AI-FML Agent for Robotic Game of Go and AIoT Real-World Co-Learning Applications

Chang-Shing Lee, Yi-Lin Tsai, Mei-Hui Wang, Wen-Kai Kuan, Zong-Han Ciou
Dept. of Computer Science and Information Engineering
National University of Tainan
Tainan, Tawain
leecs@mail.nutn.edu.tw

Naoyuki Kubota
Dept. of Mechanical Sytems Engineering
Tokyo Metropolitan University
Tokyo, Japan
kubota@tmu.ac.jp

***Abstract*—In this paper, we propose an AI-FML agent for robotic game of Go and AIoT real-world co-learning applications. The fuzzy machine learning mechanisms are adopted in the proposed model, including fuzzy markup language (FML)-based genetic learning (GFML), eXtreme Gradient Boost (XGBoost), and a seven-layered deep fuzzy neural network (DFNN) with backpropagation learning, to predict the win rate of the game of Go as Black or White. This paper uses Google AlphaGo Master sixty games as the dataset to evaluate the performance of the fuzzy machine learning, and the desired output dataset were predicted by Facebook AI Research (FAIR) ELF Open Go AI bot. In addition, we use IEEE 1855 standard for FML to describe the knowledge base and rule base of the Open Go Darkforest (OGD) prediction platform in order to infer the win rate of the game. Next, the proposed AI-FML agent publishes the inferred result to communicate with the robot Kebbi Air based on MQTT protocol to achieve the goal of human and smart machine co-learning. From Sept. 2019 to Jan. 2020, we introduced the AI-FML agent into the teaching and learning fields in Taiwan. The experimental results show the robots and students can co-learn AI tools and FML applications effectively. In addition, XGBoost outperforms the other machine learning methods but DFNN has the most obvious progress after learning. In the future, we hope to deploy the AI-FML agent to more available robot and human co-learning platforms through the established AI-FML International Academy in the world.**

Keywords*—Intelligent Agent, Robot, Fuzzy Machine Learning, Game of Go, AIoT

I. Introduction

With the success of AlphaGo [1], there has been a lot of interest among students and professionals to apply machine learning to gaming and in particular to the game of Go. Go is a highly competitive and time-consuming activity and each move played on the Go board can strongly make an influence on the probability of winning or losing the game [4]. The Open Go Darkforest (OGD) prediction platform has been constructed since 2016 [3, 4, 5, 6]. The OGD cloud platform, including a Facebook AI Research (FAIR) dynamic Darkforest (DDF) AI bot and a FAIR ELF Open Go AI bot prediction mechanisms, has the ability to predict the top five selections for the next move. The AI bot provides each selection with real-time win rate, simulation numbers, and the top-move matching rate [6]. The IEEE Computational Intelligence Society (CIS) funded Fuzzy Markup Language (FML)-based machine learning competition for human and smart machine co-learning on game of Go in IEEE CEC 2019 and FUZZ-IEEE 2019. The goal of the competition is to understand the basic concepts of an FML-based fuzzy inference system, to use the FML intelligent decision tool to establish the knowledge base and rule base of the fuzzy inference system, and to optimize the FML knowledge base and rule base through the methodologies of evolutionary computation and machine learning [2].

Machine learning has been a hot topic in research and industry and new methodologies keep developing all the time. Regression, ensemble methods, and deep learning are important machine learning methods for data scientists [10]. Extreme gradient boost (XGBoost) is a scalable tree boosting system that provides state-of-the-art results on many problems [11]. For example, Song et al. [12] proposed an optimization model combining XGBoost algorithm with improved particle swarm optimization (PSO) to address the continuous multivariable optimization problem. Jiang et al. [13] proposed a genetic algorithm (GA)-XGBoost classifier for pedestrian detection to improve the classification accuracy. Zhang et al. [14] proposed XGBoost-based algorithm to recognize five indoor activities and its performance is better than the other ensemble learning classifier and single classifiers. Deep learning architectures have been applied to many research fields and in some cases surpass human expert performance [15]. Based on deep reinforcement learning, Ding et al. [16] proposed a novel intelligent diagnosis method to overcome the shortcomings of the diagnosis methods. Huang et. al. [17] developed a neural network scheme to extract information from emails to enable its transformation into a multidimensional vector.

It is estimated that there will be 4.5 billion Internet of Things (IoTs) joining the Internet by 2020 [18]. Alaiz-Moreton et al. [19] created classification models based on ensemble methods and deep learning models to classify the attacks of an IoT system that uses the MQTT protocol. Al-Ali et al. [19] presented an energy management system for smart homes to better manage energy consumption by utilizing the technologies of IoT and big data. In 2020, in addition to Go, the organizers of the FML-based machine learning competition for human and smart machine co-learning on game of Go/AIoT at IEEE WCCI 2020 further propose the AI-FML agent to integrate AI tools, i.e., using AI-FML tools to construct the knowledge base and rule base of the fuzzy inference system, with Internet of Things (IoT). The AI-FML agent can communicate with IoT devices or robots, and it has been introduced to the learning course of 5-grade computer studies at Rende elementary school in Taiwan from Sept. 2019

The authors would like to thank the financially support sponsored by the Ministry of Science and Technology of Taiwan under the grants MOST 108-2218-E-024-001.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

to Jan. 2020. This paper uses AlphaGo Master sixty games [7] as the experimental dataset downloaded from the website of the competition @ IEEE WCCI 2020. We use the data predicted by DDF AI bot [8] as the training data, and the data predicted by ELF Open Go AI bot [9] are as the desired output of the training data. The goal of the fuzzy machine learning mechanism in this paper is to make the win rates predicted by the DDF AI bot closer to those predicted by the ELF Open Go AI bot. In addition, the proposed AI-FML agent communicates with the robot Kebbi Air through MQTT protocol to allow the students who joined the human and smart machine co-learning course to directly interact with the robot to simulate their learning motivation.

The remainder of the paper is organized as follows. We first introduce the two-stage system structure for game of Go and AIoT applications in Section II. We then describe the dataset used in the FML-based machine learning competition at IEEE WCCI 2020 in Section III as well as the adopted fuzzy machine learning mechanisms in Section IV. Experimental results will be given in Section V and finally we conclude the paper in Section VI.

