On the Use of Simulated Future Information for Evaluating Game Situations

Yudai Suzuki and Tomoharu Nakashima

Graduate School of Humanities and Sustainable System Sciences
Osaka Prefecture University
{yudai.suzuki, tomoharu.nakashima}@kis.osakafu-u.ac.jp

Abstract. A FOrward Simulation for Situation Evaluation (FOSSE) approach for evaluating game situations is proposed in this paper. FOSSE approach considers multiple future situations to quantitatively evaluate the current game situations. Since future situations are not available during an ongoing game in real time, they are generated by what is called forward simulation. Then the current game situation is evaluated using the future game situations as well as the current situation itself. First, we show the evaluation performance can be increased by using successive situations in time through preliminary experiments. Especially, the effectiveness of using future information rather than using past information is shown. Then, we present FOSSE approach where both the current and the future information of game situations are used to evaluate the current game situation. In the FOSSE approach, the future game situations are generated by forward simulation. Computational experiments are conducted to investigate the effectiveness of the proposed approach.

Keywords: Evaluating Situation · Forward Simulation · Recurrent Neural Network · Deep Learning· Time Series Data · Soccer Simulation

1 Introduction

In sports, it is useful to perceive the superiority during a game. If the game situation can be evaluated quantitatively, the degree of dominance for teams can be accurately grasped. Furthermore, it is considered that the quantitative evaluation can be applied to strategy switching guidelines in a game and to the field of automatic live broadcasting of sports. However, quantitative evaluation is difficult in the dynamic game situation. For this problem, we employ machine learning method for quantitative evaluation.

As an experimental environment of this research, we use RoboCup Soccer Simulation 2D League [1]. For the metric of evaluating game situation, Nakashima and Pomas [2] proposed a metric called *SituationScore* which represents the degree of dominance in a soccer game. This paper also uses *SituationScore* with a minor modification to evaluate the game situation.

In general, only the current situation is considered when evaluating the game situation. However, since the game progresses dynamically, and accordingly the

situation drastically changes, especially in soccer, it is difficult to capture the degree of the dominance in the game with only a single situation information. In this paper, we investigate the use of multiple situations to capture the degree of the dominance in a game.

If future information is available during a game, it is possible to evaluate the game situation with higher accuracy than with only the current information as well as the past information. However, such future information is not available during an ongoing game. To solve this problem, we propose a FOrward Simulation for Situation Evaluation (FOSSE) approach for learning a machine learning model that generates future situations by simulation and then evaluates the current situation by using the generated future situations as well as the current one itself.

The proposed FOSSE approach consists of two parts. The first part is forward simulation for generating the estimated future game situations. The other one is situation evaluation for producing the value of $SituationScore$ from the time series of game situations. We employ a Recurrent Neural Network (RNN) as the simulation model and a Deep Neural Network (DNN) as the evaluation model.

In the following sections, firstly, we show that the prediction accuracy of an evaluation model can be improved by using multiple-situation information comparing with a single-situation model. Secondly, the experiment using actual data shows that future information is more helpful than past information in an evaluation model. Finally, we indicate the effectiveness of the proposed method based on FOSSE approach through computational experiments.

It should be noted that the meaning of "evaluation" in this paper is to understand the field situation such as the degree of domination by a currently attacking team and the likeliness of scoring by that team. There are other research where the evaluation means the value of a state or an action in determining the next action by an individual player agent. Although the situation evaluation in this paper can also be used for such purpose in the future, this is not the focus of this paper. We focus on the evaluation of a field situation not from the view point of the soccer players who can only see the situation in their visual area, but from the view point of a coach or spectators who can watch the whole soccer field.

2 Quantitative Evaluation of Game Situations in RoboCup Soccer

We employ RoboCup Soccer Simulation 2D League [1] as the subject of study in this paper. Generally, as a measure to represent the degree of team dominance in a game, commonly used information would be the ball-possessing team and the ball location in the soccer field. However, such simple indices cannot accurately grasp the degree of team dominance. Therefore, another index that quantitatively expresses the game situation is required.

