to propose the chosen next offer.

6.1.3 Offering Behavior

The offering behavior of the agent is governed by a new approach, which we name as 'Pareto Walker'. The name Pareto Walker is derived from its emphasis on the Pareto frontier. Since the frontier contains all the maximal offer pairings, the goal of Negoformer is to generate the candidate offer circles and traverse along this curve.

The frequency of updating the opponent model is aligned with the size of input window employed by the predictors. Consequently, the agent collects data until it populates the initial input window and acquires its first opponent model outputs. Until then, the agent continues to offer the best deals for its own utility. Once the agent receives the required number of offers to construct the frontier, the strategy shifts to the Pareto Walker approach, which is illustrated in Figure 26.

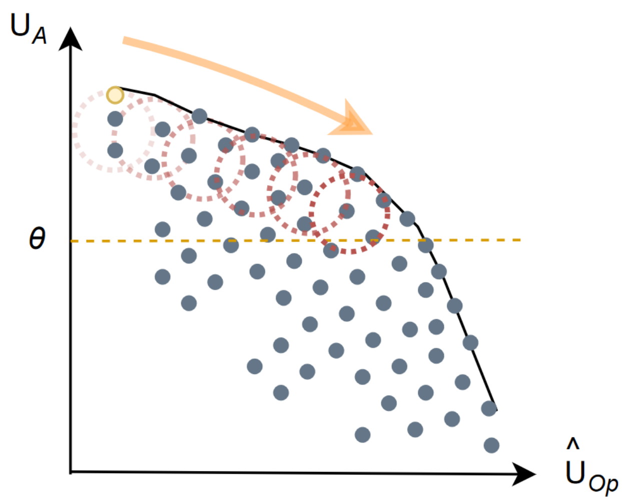


Figure 26: Pareto Walker approach

Starting from the moment the frontier is formed, the agent estimates the number of offers on the Pareto Frontier, known as Pareto points, up to the Nash point, which is positioned along the same frontier. The remaining time until the 90 percent of the negotiation duration is divided equally among these Pareto points. This means that each Pareto point is dedicated to specific time interval. Therefore, until the negotiation time reaches to 90 percent of the total negotiation time, the agent concedes over time.

Initially, the agent selects a target offer for comparison with the received offer by drawing a circle tangential to the Pareto frontier, following its standard procedure. Once 90 percent of the session duration has elapsed, the strategy shifts to a more behavior-based approach. If the opponent concedes, the agent continues to navigate along the Pareto Frontier, redrawing and recentering the circle. However, if the opponent does not seem inclined to concede, Negoformer remains within the current circle and refrains from moving to subsequent ones. Negoformer continues to pick offers within current circle and does not shift to the next circles.

The Pareto Walker approach allows the agent to navigate through potential offers that are profitable and safe for the agent. The lower bound of the circles is restricted with a fixed utility value, θ, for the agent. That means, the agent cannot accept or propose an offer that offers less than the θ utility value. If any portion of the drawn circle results in utility values falling below θ for the agent, that segment is excluded from the pool of candidate offers. In cases where there are fewer than five candidate offers in the unaffected part of the circle, the offers closest to the upper circle are selected as candidates, ensuring there are always at least five options available. Since the candidate offers can never yield less than the θ utility value for the agent, and the agent applies AC_{next} acceptance strategy, this strategic design guarantees that the agent can never have utility values less than θ.

6.2 Experimental Setup

The Negoformer agent best utilizes the Autoformer prediction model according to the *Objective 3* (i.e., guessing the opponent's utility for the opponent's next offer), as evidenced by model evaluations. Autoformer models that are used in Chapter 5 utilizes a tournament setting, which makes it directly adaptable for the agent. Out of the three outlined objectives, the most successful and robust scenario was observed as *Objective 3*. Additionally, employing *Objective 3* proves convenient as it aligns with the objective of capturing opponent behaviors by utilizing potential offers along the Pareto Frontier. Consequently, Negoformer adopts the trained Autoformer model for predicting its opponent's utility value for the opponent's upcoming offer by using the opponent related variables obtained from the frequency-based opponent model that considers the window of offers [20]. In other words, the new agent harnesses the predictions obtained from *Objective 3* to maximize its performance on the decision-making process. As used in the second study, the prediction module applies a sliding window mechanism to produce results. This means predictions become available only after accruing enough offers to fill the initial input window, which occurs no earlier than the 96^{th} round. The details on the use of prediction module can be found in Table 4.

Table 4: Prediction module in Negoformer

Prediction Module in Negoformer	
Reference Study	Learning opponent's behaviors from enriched negotiation data
Prediction Objective	Predicting opponent's utility values considering opponent model outputs
Prediction Model	Autoformer
Opponent Model	Frequency-based window
Opponent Model Frequency	Once in every 96 offers
Prediction Input Window Size	96
Prediction Output Length	336
Prediction Approach	Sliding window
Prediction Frequency	Every round with updated window of offers

The negotiation tournaments are conducted within 12 different domains involving 12 agents, including the Negoformer agent, and each agent negotiated with both profiles of the domains. The 11 employed agents are selected from the top-performing agents in the International Automated Agents Competition (ANAC) in the past, and detailed information is provided in Table 5 [37]. Every agent participated in 264 negotiation sessions.