II. Two-Stage System Structure for Game of Go and AIoT Applications

A. Two-Stage System Structure

Fig. 1 shows the two-stage system structure for robotic game of Go and AIoT real-world co-learning applications which describes as follows:

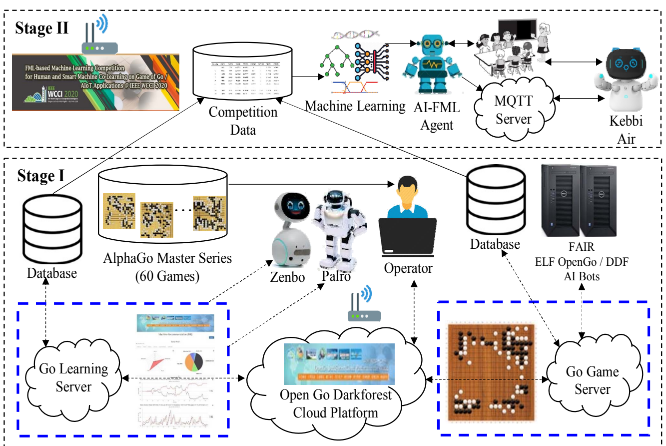


Fig. 1. Two-stage system structure for robotic game of Go and AIoT real-word co-learning applications.

- Stage I: The operator uploads the game records of AlphaGo Master series to the OGD cloud platform to predict the top five selections for the next move. The OGD cloud platform communicates with Go Game server and Go learning server through the developed intelligent agents [4]. The FAIR ELF Open Go/DDF AI bots provide each move suggestion with real-time variances in win rate, simulation number, and top-move matching rate which will be stored in the database. The robot Zenbo or Palro reports the information and suggestions to the Go player or operator for human and smart machine co-learning.
- Stage II: The competition data of FML-based machine learning competition for human and smart machine co-learning on game of Go/AIoT at IEEE WCCI 2020 will be available from the database. In this paper, we train the game dataset based on machine learning mechanisms to make the proposed AI-FML agent work on various application domains. Finally, students in the classroom are able to interact with the intelligent agent and the robot like the Kebbi Air for real-world novel applications through the MQTT server constructed by NUWA Robotics or KingKit Technology-Ltd. in Taiwan.

B. Diagram of AI-FML Agent with AIoT Real-World Applications

Fuzzy Markup Language (FML) is based on XML and has been a standard of IEEE 1855 since 2016. It is a powerful tool for human to construct the knowledge base and rule base on various real-world applications [20, 21]. MQTT protocol is designed as an extremely lightweight publish/subscribe messaging transport which is usually widely used in IoT [18]. The AI-FML agent is developed based on the technologies of fuzzy inference mechanism and the concepts of the MQTT protocol. The AI-FML agent is deployed on the MQTT client/server system, which is a communication protocol for IoT and was originally developed by IBM and Eurotech. The MQTT protocol has been the OASIS international standard in 2014, and it can work in narrow band and low power environment for transmission and receiving message based on Publisher and Subscriber mechanism via Broker. The robot will receive the fuzzy machine learning results based on the AI-FML agent and MQTT client/server system.

Fig. 2 shows the diagram of the AI-FML agent with AIoT for real-world applications which is composed of a data preprocessing mechanism, a genetic algorithm (GA)-based FML fuzzy inference mechanism, an AI-FML agent, a MQTT broker, a blockly program on CodeLab of NUWA Robotics or Webduino:Bit of KingKit Technology-Ltd., and IoT devices, such as a robot Kebbi Air. First, the data pre-processing mechanism, including standardization and outlier analysis, is performed. Then, the GA-based FML fuzzy inference mechanism, including optimization process for knowledge base and rule base, infers the suitable result and sends it to the AI-FML agent. In addition, the AI-FML agent deals with the communication with the MQTT broker in the cloud server through TCP/IP. The user edits his/her blockly program on CodeLab or Webduino:Bit and deploys it to the IoT device or a robot. When the AI-FML agent receives the message, the robot reflects its corresponding action/behavior based on the designed blockly program [22] in the client.

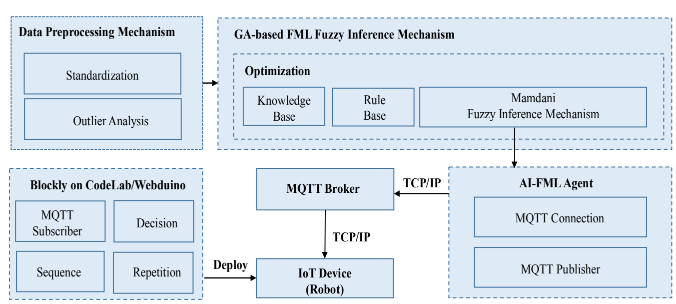


Fig. 2. Diagram of AI-FML agent for AIoT real-world application.

C. Message Delivery of AI-FML Agent with AIoT Real-World Applications

Fig. 3 is the basic concept of message delivery method of AI-FML agent through the MQTT protocol which is described as follows.

- In the edge computing, the user first activates the FML system to construct the knowledge base (KB) and rule base (RB) of the real-world application domain, for example, Go game, which are stored in the KB/RB repository. Then, the user surfs on the CodeLab/Webduino:Bit visual Kebbi-Air blockly program editor to design the robot's corresponding behaviors/actions to the inferred result, and deploys the blockly program to the robot by subscribing the topic like "fml." After that, the user inputs fuzzy variable with its values into the constructed fuzzy inference system of Go game.
- AI-FML agent then acts as a publisher to publish the message, i.e., the inferred result, to the topic "fml" in the MQTT broker. Then, the server converts the published message to a string data and delivers the message to the subscriber, i.e., the robot Kebbi Air.
- The robot subscriber "fml" gets this message and activates its corresponding actions to that number according to the designed blockly program such as speaking, dancing, singing, smiling, …, and so on. For example, if the win rate of the Black is very high, then Kebbi Air speaks "Black has an obvious advantage" with smile.