Nakashima and Pomas [2] proposed $SituationScore$, which represents the value of a game situation. In their work, a game situation is quantitatively eval-

uated by using time cycles until the next goal. This paper uses the same idea of *SituationScore* with a minor modification.

This section first introduces the RoboCup Soccer Simulation 2D League. Then, some modification to *SituationScore* is presented.

2.1 RoboCup Soccer Simulation 2D League

RoboCup [1] is a research project that focuses on the development of robotics and artificial intelligence. There are various leagues in this project. RoboCup Soccer Simulation 2D League is one of such leagues. It does not use real soccer robots but simulated soccer players. The players are represented by a two-dimenional circle as shown in Fig. 1. They play soccer in a two-dimensional virtual soccer field that is set up on a computer. The positions of the players and the ball are represented as a two-dimensional vector. Each player is programmed as an independent agent unlike a video soccer game where there is one central system that control all the objects such as all players. A game consists of 6,000 time cycles and one cycle is discretized in 0.1 second. When the game is over, a game log is generated in which all the game information such as the position coordinates of the player and the ball in each cycle are included.

2.2 Modification to *SituationScore*

Nakashima and Pomas [2] proposed a metric called *SituationScore*. This metric represents the value of a game situation. The value of *SituationScore* increases as the game situation is close to the time of goal scoring. In its original definition, the maximum value of *SituationScore* was 100 (when the left team scores), and the minimum value was -100 (when the right team scores). In the original definition of *SituationScore*, the superiority and inferiority of the teams are considered for all the time steps. However, it is difficult to predict the value of *SituationScore* when it is close to zero, which is a boundary situation between the superiority and the inferiority of the teams. Due to this problem, some changes were made to *SituationScore* in this paper so that the lower limit is set to 0 assuming that *SituationScore* presents only the degree of dominance of either one team. Also, because it was also difficult to correctly predict the value of situations far from the goal, we only consider those situations where a goal is scored within 50 time cycles. As a result, in this paper, we slightly modify the definition of *SituationScore* as follows:

$$SituationScore(t) = 50 - n, \tag{1}$$

where n represents the number of remaining cycles from t until the next score. In this paper, the range of *SituationScore* is $0 \leq SituationScore \leq +50$, which means that we only consider the goals by the left team. The value of *SituationScore* for the right team can be separately defined by switching the sign of the value (i.e., positive to negative). Figure 2 shows an example game situation which is nine time cycles before the left team scores a goal along with its *SituationScore*.

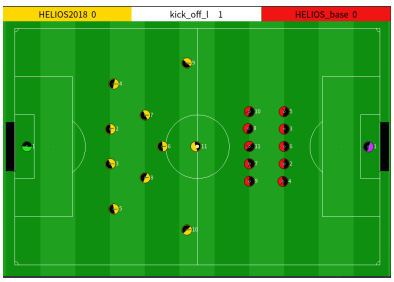

Fig. 1: Game screen of the RoboCup Soccer Simulation 2D League.

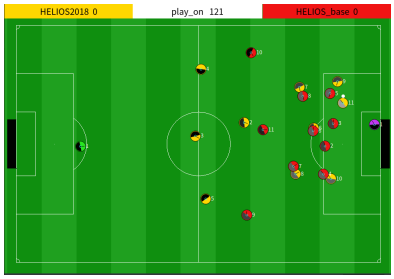

Fig. 2: Situation that is nine time cycles before the left team scores. The value of *SituationScore* is +41.

2.3 Dataset

The dataset in the computational experiments in this paper was generated by the following steps:

1. A game between HELIOS2018 [3] and agent2d [4] is performed for a specified number of times.
2. The log files of the games are analyzed by using Python scripts to detect at which cycles goals were scored.
3. The numerical information of the soccer field for 50 time cycles before each goal of the left team (i.e., HELIOS2018 [3]) are recorded as well as their corresponding *SituationScore* values. The recorded information is saved in a file for each of the time cycles. The numerical information includes the position of 22 players and the ball. The value of *SituationScore* is calculated as in (1). This *SituationScore* value is used as the ground truth for the situation evaluation.