Table 5: Competing agents' information

Agent Name	Agent Information
Conceder	Genius baseline agent [8]
Boulware	Genius baseline agent [8]
Hybrid	ANAC 2019 finalist [9]
SAGA	ANAC 2019 individual utility category third-place [38]
AgentKN	ANAC 2017 Nash category runner-up [39]
RubickAgent	ANAC 2017 individual utility category third-place [40]
Kawaii	ANAC 2015 finalist [41]
PonPoko	ANAC 2017 individual utility category winner [39]
YXAgent	ANAC 2016 individual utility category runner-up [39]
IAMhaggler	ANAC 2012 Nash category winner [42]
MICRO	ANAC 2022 individual utility category runner-up [43]

The domains used for the tournaments are generated based on their domain size, opposition, and whether they are balanced or unbalanced. The specifications of the generated domains for the negotiation tournaments are presented in Table 6.

Table 6: Domain distribution information

ID	Category	Domain Size	Issue Values	Opposition
Small-1	Small	256	[4, 4, 4, 4]	Low
Small-2	Small	729	[3, 3, 3, 3, 3, 3]	High
Small-3	Small	1024	[4, 4, 4, 4, 4]	Low
Small-4	Small	1296	[6, 6, 6, 6]	High
Medium-1	Medium	4096	[8, 8, 8, 8]	Low
Medium-2	Medium	6561	[9, 9, 9, 9]	High
Medium-3	Medium	7776	[6, 6, 6, 6, 6]	Low
Medium-4	Medium	10000	[10, 10, 10, 10]	High
Large-3	Large	16384	[4, 4, 4, 4, 4, 4, 4]	Low
Large-1	Large	19683	[3, 3, 3, 3, 3, 3, 3, 3, 3]	Low
Large-4	Large	32768	[8, 8, 8, 8, 8]	High
Large-2	Large	46656	[6, 6, 6, 6, 6, 6]	High

Note that the threshold of the agent's minimum utility, θ, is set as 0.5 for this experiment. When the agent draws a circle to define its candidate offers, it uses a radius, ϵ. This radius takes values in the interval [0.025, 0.05], where 0.025 is the lower limit selected for large domains, and 0.05 is the upper limit for small domains.

6.3 *Evaluation of the Proposed Strategy*

A series of tournaments has been executed according to the given experimental setup instructions. This chapter provides a comprehensive description and evaluation of Negoformer's performance in these tournaments compared to other agents.

Figure 27 presents a comprehensive overview of the negotiation tournament outcomes by displaying both the average utility values acquired by the agent and its opponents, along with the average Nash distances. In the visualization, blue bars denote the average utility values obtained by the corresponding agent, while red bars represent the average utilities made by its negotiating counterparts when engaging with the same agent. Each bar is accompanied by the corresponding deviation. Moreover, the green line shows the Nash distances on the right axis, indicating the distance from the optimal offer for both agents.

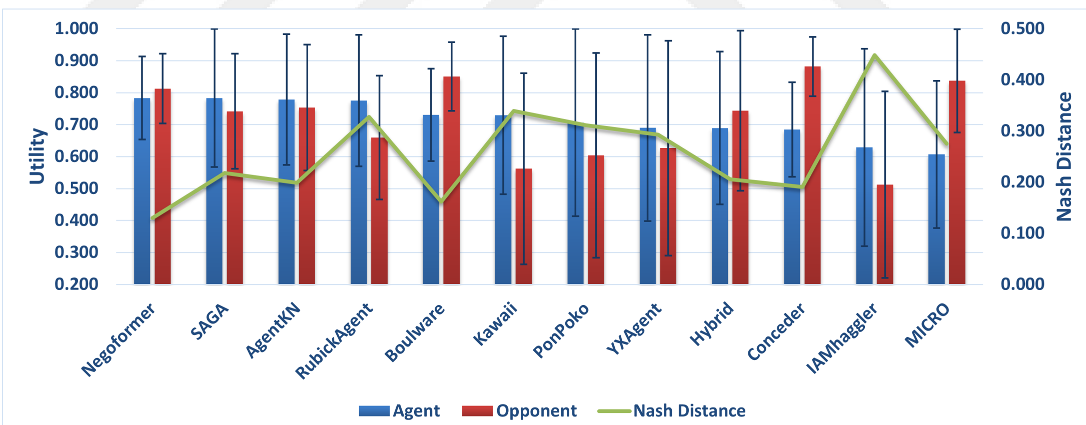


Figure 27: Average tournament utility values and Nash distances

The agents in Figure 27 are arranged based on their individual average utility values. The outcomes indicate that Negoformer and SAGA agents demonstrate the highest average values with a utility of 0.78 across all negotiation sessions. However, considering the deviations in agent utilities, Negoformer exhibits a more consistent performance by showing lower deviation compared to the SAGA Agent. The difference in deviations can be observed clearer in Figure 28 where the utility values of the agents are illustrated in a heat map. AgentKN and Rubick Agent follow them in terms of utility values.