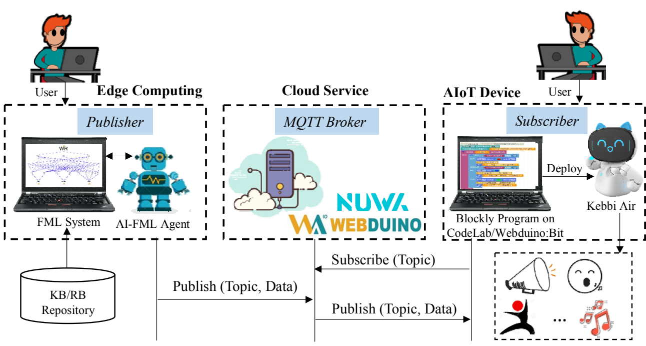


Fig. 3. Message delivery method of AI-FML agent through the MQTT protocol.

III. FML-BASED MACHINE LEARNING COMPETITION ON GAME OF GO AT IEEE WCCI 2020

A. Introduction to DataSet

In this paper, we adopt the dataset from Google DeepMind AlphaGo Master sixty games [7] to test the performance of the fuzzy machine learning (FML) mechanism. There are five steps for the fuzzy machine learning, including 1) real-world problem definition, 2) fuzzy feature extraction and data pre-processing, 3) selection and construction for the FML model, 4) analysis and modification for the FML model, and 5) optimization for the FML on the real-world problem. There are six fuzzy features in the dataset including *DBSN*, *DWSN*, *DBWR*, *DWWR*, *DBTMR*, and *DWTMR*, predicted by Darkforest AI bot. The outputs are *EBWR* and *EWWR* predicted by ELF Open Go AI bot. *DBSN* and *DWSN* denote the number of simulations for Black and White, respectively. *DBWR* and *DWWR* are the win rates of Black and White, respectively. *DBTMR* and *DWTMR* are the top-move matching rates of Black and White, respectively. *EBWR* and *EWWR* are the win rates of Black and White, respectively.

TABLE I. PARTIAL DATA FROM GAME 1 TO GAME 40.

No	*DBSN*	*DWSN*	*DBWR*	*DWWR*	*DBTMR*	*DWTMR*	*EBWR*	*EWWR*
1	3863	2274	0.52733	0.48533	0	1	0.49636	0.50884
2	9283	7866	0.51529	0.48532	0.5	1	0.45779	0.54844
3	11395	6798	0.51265	0.47717	0.6667	1	0.45671	0.57235
4	4499	10703	0.51885	0.46988	0.75	1	0.49086	0.5201
5	7388	20017	0.5288	0.46679	0.8	1	0.47218	0.53649
6	20098	9693	0.53309	0.46602	0.8333	1	0.47249	0.52224
					⋮			
3754	7211	12038	0.92754	0.28941	0.797	0.812	0.9566	0.08276
3755	20017	20134	0.81634	0.18352	0.7985	0.8134	0.87651	0.04687
3756	7691	12699	0.80645	0.15954	0.7926	0.8074	0.92438	0.18538
3757	17076	4603	0.85503	0.09789	0.7941	0.8088	0.9107	0.18915
3758	9015	20017	0.90843	0.10819	0.7956	0.8102	0.98549	0.00822

Participants of the IEEE WCCI 2020 competition [2] can choose any 40 games from 60 games as the training data and the remaining 20 games as the testing data. In this paper, we choose Game 1 to Game 40 as the training data and Game 41 to Game 60 as the testing data. Table I shows partial data from Game 1 to Game 40 and its sample size is 3758. Additionally, we compose the dataset from Game 1 to Game 40 in two different training datasets, namely training data set 1 (TDS1) and training dataset 2 (TDS2). In TDS1, fuzzy feature variables include *DSN*, *DTMR*, and *DWR* while in TDS2, fuzzy features variables are *DBSN*, *DWSN*, *DBTMR*, *DWTMR*, *DBWR*, and *DWWR*. The labeled data of TDS1 is *EWR* while the ones of TDS2 is *EBWR* and *EWWR*. Table II shows fuzzy feature variables and labeled variable(s) of TDS1 and TDS2. Additionally, TDS2 has two choices including TDS2-1 with two labeled variables and TDS2-2 with only one labeled variable. The number of fuzzy feature variables (F. No.) and labeled variable(s) (L. No.) of TDS1, TDS2-1, and TDS2-2 are (3, 1), (6, 2), and (6, 1), respectively. The total sample size of TDS1 and TDS2 is 7,494 and 3,736, respectively, before data preprocessing.

TABLE II. FEATURE VARIABLES AND LABELED VARIABLE(S).

Training Dataset Name	Feature Variables	Labeled Variable(s)	(F.No, L. No)
TDS1	*DSN*, *DTMR*, *DWR*	*EWR*	(3, 1)
TDS2	*DBSN*, *DWSN*, *DBTMR*, *DWTMR*, *DBWR*, *DWWR*	TDS2-1	
		EBWR, *EWWR*	(6, 2)
		TDS2-2	
		EBWR	(6, 1)

B. Data Standardization

- Training Data Set TDS1

Fig. 4 shows the correlation among fuzzy features *DSN*, *DTMR*, *DWR*, and *EWR* of TDS1 in a heat map. We can observe that the correlation between *DWR* and *EWR* is about 0.7 while the relationship between *DTMR* and *EWR* is negative correlation. Next, we observe the distribution of *DSN* in TDS1 and find that the difference between minimum and maximum of each game is from 100 to millions so that we utilize Quantile Transformer to transform and normalize the *DSN* between 0 and

1. Figs. 5(a)-5(b) show the box plots of *DSN* of TDS1 before and after standardization, respectively.

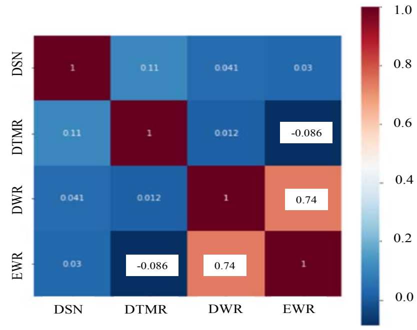


Fig. 4. Heat map of fuzzy feture variables and labeled variable of TDS1.

