The Prediction of the Best Contract in Competitive Bridge Bidding

Yun-An Chen and Chang-Biau Yang

Abstract—**In bridge, the bidding phase is crucial for determining the outcome of the game, as the final contract significantly impacts the result. Previous studies often neglect to provide meaningful hand information implied by the convention card. To address this issue, we incorporate the implications of auctions from the convention card and label the data to improve decision-making. We train a neural network model using a dataset of 450,000 boards collected from Bridge Base Online (BBO). By analyzing the double dummy results, we identify the best contract, which is then adjusted according to the bidding sequence to reflect the most suitable choice. To evaluate the model's performance, we perform stratified 10-fold cross-validation ten times. Each time, the dataset is divided into 10 folds randomly. Our neural network model demonstrates high accuracy, with an average test accuracy of 0.87 and a standard deviation of 0.004. Furthermore, we compare our model against the existing BBO bidding system. Evaluated by International Match Points (IMPs), our model shows competitive performance, with a win rate of 6.4%, a tie rate of 87.6%, and a loss rate of 6%, achieving an average IMPs per board of 0.01. These results indicate that our neural network model is competitive with the BBO system.**

Index Terms—**contract bridge, bridge bidding, convention card, double dummy solver, neural network, 10-fold cross validation**

I. Introduction

In bridge, there are two main phases: bidding and playing. The bidding phase is crucial, as players engage in an auction to determine the best contract. Once the contract is established, the game transitions to the playing phase. The declarer needs to achieve the required number of tricks to make the contract, while the defenders use strategic methods to prevent the declarer from succeeding. This combination of strategic planning and the need to infer hidden information makes bridge a fascinating and complex challenge for AI development.

In both the bidding and playing phases of bridge, limited information is available. Therefore, effectively utilizing the acquired information is crucial. The *double dummy bridge* (DDB) problem assumes each player can see all four hands on the table. The *double dummy solver* (DDS) software is widely used for solving bridge playing problems, analyzing the optimal contract under specific conditions, and offering players detailed strategies. Some studies aim to optimize the efficiency of DDS by using neural networks as DDB estimators, as listed in Table I.

W	N	E	S
P	1♠	P	1NT
P	2♦	P	2♠
P	4♠	P	P
P			

(a)

W	N	E	S
P	1♠	1NT	P
2♦	X	2♥	2♠
P	4♠	P	P

(b)

Fig. 1: Two examples of bidding sequence on BBO. (a) An example of non-competitive bidding, where only the NS side participates in the auction. (b) An example of competitive bidding, where both NS and EW sides participate in the auction.

Compared to the DDS algorithm in playing strategies, the complexity of bidding systems has led to a lack of specific algorithms for bidding. In recent years, some researchers have focused on bidding challenges, as listed in Table II. However, most of these studies often neglect the implicit meanings within bidding systems, making the results difficult to apply in real bridge competitions.

Lin *et al.* [11] proposed the two-stage random forest models to predict the final contract in non-competitive bidding. They extracted the hidden hand features embedded in the bidding sequence with BBO's bidding system. The main features include the player's hand features, the suit lengths for each suit of the player and partner, balanced labels, stopper conditions, vulnerability, and bidding history. The two-stage models are designed to first predict the suit (or NT) and then the game category (partial game, game, slam, or grand slam).

In addition to the concepts proposed by the aforementioned research, integrating convention cards into competitive bidding models is also an important issue. Figure 1 shows two examples of bridge bidding sequence. In competitive bidding, the meaning of bids may change due to opponents' interference.

In this paper, we focus on encoding the bidding history using a convention card and employing neural networks to predict the best contract in competitive bidding. We collect bidding history data from *Bridge Base Online* (BBO), extracting players' hand information. The main features include *high card points* (HCP), card distribution, the bidding sequence, and the upper and lower bounds of card numbers in each suit. We use the results of *double dummy solver* (DDS) to determine the best contract and then use neural networks to predict the final contract.

This research work was partially supported by National Science and Technology Council of Taiwan under contract NSTC 112-2221-E-110-026-MY2.

Yun-An Chen is with Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan.