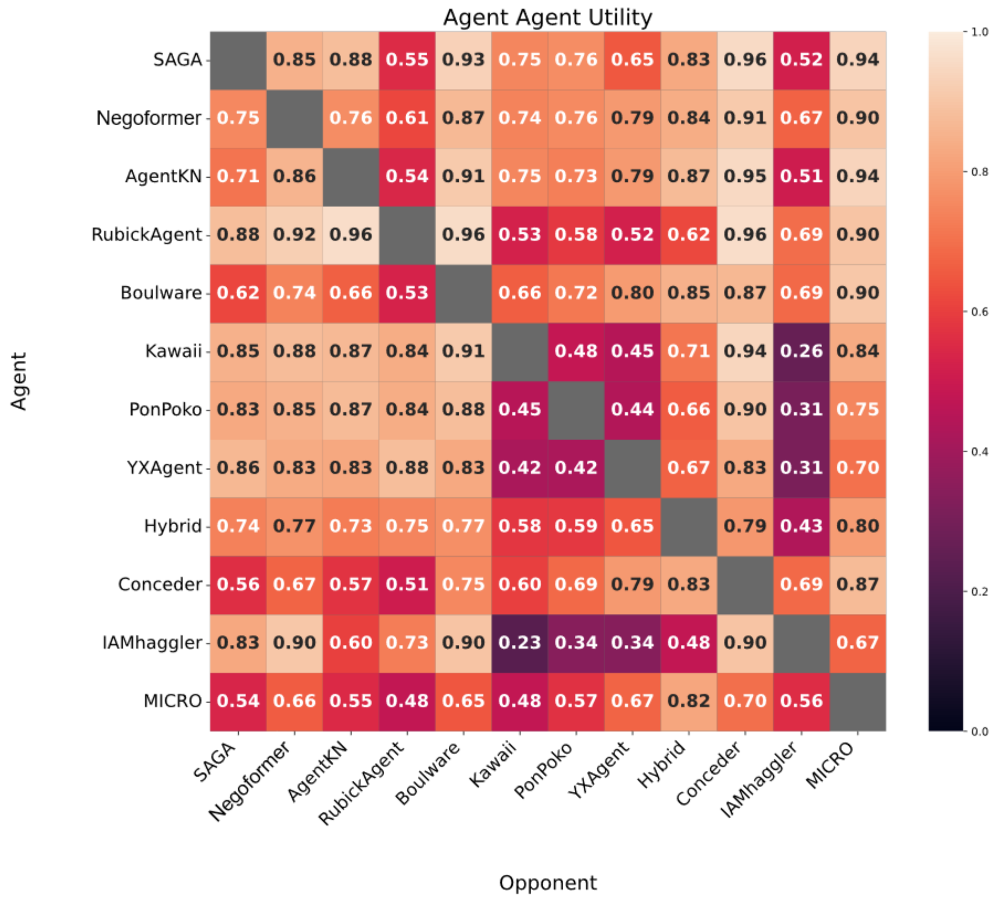


Figure 28: Heat map of agent-to-agent utility values

It is essential to consider effect of the domain characteristics while evaluating the utility values of the agents. Figure 29 demonstrates the heat map of average utility values of the agents for each domain used for the negotiations. The average utility values gained within each domain depict domains with high opposition values yield low utility values for all agents. Recall that the domains with high oppositions are *Small-2*, *Small-4*, *Medium-2*, *Medium-4*, *Large-2*, and *Large-4* as previously shown in Table 6.

Another important insight drawn from this output is the average utility values received by the opponents. It is evident that Negoformer offers favorable utility

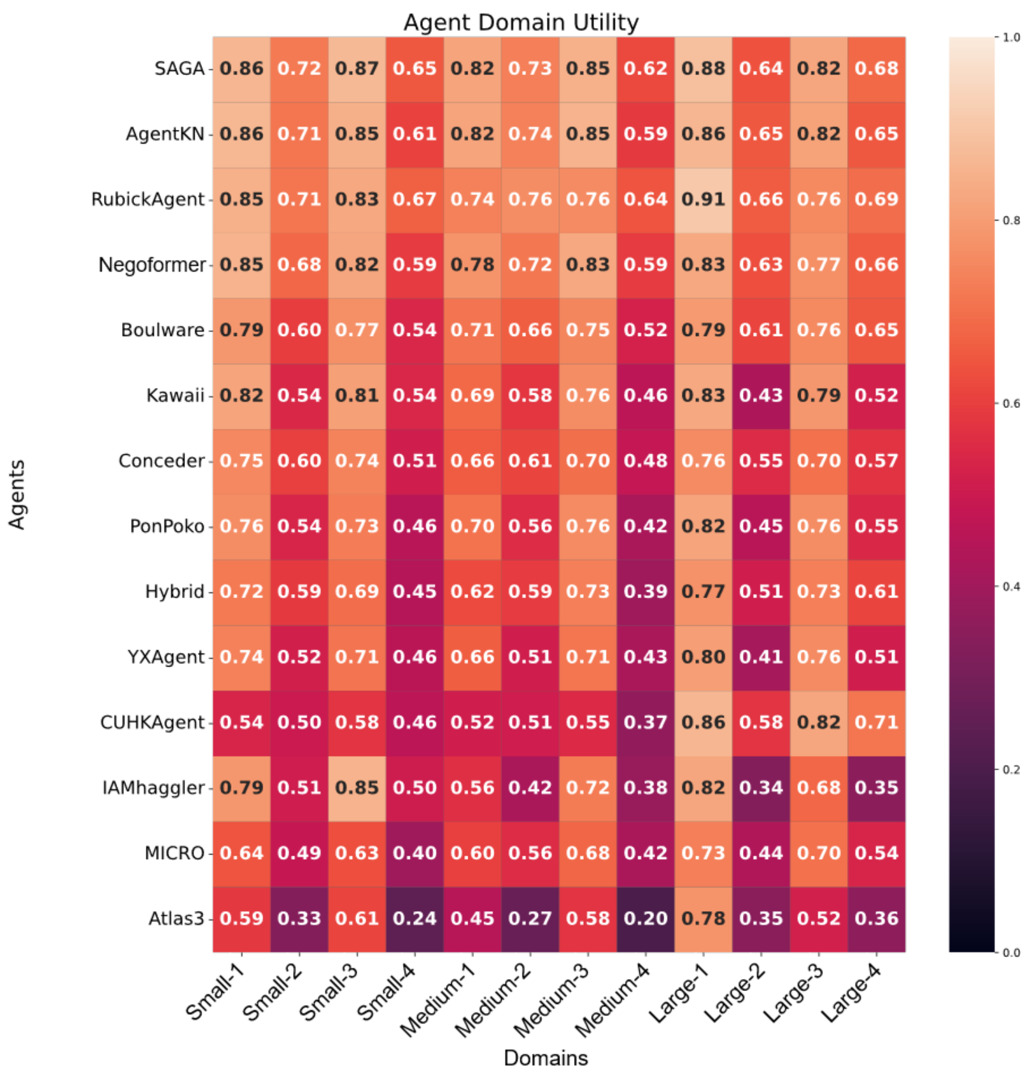


Figure 29: Heat map of agent utility values for each domain

values to its counterparts while also preserving its own interests. Among the first four agents shown in the graph, Negoformer notably prioritizes the satisfaction of its opponents the most.