- Training Data Set TDS2

Fig. 6 shows the scatter matrix among fuzzy feature variables and labeled variables of TDS2. The diagonal plots show the histogram of each fuzzy variable, while the non-diagonal plots show the scatter diagram for analyzing relationships between two fuzzy variables. For example, the pairs of (*DBWR*, *DWWR*) and (*EBWR*, E*WWR*) are strongly negative correlation. And, the pairs of (*DBWR*, *EBWR*) and (*DWWR*, *EWWR*) are scattered from left-bottom to right-top which indicates there exists the relationship between fuzzy variables in the pair. Figs. 7(a) and 7(b) show box plots of *DBSN* and *DWSN* of TDS2, respectively, before and after standardization.

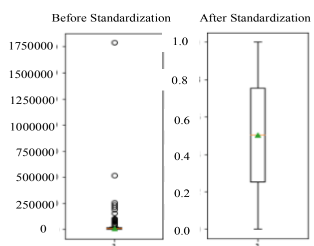


Fig. 5. Box plot of *DSN* of TDS1 before and after standardization.

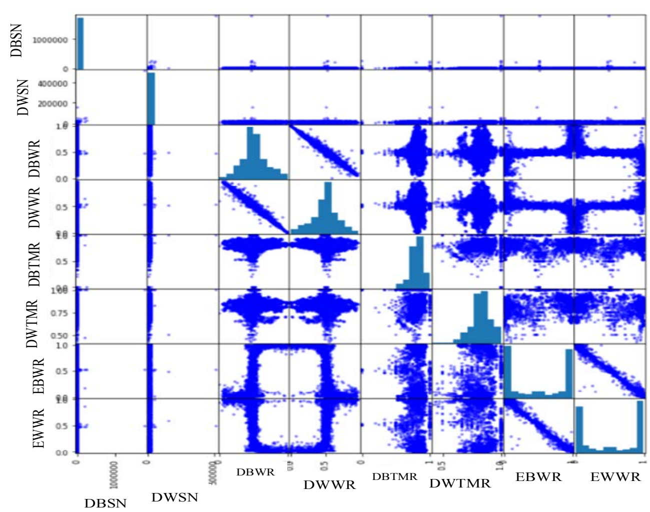


Fig. 6. Scatter matrix with histograms of fuzzy feture variables and labeled variables of TDS2.

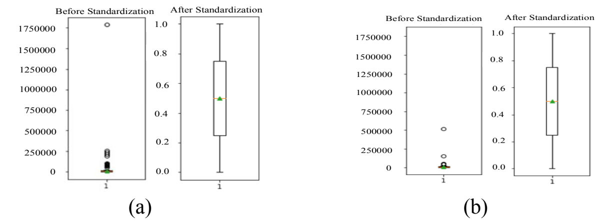


Fig. 7. Box plot of (a) *DBSN* and (b) *DWSN* of TDS2.

C. Data Outlier Analysis

Outliers are data points that are far from the other data points; therefore, this subsection tries to find outlier candidates in order not to cause problems in learning-model procedures. In this paper, we remove outliers based on measures of the interquartile range (IQR = Q3 − Q1) where Q1 and Q3 denote the lower quartile and upper quartile, respectively. When a point is beyond 1.9×IQR, it will be removed from the training dataset. Fig. 8(a) shows the box plots of fuzzy variables of TDS1 and indicates *DTMR* has outliers. Fig. 8(b) shows the box plot of *DTMR* after removing outliers. Figs. 9(a)-9(b) show the box plots of *DBTMR* and *DWTMR* of TDS2 before and after removing outliers, respectively. After removing outliers from the TDS1, the sample size of TDS1 is changed from 7,494 to 7,356. The sample size of TDS2 is reduced from 3,736 to 3,582. Fig. 10 shows graphs of the histogram and the estimated PDF over the *DBSN*, *DWSN*, *DBWR*, *DWWR*, *DBTMR*, and *DWTMR* in TDS2 after data standardization and data outlier analysis.

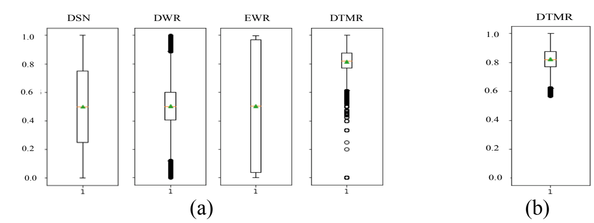


Fig. 8. Box plots of (a) *DSN*, *DWR*, *EWR*, and *DTMR* in TDS1. (b) Box plot of *DTMR* of TDS1 after remvoing outliers.

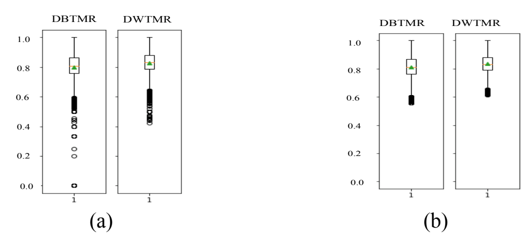


Fig. 9. Box plots of *DBTMR* and *DWTMR* of TDS2 (a) before and (b) after removing outliers.

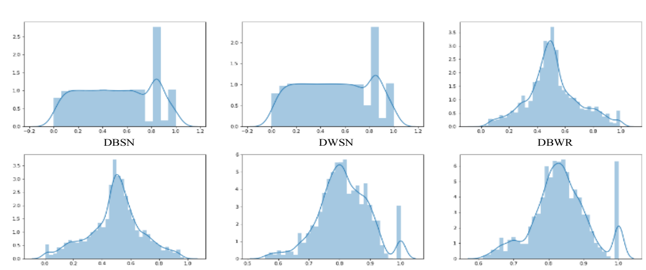


Fig. 10. Histogram and the estimated PDF over the data in TDS2 after data standardization and data outlier analysis.

IV. FUZZY MACHINE LEARNING MECHANISMS

A. GFML-based Learning Mechanism

This subsection describes the information of the GFML-based machine learning (FML) mechanism [23], where we use TDS1 as the training set of the fuzzy machine learning. Table III shows the fuzzy variables with their corresponding fuzzy sets for fuzzy knowledge base. There are three input fuzzy variables, including *DSN*, *DTMR*, and *DWR* and we set *DSN* = {**low, high**}, *DTMR* = {**low, high**}, and *DWR* = {**low, med_low, med_high, high**}. The output fuzzy variable is *EWR* and we set *EWR* = {**low, med, high**}. In this paper, we use 16 fuzzy inference rules with three input fuzzy feature variables for optimizing the predicted win rate by DDF Go AI bot based on the ELF Open Go AI bot prediction. Table IV shows its fuzzy rules.