Chang-Biau Yang is with Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. E-mail: cbyang@cse.nsysu.edu.tw. (Corresponding author)

TABLE I: The research for predicting the number of winning tricks [6].

Year	Author(s)	Method	Note
2004	Mossakowski and Mańdziuk [14]	neural network	winning trick estimation
2006	Mossakowski and Mańdziuk [15]	neural network	hand strength evaluation
2007	Mossakowski and Mańdziuk [16]	neural network	hand strength evaluation
2009	Mossakowski and Mańdziuk [17]	neural network	winning trick estimation
2018	Mańdziuk and Suchan [13]	autoencoder	feature extraction
2021	Kowalik and Mańdziuk [9]	convolutional neural networks	feature extraction
2023	Jan *et al.* [6]	genetic algorithm	hand strength evaluation

TABLE II: The bridge bidding models with machine learning [11].

Year	Author(s)	Method	Note
2006	Amit and Markovitch [1]	partial information decision making	partner's bidding
2007	DeLooze and Downey [2]	self organizing map	non-competitive bidding
2015	Ho and Lin [5]	upper confidence bound	non-competitive bidding
2018	Yeh *et al.* [21]	deep reinforcement learning	non-competitive bidding
2018	Legras *et al.* [10]	inductive logic programming	opening bid
2019	Rong *et al.* [19]	estimation neural network (ENN) policy neural network (PNN)	partner's hand competitive bidding
2019	Gong *et al.* [4]	ENN, PNN	competitive bidding
2020	Lockhart *et al.* [12]	imitation learning	competitive bidding
2023	Sztyber *et al.* [20]	bridgehand2vec reinforcement learning	opening bid
2023	Lin *et al.* [11]	random forest	non-competitive bidding
2025	This paper	neural network	competitive bidding

The results demonstrate that our model achieves high accuracy and predictive ability in competitive bidding scenarios. We perform stratified 10-fold cross-validation ten times, achieving an average test accuracy of 0.87 with a standard deviation of 0.004. Additionally, the model's performance is compared against the BBO's bidding system. When evaluated by *International Match Points* (IMPs), our model demonstrates a win rate of 6.4%, a tie rate of 87.6%, and a loss rate of 6%, resulting in an average of 0.01 IMPs per board.

This paper contains the following chapters. Section II introduces the background knowledge. Section III outlines the research methodology, data preprocessing, and the definition of the best contract. Section IV presents the experimental results. Finally, Section V provides the conclusion of the paper.

II. Preliminaries – Evaluation of Hand Strength

In bridge, accurately evaluating the strength of one's hand is crucial for considering appropriate bidding and playing strategies. Several methods [6, 15, 16] have been developed to evaluate hand strength, broadly categorized into two types: high card points (HCP) and distributional points.

The HCP method estimates the strength of honors. For example, in the Goren system, A, K, Q, and J are assigned 4, 3, 2, and 1 point(s), respectively. Some methods also assign points to 10s (T) Additionally, the short suit method primarily focuses on evaluating the non-trump suits, assigning 1 point for a doubleton, 3 points for a singleton, and 5 points for a void.

TABLE III: The hand evaluation formulas of $H + TL + NL$ for a suit contract [6].

H						TL		NL			Correlation coefficient
$2, 3, \cdots, 8$	9, 10	J	Q	K	A	a	b	void	singleton	doubleton	
0	0	0.5	1.0	2.5	4	1.5	1	3.5	2	0.5	0.918

Jan *et al.* [6] used genetic algorithms to train a series of formulas for designing a more precise method for evaluating hand strength. The suggested formulas [6] are listed as follows:

$$H : \sum_{suit \in hand} \sum_{card \in suit} M^H[card].$$

$$TL : a \times (L_t - b).$$

$$NL : \sum_{suit \in hand, suit \neq t} M^{NL}[L_{suit}].$$

$$L_{wh} : \sum_{suit \in hand} \begin{cases} a \times (L_{suit} - b) & \text{, if } L_{suit} \geq b, \\ & \text{with honors;} \\ 0 & \text{, otherwise.} \end{cases}$$

For a suit contract, the formula combination is $H + TL + NL$, as shown in Table III. For an NT contract, the formula combination is $H + L_{wh}$, as shown in Table IV.