A dataset containing the numerical field information for about 394,350 time cycles was constructed from 1,000 games. This dataset was then split into three parts as follows: training data ($5,490 \times 50$ time cycles), validation data (788×50 time cycles), and test data ($1,609 \times 50$ time cycles). In the rest of this paper, we use this dataset in all experiments.

3 Situation Evaluation with Multiple Situations

3.1 Evaluation Model

This section presents the investigation into the effect of using multiple situations on the accuracy of the trained model for situation evaluation. We employ a simple DNN as an evaluation model of game situations. This model produces the value of *SituationScore* at time cycle t. The overview of the DNN model is shown in Fig. 3. In this figure, $\boldsymbol{X}$ is the information of the game situation such as the

position of the players and the ball. $\boldsymbol{X}_t$ is the information of the current (time cycle t) game state. $\boldsymbol{X}_{t-n_p}$ is the past state information (i.e., n_p time cycles before the current time cycle). $\boldsymbol{X}_{t+n_f}$ is the future state information (i.e., n_f time cycles after the current time cycle).

Numerical experiments are conducted in the next subsection in order to evaluate the performance of the trained model with various combination of input game situations.

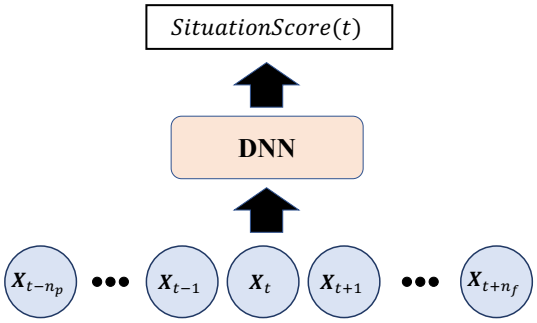

Fig. 3: The overview of Deep Neural Network.

3.2 Experiment

Experimental Settings The purpose of the experiments in this section is to examine the usefulness of using multiple field information with successive time cycles for evaluating the field situation (i.e., predicting the value of $SituationScore$). We compare the following four models with different combinations of game situations for the input of the DNN.

Model 1: Single situation (only the current game situation)
Model 2: Multiple situations (the current, past, and future game situations)
Model 3: Multiple situations (the current and past game situations)
Model 4: Multiple situations (the current and future game situations)

Each architecture is shown in Figs. 4 $\sim$ 7. The number of hidden layers is fixed to 20 for all models, each hidden layer has 16 units, and the layers are fully-connected. For the training of the DNNs, we set the batch size to 64, and used Adam [5] optimizer with the initial learning rate $= 0.001, \beta_1 = 0.9, \beta_2 = 0.999$. Table 1 indicates the experimental settings. The dimensionality of input data in each situation is one of the following three types: two (the $x - y$ coordinates of the ball position), 24 (the $x - y$ coordinates of the ball position and the left team player's positions), and 46 (the $x - y$ coordinates of the ball position and the all player's positions). Past and future information of 5 time cycles are used for Models 2, 3, and 4.

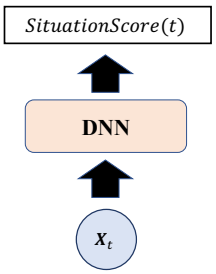

Fig. 4: Model 1.

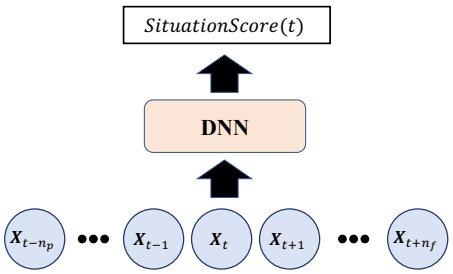

Fig. 5: Model 2.