Among the top four performing agents, Negoformer shows the lowest Nash distance, which can be interpreted from the high utility values observed for its opponents as well. While Conceder and Boulware agents also exhibit Nash distances close to Negoformer's, it is crucial to note that this is attributed to the high utility value they provide for their opponents. Achieving a balance in utility gains for both parties and agreeing on offers that are likely to be optimal presents a significant challenge, which Negoformer successfully managed.

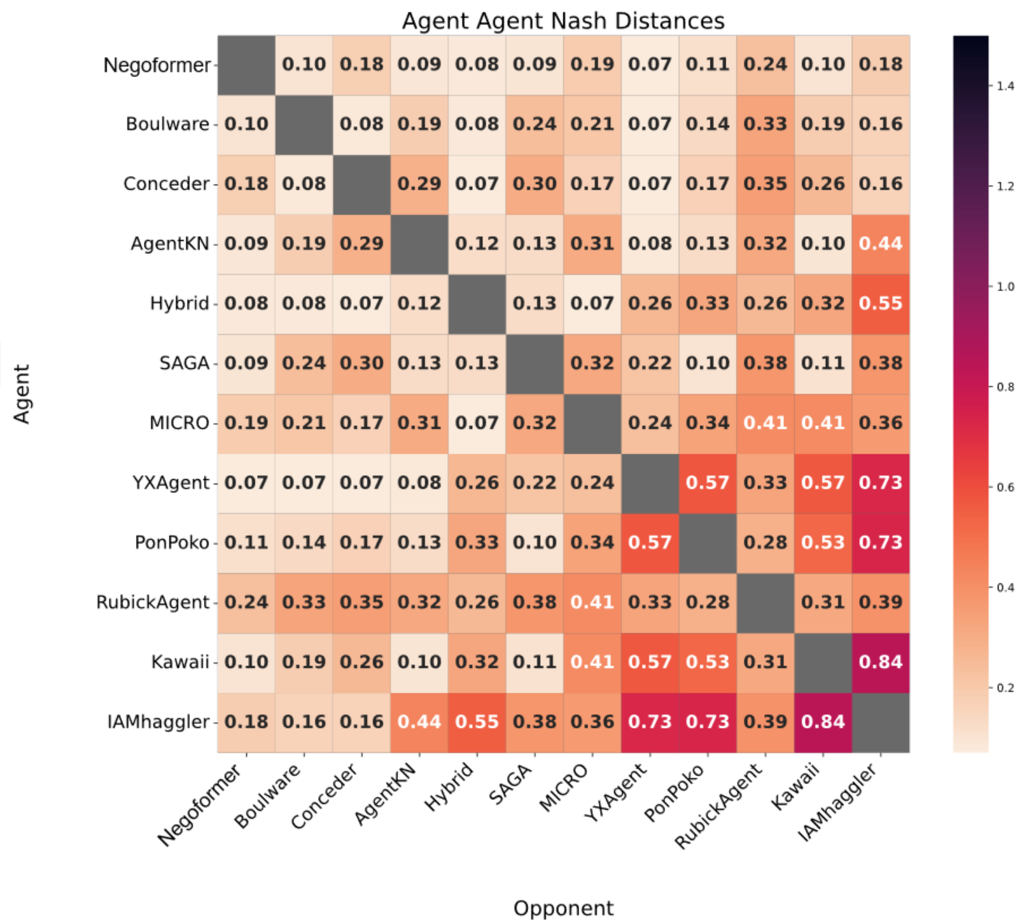


Figure 30: Heat map of agent-to-agent Nash distances

Figure 30 shows the agent-to-agent heat map of the average Nash distances. Contrast to the heat map of individual utility values shown in Figure 28, there is a notable shift in SAGA Agent's position. While initially tied with Negoformer, SAGA Agent now appears lower due to displaying higher average Nash distances. This means our agent led relatively fairer agreements compared to SAGA Agent. Negoformer's approach extends beyond merely securing agreements with optimal offers as it also prioritizes fairness towards opponents' gains. Figure 31 displays a heat map of agents showcasing average gained social welfare points, which consists of a calculation based on a utilitarian social welfare approach (i.e., sum of utilities that each negotiation partners received at the end of the negotiation). Notably, among all negotiating agents, Negoformer demonstrates the highest sensitivity to fairness in negotiations. This sensitivity is pivotal for fostering healthy interactions and sustaining positive relationships with competing agents. When comparing agents' positions in the heat maps of Nash distances and social welfare points, it becomes apparent that not all agents excel in both aspects. Recall that the fairest offer might differ from the optimal offer, particularly in unbalanced negotiation domains. Negoformer's consistent high performance in both metrics indicates that the proposed strategy can remain robust in unbalanced domains.