III. The Prediction of the Best Contract

Our approach for predicting the best contract is divided into four stages as follows:

TABLE IV: The hand evaluation formulas of $H + L_{wh}$ for the NT contract[6].

H						L_{wh}		Correlation coefficient
$2, 3, \cdots, 8$	9, 10	J	Q	K	A	a	b	
0	0.5	1	1.5	2.5	4	0	0	0.916

Fig. 2: A screenshot of *Just Declare* mode on BBO.

Stage 1: Dataset and features.
We collect approximately 450,000 boards from the *Just Declare* mode on *Bridge Base Online* (BBO).

Stage 2: The definition of the best contract.
We adjust the final contracts of each board according to the results of the *double dummy solver* (DDS) to ensure the best outcomes.

Stage 3: Neural network architecture.
We use a neural network model to predict the final contract. The input layer consists of 612 neurons, divided into three sections: player's hand features, bidding state information, and bidding explanations. The output layer consists of 37 neurons, corresponding to the possible contract outcomes in bridge bidding (1C to 7N, pass, double).

Stage 4: Performance calculation with IMP scores.

A. Dataset Generation

We collect approximately 450,000 boards from the *Just Declare* mode on *Bridge Base Online* (BBO) [3] as our dataset. To further enhance the diversity of our training data, we employ data augmentation. Our dataset includes different types of final contracts (partial, game, slam, and grand slam), but some types of final contracts may be underrepresented. By using data augmentation, we can increase the amount of data for these less common contracts, thereby balancing the dataset and improving the model's prediction accuracy for all contract types.

We use the *Fisher-Yates shuffle algorithm* [8] to ensure that each combination has an equal probability, as shown in Algorithm 1.

B. Feature Extraction

Each player's bidding history is represented by an 141-bit vector, as shown in Figures 3 and 4.

C. The Definition of the Best Contract

In our study, the term "best contract" refers to the best final contract that maximizes scores in a bridge game. To mitigate the influence of luck, we simulate 100 random distributions for each board and compute the double dummy results for each simulation. The adjustment process involves gathering the original contract from BBO for each board, using DDS to compute the maximum number of tricks of each contract for each player, and determining the best contract based on BBO and DDS results. The final contract determination involves the following two main steps:

Step 1: Suit adjustment. The trump suit is set as the suit bid by BBO. For example, if BBO's bid is 4♡, then the trump suit is set to hearts.

Step 2: Level adjustment. We primarily adjust the final

Algorithm 1 Shuffling with one player hand fixed

Input: $fixed\_hand$ = 13 cards of the target whose bid is to be predicted.
Output: A shuffled deck with one player hand fixed.

$deck \leftarrow [\mathrm{C2}, \mathrm{C3}, \ldots, \mathrm{SK}, \mathrm{SA}]$
remove $fixed\_hand$ from $deck$ ▷ fixed_hand is the 13 cards of the target player
$n \leftarrow \mathrm{length}(deck)$
for i from 1 to n **do**
  $j \leftarrow$ a random index ranging from i to n
  $\mathrm{swap}(deck[i], deck[j])$
end for
$deck \leftarrow fixed\_hand + deck$
return $deck$

W (LHO)	N (PD)	E (RHO)	S (OWN)
1♠	2♠	P	2N
P	3♣	P	3♡
P	P	P	

	Pass hand	...	1♠	1♠A	1♠P	1♠X	...	2♠	2♠A	2♠P	2♠X	2N	2NA	2NP	...	7NX	Vul. own	Vul. opp	Open
LHO	0	...	1	0	0	0	...	0	0	0	0	0	0	1	...	0	1	0	0
PD	0	...	0	0	0	0	...	1	1	0	0	0	0	0	...	0			
RHO	0	...	0	0	0	0	...	0	0	1	0	0	0	0	...	0			
OWN	0	...	0	0	0	0	...	0	0	0	0	1	1	0	...	0			

Fig. 3: An example of encoding bidding state information.