TABLE III. PARAMETERS OF FUZZY SETS FOR FUZZY KNOWLEDGE BASE.

Input Fuzzy Variables	Parameters of Fuzzy Sets	
DSN	low	[0,0,0.4,0.6]
	high	[0.4,0.6,1,1]
DTMR	low	[0,0,0.4,0.6]
	high	[0.4,0.6,1,1]
DWR	low	[0,0,0.2,0.3]
	med_low	[0.2,0.3,0.45,0.55]
	med_high	[0.45,0.55,0.7,0.8]
	high	[0.7,0.8,1,1]
Output Fuzzy Variable	Parameters of Fuzzy Set	
EWR	low	[0,0,0.3,0.4]
	med	[0.3,0.4,0.6,0.7]
	high	[0.6,0.7,1,1]

TABLE IV. FUZZY RULES.

No.	Input Fuzzy Variables			Output Fuzzy Variable
	DSN	*DTMR*	*DWR*	*EWR*
1	low	low	low	low
2	low	low	med_low	low
3	low	low	med_high	high
4	low	low	high	high
5	low	high	low	low
		⋮		
11	high	low	med_high	med
12	high	low	high	high
13	high	high	low	low
14	high	high	med_low	med
15	high	high	med_high	high
16	high	high	high	high

B. XGBoost-based Learning Mechanism

Ensemble learning [24] could be categorized into three classes, including learning in a way of bagging, boosting, and stacking, which can improve the predicted acurrancy in real-world applications. The main idea of boosting is to add new models to the ensemble sequentially. XGBoost [11] is one type of gradient boosting algorithms and its goal is to modify its previous tree and to generate a better new decision tree after learning. In addition, it normalizes the objective function and adopts L1/L2 regularization to keep loss function more smooth and avoid noise interruption. Fig 11 shows that a series of weak prediction models 1, 2, … , and *K* might not be good for the entire training dataset but is good for some part of the training dataset. All weak prediction models 1, 2, .., and *K* have been trained and they are combined to give the final result to achieve better performance of real-world predictions. The objective function of XGBoost (*Obj*) is calculated by (1) [11].

$$Obj = \sum_{i=1}^{n} loss(y_i - \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (1)$$

Where 1) a given training data *D* is with *n* samples and *m* fuzzy features D={($\mathbf{x}_i$, y_i)}, 2) each f_k corresponds to an inependent tree structure *q* and leaf weights *w*, 3) $\hat{y}_i$ is the predicted output and calculated by $\sum_{k=1}^{K} f_k(\mathbf{x}_i)$, 4) *loss* is a differentiable convex loss function that measures the difference between the prdiction $\hat{y}_i$ and the target y_i, 5) Ω is the regularization to smooth the final learned weights to avoid over-fitting [11].

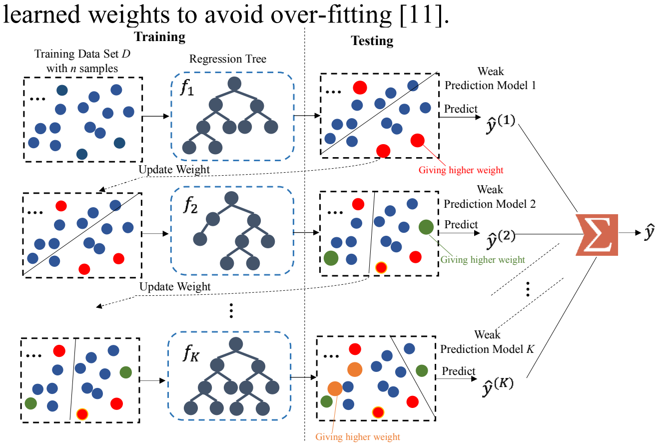


Fig. 11. Gradient boosting with *K* weak prediction models.

C. Deep Fuzzy Nueural Network Learning Mechanism

In this subsection, we utilize a deep fuzzy neural network (DFNN) learning mechanism [8] to train the model for the dataset of AlphaGo Mater sixty games. Fig. 12 shows the structure DFNN learning structure adopted in this paper. The detailed structure and parameters of DFNN will be introduced in Experimental Results section. There are six input fuzzy variables, including fuzzy variables of *DBSN*, *DWSN*, *DBWR*, *DWWR*, *DBTMR*, and *DWTMR*, in the input fuzzy layer. The number of nodes in hidden layers 1, 2, 3, 4, and 5 are 66, 128, 256, 256, and 10, respectively, followed by a ReLU activation function. The final layer has 1 node to denote the predicted win rate of Black and applies a ReLU function. Moreover, the dropout, mini-batch, and Adam optimization [25] mechanisms are also adopted in this paper.

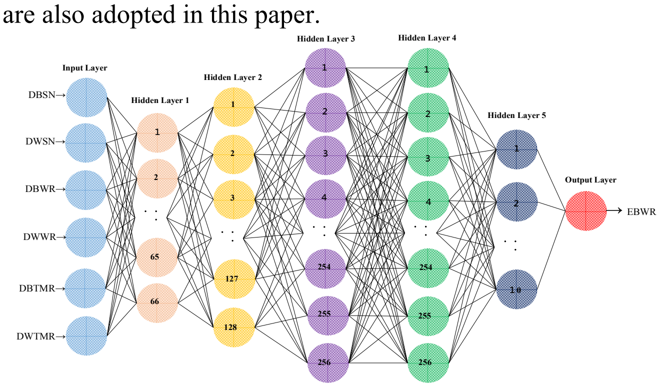


Fig. 12. Structure of DFNN learning mechanism adopted in this paper.