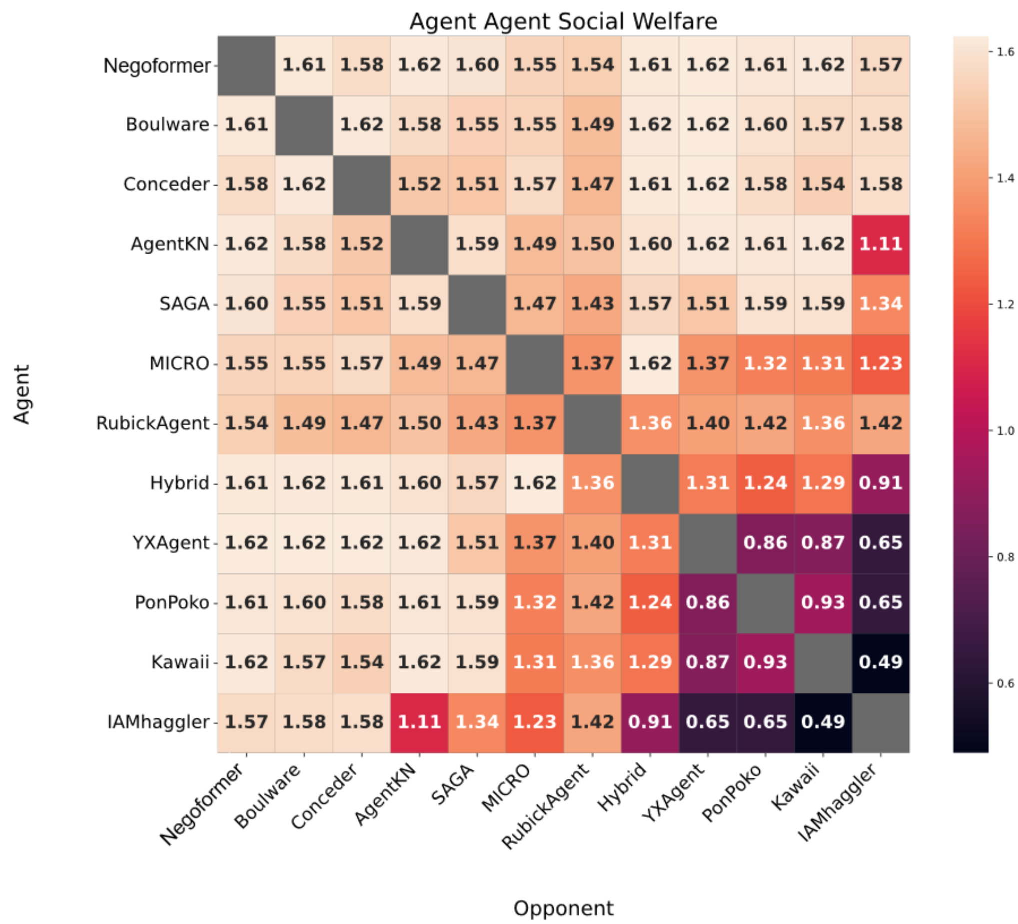


Figure 31: Heat map of agent-to-agent social welfare points

While evaluating the negotiation performances, it is essential to observe the agents' success in completing negotiation sessions with an agreement and the time taken to reach consensus with opponents. Figure 32 illustrates the average acceptance rates and negotiation rounds required for agreements among the agents. Notably, Negoformer shows a 100% success rate in achieving agreements across all negotiations. While its average agreement rounds align with those of other negotiating agents, it indicates that the agent shows a balanced approach as it does not hurry or delay decisions.