OWN	LHO	PD	RHO
1NT	X	XX	P
?			

	Pass hand	...	1♠	1♠A	1♠P	1♠X	1N	1NA	1NP	1NX	...	7NX
LHO	0	0	0	0	0	0	0	0	0	1	...	0
PD	0	0	0	0	0	0	0	0	0	1	...	0
RHO	0	0	0	0	0	0	0	0	1	0	...	0
OWN	0	0	0	0	0	0	1	0	0	0	...	0

Fig. 4: An example of a bidding sequence with both a double (X) and a redouble (XX).

TABLE V: An example of bidding explanation encoding, where L/H means lower/upper bound on the number of cards in the suit.

	♠		♡		◇		♣		HCP	
	L	H	L	H	L	H	L	H	L	H
LHO	0	5	4	7	0	6	0	7	6	15
PD	5	7	0	4	4	4	0	4	8	15
RHO	0	5	0	7	3	6	0	7	11	21

TABLE VI: Parameter values of our neural network model.

Parameter	Value	Parameter	Value
Input size	612	Activation function	ReLU
Hidden layer size 1	1224	Dropout rate	0.2
Hidden layer size 2	612	Loss function	Cross entropy
Output size	37	Optimizer	Adam
Epoch	600	Learning rate	0.0006
Batch size	4096		

contract category (partial, game, slam, or grand slam) based on the DDS results. However, for consistency in bidding sequences, we may further adjust it according to the contract level. The adjustments are made considering three cases for the final contract bid by BBO and DDS results: the same level, underbid, and overbid.

D. Neural Network Architecture

Our neural network architecture includes fully connected layers, as shown in Figure 5. It consists of an input layer, two hidden layers, and an output layer.

During training, the model uses the cross-entropy loss function [18], as defined in Equation 1.

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c} \log\left(\frac{\exp(x_{n,c})}{\sum_{i=1}^{C}\exp(x_{n,i})}\right), \quad (1)$$

where N is the total number of samples, x represents the input features, y represents the target, and C is the total number of classes. The model is optimized using the Adam optimizer [7]. Table VI lists the parameters in our model.

IV. Experimental Results

We perform the experiments on a computer running macOS 14.4.1, with an Apple M1 CPU and 16 GB of RAM. The neural network models are trained using Python version 3.9.6. Our dataset consists of approximately 490,000 boards, including data from the Just Declare mode on BBO and additional pass data. We adjust the final contracts for 59,281 boards using the DDS, representing 13% of the total dataset. Through data augmentation, the training dataset expands to about 860,000 instances. The experiments are performed with 10-fold cross-validation for ten times. The total execution time takes approximately 30 hours.

A. Main Results

All performance measurements are calculated as the average of the ten 10-fold cross-validations. Table VII presents the performance metrics for predicting the best contract, with scores averaged across the ten experiments. Our neural network model demonstrates high accuracy, with an average test accuracy of 0.872 and a standard deviation of 0.004.

As presented in Table VIII, our model demonstrates a win rate of 6.4%, a tie rate of 87.6%, and a loss rate of 6.0%, resulting in an average of 0.01 IMPs per board. These results indicate that our neural network model performs competitively with the BBO system.

TABLE VII: Performance metrics of our model in predicting the best contracts across ten 10-fold cross-validations.

	Mean μ (standard deviation σ)
Accuracy	0.872 ($\pm$ 0.004)
Macro F1-score	0.794 ($\pm$ 0.009)
Weighted F1-score	0.875 ($\pm$ 0.005)

TABLE VIII: Performance comparison of our model and BBO with IMPs across ten 10-fold cross-validations.

Win rate	Tie rate	Loss rate	IMPs per board (standard deviation σ)
6.4 %	87.6 %	6.0 %	0.01 ($\pm$ 0.014) IMPs

Additionally, Table IX provides a performance comparison of our defined best contract and BBO's contract using IMP scores for each board. Our defined best contracts demonstrate a win rate of 9.8%, a tie rate of 87.0%, and a loss rate of 3.2%, resulting in an average of 0.45 IMPs per board. This comparison highlights the effectiveness of our algorithms in finding the best contracts.