V. EXPERIMENTAL RESULTS

A. AI-FML Agent with Fuzzy Machine Learning Mechanisms

Five experiments in (Exp. 1– Exp. 5) fuzzy machine learning for testing the performance of the AI-FML agent were done in this paper. Table V shows the information of these five experiments. The first two experiments Exp. 1 and Exp. 2 adopt TDS1 to be the training data but Exp. 3 – Exp. 5 use TDS2 to do the training. The difference between TDS2-1 and TDS2-2 is the number of labeled variables (see Table II). The implemented fuzzy machine learning mechanisms in Exp. 1, Exp. 2 – Exp. 4, and Exp. 5 are GFML-based AI-FML agent [23], XGBoost-based regression model [11], and DFNN with Adam learning mechanism [25], respectively. We briefly describe the adopted parameters for the fuzzy machine learning as follows:

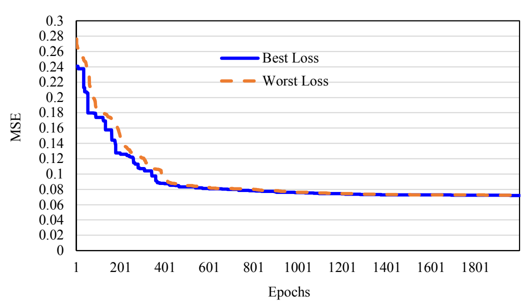
1) ***Exp. 1***: *pool size is 20, crossover rate is 0.9, and mutation rate is 0.1;*
2) ***Exp. 2** – **Exp. 4***: *maxiumun_depth of the decision tree is 6 and learning rate is 0.001;*
3) ***Exp. 5***: *dropout rate is 0.3, batch size is 140. In addition, the parameters for Adam learning optimization are as follows: learning rate is 0.001, β_1 is 0.1, and β_2 is 0.999;*
4) *The number of epoch in Exp. 1 – Exp. 5 is 2000;*
5) *The proportion which splits the training dataset into a validation dataset is 0.3 during training. There are 5,149 training data and 2,207 validating data in Exp. 1 and Exp. 2. For Exp. 3 – Exp. 5, the number of the trainind data and validate data are 2,507 and 1,075, respectively.*

TABLE V. FIVE EXPERIMENTAL INFORMATION.

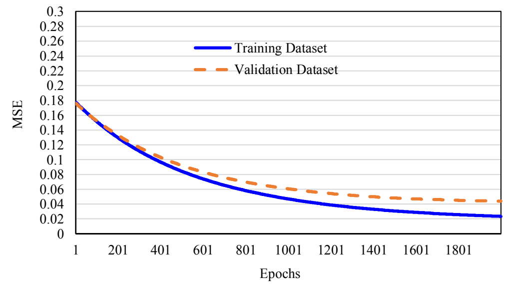
Exp. No.	1	2	3	4	5
TDS No.	TDS1	TDS1	TDS2-1	TDS2-2	TDS2-2
Machine Learning Method	GFML-based AI-FML Agent	XGBoost-based Regression Model			DFNN with Adam

TABLE VI. MSE OF EXP. 1 TO EXP. 5.

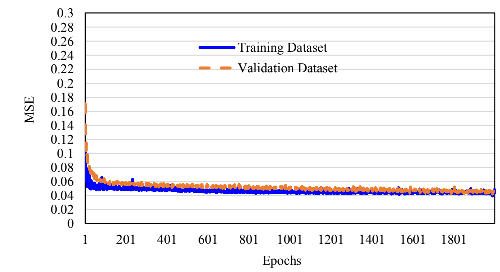
MSE	Exp. No				
	1	2	3	4	5
Before Learning	0.263				
After Learning	1 epoch				
	0.2389	0.1744	0.1776	0.17685	**0.1577**
	2000 epochs				
	0.0734	0.0488	0.03365	**0.03355**	0.04075
	Difference between 1 and 2000 epochs				
	0.1655	0.1256	0.14395	0.1433	**0.11695**



(a)



(b)



(c)

Fig. 13. (a) Best loss and worset loss curves of training dataset in Exp. 1. Loss curves of training dataset and validation dataset in (b) Exp. 4 and (c) Exp. 5.

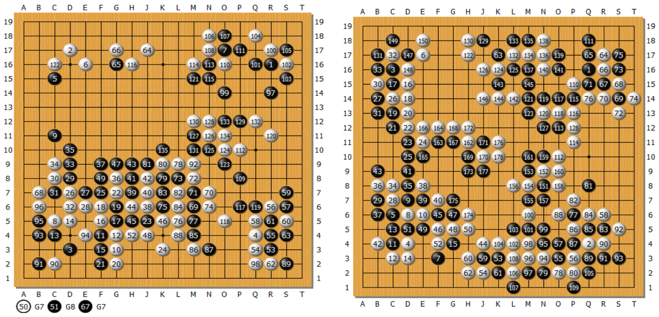
In this paper, we used mean square error (MSE) as the metric to explore the performance of the implemented five experiments. Table VI shows the MSE values without learning, with 1-epoch learning, and with 2000-epoch learning, in Exp. 1 to Exp. 5. Observe Table VI, Exp. 5 has the best performance when learning 1 epoch, and Exp. 4 performs best after learning 2000 epochs. Fig. 13(a) shows the variance in the chosen best loss and the worset loss from all of the chromosomes in the pool during learning. Figs. 13(b) and 13(c) show the loss curves for both the training dataset and validation dataset against the number of training iterations in Exp. 4 and Exp. 5, respectively.

B. Prediction Performance using the Learned Models

This subsection displays the prediction performance of the testing dataset from Game 41 to Game 60. We use the learned models in Exp. 4 and Exp. 5 to observe the performance of Game 39 (*AlphaGo Master as Black, Iyama Yuta as White, B+R*) and Game 58 (*Chang Hao as Black, AlphaGo Master as White, W+R*) with Komi = 6.5, Japanese rule, basic time = 1 minute, and overtime = 3×30 byo-yomi.