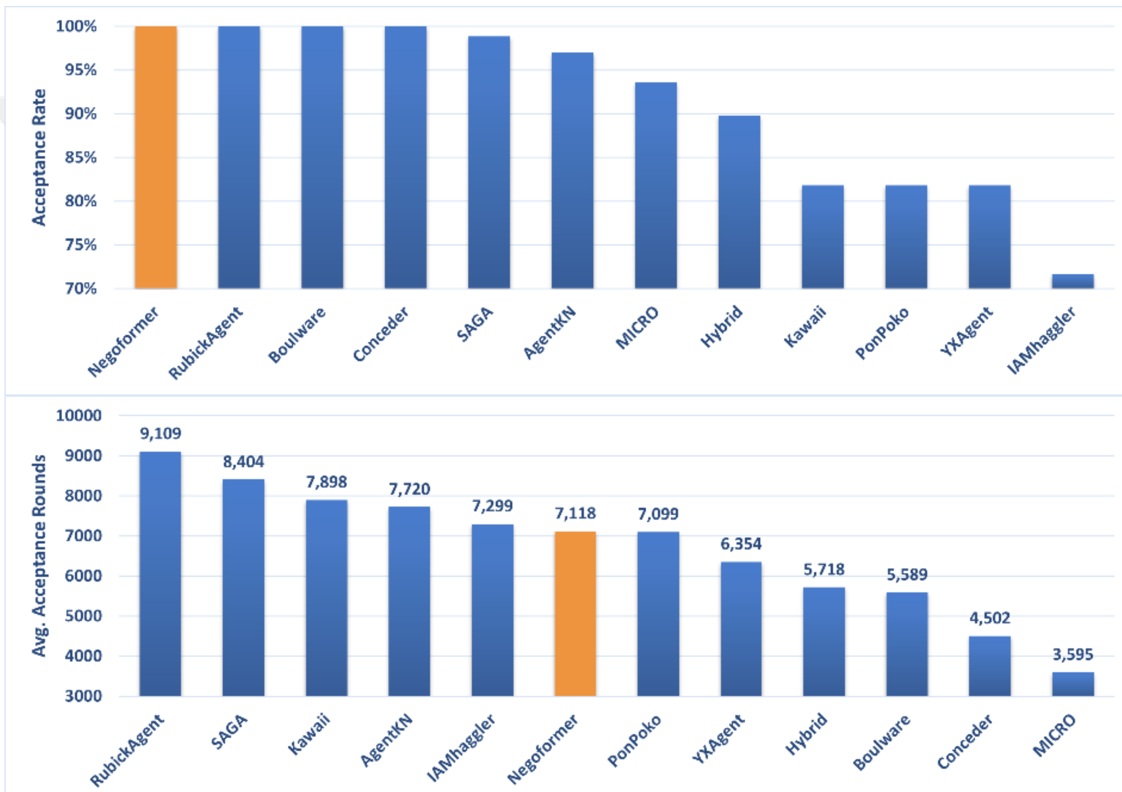


Figure 32: Average acceptance rates and rounds

In addition to the given metrics of individual agent utilities, Nash distances and social welfare points, it is essential to assess the offering behavior of the agent to confirm the alignment of Negoformer's actions with its predefined strategies. As outlined in the offering strategy, the agent operates with distinct behaviors before and after the negotiation reaches to the 90 percent of the session. Before reaching to 90 percent, the expectation for the agent is to concede in line with the Pareto frontier over time. After 90 percent, it concedes only if its opponent concedes as well. If the opponent does not concede, the agent will not move to the next Pareto point.

Accordingly, Figure 33 demonstrates the evolving behavior of the agent over time. To monitor these changes, a designated metric called the 'Walk Index' is defined. Each point of utility pair along the Pareto Frontier is assigned to an index in an order, starting from the pair where agent's utility is the highest. The figure reveals that until 90 percent of the negotiation session, the target offer shifts progressively across indexed Pareto points. However, this continual index change ceases after reaching the 90 percent mark. A static Walk Index for consecutive rounds implies a lack of concession from the opponent towards Negoformer. Consequently, this index assessment validates the agent's adherence to its expected behavior.

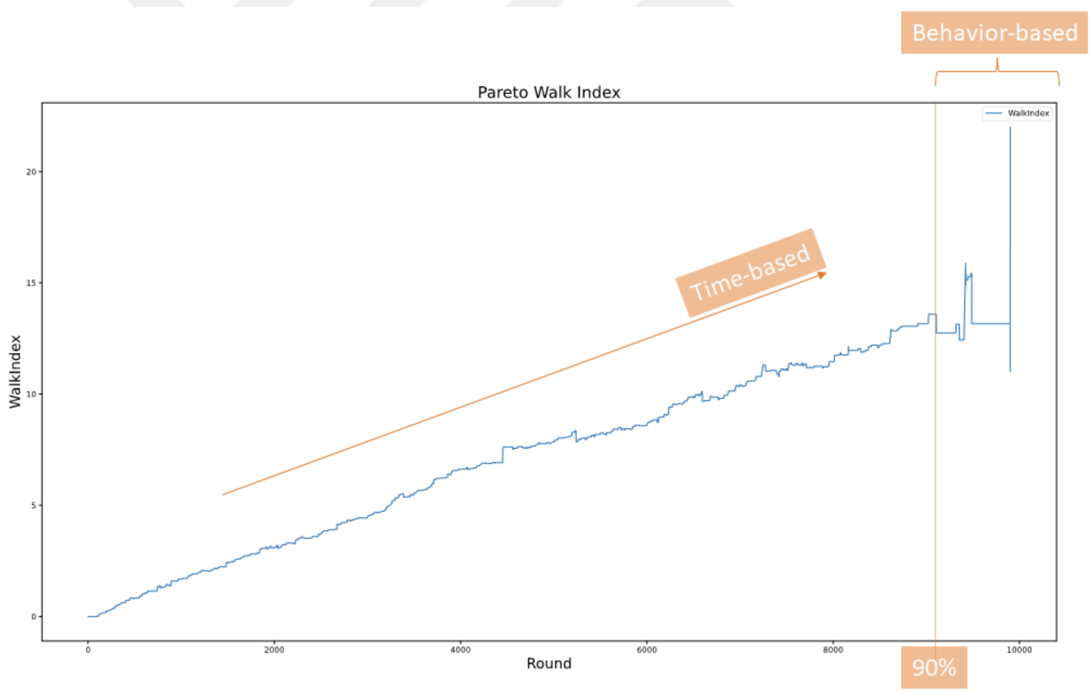


Figure 33: Pareto Walk Index

CHAPTER VII

CONCLUSION

Agents in automated negotiations engage with each other with the aim of agreeing on an offer that is most favorable for themselves, mostly without having any information about their opponents. Sophisticated agents aim to close this knowledge gap by observing and evaluating the offer exchanges so far to derive useful insights about their opponents'. This thesis focuses on the methods that might allow the agent to increase its knowledge by searching answers to the following questions:

1. *How the predictive methods that are applicable for time series data can be utilized to learn more about the opponent agent?*

2. *How successful are these predictive methods in terms of predicting the utility values that the opponent will gain?*

3. *Considering the methods are successful in predicting the opponents' utility values, how can an agent utilize these predictions while making its offers?*

Accordingly, this thesis initially employed and evaluated several machine learning methods on the negotiation data to see their performances in various conditions. To evaluate the predictive performance of these models, three distinct learning objectives are outlined ; (i) *Objective 1*: predicting the agent's utility values for the coming rounds by using negotiation session data, (ii) *Objective 2*: predicting the agent's utility values for the coming rounds by using both negotiation session data and the opponent model outputs, and (iii) *Objective 3*: predicting the future utility values of its opponent by using both negotiation session data and the opponent model outputs. For the initial experiments, two models are selected for prediction tasks: LSTM and Transformers. Later, we also examined other models: LSTM, Informer, and Autoformer.