In summary, the neural network model shows strong predictive performance for common contracts, while less frequent contracts are more challenging. This difference highlights the model's reliance on the amount of training data for accuracy.

B. Model Comparison and Analysis

We compare two different models to predict the best contract in competitive bridge bidding: neural network and random forest models. For a fair comparison, we use the same set of input features for both models, consisting of 612 features in total. The performance for these two models are presented in Table X. Table XI lists the parameters of the random forest model.

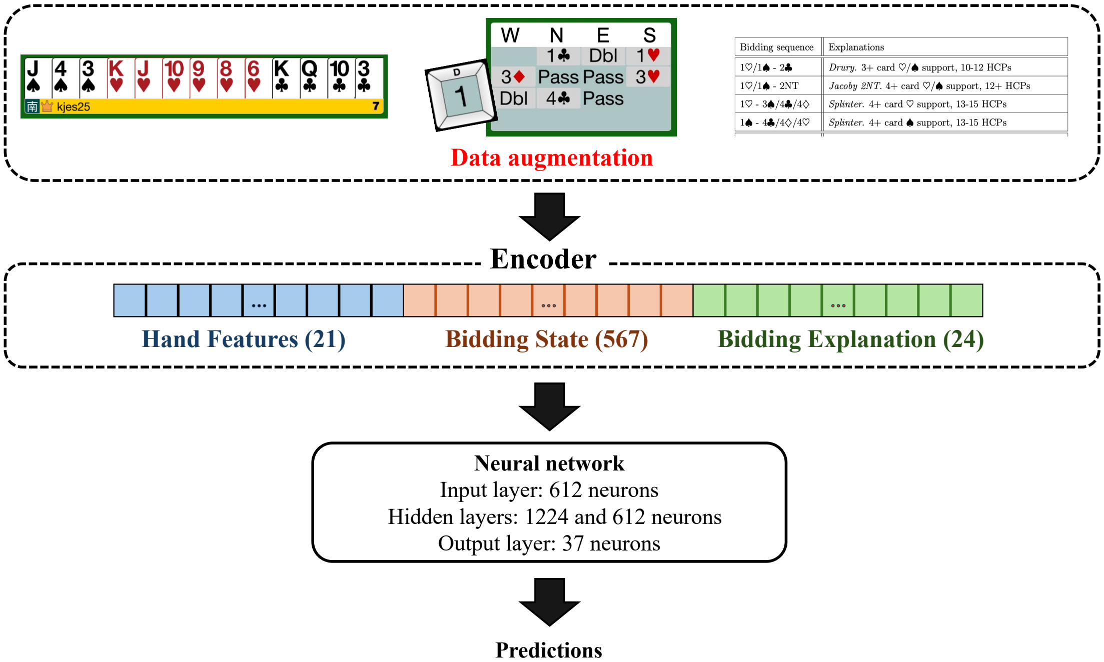
The random forest model performs slightly less well in lower-level partial contracts compared to the neural network model. In game contracts, the random forest model performs well in 3NT (F1-score = 0.92) and 4♡ (F1-score = 0.90), but

TABLE IX: The comparison of our defined best contract and BBO with IMPs.

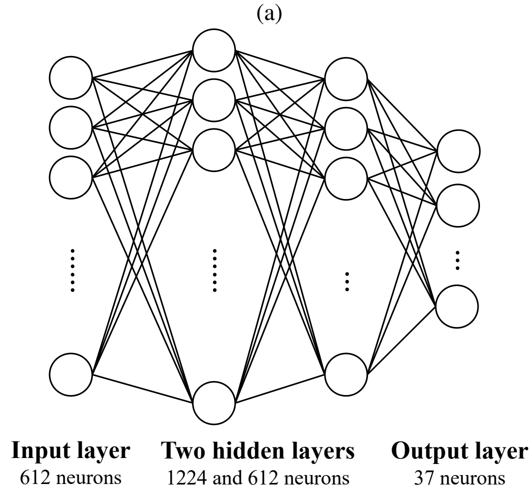
Win rate	Tie rate	Loss rate	IMPs per board
9.8 %	87.0 %	3.2 %	0.45 IMPs

TABLE X: Performance comparison of neural network and random forest models. Experiments are tested on the first 10-fold cross validation.