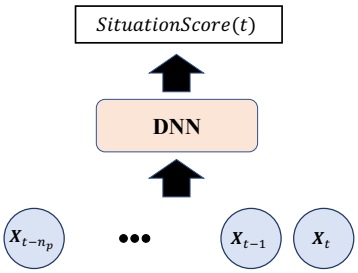

Fig. 6: Model 3.

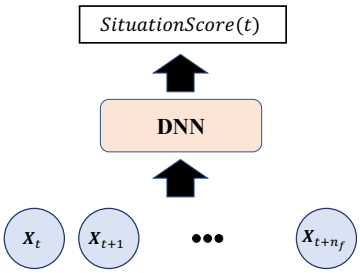

Fig. 7: Model 4.

Table 1: Experimental settings.

Input data	Single situation (current) Multiple situations (current, past, and future) Multiple situations (current and past) Multiple situations (current and future)
Dimensionality for one situation	2 inputs (Ball pos) 24 inputs (Ball pos, Left team's player pos) 46 inputs (Ball pos, All player pos)
Output	*SituationScore*

We use Mean Absolute Error (MAE) as the quality measure of the trained model's accuracy.

Results The experimental results are shown in Table 2. This table shows the effectiveness of using multiple situations compared with single situation. We can see that Model 1 with only a single situation (i.e., the current game situation) for input produced the largest value of MAE for all experimental settings. This is because the dominance trend in the dynamic game is captured by using multiple situations.

In addition, the effectiveness of using future information is represented. It turns out that future information is more effective than past information for evaluating the game situation. It is important to consider for evaluating situation, how the game is going to develop from the current situation, not how the game developed up to the current situation.

The advantage of using multiple situations with future information is also demonstrated in the experimental results. Nevertheless, it has the problem in using future information as the model's input. That is, the problem is that the future information is not available during ongoing game in the real time. If there is a way to obtain future information, that would be helpful for the situation evaluation. The next section describes the proposed method that is the solution for this problem.

Table 2: Experimental results.

Dimensionality of one situation	Model	MAE
2 inputs	1	3.94
	2	3.38
	3	3.76
	4	**3.31**
24 inputs	1	3.84
	2	3.36
	3	3.57
	4	**3.11**
46 inputs	1	3.51
	2	3.32
	3	3.45
	4	**3.07**

4 FOSSE Approach for Evaluating Field Situation

4.1 FOSSE Approach

In the last section, it was shown that using past and future multiple situations helps enhance the performance of the trained model for situation evaluation. Especially, using future situations produced the best accuracy among the considered four models. There is, however, a problem in real-time application that the future information is not available during an ongoing game. To solve this problem, we propose FOSSE (FOward Simulation for Situation Evaluation) approach. Figure 8 shows the overview of FOSSE approach. This approach consists of two parts: forward simulation part and situation evaluation part. The forward simulation part generates the estimation of the future information from the current and the past game situations. Using the generated future information as well as the past and the current field information, the situation evaluation part

produces the value of *SituationScore* at time cycle t. The following subsections explain each part of FOSSE approach.

In this section, firstly, we explain forward simulation in detail. Secondly, we explain the method to evaluate situation by FOSSE approach. Finally, the computational experiments are conducted to show the effectiveness of the proposed methods.

4.2 Forward Simulation

The forward simulation part is shown Fig. 9. The forward simulation takes the past situations as input and generates the estimated field situation of the future.

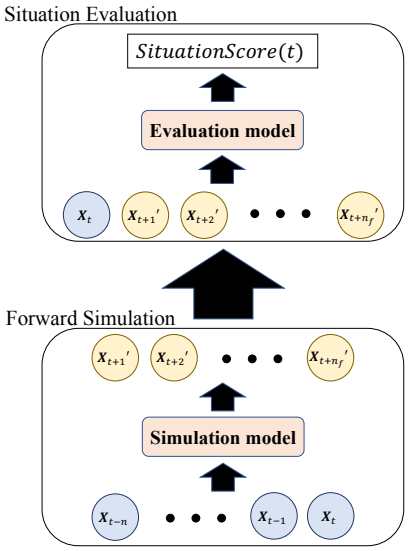

Fig. 8: FOSSE approach.