Establishing a learning framework with three distinct objectives significantly contributed to the project by enabling an exploration of how a predictive model might enhance the understanding of negotiation behaviors in future rounds. The study's results are assessed by the model performances concerning negotiation domain sizes and opponent strategies. While the results showcased no significant difference in predictive capabilities between LSTM and Transformers for the given tasks, they underscored the importance of focused objectives. Among all objectives, *Objective 3* was proved to be successful in comprehending the opponents' behavior the most with competitive metrics. Conversely, *Objective 1* exhibited relatively poorer performance compared to others, which indicates that using opponent model outputs during predictions play a pivotal role.

According to the findings from the initial phase of the study, the next phase is extended with additional agents with varying strategies. At this part of the study, the set of objectives is narrowed to *Objective 2* and *Objective 3*, where LSTM and two modified Transformer-based models, namely Informer and Autoformer models, are utilized for prediction tasks. Unlike the first prediction study, these models are trained by taking the negotiation perspectives of all agents into consideration. The outcomes indicate that the Autoformer model shows more reliable results across all categories, which might stem for the adaptability of auto-regression based approach to the given tasks.

After exploring the predictive capabilities of the models across various settings, the next step was to incorporate these predictions into the agent's decision making mechanism. Therefore, a novel agent strategy called 'Negoformer' is developed. Through its distinct methods of candidate offer selection, target offer determination via predictions, and overall offering strategy, Negoformer has demonstrated itself as a viable, resilient, and effective approach that can seamlessly be complemented with other predictive models. The negotiation outcomes affirm Negoformer's success concerning individual agent utility values, Nash distances achieved, and the overall social welfare within negotiations.

Moving forward, future research endeavors could explore the integration of adaptive learning mechanisms within negotiation strategies to enhance agents' responsiveness to dynamic environments. Investigating the impact of incorporating reinforcement learning frameworks could further refine the agents' decision-making processes in negotiating with diverse opponents. Furthermore, assessing multilateral negotiation settings for adaptation in competitive financial market scenarios, such as the energy sector or portfolio optimization, presents a promising avenue for testing and application.

The gradual evolution of experimentation and development within this study resulted in a comprehensive exploration of integrating predictive models into the realm of automated negotiations. This journey has underscored the potential of predictive capabilities in reshaping negotiation strategies by offering a nuanced understanding of their applicability and efficacy in complex decision-making scenarios. As this study draws to an end, it represents progress in exploring how predictive models might augment agents' negotiation strategies, offering potential pathways for future improvements in automated negotiation systems.

REFERENCES

[1] K. Hindriks, C. Jonker, and D. Tykhonov, "Let's dans! an analytic framework of negotiation dynamics and strategies," *Web Intelligence and Agent Systems*, vol. 9, pp. 319–335, 01 2011.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[3] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11106–11115, May 2021.

[4] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *CoRR*, vol. abs/2106.13008, 2021.

[5] S. Fatima, S. Kraus, and M. Wooldridge, *Principles of automated negotiation*. Cambridge University Press, 2014.

[6] I. Marsa-Maestre, M. Klein, C. M. Jonker, and R. Aydoğan, "From problems to protocols: Towards a negotiation handbook," *Decision Support Systems*, vol. 60, pp. 39–54, 2014.

[7] R. Aydoğan, T. Baarslag, K. Fujita, J. Mell, J. Gratch, D. de Jonge, Y. Mohammad, S. Nakadai, S. Morinaga, H. Osawa, C. Aranha, and C. M. Jonker, "Challenges and main results of the automated negotiating agents competition (anac) 2019," in *Multi-Agent Systems and Agreement Technologies*, pp. 366–381, Springer International Publishing, 2020.

[8] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Robotics and Autonomous Systems*, vol. 24, no. 3, pp. 159–182, 1998.

[9] M. O. Keskin, U. Çakan, and R. Aydoğan, "Solver agent: Towards emotional and opponent-aware agent for human-robot negotiation," in *Proceedings of the AAMAS 2021*, (Richland, SC), p. 1557–1559, International Foundation for Autonomous Agents and Multiagent Systems, 2021.

[10] T. Baarslag, M. Hendrikx, K. V. Hindriks, and C. M. Jonker, "Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques," *Autonomous Agents and Multi-Agent Systems*, vol. 30, p. 849–898, 2016.

[11] C. R. Williams, V. Robu, E. H. Gerding, and N. R. Jennings, "Using gaussian processes to optimise concession in complex negotiations against unknown opponents," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (15/07/11 - 21/07/11)*, pp. 432–438, 2011.

[12] F. Arslan and R. Aydogan, "An actor-critic reinforcement learning approach for bilateral negotiation," *Turkish Journal of Electrical Engineering and Computer Sciences*, pp. 1–20, 2022.

[13] A. Sengupta, Y. Mohammad, and S. Nakadai, "An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism," *CoRR*, vol. abs/2102.03588, 2021.

[14] S. Chen, H. Ammar, K. Tuyls, and G. Weiss, "Transfer learning for bilateral multi-issue negotiation," in *Proceedings of the BNAIC 2012*, vol. 24, pp. 59–66, 2012.

[15] C. Eran, M. O. Keskin, F. Cantürk, and R. Aydoğan, "A decentralized token-based negotiation approach for multi-agent path finding," in *Multi-Agent Systems*, vol. 12802, pp. 264–280, Springer International Publishing, 2021.