Model	Accuracy (σ)	Win rate	Tie rate	Loss rate	BBO comparison (σ)
Neural network	0.871 ($\pm$ 0.003)	6.4 %	87.6 %	6.0 %	0.01 IMPs ($\pm$ 0.016)
Neural network (w/o augmentation)	0.865 ($\pm$ 0.003)	6.5 %	86.9 %	6.6 %	-0.02 IMPs ($\pm$ 0.008)
Random forest	0.854 ($\pm$ 0.001)	6.8 %	85.1 %	8.1 %	-0.07 IMPs ($\pm$ 0.009)



(a)



(b)

Fig. 5: The neural network architecture in our study. (a) The data flow from hand features, bidding state, and bidding explanation through data augmentation into a neural network. (b) The neural network with fully connected layers.

TABLE XI: Parameter values of the random forest model.

Parameter	Value	Parameter	Value
n_estimators	100	random_state	None
criterion	gini	bootstrap	True
max_features	auto	stratify	True
max_depth	None	class_weight	balanced
min_samples_split	2	test_size	0.1
min_samples_leaf	1	train_size	0.9

these results are still not as good as those of the neural network model.

In the comparison between the neural network model and the random forest model for predicting high-level contracts, including game, slam, and grand slam contracts, the neural network model consistently demonstrates superior performance.

The results indicate that the neural network model exhibits a significantly superior performance compared to the random forest model in predicting the best contracts. Specifically, the neural network demonstrates enhanced effectiveness in handling high-level contracts, consistently providing more accurate predictions. Although both models achieve comparable levels of accuracy, the neural network's superior performance in competitive scenarios suggests that it is a more appropriate

TABLE XII: Comparison of model performance using different hand feature encoding methods. Experiments are tested on the first fold of the first 10-fold cross validation.

Encoding method	Accuracy (%)
Binary encoding	0.80
Feature extraction (Goren point system)	0.85
Feature extraction ($H + TL + NL$ and $H + L_{wh}$)	0.87

TABLE XIII: Comparison of model performance using different bidding encoding methods. Experiments are tested on the first fold of the first 10-fold cross validation.

Encoding method	Accuracy (%)
Our encoding method	0.87
Double indicator	0.81
Label encoding	0.66

choice for predicting the best contracts in competitive bridge bidding.

C. Feature Representations Analysis

Accurate and effective feature representation is crucial for the performance of a model when predicting bridge hands. To determine the most suitable method for representing the information of a hand, we employ several feature representation techniques. Experiments are tested on the first fold of the first 10-fold cross validation.

Table XII presents the results of different methods for encoding hand features: binary encoding and feature extraction. In the binary encoding method, each card in a 13-card hand is represented by a 52-dimensional binary vector. The feature extraction method includes key information such as HCP, the number of cards in each suit, and the presence of honor cards (A, K, Q, J). Additionally, we test two methods for calculating HCP: the hand strength evaluation formulas $H + TL + NL$ and $H + L_{wh}$ proposed by Jan *et al.*[6], as well as the Goren point system (A=4, K=3, Q=2, and J=1). As one can see, the feature extraction method with $H + TL + NL$ and $H + L_{wh}$ has the superior performance.

For the bidding state information, examples of double indicator and label encoding are shown in Figure 6. Our encoding method ensures the preservation of both the order and contextual information of each bid, which contains the complete information of the bidding sequence. This approach addresses the aforementioned issues: (1) It accurately identifies the position of each double, even within competitive bidding sequences. (2) It avoids erroneous ranking based on numerical order, thus preventing unnecessary biases in the model's learning process. (3) It captures the implicit concepts of convention cards within bids, allowing these hidden pieces of information to influence the model's predictions. This is particularly important when dealing with complex bidding strategies.

OWN	LHO	PD	RHO
	1♠	X	P
P	2♣	P	P
?			