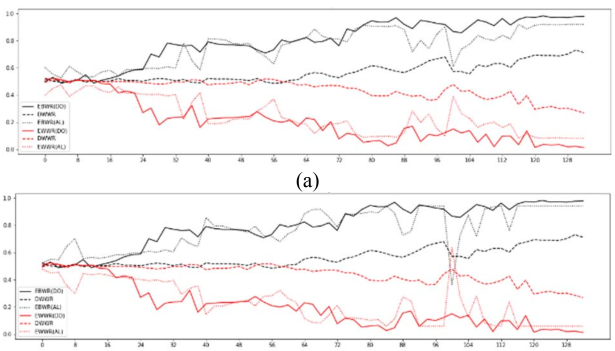
Fig. 14 shows the game records of Game 39 and Game 58. Fig. 15 shows Game 39's win rates of Black (black lines) and White (red lines) which are predicted by DDF AI bot (dashed lines), by ELF Open Go AI bot (solid lines), and by the learned models (dotted lines). The loss values of Game 39 in Exp. 4 and Exp. 5 are 0.007856 and 0.010373, respectively. Fig. 16 shows the corresponding win rate curves of Game 58. The loss values of Game 58 in Exp. 4 and Exp. 5 are 0.012748 and 0.011995, respectively. Observe Fig. 15 and Fig. 16, the results indicate as follows:

1) The trend for win rates predicted by the learned models is approaching to those predicted by the ELF Open Go AI bot.
2) Exp. 4 performs better than Exp. 5 for Game 39.
3) Exp. 5 performs better than Exp. 4 for Game 58.



(a) (b)

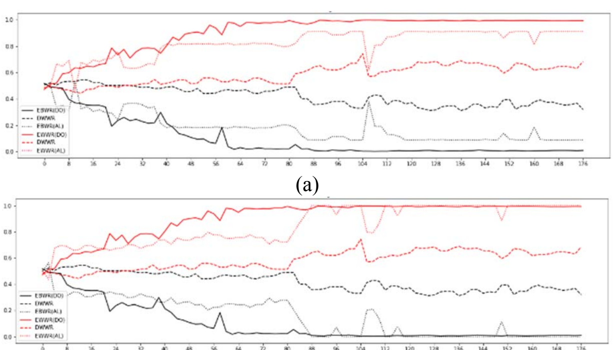
Fig. 14. Game records of (a) Game 39 and (b) Game 58.



(a)

(b)

Fig. 15. Prediction performance of Game 39 using the models learned in (a) Exp. 4 and (b) Exp. 5.



(a)

(b)

Fig. 16. Prediction performance of Game 58 using the models learned in (a) Exp. 4 and (b) Exp. 5.

C. Performance of AI-FML Agent in Teaching and Learning Fields

This subsection shows the real-world applications that GFML-based AI-FML agent works with the AIoT devices or robots in elementary schools in Taiwan. This study cooperates with NUWA Robotics and KingKit Technology-Ltd. to deploy the AI-FML agent to the robot Kebbi Air and human co-learning platform to the teaching and learning fields. Fig. 17 shows the learned fuzzy sets of all adopted fuzzy variables in Exp. 1. Fig. 18 shows the blockly program on the CodeLab of NUWA Robotics which indicates 1) if the output fuzzy variable *EWR* with *fuzzy number* between 0 and 0.35, then Kebbi Air speaks "*Catch up with the game situation and cheer up according to FB ELF Open Go Prediction*," 2) if the output fuzzy variable *EWR* with *fuzzy number* between 0.35 and 0.65, then Kebbi Air speaks "*It is hard to say who will be the winner according to FB ELF Open Go Prediction*," and 3) if the output fuzzy variable *EWR* with *fuzzy number* is greater than 0.65, then Kebbi Air speaks "*I am going to win according to FB ELF Open Go Prediction*."

From Sept. 2019 to Jan. 2020, we introduced the developed AI-FML agent to the learning course of 5-grade computer studies at Rende elementary school in Taiwan. Three classes with total 74 students joined this human and robot co-learning cours every Friday afternoon. Before the end of the class, each student surfs on the constructed website to answer the questionnaire with six questions. Fig. 19 shows the feedback given by three-class students. Despite red-circle feedback from the students of Grade 5C, on the average, most students gave positive feedback on course of human and smart machine co-learning.

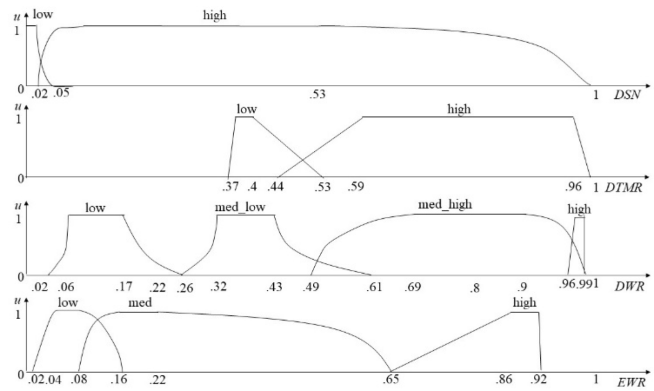


Fig. 17. Learned fuzzy sets of all adopted fuzzy variables in Exp. 1.

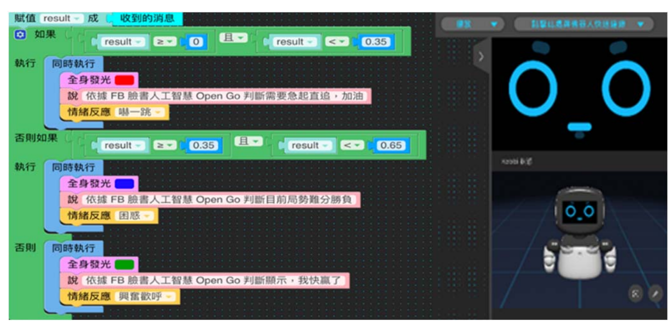


Fig. 18. Blockly program on the CodeLab of NUWA Robotics.

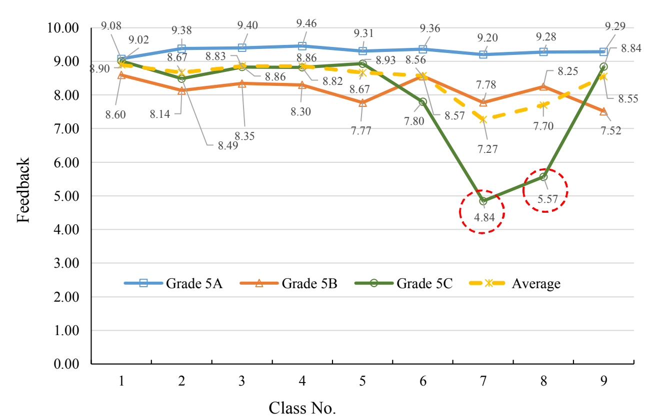


Fig. 19. Feedback given by three-class students for AI-FML learning course.