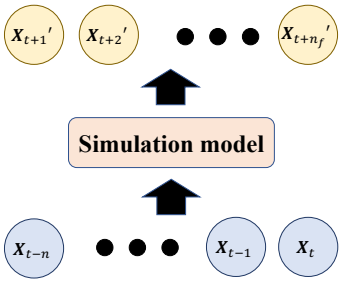

Fig. 9: Forward simulation.

Recurrent Neural Network RNN is a type of neural networks that deals with time series data through its iterative use. It takes the output vector from the previous RNN block at time $t-1$ and the game situation at time cycle t as input. The output vector is used as the input of the next RNN block at time $t+1$. The future game situations are simulated through the above process. This process is called forward simulation for predicting the field situation of future time cycles (i.e., the future game situations). This process is shown in Fig. 10. This figure shows the process of generating the future game situation at time cycle $t+1$ with a time series of previous game situations from time cycle $t-n$ to time cycle t. Each piece of information in the time series $\{\boldsymbol{X}_{t-n}, \ldots, \boldsymbol{X}_t\}$ is

processed by the same block. The block is generally represented as a hidden layer of the RNN. After the last piece of the time series is processed by the block, the estimated next situation is generated after a fully-connected layer (FC).

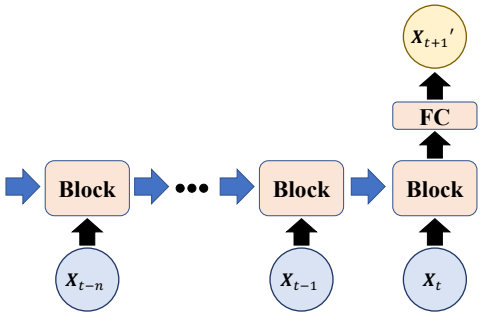

Fig. 10: Recurrent Neural Network for forward simulation.

Related Work There are several works that are related to the forward simulation using RNNs. Aida et al.[6] realized the trajectory prediction of surrounding vehicles. They look ahead for automatic driving of vehicles. Also, in the field of health care, Edward et al.[7] presented a work where physicians' diagnosis and dosing order for patients were predicted by RNNs using the vast amount of time series data obtained from electronic medical records.

Zhiyuan et al.[8] uses Long Short-Term Memory (LSTM) [9], an extended version of the RNN. They applied the LSTM for flight trajectory prediction. Although this task seemed more difficult than simple vehicle trajectory prediction, a high prediction performance was demonstrated. Alexandre et al.[10] tackled the problem of tracking the people in the crowd with the LSTM by introducing social pooling which shares the information of the neighboring persons.

In the above related works, they indicated that the RNN can successfully predict future situations from time series data. Furthermore, they also indicated that the effectiveness of the LSTM even in the difficult tasks. Based on these discussions, this paper also employs the LSTM as an architecture of RNN for the forward simulation part (i.e., we use the LSTM for the iterative block in Fig. 10).

Experiment In the computational experiments of this subsection, we investigate the accuracy of the forward simulation using the LSTM. Specifically, we investigate the prediction accuracy of the future game situation that are generated by iteratively applying the trained LSTM. For training the LSTM model we set batch size to 512 and used Adam optimizer [5] with the initial learning rate $= 0.001, \beta_1 = 0.9, \beta_2 = 0.999$. The LSTM generates a 512-dimensional output vector after taking a single-situation information and the output from the previous

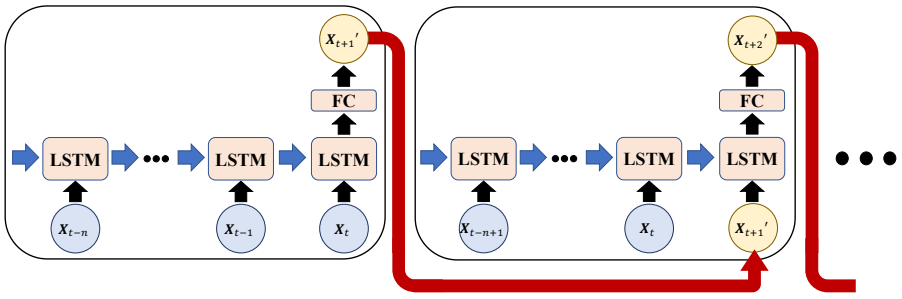

Fig. 11: The architecture of forward simulation by using LSTM.