	Pass hand	1C	1CP	1D	1DP	1H	1HP	1S	1SP	1N	1NP	2C	2CP	...	7NP
LHO	0	0	0	0	0	0	0	1	0	0	0	1	0	...	0
PD	0	0	0	0	0	0	0	0	0	0	0	0	0	...	0
RHO	0	0	0	0	0	0	0	0	1	0	0	0	0	...	0
OWN	0	0	0	0	0	0	0	0	1	0	0	0	0	...	0

1C_double	1D_double	1H_double	1S_double	1N_double	...	7N_double
0	0	0	1	0	...	0

(a)

	Pass hand	1C	1D	1H	1S	1N	2C	2D	2H	2S	2N	3C	3D	...	7N
LHO	0	0	0	0	3	0	3	0	0	0	0	0	0	...	0
PD	0	0	0	0	2	0	1	0	0	0	0	0	0	...	0
RHO	0	0	0	0	1	0	1	0	0	0	0	0	0	...	0
OWN	0	0	0	0	1	0	0	0	0	0	0	0	0	...	0

(b)

Fig. 6: Examples of encoding bidding sequence with different methods. (a) An example of double indicators. (b) An example of label encoding.

TABLE XIV: Accuracy comparison of different neural network architectures. Experiments are tested on the first fold of the first 10-fold cross validation.

Neural network type	Acc. Training (%)	Acc. Testing (%)
612 - 306 - 37	97.77	81.35
612 - 1224 - 37	99.17	84.78
612 - 306 - 153 - 37	96.74	79.95
612 - 1224 - 408 - 37	98.95	86.42
612 - 1224 - 612 - 37	99.18	87.38
612 - 256 - 128 - 64 - 37	92.94	72.42
612 - 512 - 256 - 128 - 37	96.85	80.03
612 - 1224 - 612 - 306 - 37	98.75	84.66
612 - 512 - 256 - 128 - 64 - 37	95.45	77.81
612 - 1024 - 512 - 256 - 128 - 37	97.68	83.95

D. Analysis of Hidden Layer Configuration

We conduct experiments with different configurations of hidden layers and neurons to determine the best architecture for our model based on previous studies. Table XIV shows that the model performs best with two hidden layers consisting of 1224 and 612 neurons, achieving a training accuracy of 99.18% and a testing accuracy of 87.38%. Additionally, other configurations, such as the 612-1024-512-256-128-37 and 612-1224-612-306-37 models, also achieve high testing accuracies. This indicates that increasing the number of neurons in the first hidden layer can effectively enhance the model's ability to learn complex patterns and improve predictive performance.

V. Conclusion

In this paper, we present a comprehensive approach to predicting the best contract in competitive bridge bidding using a neural network model. The bidding phase in bridge is crucial for determining the outcome of the game, as the final contract significantly impacts the result. Our work addresses the challenge of incorporating meaningful hand information implied by convention cards into the bidding process. We achieve this by labeling data to reflect the implications of auctions from the convention card, thereby improving decision-making in competitive scenarios.

We train a neural network model on a dataset of approximately 490,000 boards collected from BBO. By analyzing double dummy results, we identify the best contracts and adjust them according to the bidding sequence to reflect the most suitable choices. Our neural network model demonstrates high accuracy, with an average test accuracy of 0.87 and a standard deviation of 0.004.

Additionally, we compare our model's performance with the existing BBO bidding system using IMPs as the evaluation metric. The results indicate that our model demonstrates competitive performance, with a win rate of 6.4%, a tie rate of 87.6%, and a loss rate of 6%, resulting in an average of 0.01 IMPs per board.

Given that our research focuses on predicting the best contract for a given board state, which entails determining whether to bid based on available information and identifying the most suitable contract under those circumstances, we emphasize data processing and employ linear models for prediction. Future work could extend to predicting each bid in the competitive bidding sequence, employing different models such as long short-term memory (LSTM) or recurrent neural networks (RNN) for time-series analysis.

References

[1] A. Amit and S. Markovitch, "Learning to bid in bridge," *Machine Learning*, Vol. 63, No. 3, pp. 287–327, 2006.