[16] M. Li, P. K. Murukannaiah, and C. M. Jonker, "A data-driven method for recognizing automated negotiation strategies," *ArXiv*, vol. abs/2107.01496, 2021.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, nov 1997.

[18] R. Vahidov, G. Kersten, and B. Vu, "Human-agent negotiations: The impact agents concession schedule and task complexity on agreements," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, vol. 50, pp. 412–420, 01 2017.

[19] T. Baarslag, K. Hindriks, and C. Jonker, "A tit for tat negotiation strategy for real-time bilateral negotiations," *Studies in Computational Intelligence*, vol. 435, p. 229–233, 2013.

[20] V. Sanchez-Anguix, O. Tunalı, R. Aydoğan, and V. Julian, "Can social agents efficiently perform in automated negotiation?," *Applied Sciences*, vol. 11, no. 13, 2021.

[21] T. Baarslag, K. Hindriks, and C. Jonker, *Acceptance Conditions in Automated Negotiation*, vol. 435, pp. 95–111. Springer Berlin Heidelberg, 2013.

[22] T. Ito, M. Zhang, V. Robu, and T. Matsuo, "Complex automated negotiations: Theories, models, and software competitions," in *Complex Automated Negotiations*, 2012.

[23] O. Tunalı, R. Aydoğan, and V. Sanchez-Anguix, “Rethinking frequency opponent modeling in automated negotiation,” in *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, pp. 263–279, Springer International Publishing, 2017.

[24] K. Hindriks, C. Jonker, and D. Tykhonov, “The benefits of opponent models in negotiation,” in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 439–444, 09 2009.

[25] M. O. Keskin, B. Buzcu, and R. Aydogan, “Conflict-based negotiation strategy for human-agent negotiation,” *Applied Intelligence*, vol. 53, p. 29741–29757, Nov 2023.

[26] K. Hindriks, C. Jonker, and D. Tykhonov, “Negotiation dynamics: Analysis, concession tactics, and outcomes,” in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2007*, pp. 427–433, 12 2007.

[27] K. Hindriks, C. Jonker, and D. Tykhonov, “Analysis of negotiation dynamics,” in *Proceedings of the 11th international workshop on Cooperative Information Agents XI*, pp. 27–35, 04 2007.

[28] M. Cao, G. A. Wang, and M. Y. Kiang, “Modeling and prediction of human negotiation behavior in human-computer negotiation,” *Electronic Commerce Research and Applications*, vol. 50, p. 101099, 2021.

[29] C. Hou, “Predicting agents tactics in automated negotiation,” in *Proceedings - IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IAT 2004*, pp. 127– 133, 10 2004.

[30] R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. V. Hindriks, and C. M. Jonker, “Genius: An integrated environment for supporting the design of generic automated negotiators,” *Computational Intelligence*, vol. 30, pp. 48–70, 02 2014.

[31] R. Aydoğan, D. Festen, K. V. Hindriks, and C. M. Jonker, “Alternating offers protocols for multilateral negotiation,” in *Modern Approaches to Agent-based Complex Automated Negotiation*, pp. 153–167, Springer, 2017.

[32] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Australia: OTexts, 2nd ed., 2018.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] G. Yesevi, M. O. Keskin, A. Doğru, and R. Aydogan, “Time series predictive models for opponent behavior modeling in bilateral negotiations,” in *PRIMA 2022: Principles and Practice of Multi-Agent Systems*, pp. 381–398, 11 2022.

[35] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," *The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)*, vol. 37, 2023.

[36] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," *CoRR*, 2020.

[37] C. M. Jonker, R. Aydoğan, T. Baarslag, K. Fujita, T. Ito, and K. Hindriks, "Automated negotiating agents competition (ANAC)," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 5070–5072, AAAI Press, 2017.

[38] R. Aydoğan, T. Baarslag, K. Fujita, J. Mell, J. Gratch, D. de Jonge, Y. Mohammad, S. Nakadai, S. Morinaga, H. Osawa, C. Aranha, and C. M. Jonker, "Challenges and main results of the automated negotiating agents competition (anac) 2019," in *Multi-Agent Systems and Agreement Technologies* (N. Bassiliades, G. Chalkiadakis, and D. de Jonge, eds.), (Cham), pp. 366–381, Springer International Publishing, 2020.

[39] R. Aydoğan, K. Fujita, T. Baarslag, C. M. Jonker, and T. Ito, "ANAC 2017: Repeated multilateral negotiation league," in *Advances in Automated Negotiations*, (Singapore), pp. 101–115, Springer Singapore, 2021.

[40] O. Tunalı, R. Aydoğan, and V. Sanchez-Anguix, "Rethinking frequency opponent modeling in automated negotiation," in *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 263–279, 10 2017.

[41] Y. HOSOKAWA and K. Fujita, "Opponent's preference estimation considering their offer transition in multi-issue closed negotiations," *IEICE Transactions on Information and Systems*, vol. E103.D, pp. 2531–2539, 12 2020.

[42] C. R. Williams, V. Robu, E. H. Gerding, and N. R. Jennings, *IAMhaggler: A Negotiation Agent for Complex Environments*, vol. 383, pp. 151–158. Springer Berlin Heidelberg, 2012.

[43] D. de Jonge, "An analysis of the linear bilateral ANAC domains using the MiCRO benchmark strategy," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022* (L. D. Raedt, ed.), pp. 223–229, ijcai.org, 2022.

VITA

Gevher Yesevi holds a Bachelor's Degree from the Industrial Engineering Department at Ozyegin University. She is currently pursuing a Master's Degree in Artificial Intelligence at the same university. Additionally, Gevher works as a data scientist at an energy software company, where she applies artificial intelligence solutions for the energy sector.