LSTM block as the input for the next LSTM block. The output vector is used as a part of input for the next LSTM.

In the computational experiments, the number of generated future field situations by the forward simulation is specified to the number of the past situations in the input time series. Figure 11 shows this procedure. For example in the case of four past situations, first, the four past situations $\boldsymbol{X}_{t-4}$, $\boldsymbol{X}_{t-3}$, $\boldsymbol{X}_{t-2}$, $\boldsymbol{X}_{t-1}$, and the current situation $\boldsymbol{X}_t$ are given as the input to the model in order to generate the estimated next situation $\boldsymbol{X}'_{t+1}$. A full-connection layer (FC) is used to generate the estimated next situation $\boldsymbol{X}_{t+1}$ after processing the last piece of the input time series. Then, $\boldsymbol{X}'_{t+2}$ is predicted with another five situations of $\boldsymbol{X}_{t-3}$, $\boldsymbol{X}_{t-2}$, $\boldsymbol{X}_{t-1}$, $\boldsymbol{X}_t$, $\boldsymbol{X}'_{t+1}$ (i.e., the predicted values in the last iteration). This procedure is repeated four times to finally generate the estimated future game situation $\boldsymbol{X}'_{t+4}$. The error between the last predicted $\boldsymbol{X}'_{t+4}$ and the actual value is investigated. Evaluation of the each models is made based on MAE between the models output and the ground truth of each of the corresponding objects' positions.

Table 3 indicates the results of the experiment. The results show that the prediction for three situations has less error than that for five situations. As a matter of course, the results indicate that prediction is more difficult as the number of situations increases, since the predicted values is repeatedly stacked as input instead of actual values.

Table 3: Experimental results of forward simulation using LSTM.

# of situations	Dimensionality of one field information	MAE
Three	2 inputs	1.27
	24 inputs	0.67
	46 inputs	0.62
Five	2 inputs	1.89
	24 inputs	1.12
	46 inputs	1.34

4.3 Evaluation of Game Situations by FOSSE Approach

In Section 3, it was shown that the architecture of the DNN that uses future information produced the best situation evaluation among the four investigated models. Thus, we employ that type of the DNN model as an evaluation model. Since the future information is not available during a game, we estimate it by forward simulation that was described in Section 4.2. Moreover, based on the the results of the computational experiments in Section 4.2, the LSTM is employed as a forward simulation part in our FOSSE architecture. The overview of the resultant FOSSE architecture that we employ in this paper is shown in Fig. 12. In our FOSSE architecture, a field situation is evaluated by using the predicted future information (i.e., $\boldsymbol{X}'_{t+1}, \ldots, \boldsymbol{X}'_{t+n_f}$) generated by forward simulation. The DNN model and the RNN model are separately constructed. In the forward simulation, the LSTM model predicts the field situation of the next time cycle as shown the bottom part of Fig. 12. Then, in the situation evaluation part, the *SituationScore* of the current game situation is estimated using the current game situation as well as the estimated future game situations that are generated by the forward simulation.