[2] L. L. DeLooze and J. Downey, "Bridge bidding with imperfect information," *Proceedings of 2007 IEEE Symposium on Computational Intelligence and Games*, pp. 368–373, Honolulu, Hawaii, USA, 2007.

[3] F. Gitelman, "Bridge Base Oline." https://www.bridgebase.com/.

[4] Q. Gong, Y. Jiang, and Y. Tian, "Simple is better: Training an end-to-end contract bridge bidding agent without human knowledge," *Proceedings of Real-world Sequential Decision Making Workshop in International Conference on Machine Learning*, Long Beach, USA, 2019.

[5] C.-Y. Ho and H.-T. Lin, "Contract bridge bidding by learning," *Proceedings of the 2015 Association for the Advancement of Artificial Intelligence (AAAI) Workshop*, Hyatt Regency in Austin, Texas, USA, pp. 1–7, 2015.

[6] J.-P. Jan, K.-S. Huang, and C.-B. Yang, "Hand strength evaluation in bridge games with the genetic algorithm," *Proceedings of the 40th Symposium on Combinatorial Mathematics and Computational Theory*, pp. 50–55, Taoyuan, Taiwan, 2023.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2017.

[8] D. E. Knuth, *The art of computer programming*, Vol. 3. Pearson Education, 1997.

[9] S. Kowalik and J. Mańdziuk, "Towards human-level performance in solving double dummy bridge problem," *Proceedings of the 28th International Conference on Neural Information Processing*, pp. 15–27, Sanur, Bali, Indonesia, 2021.

[10] S. Legras, C. Rouveirol, and V. Ventos, "The game of bridge: A challenge for ILP," *Proceedings of the International Conference on Inductive Logic Programming*, pp. 72–87, Ferrara, Italy, 2018.

[11] Y.-Y. Lin, C.-B. Yang, and K.-S. Huang, "The prediction of the most suitable contract in non-competitive bridge bidding," *Proceedings of the 28th International Conference on Technologies and Applications of Artificial Intelligence*, Yunlin, Taiwan, 2023.

[12] E. Lockhart, N. Burch, N. Bard, S. Borgeaud, T. Eccles, L. Smaira, and R. Smith, "Human-agent cooperation in bridge bidding," *arXiv preprint arXiv:2011.14124*, 2020.

[13] J. Mańdziuk and J. Suchan, "Solving the double dummy bridge problem with shallow autoencoders," *Proceedings of the 25th International Conference on Neural Information Processing*, pp. 268–280, Siem Reap, Cambodia, 2018.

[14] K. Mossakowski and J. Mańdziuk, "Artificial neural networks for solving double dummy bridge problems," *Proceedings of the Artificial Intelligence and Soft Computing*, pp. 915–921, Zakopane, Poland, 2004.

[15] K. Mossakowski and J. Mańdziuk, "Neural networks and the estimation of hands' strength in contract bridge," *Proceedings of the Artificial Intelligence and Soft Computing*, pp. 1189–1198, Zakopane, Poland, 2006.

[16] K. Mossakowski and J. Mańdziuk, "Example-based estimation of hand's strength in the game of bridge with or without using explicit human knowledge," *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 413–420, Honolulu, Hawaii, USA, 2007.

[17] K. Mossakowski and J. Mańdziuk, "Learning without human expertise: A case study of the double dummy bridge problem," *IEEE Transactions on Neural Networks*, Vol. 20, No. 2, pp. 278–299, 2009.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *ArXiv*, Vol. abs/1912.01703, 2019.

[19] J. Rong, T. Qin, and B. An, "Competitive bridge bidding with deep neural networks," *Proceedings of the International Conference On Autonomous Agents and Multi-Agent Systems*, pp. 1–9, Montreal, Canada, 2019.

[20] A. Sztyber-Betley, F. Kołodziej, J. Betley, and P. Duszak, "Bridgehand2vec bridge hand representation," *arXiv preprint arXiv:2310.06624*, 2023.

[21] C.-K. Yeh, C.-Y. Hsieh, and H.-T. Lin, "Automatic bridge bidding using deep reinforcement learning," *IEEE Transactions on Games*, Vol. 10, No. 4, pp. 365–377, 2018.