VI. CONCLUSIONS

This paper proposes an AI-FML agent to integrate AI tools with IoT for real-world applications. The AI-FML agent can communicate with IoT devices or robots, and it has been introduced to the learning course of 5-grade computer studies at Rende elementary school in Taiwan from Sept. 2019 to Jan. 2020. Additionally, this paper uses AlphaGo Master sixty games as the experimental dataset to make the win rates predicted by the DDF AI bot closer to those predicted by the ELF Open Go AI bot based on FML-based genetic learning (GFML), XGBoost learning, and DFNN learning. The experimental results show that XGBoost learning and DFNN learning have the good performance as well as AI-FML agent integrating with the robot Kebbi Air is popular with the involved students. In the future, we will discuss the difference in the performance among the used learning mechanisms, expand the discussion to help provide additional insight to the readers, and deploy the AI-FML agent to more available robot and human co-learning platforms through the established AI-FML International Academy in the world.

ACKNOWLEDGMENT

The authors would like to thank the staff of Center for Research of Knowledge Application & Web Service (KWS Center) of NUTN and the involved faculty and students of Rende elementary school in Taiwan. Finally, we also would like to thank the involved staff of NUWA Robotics and KingKit Technology-Ltd. for helping connect the developed AI-FML agent to Kebbi Air through the constructed MQTT server.

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[2] OASE Lab., "FML-based machine learning competition for human and smart machine co-learning on game of Go/AIoT applications at IEEE WCCI 2020," Jan. 2020, [Online] Available: http://oase.nutn.edu.tw/wcci2020-fmlcompetition/overview.php.

[3] C. S. Lee, M. H. Wang, L. W. Ko, Y. Hsiu Lee, H. Ohashi, N. Kubota, Y. Nojima, and S. F. Su, "Human intelligence meets smart machine: a special event at the IEEE International Conference on Systems, Man, and Cybernetics 2018," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 1, pp. 23-31, Jan. 2020.

[4] C. S. Lee, M. H. Wang, Y. L. Tsai, L. W. Ko, B. Y. Tsai, P. H. Hung, L. A. Lin, and N. Kubota, "Intelligent agent for real-world applications on robotic edutainment and humanized co-learning," *Journal of Ambient Intelligence and Humanized Computing*, 2019. (DOI: 10.1007/s12652-019-01454-4).

[5] C. S. Lee, M. H. Wang, L. W. Ko, N. Kubota, L. A. Lin, S. Kitaoka, Y. T Wang, and S. F. Su, "Human and smart machine co-learning: brain-computer interaction at the 2017 IEEE International Conference on Systems, Man, and Cybernetics," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 4, no. 2, pp. 6-13, Apr. 2018.

[6] C. S. Lee, M. H. Wang, S. C. Yang, P. H. Hung, S. W. Lin, N. Shuo, N. Kubota, C. H. Chou, P. C. Chou, and C. H. Kao, "FML-based dynamic assessment agent for human-machine cooperative system on game of Go," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 5, pp. 677-705, 2017.

[7] DeepMind, AlphaGo Master series: 60 online games, Jan. 2020, [Online] Available: https://deepmind.com/alphago-master-series.

[8] Y. Tian and Y. Zhu, "Better computer Go player with neural network and long-term prediction," Feb. 2016, [Online] Available: https://arxiv.org/abs/1511.06410.

[9] Y. Tian, J. Ma, Q. Gong, S. Sengupta, Z. Chen, J. Pinkerton, and C. L. Zitnick, "ELF Open Go: An analysis and open reimplementation of AlphaZero," May 2019, [Online] Available: https://arxiv.org/abs/1902.04522.

[10] J. Castanon, "10 machine learning methods that every data scientist should know," May 2019, [Online] Available: https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9.

[11] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," Jun. 2016, [Online] Available: https://arxiv.org/abs/1603.02754.

[12] K. Song, F. Yan, T. Ding, L. Gao, and S. Lu, "A steel property optimization model based on the XGBoost algorithm and improved PSO," *Computational Materials Science*, vol. 174, pp. 1-13, Mar. 2020.

[13] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters," *IEEE Access*, vol. 7, pp. 118310-118321, 2019.

[14] W. Zhang, X. Zhao, and Z. Li, "A comprehensive study of smartphone-based indoor activity recognition via Xgboost," *IEEE Access*, vol. 7, pp. 80027-80042, 2019.

[15] Wikipedia, "Deep learning," Feb. 2016, [Online] Available: https://en.wikipedia.org/wiki/Deep_learning.

[16] Y. Ding, L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng, and C. Lu, "Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach," *Advanced Engineering Informatics*, vol. 42, pp. 1-11, 2019.

[17] J. W. Huang, C. W. Chiang, and J. W. Chang, "Email security level classification of imbalanced data using artificial neural network: The real case in a world-leading enterprise," *Engineering Applications of Artificial Intelligence*, vol. 75, pp. 11-21, 2018.

[18] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Munoz-Castaneda, I. Garcia, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, vol. 2019, pp. 1-11, 2019.

[19] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approcah," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426-434, Nov. 2017.

[20] G Acampora and V Loia, "Fuzzy control interoperability and scalability for adaptive domotic framework," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 2, pp. 97-111, 2005.

[21] IEEE CIS, 1855-2016-IEEE Standard for Fuzzy Markup Language, 2016 [Online] Available: https://ieeexplore.ieee.org/document/7479441.

[22] Google, "Blockly Programming," Jan. 2020, [Online] Available: https://developers.google.com/blockly.

[23] C. S. Lee, M. H. Wang, M. J. Wu, O. Teytaud, and S. J. Yen, "T2FS-based adaptive linguistic assessment system for semantic analysis and human performance evaluation on game of Go," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 2, pp. 400-420, Apr. 2015.

[24] Y. Yang, Temporal data mining via unsupervised ensemble learning, Elsevier, 2017.

[25] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," *2015 International Conference on Learning Representation* (*ICLR 2015*), San Diego, CA, USA, May 7-9, 2015.