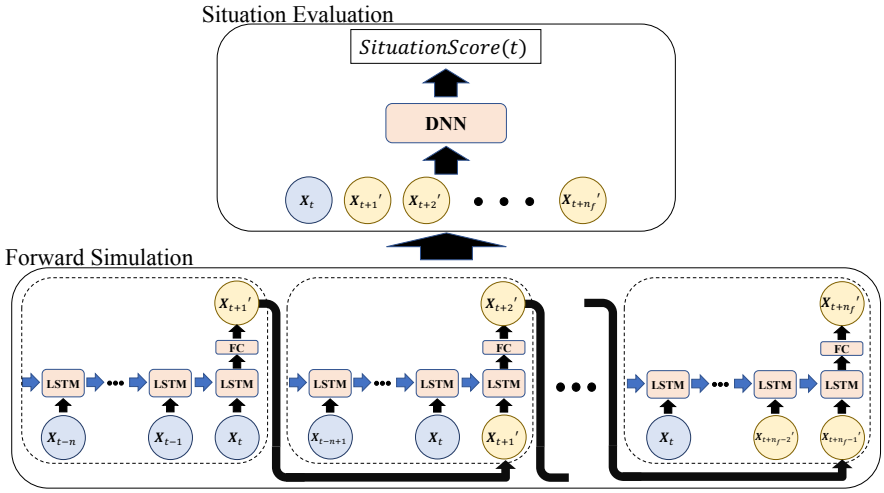

Fig. 12: The overview of our FOSSE architecture with DNN and LSTM.

4.4 Experiment

Experimental Settings Table 4 indicates the settings of the models that are used in the computational experiment in this section. We compare the accuracy performance of four trained DNN models using single situation, multiple past situations, multiple future situations, and multiple predicted future situations (i.e., the proposed method).

Table 4: Experimental settings of evaluating situation.

Input	Single Past+current (Multiple-Past) Future+current (Multiple-Future) Predicted future+current (Multiple-Predict)
Dimensionality per situation	2 inputs (Ball pos.) 24 inputs (Ball pos. and Left team's player pos.) 46 inputs (Ball pos. and All player pos.)
# of input situations	1, 3, 5

Results Table 5 shows the results of the experiments. It is shown that higher accuracy is demonstrated by the proposed method than using the field situation of a single situation. Besides, the performance of the proposed method is better than using multiple past situations when the field situations of three successive time cycles are used as input. As a result, it is shown that evaluation using predicted future situations by simulation model is more useful in the situation evaluation than using already-known past situations. Although using multiple future situations leads to a high accuracy of the trained model, this is only ideal because the future information is not available at the current time. Thus, using multiple future situations is not a real option for model building in real-time games. On the other hand, the proposed method can be used during ongoing games because the field situation of future time cycles is generated by forward simulation.

On the contrary, when the field situations of five successive time cycles are used, the proposed method outperforms using single situation. This, however, cannot show an effectiveness compared with the model using past situations. This is considered to be due to the fact that the error of the forward simulation model's output increases as the number of situations increases as described in Section 4.2. Although this paper employs a simple simulation model, it can be expected that the accuracy will be improved by elaborating more on the forward simulation model. The improvement of the forward simulation model is left for our future work.

The results of the computational experiments show the effectiveness of the proposed method that evaluates the situation combined with forward simulation. Accurate evaluation of game situations is important for the victory in many sports, not just in soccer. The other sports can be also benefitted by the FOSSE approach in evaluating the game situations.

5 Conclusion

In this paper, we proposed FOSSE approach for evaluating game situation of RoboCup Soccer Simulation 2D League. Three contributions in evalating a game situation were presented. The first contribution is to show the effectiveness of using the field situations of multiple time cycles rather than only a single situa-

Table 5: Experimental results (FOSSE model).

Dimensionality	Input	MAE (# of time cycles)		
		One	Three	Five
2 inputs	Single	3.94	-	-
	Multiple-Past	-	3.70	3.76
	Multiple-Future	-	3.33	3.31
	Multiple-Predict	-	3.63	3.77
24 inputs	Single	3.84	-	-
	Multiple-Past	-	3.62	3.57
	Multiple-Future	-	3.35	3.11
	Multiple-Predict	-	3.55	3.59
46 inputs	Single	3.51	-	-
	Multiple-Past	-	3.53	3.45
	Multiple-Future	-	3.28	3.07
	Multiple-Predict	-	3.44	4.09

tion. The second contribution is to show that future information is more valuable than past information. The third contribution, which is the main contribution, is to propose FOSSE approach where simulated future information was generated by forward simulation. The FOSSE approach consists of two parts: Forward simulation part and situation evaluation part.

In our FOSSE approach, a DNN with multiple future situations was used as the situation evaluation part, and the LSTM was used for the forward simulation part. From the computational experiments, the effectiveness of our model was shown. This achievement allows us to evaluate the game situation during the ongoing game in real time. It is expected that the FOSSE approach can be applied to other sports as well as soccer such as rugby and basket ball.

The idea of this approach is similar to human thinking processes. People often unconsciously perform forward simulation when evaluating the situation in real life. When humans guess $SituationScore$ in a certain situation, it has possibility that they consider not only the current game situation but also expected future game situations. If it is proved that the proposed method is the same process as human thought process, it is considered effective to reproduce human thinking process by machine learning method.

6 Future Work

This paper conducted the computational experiments with only two teams. That is, only two teams were involved in the generation of training and test datasets. Considering the practical application where various teams are involved in a tournament, it is necessary to show that the proposed method works in general for any other teams. In the future, we will investigate the generalization of the proposed method. That is, it is necessary to examine the performance of the trained model to unknown teams that are not included in the generation process of the training dataset.

Furthermore, as already mentioned in the experiments of Section 4.4, it is necessary to consider improving the prediction accuracy of the forward simulation. For instance, different architectures of the forward simulation model and evaluation situation model can be used by increasing the number of hidden layers or by changing the number of situations for the input. Another idea is to adapt the FOSSE approach for accommodating field image data because it was indicated in [2] that using image data could lead to a better accuracy performance in evaluating game situations. In addition, we will consider machine learning method that computationally realizes human thinking processes. Incorporating human thought processes into machine learning method has the potential to contribute to the development of artificial intelligence.

Ultimately, we would like to implement it on the RoboCup soccer team and apply it as an indicator of tactical switching during the game. In addition to that, we would like to apply it for enhancing the game-watching experience, which is not related to the implementation of a team.

References

1. Hitoaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa and Hitoshi Matsubara, "RoboCup: A Challenge Problem for AI", *AI Magazine*, Vol.18, No.1, pp.73–85, 1997.
2. Tomoharu Nakashima and Tanguy Pomas, "Evaluation of Situation in RoboCup 2D Simulations using Soccer Field Images", *Proc. of the RoboCup Symposium*, 12 pages, 2018.
3. Hidehisa Akiyama, Tomoharu Nakashima, Yudai Suzuki, An Ohori, and Takuya Fukushima: "HELIOS2018: Team Description Paper", *RoboCup2018 Montreal*, 6 pages, 2018.
4. Hidehisa Akiyama, and Tomoharu Nakashima, "HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation", *Proc. of the 17th annual RoboCup International Symposium*, 2013.
5. Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization", *Proc. of International Conference on Learning Representations*, 2015.
6. Aida Khosroshahi, Eshed Ohn-Bar and Mohan Manubhai Trivedi, "Surround Vehicles Trajectory Analysis with Recurrent Neural Networks", *Proc. of the IEEE 19th Conference on Intelligent Transportstion Systems (ITSC)*, pp.2267–2271, 2016.
7. Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart and Jimeng Sun, "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks", *Proc. of the Machine Learning for Healthcare 2016*, pp.301-318, 2016.
8. Zhiyuan Shi, Min Xu, Quan Pan, Bing Yan and Haimin Zhang, "LSTM-based Flight Trajectory Prediction", *Proc. of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp.1–8, 2018.
9. Sepp Hochreiter and Jrgen Schmidhuber,"Long Short-Term Memory", *Journal of Neural Computation*, vol.9, No.8, pp.1735–1780, 1997.
10. Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei and Silvio Savarese, "Social LSTM : Human Trajectory Prediction in Crowded Space", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.961–971, 2016.